Mathematical Logic: From Formal Language to Gödel and Cohen
Volumes I–X

Student Edition

Contents

How to read the series vii

Notation and conventions ix

I Volume I: Propositional and First-Order Logic 1

Preface 2

1 Syntax and propositional semantics 3

2 Propositional proof systems 7

3 First-order syntax, substitution, and equality 10

4 Structures and satisfaction 13

5 Formal deduction 16

6 Henkin completeness 19

7 Compactness and Löwenheim–Skolem 23

End-of-volume perspective 26

II Volume II: Computability and Arithmetized Syntax 27

Preface 28

8 Primitive recursion 29

9 Coding finite objects 32

10 Recursive and computably enumerable relations 35

11 Weak arithmetic 38

12 Representability **42**

13 Gödel numbering of syntax **46**

14 Proof and provability predicates **49**

End-of-volume perspective **52**

III Volume III: Incompleteness, Provability, and Truth 53

Preface **54**

15 Self-reference **55**

16 Gödel's first theorem **60**

17 Rosser and essential undecidability **64**

18 Consistency and soundness notions **68**

19 Formalized provability **72**

20 Löb and second incompleteness **76**

21 Truth, reflection, and iterated consistency **81**

End-of-volume perspective **85**

IV Volume IV: ZF Set Theory and the Transfinite 86

Preface **87**

22 ZF axioms and elementary constructions **88**

23 Natural numbers and recursion **91**

24 Foundation and rank **94**

25 Ordinals **97**

26 Ordinal arithmetic **100**

27 Cardinals without assuming choice **104**

28 Well-founded relations and collapse **108**

End-of-volume perspective **111**

V Volume V: Choice, Models, Satisfaction, and Absoluteness 112

Preface **113**

29 Axiom of Choice and equivalents **114**

30 Cardinal arithmetic with choice **117**

31 Satisfaction for set-sized structures **120**

32 Absoluteness **124**

33 Reflection **128**

34 Elementary hulls **131**

35 Inner models and the Skolem paradox **135**

End-of-volume perspective **139**

VI Volume VI: The Constructible Universe 140

Preface **141**

36 Definability and Def(M) **142**

37 The L-hierarchy **146**

38 Uniform definability and absoluteness of L **150**

39 ZF axioms in L, part I **154**

40 ZF axioms in L, part II **158**

41 Canonical well-order and Choice **162**

42 Condensation **165**

43 GCH in L **169**

End-of-volume perspective **173**

VII Volume VII: Forcing I: Generic Extensions and the Forcing Theorem 174

Preface 175

44 Posets, dense sets, and generics 176

45 Names and valuation 180

46 Atomic forcing 184

47 General forcing relation 188

48 Truth and generic extension theorem 192

49 ZFC in generic extensions 197

End-of-volume perspective 202

VIII Volume VIII: Forcing II: Boolean Values, Preservation, and Relative Consistency 203

Preface 204

50 Separative quotients and Boolean completion 205

51 Boolean-valued semantics 208

52 Boolean-valued ZFC 213

53 Consistency transfer 217

54 Chain conditions and closure 222

55 Nice names and preservation bookkeeping 226

End-of-volume perspective 230

IX Volume IX: Cohen Forcing and Independence of CH 231

Preface 232

56 Adding Cohen reals 233

57 The ccc 236

58 Lower bound for the continuum 240

59 Upper bound by nice names **243**

60 CH and its failure **248**

61 Homogeneity and relative consistency **252**

End-of-volume perspective **255**

X Volume X: Symmetric Extensions and Independence of Choice 256

Preface **257**

62 Fraenkel–Mostowski precursor **258**

63 Automorphisms of forcing and names **261**

64 Symmetric and hereditarily symmetric names **265**

65 ZF in a symmetric extension, elementary axioms **269**

66 ZF in a symmetric extension, hard axioms **273**

67 The basic Cohen symmetric model **278**

68 Failure of Choice **281**

End-of-volume perspective **285**

Glossary of recurring terms and notation **286**

How to read the series

The ten volumes form two large arcs. Volumes I–III develop logic, effective syntax, and incompleteness. Volumes IV–X develop set theory, constructibility, forcing, and the independence of CH and AC. The books are cumulative, but not every chapter has to be read with the same intensity on a first pass.

For a first route to Gödel's theorems, read Volume I through completeness, then the coding and provability chapters of Volume II, followed by all of Volume III. For a first route to Cohen's theorems, read the transfinite core of Volume IV, the satisfaction/absoluteness/hull chapters of Volume V, then Volumes VI–X in order.

Major theorem roadmap

The following landmarks indicate where the central proof dependencies close. Page numbers refer to the unified binder.

First-order completeness (p. 21)
: Henkin witnesses and the quotient term model convert consistency into satisfiability.

Downward Löwenheim–Skolem (p. 24)
: Skolem closure plus Tarski–Vaught gives elementary submodels of controlled size.

Diagonal lemma (p. 56)
: Represented substitution produces formal self-reference.

Gödel I (p. 62)
: The canonical Gödel sentence is undecided under the original consistency hypotheses.

Löb's theorem (p. 76) and Gödel II (p. 77)
: The derivability conditions turn arithmetized provability into the second incompleteness theorem.

Reflection (p. 129) and Skolem hulls (p. 132)
: These are the model-theoretic tools later used in constructibility.

$L \models \mathrm{ZF}$ **(p. 160)**
: Separation, Power Set, Collection, and Replacement are verified inside the constructible universe.

Condensation (p. 166) and $L \models \mathrm{GCH}$ **(p. 171)**
: Small elementary hulls collapse to earlier L-levels, yielding the cardinal localization needed for GCH.

Truth Lemma (p. 192) and Fundamental Theorem of Forcing (p. 193)
: Truth in a generic extension is matched with the ground-model forcing relation.

Formal consistency transfer (p. 219)
Boolean-valued/finite-fragment semantics supplies the genuine $\mathrm{Con}(T) \Rightarrow \mathrm{Con}(T + \varphi)$ bridge.

Cohen continuum calculation (p. 249)
ccc preservation, coordinate reals, and nice-name counting yield $2^{\aleph_0} = \aleph_2$ in the target extension.

Symmetric Power Set (p. 274) and failure of Choice (p. 282)
The symmetric submodel first satisfies ZF; only then does the finite-support argument produce $\neg$AC.

Conventions for hypotheses

Whenever a model M is assumed transitive, externally countable, or a model of Choice, that hypothesis is meant literally and is not inferred merely from consistency. Likewise, consistency, ω-consistency, Σ_1-soundness, and full soundness are distinct notions. Later chapters rely on these distinctions.

Notation and conventions

Throughout the series, structures are written with roman letters such as M, N, while theories are usually T, S. Semantic consequence is $T \models \varphi$ and formal derivability is $T \vdash \varphi$. The symbol $\mathrm{Con}(T)$ always denotes the canonical arithmetical consistency sentence attached to the proof predicate fixed in Volume II; it is not an informal metalinguistic abbreviation.

For set theory, Ord is the class of ordinals, V_α is the cumulative hierarchy, and L_α is the constructible hierarchy. Cardinal successors are written κ^+, always in the model currently under discussion unless a superscript such as κ^{+L} is displayed.

For forcing, the convention is that $q \leq p$ means that q is stronger than p. Every forcing notion used in Volumes VII–X has a largest (weakest) condition $1_{\mathbb{P}}$; when a given poset has no largest condition, one is adjoined before names are defined. A generic extension is $M[G]$; canonical ground-model names are $\check{x}$. In Boolean-valued semantics, $\|\varphi\|$ is the Boolean truth value of φ. In symmetric extensions, N denotes the symmetric submodel of the ambient generic extension.

Historical names are used only after the underlying assertion has been stated precisely. Source notes indicate provenance and useful comparison points; they are never substitutes for proofs in the text.

A five-question checklist for reading a dense formula

When a displayed line feels opaque, pause before manipulating it and answer these five questions.

1. **What is the type of each object?** Is it a formula, a natural-number code, a set, an ordinal, a forcing name, a condition, a Boolean value, or an actual object of a model?
2. **Which universe or theory is speaking?** Distinguish assertions inside T, truth in a structure M, truth in an extension $M[G]$, and statements in the surrounding metatheory.
3. **What operation occurs on this line?** Substitution, recursion, coding, valuation, forcing, collapse, taking a supremum, or cardinal counting should be named before it is used.
4. **What licenses the operation?** Look for the induction hypothesis, freshness condition, rank decrease, density, genericity, transitivity, chain condition, or consistency assumption.
5. **What is the local goal?** A long proof becomes easier when each displayed transformation is read as solving one small typed subproblem rather than as a mysterious global leap.

Part I

Volume I: Propositional and First-Order Logic

Preface

We begin with the formal distinction between syntax and semantics. Every recursive definition is made explicit, because later Gödel coding depends on the exact shape of syntax.

Position in the series

This opening volume supplies the proof-theoretic and model-theoretic language used everywhere else. Its completeness and Löwenheim–Skolem results are prerequisites for the metamathematical and set-theoretic model constructions that follow.

Sources and further reading

The presentation follows the standard Henkin route to completeness, with the term-model proof written so that equality and witness introduction are visible. For historical context see [1]; for a modern systematic treatment see [2].

Notation for this volume

$\models$ denotes semantic consequence and $\vdash$ formal derivability. Structures are denoted by M, N; assignments by s.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 1

Syntax and propositional semantics

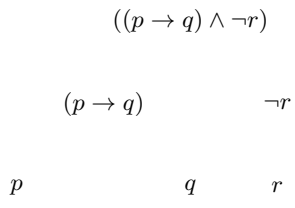

Figure 1.1: A parse tree makes structural recursion visible. A recursive definition evaluates the leaves first and then moves upward through the constructors.

Definitions and setup

Formal definitions used in this chapter

Propositional formula. The least set containing the propositional variables and closed under the chosen connectives.
Valuation. A function $v : \mathsf{PropVar} \to \{0, 1\}$, extended recursively to formulas.
Semantic consequence. $\Gamma \models \varphi$ means every valuation satisfying every member of Γ also satisfies φ.

A propositional language begins with a set of propositional variables and a fixed finite list of connectives. Formulas are the least strings generated by the formation rules; this "least" clause is what licenses structural induction. A valuation is a function from variables to $\{0, 1\}$, extended recursively to formulas. Semantic consequence $\Gamma \models \varphi$ means that every valuation satisfying all formulas in Γ also satisfies φ. Syntactic derivability has not yet been introduced, so throughout this chapter the words valid and consequence are purely semantic. The complexity of a formula is the height of its parse tree, and a literal is a variable or its negation. CNF means a conjunction of disjunctions of literals; DNF means a disjunction of conjunctions of literals.

How this chapter fits together

The first task is to make the recursive nature of formulas completely explicit. Unique readability is what turns an informal string of symbols into a tree on which induction and recursive definitions can safely operate. Once that foundation is in place, valuations become recursive maps on formulas, and normal forms become

constructive transformations rather than merely semantic slogans. Keep syntax and semantics separate: at this stage $\Gamma \models \varphi$ refers only to valuations, not to formal proofs.

Lemma 1.1 (Unique readability for propositional formulas)**.** *Let* $\mathsf{Form}_{\mathrm{prop}}$ *be the formulas generated from propositional variables by* $\neg, \wedge, \vee, \rightarrow$*. Every* $\varphi \in \mathsf{Form}_{\mathrm{prop}}$ *has a unique parse tree. Equivalently, if* φ *is non-atomic, then its outermost connective and its immediate constituent formula(s) are uniquely determined.*

Proof. Define the construction height $h(\varphi)$ recursively: variables have height 0, $h(\neg\psi) = h(\psi) + 1$, and $h(\psi \circ \chi) = 1 + \max(h(\psi), h(\chi))$ for a binary connective $\circ$. We prove uniqueness of the immediate constituents by induction on the length of the written formula. If φ is a variable, there is no decomposition. If the first symbol is $\neg$, the formation convention forces $\varphi = \neg\psi$, and the remaining well-formed substring ψ is unique by the induction hypothesis. If the outermost symbol is binary, the matching-parenthesis depth determines a unique split

$$\varphi = (\psi \circ \chi).$$

A second decomposition would have the same outer connective and the same split point; induction then gives the same ψ, χ. Thus the parse tree is unique. In particular $h(\varphi)$, the set of subformulas, and all recursive definitions on formulas are well defined. *Point specific to this result.* For propositional formulas, the decisive point is that the outermost connective and its matching subexpressions are recoverable from the written string. The induction therefore proves uniqueness of the parse itself, not merely existence of some parse. □

Theorem 1.2 (Structural induction and structural recursion on formulas)**.** *Let* P *be a property of propositional formulas. If* P *holds for every propositional variable and is preserved by each formation operation, then* $P(\varphi)$ *holds for every formula* φ*. Moreover, given values on variables and prescribed operations corresponding to the connectives, there is a unique function on formulas satisfying the associated recursive clauses.*

Proof. For structural induction, let S be the set of formulas satisfying P. By hypothesis every propositional variable lies in S, and S is closed under every formation operation. Since $\mathsf{Form}_{\mathrm{prop}}$ is the least set with those properties,

$$\mathsf{Form}_{\mathrm{prop}} \subseteq S,$$

so P holds for every formula. For recursion, prescribe a value a_p for each variable p, an operation N for negation, and operations $B_\circ$ for the binary connectives. Define F by induction on formula height:

$$F(p) = a_p, \qquad F(\neg\psi) = N(F(\psi)), \qquad F(\psi \circ \chi) = B_\circ(F(\psi), F(\chi)).$$

Unique readability guarantees that a non-atomic formula has only one outer constructor and therefore receives only one value. Induction on height proves that any other function satisfying the same clauses agrees with F, establishing uniqueness. *Point specific to this result.* The recursion part needs uniqueness of the immediate constituents: once the outer constructor is known, the value of the recursively defined function is forced by the already determined values on those constituents. This is why unique readability logically precedes structural recursion. □

Theorem 1.3 (Disjunctive and conjunctive normal-form theorems)**.** *For every propositional formula* φ *there exist a disjunctive normal form* D_φ *and a conjunctive normal form* C_φ*, built from literals using only* $\wedge$ *and* $\vee$*, such that* $\models \varphi \leftrightarrow D_\varphi$ *and* $\models \varphi \leftrightarrow C_\varphi$*.*

Proof. Let $p_1, \ldots, p_n$ be the variables occurring in φ. For each valuation $v \in 2^n$, define the minterm

$$m_v := \bigwedge_{i=1}^{n} \ell_i^v, \qquad \ell_i^v = \begin{cases} p_i, & v(p_i) = 1, \\ \neg p_i, & v(p_i) = 0. \end{cases}$$

Then m_v is true under exactly the valuation v. Hence

$$D_\varphi := \bigvee_{v \models \varphi} m_v$$

has the same truth table as φ, so $\models \varphi \leftrightarrow D_\varphi$. Dually, for each falsifying valuation let

$$c_v := \bigvee_{i=1}^{n} \neg \ell_i^v;$$

this clause is false exactly at v. Therefore

$$C_\varphi := \bigwedge_{v \not\models \varphi} c_v$$

is equivalent to φ. Empty disjunction/conjunction conventions cover contradictions and tautologies. Both constructions are finite because φ contains only finitely many variables. *Further explanation.* A useful way to separate the two normal forms is to notice what the distribution step is doing. DNF records the valuations on which the formula is true, one conjunction of literals per satisfying row; CNF records the valuations on which it is false, one disjunction excluding each falsifying row. Thus the semantic construction and the syntactic distribution construction lead to the same endpoint. This also explains why the theorem concerns finite formulas: only finitely many propositional variables occur in a fixed formula. □

Proposition 1.4 (Compact truth-table decision procedure for propositional consequence)**.** *For every finite set Γ of propositional formulas and every formula φ, there is a terminating algorithm that decides whether $\Gamma \models \varphi$: enumerate the finitely many valuations of the variables occurring in $\Gamma \cup \{\varphi\}$ and test whether any valuation satisfies $\Gamma \cup \{\neg\varphi\}$.*

Proof. Let V be the finite set of propositional variables occurring in $\Gamma \cup \{\varphi\}$. If $|V| = n$, there are exactly 2^n assignments $v : V \to \{0, 1\}$. For each assignment, recursively evaluate every formula using the truth functions of the connectives. The algorithm outputs "yes" iff

$$\forall v : V \to 2 \quad \Big[\big(\forall \gamma \in \Gamma\ v(\gamma) = 1\big) \Longrightarrow v(\varphi) = 1\Big].$$

Equivalently, it searches for a valuation satisfying $\Gamma \cup \{\neg\varphi\}$. Coincidence for propositional evaluation shows that extending v arbitrarily to variables outside V cannot alter the formulas being tested. Thus the finite search is exhaustive, terminates after 2^n rows, and decides $\Gamma \models \varphi$. *Further explanation.* Correctness depends on restricting attention to the variables that actually occur. Extending a valuation on $p_1, \ldots, p_n$ arbitrarily to all other propositional variables cannot change the truth value of any formula in $\Gamma \cup \{\varphi\}$. Hence the finite table is exhaustive for the consequence question at hand. The procedure therefore proves decidability of finite propositional consequence without making any appeal to the later completeness theorem. □

Worked example

Worked Example 1.5 (A truth table becomes an algorithm)**.** Consider $\varphi = (p \to q) \wedge (p \vee r)$. Eliminate the implication to obtain $(\neg p \vee q) \wedge (p \vee r)$, already a CNF. With variables p, q, r, only eight valuations need inspection. If a finite set Γ uses the same variables, semantic consequence is decided by checking precisely those rows on which every member of Γ is true. This small calculation is the finite prototype of later decision and coding arguments.

A useful warning

Do not infer proof-theoretic completeness from the existence of a truth-table procedure. Deciding whether a finite propositional consequence is valid is a semantic fact; proving that every valid consequence has a derivation is a separate theorem.

Looking ahead

The next chapter, *Propositional proof systems*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 2

Propositional proof systems

Definitions and setup

Formal definitions used in this chapter

Derivation. A finite sequence of formulas each of which is an axiom, an assumption, or follows from earlier lines by a rule of the fixed calculus.
Consistency. Γ is consistent if no contradiction is derivable from Γ.
Complete theory. Δ is complete if for every formula φ, either $\varphi \in \Delta$ or $\neg\varphi \in \Delta$.

Fix one Hilbert-style propositional calculus. A derivation from Γ is a finite sequence whose lines are logical axioms, members of Γ, or follow from earlier lines by modus ponens. Write $\Gamma \vdash \varphi$ if such a derivation ends in φ. A theory is syntactically consistent if no contradiction is derivable. It is complete if for every formula φ, either φ or $\neg\varphi$ is derivable. A maximal consistent theory is a consistent set with no proper consistent extension. These notions concern formal proofs and must not be confused with satisfiability, which is semantic.

How this chapter fits together

We now introduce a formal calculus and ask whether it matches the semantics just developed. Soundness runs from proofs to truth. Completeness runs in the opposite direction and is subtler: a consistent theory is extended to a maximal one, and that maximal theory itself determines a valuation. Compactness then falls out because formal proofs are finite objects. This chapter is the first place where the syntactic/semantic distinction does real work.

Theorem 2.1 (Soundness of the chosen propositional calculus)**.** *For every set Γ of propositional formulas and every formula φ, if $\Gamma \vdash_{\mathrm{Prop}} \varphi$, then $\Gamma \models \varphi$.*

Proof. We induct on the length of a derivation

$$\psi_0, \dots, \psi_m = \varphi$$

from Γ. If $\psi_i \in \Gamma$, every valuation satisfying Γ satisfies ψ_i. If ψ_i is an instance of a propositional axiom scheme, a finite truth-table check of that scheme shows $\models \psi_i$. The only inference rule is modus ponens. If earlier lines are α and $\alpha \to \beta$, the induction hypothesis gives, for every $v \models \Gamma$,

$$v(\alpha) = 1, \qquad v(\alpha \to \beta) = 1.$$

The truth table for implication then forces $v(\beta) = 1$. Thus each line of the derivation is true under every valuation satisfying Γ, and in particular

$$\Gamma \vdash_{\mathrm{Prop}} \varphi \Longrightarrow \Gamma \models \varphi.$$

No quantifier rule or first-order side condition occurs in this argument. *Point specific to this result.* Here the induction is on derivation length. The content is semantic preservation: axioms are tautologies and each rule sends truth-preserving premises to a truth-preserving conclusion. No maximal-consistency construction is used in this direction. □

Lemma 2.2 (Maximal consistent extension for propositional theories)**.** *If Γ is a consistent set of propositional formulas, then there exists a consistent set $\Delta \supseteq \Gamma$ such that for every propositional formula φ, exactly one of φ and $\neg\varphi$ belongs to Δ. Such a Δ is maximal consistent.*

Proof. Enumerate all propositional formulas as $\varphi_0, \varphi_1, \ldots$. Put $\Gamma_0 = \Gamma$ and recursively define

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\}, & \text{if this set is consistent,} \\ \Gamma_n \cup \{\neg\varphi_n\}, & \text{otherwise.} \end{cases}$$

At least one choice is consistent: if both extensions were inconsistent, the deduction theorem would give

$$\Gamma_n \vdash \neg\varphi_n \quad \text{and} \quad \Gamma_n \vdash \varphi_n,$$

contradicting consistency of Γ_n. Let $\Delta = \bigcup_n \Gamma_n$. A contradiction derived from Δ uses only finitely many assumptions and hence already occurs in some Γ_n, impossible. Thus Δ is consistent. Every formula is decided at its enumeration stage, so exactly one of $\varphi, \neg\varphi$ belongs to Δ. Any proper extension would add the opposite of a decided formula and become inconsistent; hence Δ is maximal consistent. *Point specific to this result.* The extension step uses the elementary dichotomy that, for consistent T, at least one of $T \cup \{\varphi\}$ and $T \cup \{\neg\varphi\}$ is consistent. Finiteness of proofs is then what makes the union of the stage-by-stage construction consistent. □

Theorem 2.3 (Completeness of propositional logic)**.** *For every set Γ of propositional formulas and every formula φ, $\Gamma \models \varphi$ implies $\Gamma \vdash_{\mathrm{Prop}} \varphi$. Hence syntactic and semantic consequence coincide for the chosen propositional calculus.*

How to read the symbols: $\Gamma \models \varphi$ versus $\Gamma \vdash \varphi$

The symbol $\models$ belongs to semantics: it quantifies over valuations. The symbol $\vdash$ belongs to syntax: it asserts the existence of a finite derivation. Completeness proves that the semantic condition implies the syntactic one. The proof does not transform a truth table into a derivation directly; instead it assumes $\Gamma \nvdash \varphi$, constructs a valuation satisfying $\Gamma \cup \{\neg\varphi\}$, and contradicts $\Gamma \models \varphi$.

Proof. Assume $\Gamma \models \varphi$ but $\Gamma \nvdash \varphi$. By the propositional deduction theorem,

$$\Gamma \cup \{\neg\varphi\}$$

is consistent. Extend it to a maximal consistent set Δ. Define a valuation on variables by

$$v(p) = 1 \iff p \in \Delta.$$

We prove the truth lemma

$$v(\psi) = 1 \iff \psi \in \Delta \tag{*}$$

by structural induction. For negation, maximal consistency gives $\neg\psi \in \Delta$ iff $\psi \notin \Delta$. For each binary connective, the corresponding propositional tautologies and closure of Δ under modus ponens give exactly the truth-table clause. Thus (*) holds. Since $\Gamma \subseteq \Delta$ and $\neg\varphi \in \Delta$, the valuation satisfies Γ and falsifies φ, contradicting $\Gamma \models \varphi$. Therefore $\Gamma \vdash \varphi$. *Further explanation.* The key point is the truth lemma for the valuation extracted from T^*. Maximal consistency gives exactly the Boolean closure properties needed by the induction: for instance, $\varphi \wedge \psi \in T^*$ iff both conjuncts belong to T^*, and $\neg\varphi \in T^*$ iff $\varphi \notin T^*$. Once that equivalence is established for every formula, the contradiction argument converts the semantic hypothesis $\Gamma \models \varphi$ into a formal derivation. Thus completeness is not an enumeration of proofs; it is a model construction from failure of derivability. □

Theorem 2.4 (Compactness of propositional logic)**.** *A set* Γ *of propositional formulas is satisfiable if and only if every finite subset* $\Gamma_0 \subseteq \Gamma$ *is satisfiable.*

Proof. Assume every finite subset of Γ is satisfiable. If Γ were inconsistent, there would be a finite formal derivation of a contradiction from Γ. Only finitely many assumptions $\gamma_1, \ldots, \gamma_n \in \Gamma$ occur in that derivation, so

$$\{\gamma_1, \ldots, \gamma_n\} \vdash \bot.$$

By soundness, this finite subset would be unsatisfiable, contrary to hypothesis. Hence Γ is consistent. Propositional completeness supplies a valuation v with

$$v \models \Gamma.$$

Conversely, if Γ has a satisfying valuation, the same valuation satisfies every finite subset. Thus

$$\Gamma \text{ satisfiable} \iff \forall \Gamma_0 \in [\Gamma]^{<\omega}\ \Gamma_0 \text{ satisfiable}.$$

The proof isolates the key compactness mechanism: formal derivations are finite. *Point specific to this result.*

The compactness conclusion is a corollary of proof finiteness plus completeness: if a whole theory produced a contradiction, one finite set of premises occurring in that proof would already do so. The argument is therefore different in direction from the maximal-consistency construction that establishes completeness. □

Worked example

Worked Example 2.5 (From a maximal theory to a valuation)**.** Let T^* be maximal consistent and define $v(p) = 1$ exactly when $p \in T^*$. If $\neg\psi \in T^*$, consistency prevents $\psi \in T^*$; maximality gives the converse. Similar closure facts handle $\wedge$ and $\vee$. An induction therefore proves $v(\psi) = 1$ iff $\psi \in T^*$. The abstract maximal set has become a concrete semantic object.

A useful warning

Maximal consistency is not the same as containing every formula. It means deciding every formula by exactly one of φ and $\neg\varphi$, while remaining closed under the deductive calculus.

Looking ahead

The next chapter, *First-order syntax, substitution, and equality*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 3

First-order syntax, substitution, and equality

Definitions and setup

Formal definitions used in this chapter

Term/formula. Terms and formulas are generated recursively from the symbols of a fixed first-order language $\mathcal{L}$.
Free variable. A variable occurrence is free when it is outside the scope of a quantifier binding that variable.
Capture-avoiding substitution. $\varphi[t/x]$ replaces free occurrences of x by t, first renaming bound variables when necessary so no free variable of t becomes bound.

A first-order signature consists of relation symbols, function symbols, and constant symbols with specified arities. Terms are generated from variables and function symbols. Atomic formulas are relation applications and equalities of terms; formulas are generated from atomic formulas by connectives and quantifiers. An occurrence of a variable is free unless it lies in the scope of a quantifier binding that variable. A term t is substitutable for x in φ when replacing free occurrences of x by t causes no variable of t to become captured. We use alpha-renaming of bound variables whenever necessary to meet this condition.

How this chapter fits together

First-order syntax introduces two complications absent propositionally: variables can be bound, and equality identifies terms semantically even when their strings differ. Capture-avoiding substitution must therefore be defined before it is used, with freshness conditions stated explicitly. Equality is then handled algebraically: provable equality becomes a congruence, so the eventual term model uses equivalence classes of terms rather than raw strings.

Lemma 3.1 (Unique readability for first-order terms and formulas)**.** *Fix a first-order language $\mathcal{L}$. Every $\mathcal{L}$-term and every $\mathcal{L}$-formula has a unique parse tree; in particular, the outermost constructor and immediate constituent term(s) or formula(s) are uniquely determined.*

Proof. Define terms and formulas simultaneously by their formation trees. For terms, variables and constants are atomic and $f(t_1, \dots, t_n)$ has immediate subterms $t_1, \dots, t_n$. For formulas, atomic formulas have the form $R(t_1, \dots, t_n)$ or $t = s$, Boolean connectives attach formula constituents, and $\forall x\, \psi$ or $\exists x\, \psi$ has immediate constituent ψ. We prove by induction on string length that the outermost syntactic category and constructor

are unique. Parenthesis depth determines the split points of argument lists and binary connectives; a leading quantifier determines its bound variable and scope. The induction hypothesis gives unique parses of all immediate constituents. Hence every term and formula has a unique syntax tree. This makes notions such as free occurrence, substitution, complexity, and subformula unambiguous. *Point specific to this result.*

First-order syntax adds terms, quantifiers, and variable binding, but the uniqueness argument still begins by recovering the outer constructor. Parentheses and arities determine the immediate term/formula constituents; bound-variable scope is part of that recovered parse. □

Theorem 3.2 (Structural recursion on terms and formulas)**.** *Any assignment of values to variables together with recursive clauses for the function, relation, connective, and quantifier constructors extends uniquely to recursively defined functions on all $\mathcal{L}$-terms and formulas, provided the clauses depend only on immediate constituents.*

Proof. Let target sets and operations be specified for each syntactic constructor. Define a function on terms by

$$F(x) = a_x, \qquad F(f(t_1, \ldots, t_n)) = F_f(F(t_1), \ldots, F(t_n)).$$

Unique readability makes this recursive clause unambiguous, and induction on term height proves existence and uniqueness. For formulas, prescribe values on atomic formulas and operations $N, B_\circ, Q_x$ corresponding to negation, binary connectives, and quantification; then set

$$G(\neg\psi) = N(G(\psi)), \quad G(\psi \circ \chi) = B_\circ(G(\psi), G(\chi)), \quad G(\forall x\psi) = Q_x(G(\psi)).$$

Again unique readability identifies the single applicable clause. Structural induction is the associated proof principle: a property holding for atomic syntax and preserved by every constructor holds for all terms/formulas. *Point specific to this result.* Because formulas contain terms and terms may occur inside atomic formulas, the clean formulation is a mutual recursion on the two syntactic classes. The recursion is legitimate only after unique readability guarantees a unique decomposition at each step. □

Lemma 3.3 (Capture-avoiding substitution and substitution composition)**.** *Let $x \neq y$, let s, t be terms, and let φ be a formula. Before substitution, alpha-rename bound variables of φ so that no bound variable occurring on a substitution path belongs to* $\mathrm{FV}(s) \cup \mathrm{FV}(t)$*. Then capture-avoiding substitution is well defined and*

$$\varphi[s/x][t/y] = \varphi[s[t/y]/x,\, t/y],$$

where the right-hand side is simultaneous substitution: free occurrences of x are replaced by $s[t/y]$, free occurrences of y are replaced by t, and no substitution is performed inside the replacement terms after they are inserted.

Commented symbolic trace: capture-avoiding substitution

Read $\varphi[s/x][t/y]$ from left to right. First replace free occurrences of x by s; only after that replacement do we substitute t for the free occurrences of y. A bound variable may have to be renamed before either step so that a free variable of the inserted term does not become accidentally bound. That renaming is not cosmetic: it preserves the set of free variables and therefore preserves the intended semantics.

Proof. Define substitution on terms by structural recursion. For formulas, recurse through the Boolean connectives. At a quantified formula $Qz\,\psi$, where $Q \in \{\forall, \exists\}$, first alpha-rename z when necessary so that

$$z \notin \mathrm{FV}(s) \cup \mathrm{FV}(t) \cup \{x, y\}.$$

After this renaming, recursion under the quantifier cannot capture a variable that was free in either replacement term. Thus the capture-avoiding operation is defined at every node of the syntax tree.

For the composition formula, argue by structural induction on φ. The variable case exhibits the three possibilities. If the variable is x, the left side is

$$x[s/x][t/y] = s[t/y],$$

which is exactly the simultaneous replacement assigned to x. If it is y, then

$$y[s/x][t/y] = t,$$

and any other variable is unchanged. Function and relation symbols reduce componentwise to the term induction hypothesis. Boolean connectives inherit the equality from their immediate subformulas.

For a quantified formula $Qz\,\psi$, use the alpha-renamed representative with z fresh as displayed above. Both sides leave the binder Qz fixed and perform the two substitutions only in ψ. The induction hypothesis gives

$$\psi[s/x][t/y] = \psi[s[t/y]/x,\, t/y],$$

so prefixing the same quantifier yields the desired equality for $Qz\psi$. Alpha-equivalent preliminary renamings lead to alpha-equivalent outputs, so the construction is independent of the particular fresh name chosen. This proves both well-definedness and the stated composition law without an unspecified "freshness condition." □

Theorem 3.4 (Equality congruence and quotient-term construction)**.** *Let T be a theory with equality and let $\sim$ be the relation on closed terms defined by $s \sim t \iff T \vdash s = t$. Then $\sim$ is an equivalence relation and a congruence for every function and relation symbol. Consequently the quotient of the closed-term algebra by $\sim$ carries a well-defined $\mathcal{L}$-structure.*

Proof. On closed terms define

$$s \sim t \Longleftrightarrow T \vdash s = t.$$

Equality axioms give reflexivity, symmetry, and transitivity, so $\sim$ is an equivalence relation. Substitutivity gives congruence: if $s_i \sim t_i$ for $1 \le i \le n$, then

$$T \vdash f(s_1, \dots, s_n) = f(t_1, \dots, t_n),$$

and for a relation symbol R,

$$T \vdash R(\bar{s}) \leftrightarrow R(\bar{t}).$$

Hence functions and relations are well defined on equivalence classes $[t]$. The quotient of closed terms by $\sim$ is therefore a legitimate first-order structure. This quotient is essential in the Henkin model: syntactically different terms that T proves equal represent one element rather than two. *Further explanation.* There are two well-definedness checks. First, if $s_i \sim t_i$, substitutivity proves that a function symbol has the same equivalence class whether it is applied to the s_i's or the t_i's. Second, the same argument for relation symbols shows that truth of an atomic relation depends only on the equivalence classes of its arguments. These checks are what make the quotient a genuine structure rather than merely a set of equivalence classes. □

Worked example

Worked Example 3.5 (Why capture avoidance is necessary)**.** In $\forall y\, R(x, y)$, substituting the term y naively for x gives $\forall y\, R(y, y)$, where the formerly free y has been captured. Rename the bound variable first: $\forall z\, R(x, z)$, then substitute to obtain $\forall z\, R(y, z)$. The two results have very different meanings; freshness is therefore a mathematical condition, not typographical etiquette.

A useful warning

Never identify two closed terms merely because the intended interpretation makes them equal. The term model is built from syntax, so equality must first be shown to be a provable congruence and then quotiented.

Looking ahead

The next chapter, *Structures and satisfaction*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 4

Structures and satisfaction

Definitions and setup

Formal definitions used in this chapter

Structure. An $\mathcal{L}$-structure M consists of a nonempty domain together with interpretations of the nonlogical symbols.
Assignment. A map from variables to M.
Satisfaction. The relation $M, a \models \varphi$ is defined recursively by the interpretations of atomic formulas, truth-functional clauses, and quantifier clauses.

An L-structure M consists of a nonempty domain together with interpretations of every symbol in the signature. A variable assignment s maps variables to elements of M. Terms receive values $t^M[s]$ by recursion. Satisfaction $M, s \models \varphi$ is then defined recursively; for a sentence the assignment is irrelevant and we write $M \models \varphi$. An embedding preserves functions, relations, and constants; an isomorphism is a bijective embedding whose inverse is also an embedding. A reduct forgets interpretations of symbols, while an expansion adds them.

How this chapter fits together

Satisfaction is a recursion on formulas, but the recursion interacts with assignments. The substitution theorem connects syntactic replacement with semantic reassignment; the coincidence lemma says truth depends only on values of free variables. These two facts make the quantifier clauses manageable and are reused whenever formulas are transported between structures, reducts, expansions, and isomorphic copies.

Theorem 4.1 (Term-evaluation substitution theorem)**.** *For every $\mathcal{L}$-structure M, assignment a, term t, term s, and variable x,*

$$(t[s/x])^M[a] = t^M\big[a\big(x := s^M[a]\big)\big].$$

The analogous substitution lemma holds for formulas:

$$M, a \models \varphi[s/x] \iff M, a(x := s^M[a]) \models \varphi,$$

whenever the substitution is capture avoiding.

Proof. We prove by induction on the term t that for every structure M and assignment a,

$$(t[s/x])^M[a] = t^M[a(x \mapsto s^M[a])]. \tag{1}$$

For a variable this is immediate from the two assignment cases; for $f(t_1, \dots, t_n)$, apply the induction hypothesis to each argument and then the interpretation f^M. The formula substitution theorem follows by structural induction on φ:

$$M, a \models \varphi[s/x] \iff M, a(x \mapsto s^M[a]) \models \varphi. \tag{2}$$

Atomic formulas use (1), Boolean connectives use the induction hypothesis, and quantifiers use capture avoidance to ensure the modified assignments agree on the correct free variables. Equations (1)–(2) are the semantic correctness theorem for syntactic substitution. *Point specific to this result.* The term case is the engine of the theorem: evaluating $t[s/x]$ under an assignment equals evaluating t after changing the value of x to the value of s. The formula substitution lemma later inherits this identity in its atomic case. □

Lemma 4.2 (Coincidence lemma)**.** *If assignments a, b agree on every variable free in a term t, then $t^M[a] = t^M[b]$. If they agree on every variable free in a formula φ, then*

$$M, a \models \varphi \iff M, b \models \varphi.$$

Proof. Let a, b be assignments in a structure M. First prove by induction on terms that if a and b agree on all variables occurring in t, then

$$t^M[a] = t^M[b].$$

For formulas, prove by structural induction that agreement on the free variables of φ implies

$$M, a \models \varphi \iff M, b \models \varphi.$$

Atomic formulas use the term result. Boolean cases are direct. If $\varphi = \exists x \psi$, any witness assignment $a(x \mapsto m)$ and $b(x \mapsto m)$ still agree on the free variables of ψ; the induction hypothesis transfers truth of ψ. Thus bound-variable assignments and variables absent from the formula cannot affect satisfaction. This is the coincidence lemma. *Point specific to this result.* For coincidence, the induction tracks free variables rather than substitution. At the quantifier step, assignments may differ at the newly bound variable; the induction hypothesis applies because every other free coordinate still agrees. □

Theorem 4.3 (Isomorphism invariance of satisfaction)**.** *If $h : M \cong N$ is an isomorphism of $\mathcal{L}$-structures, then for every formula $\varphi(\bar{x})$ and every tuple $\bar{a} \in M$,*

$$M \models \varphi(\bar{a}) \iff N \models \varphi(h\bar{a}).$$

Proof. Let $h : M \to N$ be an isomorphism. Extend it to assignments by $(h \circ a)(x) = h(a(x))$. Induct on terms to prove

$$h(t^M[a]) = t^N[h \circ a]. \tag{1}$$

For a function application, (1) follows from preservation of the function interpretation by h. Now induct on formulas. For an atomic relation,

$$M \models R(t_1, \dots, t_n)[a] \iff N \models R(t_1, \dots, t_n)[h \circ a]$$

by (1) and preservation/reflection of R. Equality is similar; Boolean cases follow immediately. For $\exists x \psi$, a witness $m \in M$ is carried to $h(m) \in N$, and surjectivity of h gives the converse. Therefore

$$M, a \models \varphi \iff N, h \circ a \models \varphi.$$

Point specific to this result. The key extra statement is the term identity $h(t^M(\bar{a})) = t^N(h\bar{a})$. Once proved by term induction, it transports atomic truth; the formula induction then shows that quantification is preserved because an isomorphism is bijective. □

Proposition 4.4 (Expansion/reduct compatibility of satisfaction)**.** *Let $\mathcal{L}_0 \subseteq \mathcal{L}$, let M be an $\mathcal{L}$-structure, and let $M \restriction \mathcal{L}_0$ be its reduct. For every $\mathcal{L}_0$-formula φ and assignment a,*

$$M, a \models \varphi \iff M \restriction \mathcal{L}_0, a \models \varphi.$$

Equivalently, expanding a structure does not change the truth of formulas in the old language.

Proof. Suppose $L \subseteq L'$, M' is an L'-structure, and $M = M' \restriction L$ is its reduct. We prove by induction on L-terms that

$$t^{M'}[a] = t^{M}[a],$$

because every symbol occurring in t has the same interpretation in the reduct. Induction on L-formulas then gives

$$M', a \models \varphi \iff M, a \models \varphi.$$

Atomic cases use the term identity; Boolean and quantifier clauses quantify over the same underlying set. Conversely, an expansion may choose arbitrary interpretations only for symbols in $L' \setminus L$; it cannot change the truth of an L-formula. Thus satisfaction of formulas in the smaller signature is invariant under expansion/reduct. *Point specific to this result.* Only symbols occurring in the smaller language can affect the value of its terms or formulas. Thus an expansion and its reduct give identical interpretations to every symbol visible to the formula, and the satisfaction recursion yields the same truth value. □

Worked example

Worked Example 4.5 (Substitution as reassignment)**.** Let $t(x) = f(x)$ and let s assign a to x. If a term u has value b under s, then evaluating $t[u/x]$ under s gives $f^M(b)$. The substitution lemma says the same value is obtained by evaluating t under the assignment $s[x \mapsto b]$. The formula version is the same principle with truth values instead of term values.

A useful warning

The quantifier step always changes one assignment coordinate. Omitting that bookkeeping is the most common source of false substitution and isomorphism arguments.

Looking ahead

The next chapter, *Formal deduction*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 5

Formal deduction

Definitions and setup

Formal definitions used in this chapter

Syntactic consequence. $\Gamma \vdash \varphi$ means there is a finite derivation of φ from assumptions in Γ.
Semantic consequence. $\Gamma \models \varphi$ means every structure satisfying Γ satisfies φ. Soundness compares these two notions.

The first-order calculus extends the propositional calculus by quantifier axioms/rules and equality axioms. We fix the eigenvariable side conditions explicitly: generalization may not bind a variable that is free in an undischarged assumption when the deduction theorem is being used. A theory is simply a set of sentences in the fixed language. Consistency, completeness, and derivability are syntactic notions. The deduction theorem is a transformation of finite derivations; it does not assert semantic implication until soundness has been proved.

How this chapter fits together

The formal calculus now has to support the later completeness construction. Soundness is proved line by line, while the deduction theorem converts a derivation with an extra assumption into an implication. Consistency extension and Lindenbaum's lemma then prepare a theory that decides every sentence. Side conditions on generalization are essential because a free variable in an undischarged assumption cannot be varied arbitrarily.

Theorem 5.1 (Soundness of first-order deduction)**.** *For every first-order theory Γ and sentence φ, if $\Gamma \vdash \varphi$ in the chosen first-order calculus, then every $\mathcal{L}$-structure satisfying Γ also satisfies φ; that is, $\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi$.*

Proof. Induct on the length of a first-order derivation from Γ. Logical axiom schemes are valid under every assignment by the semantic clauses; equality axioms are valid because equality in a structure is actual identity. Assumptions in Γ hold in every model of Γ. Modus ponens preserves truth. For generalization, suppose the rule passes from ψ to $\forall x \psi$ and the calculus requires that x not occur free in any undischarged assumption. If $M, a \models \Gamma$, then for every $m \in M$, the coincidence lemma gives

$$M, a(x \mapsto m) \models \Gamma.$$

By the induction hypothesis,

$$M, a(x \mapsto m) \models \psi.$$

Since m was arbitrary, $M, a \models \forall x \psi$. Hence

$$\Gamma \vdash \varphi \Longrightarrow \Gamma \models \varphi.$$

Point specific to this result. The quantifier rules are the only new semantic issue beyond propositional soundness. Their eigenvariable and free-variable side conditions ensure that varying a quantified coordinate cannot change an undischarged assumption, which is exactly what the semantic proof requires. □

Theorem 5.2 (Deduction theorem in the chosen calculus)**.** *For sentences φ, ψ and theory Γ, subject to the eigenvariable restrictions of the chosen calculus,*

$$\Gamma \cup \{\varphi\} \vdash \psi \quad \Longleftrightarrow \quad \Gamma \vdash \varphi \to \psi.$$

Proof. We prove the nontrivial direction by induction on a derivation of ψ from $\Gamma \cup \{\varphi\}$. If a line α is an axiom or belongs to Γ, propositional tautologies give

$$\Gamma \vdash \varphi \to \alpha.$$

If $\alpha = \varphi$, derive $\varphi \to \varphi$. If β follows from α and $\alpha \to \beta$ by modus ponens, combine the induction hypotheses using

$$(\varphi \to (\alpha \to \beta)) \to ((\varphi \to \alpha) \to (\varphi \to \beta)).$$

For generalization, the eigenvariable restriction guarantees that the quantified variable is not free in φ or the undischarged assumptions, so the quantifier axiom lifts the induction hypothesis. Thus

$$\Gamma \cup \{\varphi\} \vdash \psi \Longrightarrow \Gamma \vdash \varphi \to \psi.$$

The converse is one application of modus ponens. *Further explanation.* The generalization case is the only place where a careless formulation fails. If x occurs free in the assumption φ, then from a derivation of ψ under that assumption one cannot in general infer $\forall x\, \psi$ while keeping the assumption fixed. The side condition prevents exactly this change of meaning. With it in place, the induction converts every line of the original derivation into a theorem of the form $\varphi \to \chi$, so the final line is $T \vdash \varphi \to \psi$. □

Lemma 5.3 (Consistency extension lemma)**.** *If Γ is consistent and φ is any sentence, then at least one of $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ is consistent. In particular, if $\Gamma \nvdash \varphi$, then $\Gamma \cup \{\neg\varphi\}$ is consistent.*

Proof. Let T be consistent and let σ be a sentence not decided by T. If both

$$T \cup \{\sigma\} \quad \text{and} \quad T \cup \{\neg\sigma\}$$

were inconsistent, the deduction theorem would give

$$T \vdash \neg\sigma \quad \text{and} \quad T \vdash \neg\neg\sigma,$$

and classical logic yields $T \vdash \sigma$, contradicting consistency. Hence at least one of the two one-sentence extensions is consistent. Repeating this decision step along an enumeration of formulas and taking unions at limit stages yields consistent extensions. The finiteness of proofs is what preserves consistency at unions: any contradiction in the union appears at a finite stage. This lemma is the local consistency fact used by the Lindenbaum construction. *Further explanation.* For the extension step, suppose both $T \cup \{\varphi\}$ and $T \cup \{\neg\varphi\}$ were inconsistent. By the deduction theorem the first would give $T \vdash \neg\varphi$ and the second $T \vdash \neg\neg\varphi$, hence T itself would be inconsistent. Therefore at least one of the two one-sentence extensions preserves consistency. This elementary dichotomy is the engine behind the Lindenbaum construction and later Henkin completions. □

Theorem 5.4 (Lindenbaum maximal-consistency theorem)**.** *Every consistent first-order theory Γ in a countable language has a complete consistent extension $\Delta \supseteq \Gamma$: for every sentence φ, exactly one of φ and $\neg\varphi$ belongs to Δ.*

Proof. Enumerate all sentences as $\sigma_0, \sigma_1, \ldots$. Starting from consistent $T_0 = T$, use the consistency-extension lemma to choose

$$T_{n+1} = T_n \cup \{\sigma_n\} \quad \text{or} \quad T_n \cup \{\neg\sigma_n\}$$

so that T_{n+1} remains consistent. Set

$$T^* := \bigcup_{n<\omega} T_n.$$

A contradiction from T^* uses finitely many assumptions and hence occurs in some T_n, impossible. Every sentence is decided when it is enumerated, so T^* is complete. If $S \supsetneq T^*$ were consistent, choose $\sigma \in S \setminus T^*$; completeness gives $\neg\sigma \in T^* \subseteq S$, contradiction. Thus T^* is maximal consistent. *Point specific to this result.*

At each enumeration stage we decide one formula while preserving consistency. Completeness of the resulting theory means syntactic decision of every sentence, not semantic completeness of the proof system; that latter theorem comes only after the Henkin model is built. □

Worked example

Worked Example 5.5 (A side condition that matters)**.** From the assumption $P(x)$ one may not in general infer $\forall x\, P(x)$: the assumption may be true only under the present assignment to x. This is why generalization requires the quantified variable not to occur free in the undischarged assumptions. The deduction theorem preserves precisely this restriction.

A useful warning

When a proof uses generalization, check the eigenvariable condition before applying the semantic argument. Dropping it makes the formal system unsound.

Looking ahead

The next chapter, *Henkin completeness*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 6

Henkin completeness

Definitions and setup

Formal definitions used in this chapter

Henkin witness property. A theory has the witness property when every provable existential sentence $\exists x \varphi(x)$ has a closed term t with $\varphi(t)$ in the theory.
Term model. Its elements are equivalence classes of closed terms modulo provable equality, with symbols interpreted by term formation.

A Henkin theory has enough names for existential witnesses: for every formula $\exists x\, \varphi(x)$ there is a constant c such that the theory contains $\exists x\, \varphi(x) \to \varphi(c)$. A complete Henkin theory is also syntactically complete. Its term model has equivalence classes of closed terms as elements, with two terms identified exactly when the theory proves them equal. The central truth lemma says that semantic truth in this term model agrees with membership/provability in the complete Henkin theory.

How this chapter fits together

Completeness is proved by manufacturing a model from a consistent theory. Henkin constants provide witnesses for existential formulas; maximal consistency decides every sentence; provable equality is quotiented; and the truth lemma identifies semantic truth in the term model with membership in the complete Henkin theory. The proof should be read as a construction with four stages: witnesses, completion, quotient, and truth lemma.

Lemma 6.1 (Henkin witness extension preserves consistency)**.** *Let T be a consistent theory in a language $\mathcal{L}$, let $\varphi(x)$ be an $\mathcal{L}$-formula, and let $c \notin \mathcal{L}$ be a new constant symbol. Then*

$$T \cup \{\exists x\, \varphi(x) \to \varphi(c)\}$$

is consistent. Iterating the construction with a fresh constant for each enumerated existential formula preserves consistency at every finite stage and at the union stage.

Proof. Let T be consistent, let $\exists x \varphi(x)$ be a sentence, and let c be a new constant. Add the Henkin axiom

$$H: \quad \exists x \varphi(x) \to \varphi(c).$$

Suppose $T \cup \{H\}$ were inconsistent. By the deduction theorem,

$$T \vdash \neg H.$$

Classical propositional reasoning gives

$$T \vdash \exists x \varphi(x) \quad \text{and} \quad T \vdash \neg\varphi(c).$$

Because c does not occur in T or φ, replace it by a fresh variable and generalize to obtain

$$T \vdash \forall x \neg\varphi(x),$$

contradicting the existential theorem. Therefore adjoining one fresh witness axiom preserves consistency. Iterating this construction over an enumeration preserves consistency because every contradiction is finite. *Further explanation.* Freshness of c is indispensable. It lets us replace the new constant by a variable after deriving $\neg\varphi(c)$, because no previous axiom or assumption can depend on that constant. The contradiction therefore returns to the old language and conflicts with $\exists x\, \varphi(x)$. Repeating the step over an enumeration of existential formulas is safe because any finite contradiction in the union would already appear at a finite stage. □

Proposition 6.2 (Construction of a complete Henkin theory)**.** *Every consistent theory T in a countable language has a consistent extension T^H in a countable expansion of the language such that (i) T^H is complete, and (ii) for every sentence $\exists x\, \varphi(x)$, if $T^H \vdash \exists x\, \varphi(x)$, then $T^H \vdash \varphi(c)$ for some closed Henkin constant c.*

Proof. First expand the language by countably many fresh constants and enumerate all existential formulas. Repeatedly apply the witness-consistency lemma, obtaining a consistent Henkin theory T_H such that

$$T_H \vdash \exists x \varphi(x) \Longrightarrow T_H \vdash \varphi(c)$$

for an assigned witness constant c. Next enumerate all sentences of the expanded language and apply the Lindenbaum construction to obtain a complete consistent extension $T^* \supseteq T_H$. Witness axioms already in T_H remain in T^*, so the Henkin property is preserved. Hence T^* is simultaneously complete, consistent, and Henkin. The two stages are logically separate: witness constants guarantee existential realization; maximal completion guarantees every sentence or its negation is available for the truth lemma. *Further explanation.*

The two requirements–deciding every sentence and supplying witnesses–must be interleaved rather than performed independently. A convenient enumeration lists, at each stage, either a sentence to be decided or an existential formula whose witness axiom has not yet been added. The previous consistency lemmas handle the two cases. Because every requirement appears eventually, the union is simultaneously complete and Henkin, which is exactly the combination needed by the term-model truth lemma. □

Lemma 6.3 (Truth lemma for the Henkin term model)**.** *Let T^H be complete, consistent, and Henkin, and let M_{T^H} be its quotient term model. For every formula $\varphi(\bar{x})$ and closed terms $\bar{t}$,*

$$M_{T^H} \models \varphi([\bar{t}]) \quad \Longleftrightarrow \quad T^H \vdash \varphi(\bar{t}).$$

How to read the symbols: the Henkin truth lemma

The left side $M_T \models \varphi([\bar{t}])$ is a semantic statement about the term model. The right side $T^* \vdash \varphi(\bar{t})$ is a syntactic statement about the complete Henkin theory. The brackets $[t]$ matter because the model element is an equivalence class of closed terms modulo provable equality, whereas t itself is a piece of syntax. The induction proves that these two levels agree formula by formula.

Proof. Let the universe of the term model M_T be equivalence classes $[t]$ of closed terms modulo

$$s \sim t \iff T^* \vdash s = t.$$

Interpret functions by term formation and relations by

$$M_T \models R([t_1], \ldots, [t_n]) \iff R(t_1, \ldots, t_n) \in T^*.$$

Equality congruence makes these definitions independent of representatives. We prove by induction on φ the truth lemma

$$M_T \models \varphi([\bar{t}]) \iff \varphi(\bar{t}) \in T^*.$$

Atomic cases are definitions; Boolean cases use completeness and consistency of T^*. For $\exists x\psi$, if the left side holds, some closed term witnesses it and induction gives the corresponding instance in T^*. Conversely, if $\exists x\psi \in T^*$, the Henkin property supplies a witness constant c with $\psi(c) \in T^*$, and induction gives a model witness. *Further explanation.* The existential step is the reason for having constructed a Henkin theory. If the model satisfies $\exists x\, \psi(x, \bar{t})$, some closed term class $[s]$ witnesses it, and the induction hypothesis yields $T^* \vdash \psi(s, \bar{t})$. Conversely, if $T^* \vdash \exists x\, \psi(x, \bar{t})$, the witness property gives a designated closed term c with $T^* \vdash \psi(c, \bar{t})$. The induction hypothesis then turns that derivation back into a model witness. All other connectives follow from maximal consistency. □

Theorem 6.4 (Gödel completeness theorem for first-order logic with equality)**.** *For every first-order theory Γ with equality and every sentence φ,*

$$\Gamma \models \varphi \quad \iff \quad \Gamma \vdash \varphi.$$

Equivalently, a first-order theory is syntactically consistent if and only if it has a model.

Commented symbolic trace: the completeness contradiction

The proof has three typed steps:

1. $T \nvdash \varphi$ is syntactic information.
2. By the deduction theorem, this makes $T \cup \{\neg\varphi\}$ syntactically consistent.
3. The Henkin construction converts that consistency into a semantic model of $T \cup \{\neg\varphi\}$, contradicting the semantic assumption $T \models \varphi$.

Keeping these types separate prevents the circular mistake of using completeness in order to prove completeness.

Proof. Assume $\Gamma \models \varphi$ and suppose $\Gamma \nvdash \varphi$. By the deduction theorem,

$$\Gamma \cup \{\neg\varphi\}$$

is consistent. Extend this theory to a complete Henkin theory T^*, and form its equality-quotiented term model M_{T^*}. The Henkin truth lemma gives

$$M_{T^*} \models T^*.$$

In particular,

$$M_{T^*} \models \Gamma \quad \text{and} \quad M_{T^*} \models \neg\varphi,$$

contradicting $\Gamma \models \varphi$. Therefore $\Gamma \vdash \varphi$. Combining this with soundness yields

$$\Gamma \vdash \varphi \iff \Gamma \models \varphi.$$

The equality axioms are already built into the quotient term model, so no separate equality-completeness argument is hidden. *Further explanation.* This proof is most transparent in contrapositive form. Failure of derivability says that adding $\neg\varphi$ does not create a contradiction; Henkinization converts that syntactic consistency into a concrete model of $T \cup \{\neg\varphi\}$. Such a model is precisely a counterexample to semantic consequence. Equality causes no additional gap because the term model was already quotiented by provable equality, so the equality axioms are represented by genuine identity in the constructed structure. □

Remark 6.5 (Source note)**.** Gödel proved first-order completeness in 1930; the witness-constant proof used here follows the later Henkin organization. Compare [3, 1, 2].

Worked example

Worked Example 6.6 (A tiny Henkin witness)**.** Suppose the language contains a unary predicate P and the theory contains $\exists x\, P(x)$. Add a fresh constant c together with $\exists x\, P(x) \to P(c)$. In any complete Henkin extension containing the existential sentence, $P(c)$ is present. The class of c in the term model is therefore an actual witness. The general construction simply performs this operation for every existential formula.

A useful warning

A term model with equality cannot use raw closed terms. Without quotienting by provable equality, the equality axioms may be true syntactically but false in the constructed structure.

Looking ahead

The next chapter, *Compactness and Löwenheim–Skolem*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 7

Compactness and Löwenheim–Skolem

Definitions and setup

Formal definitions used in this chapter

Diagram. The atomic diagram of M records all atomic and negated atomic facts about constants naming elements of M; the elementary diagram records all first-order truths with such parameters.
Elementary substructure. $N \preccurlyeq M$ means $N \subseteq M$ and every formula with parameters from N has the same truth value in N and M.

Compactness states that a first-order theory has a model exactly when every finite subtheory has a model. The diagram of a structure is obtained by adding a constant name for each element and recording the atomic facts true of those names; the elementary diagram records all first-order facts with parameters. An elementary substructure $A \preccurlyeq M$ is a substructure in which every formula with parameters from A has the same truth value as in M. Löwenheim–Skolem theorems control the possible cardinalities of elementary models; they do not say that an externally small model regards all of its sets as small.

How this chapter fits together

Completeness immediately yields compactness because a contradiction has a finite proof. Compactness, diagrams, and Skolem hulls then control model size. Downward Löwenheim–Skolem constructs small elementary substructures; upward Löwenheim–Skolem uses compactness to force many distinct elements. The resulting non-categoricity phenomena are a feature of first-order expressiveness rather than a defect in the proofs.

Theorem 7.1 (First-order compactness theorem)**.** *A first-order theory Γ is satisfiable if and only if every finite subset $\Gamma_0 \subseteq \Gamma$ is satisfiable.*

Proof. Suppose every finite subset of a first-order theory T is satisfiable. If T were inconsistent, a finite formal proof of contradiction would use only finitely many sentences $T_0 \subseteq T$, giving

$$T_0 \vdash \bot.$$

By soundness, T_0 would be unsatisfiable, contrary to hypothesis. Thus T is consistent. By the first-order Completeness Theorem, T has a model. The converse is immediate: a model of T is also a model of every finite subset. Hence

$$T \text{ satisfiable} \iff \forall T_0 \in [T]^{<\omega}\ T_0 \text{ satisfiable}.$$

Unlike propositional compactness, the conclusion may concern an infinite structure and cannot be obtained by a finite truth table; completeness supplies the model. *Point specific to this result.* The first-order theorem is obtained only after Henkin completeness is available. A contradictory derivation from an infinite theory uses finitely many assumptions, so finite satisfiability implies syntactic consistency; completeness then supplies a model of the entire theory. □

Remark 7.2 (Source note)**.** Compactness is presented here as a corollary of completeness plus finiteness of formal derivations; see [2].

Lemma 7.3 (Diagram lemma)**.** *Let M be an $\mathcal{L}$-structure. If an $\mathcal{L}(M)$-structure N satisfies the atomic diagram of M, then the map $a \mapsto c_a^N$ is an embedding $M \hookrightarrow N$. If N satisfies the elementary diagram of M, this embedding is elementary.*

Proof. Expand the language by a constant c_a for each $a \in M$. The atomic diagram $\mathrm{Diag}(M)$ contains every atomic or negated atomic sentence with these constants that is true in M. If an expanded structure N satisfies the diagram, define

$$e : M \to N, \qquad e(a) = c_a^N.$$

The equality sentences in the diagram make e injective. Function and relation sentences show that e preserves all symbols, so it is an embedding. If instead N satisfies the elementary diagram, which contains every first-order sentence $\varphi(c_{a_1}, \dots, c_{a_n})$ true in M, then

$$M \models \varphi(\bar{a}) \iff N \models \varphi(e\bar{a}),$$

so e is elementary. Conversely any (elementary) embedding makes the corresponding diagram true. *Further explanation.* For the elementary diagram, the converse direction is equally important. If $j : M \to N$ is elementary and N interprets each new constant c_a as $j(a)$, then every sentence in the elementary diagram is true in N. Thus realizing the elementary diagram and admitting an elementary embedding are equivalent descriptions of the same situation. The atomic diagram gives only an embedding because it records only the atomic structure. □

Theorem 7.4 (Downward Löwenheim–Skolem theorem)**.** *Let M be an infinite $\mathcal{L}$-structure and $A \subseteq M$. Then there is an elementary substructure $N \preccurlyeq M$ with $A \subseteq N$ and*

$$|N| \leq \max\{|A|, |\mathcal{L}|, \aleph_0\}.$$

How to read the symbols: what the Skolem hull is doing

The hull construction is a closure algorithm. Start with the parameters you insist on keeping. Repeatedly apply every basic function symbol and every chosen Skolem function. The union after ω stages is small because each stage adds only values of finitely many arguments, and it is elementary because every existential statement true in the large structure has a chosen witness produced by a Skolem function inside the hull.

Proof. Expand the language by Skolem functions, one for every existential formula. Starting from the prescribed set $A \subseteq M$, define

$$H_0 = A, \qquad H_{n+1} = H_n \cup \{f^M(\bar{a}) : f \text{ a language/Skolem function}, \ \bar{a} \in H_n^{<\omega}\},$$

and $H = \bigcup_{n<\omega} H_n$. If $|L| = \lambda$, then

$$|H| \leq \max(|A|, \lambda, \aleph_0).$$

By construction, whenever

$$M \models \exists x \varphi(x, \bar{a}) \quad (\bar{a} \in H),$$

the chosen Skolem function produces a witness in H. Tarski–Vaught therefore gives

$$H \preccurlyeq M.$$

Taking the reduct to the original language yields the required elementary substructure of the stated cardinal bound. *Further explanation.* The cardinal estimate comes from the closure process itself. At each finite stage there are at most $\max(|A|, |L|, \aleph_0)$ tuples to which the countably/ $|L|$-many Skolem functions can be applied, so that bound is preserved. Taking the union of ω stages does not increase it. Tarski–Vaught then supplies elementarity, so the construction controls size without sacrificing any first-order truths with parameters from the smaller structure. □

Theorem 7.5 (Tarski–Vaught criterion and upward Löwenheim–Skolem theorem)**.** *(Tarski–Vaught.) If $N \subseteq M$ is a substructure and every existential formula with parameters from N that has a witness in M has a witness in N, then $N \preccurlyeq M$. (Upward Löwenheim–Skolem.) If a theory has an infinite model, then for every cardinal $\kappa \geq |\mathcal{L}| + \aleph_0$ it has a model of cardinality κ.*

Proof. For Tarski–Vaught, suppose $A \subseteq M$ is a substructure satisfying the witness condition: whenever $\bar{a} \in A$ and

$$M \models \exists x \varphi(x, \bar{a}),$$

some $b \in A$ satisfies $M \models \varphi(b, \bar{a})$. Induction on formulas proves

$$A \models \psi(\bar{a}) \iff M \models \psi(\bar{a}),$$

with the existential step exactly the witness condition. Hence $A \preccurlyeq M$. For upward Löwenheim–Skolem, let T have an infinite model and add constants c_α for $\alpha < \kappa$ together with

$$c_\alpha \neq c_\beta \qquad (\alpha \neq \beta).$$

Every finite subset is satisfiable in an infinite model, so compactness gives a model of size at least κ. Applying downward Löwenheim–Skolem to that model, while retaining all new constants, yields an elementary model of the desired cardinal size. *Further explanation.* These are two complementary uses of witnesses. Tarski–Vaught characterizes elementarity by saying that existential witnesses needed by the large structure can already be found in the substructure. Upward Löwenheim–Skolem runs in the opposite direction: compactness first creates arbitrarily many distinct named elements, and downward Löwenheim–Skolem can then reduce the resulting model to a controlled target cardinal. Separating the two steps prevents the common misconception that upward Löwenheim–Skolem is a direct substructure construction. □

Worked example

Worked Example 7.6 (A nonstandard element by compactness)**.** Expand arithmetic by a constant c and add $c > \bar{n}$ for each standard numeral $\bar{n}$. If a finite subset mentions only $c > \bar{n}_0, \ldots, c > \bar{n}_k$, interpret c as $1 + \max\{n_0, \ldots, n_k\}$; hence every finite subset is satisfiable in $\mathbb{N}$. Compactness gives a model satisfying all the sentences simultaneously, so its interpretation of c is larger than every standard numeral. The model must therefore contain nonstandard elements.

A useful warning

Downward Löwenheim–Skolem says that if M is an infinite structure in language L and $A \subseteq M$, then there is $N \preccurlyeq M$ containing A with $|N| \leq \max(|A|, |L|, \aleph_0)$; it does not say that sets externally countable in the ambient universe are internally seen as countable.

End-of-volume perspective

The logical language is now mature enough to build and compare models. The two indispensable bridges are completeness (syntax to semantics) and compactness (finite satisfiability to global satisfiability).

Part II

Volume II: Computability and Arithmetized Syntax

Preface

The aim is to make finite syntax manipulable inside arithmetic. Concrete encodings are treated as mathematical objects rather than as an appeal to 'effective coding'.

Position in the series

Volume II turns finite syntax into arithmetic data. Its coding and representability results are the technical input for the fixed-point and provability arguments of Volume III.

Sources and further reading

The arithmetization is arranged to feed directly into the incompleteness proofs of Volume III. The historical source is Gödel's 1931 paper [4]; modern background may be found in [2].

Notation for this volume

$\ulcorner \varphi \urcorner$ denotes the Gödel number of φ. $\mathrm{Prf}_T(p, q)$ is the proof predicate and $\mathrm{Prov}_T(q) \equiv \exists p \, \mathrm{Prf}_T(p, q)$.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 8

Primitive recursion

Definitions and setup

Formal definitions used in this chapter

Primitive-recursive function. The least class of functions on $\mathbb{N}$ containing zero, successor, and projections and closed under composition and primitive recursion.
Primitive-recursive relation. A relation whose characteristic function is primitive recursive.

Primitive-recursive functions are generated from zero, successor, and projections by composition and primitive recursion. The point is not merely that such functions are computable, but that their computations can be established by bounded finite data. A relation is primitive recursive when its characteristic function is primitive recursive. Bounded sums, products, and searches will be derived rather than added as new primitives. Pairing functions let several numerical parameters be packaged into one code.

How this chapter fits together

Primitive recursion supplies a deliberately restrictive language of effective constructions. The point is not merely that familiar numerical functions are computable, but that their computation histories can later be represented inside arithmetic. Closure under composition, recursion, bounded sums, bounded products, and bounded search gives a toolbox from which the coding functions of the next chapters are built.

Theorem 8.1 (Closure of primitive-recursive functions under composition)**.** *If $g_1, \dots, g_k : \mathbb{N}^m \to \mathbb{N}$ and $h : \mathbb{N}^k \to \mathbb{N}$ are primitive recursive, then the composition $f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x}))$ is primitive recursive.*

Proof. Let $\mathcal{PR}$ denote the least class of numerical functions containing the zero function, successor, and all projections, and closed under composition and primitive recursion. Assume

$$g_i \in \mathcal{PR} \quad (1 \le i \le k), \qquad h \in \mathcal{PR}.$$

The function in the statement is

$$f(\bar{x}) = h(g_1(\bar{x}), \dots, g_k(\bar{x})).$$

This is exactly one application of the composition clause in the definition of $\mathcal{PR}$. More formally, if C_h is a primitive-recursive program for h and C_i is one for g_i, the composition constructor produces the program

$$\bar{x} \longmapsto (g_1(\bar{x}), \dots, g_k(\bar{x})) \longmapsto h(g_1(\bar{x}), \dots, g_k(\bar{x})).$$

No minimization or unbounded search is introduced. Therefore the resulting function remains in $\mathcal{PR}$. Notice that this proof uses the closure clause itself; it does not assert that every computable composition is primitive recursive. The latter would require first proving that the component functions already lie in $\mathcal{PR}$.

Point specific to this result. Composition is one of the defining closure operations: substitute already primitive-recursive outputs into a primitive-recursive outer function. The point of recording it explicitly is that later coding constructions are nested compositions of projections, arithmetic, and bounded case distinctions. □

Theorem 8.2 (Closure under primitive recursion)**.** *If $g : \mathbb{N}^m \to \mathbb{N}$ and $h : \mathbb{N}^{m+2} \to \mathbb{N}$ are primitive recursive, then the uniquely defined function f satisfying*

$$f(\bar{x}, 0) = g(\bar{x}), \qquad f(\bar{x}, n+1) = h(\bar{x}, n, f(\bar{x}, n))$$

is primitive recursive.

Proof. Let $g : \mathbb{N}^m \to \mathbb{N}$ and $h : \mathbb{N}^{m+2} \to \mathbb{N}$ be primitive recursive. By the primitive-recursion constructor there is a unique function $f : \mathbb{N}^{m+1} \to \mathbb{N}$ satisfying

$$f(\bar{x}, 0) = g(\bar{x}), \qquad f(\bar{x}, n+1) = h(\bar{x}, n, f(\bar{x}, n)).$$

The construction is itself one of the defining closure operations of $\mathcal{PR}$, so $f \in \mathcal{PR}$. Uniqueness can also be checked independently by induction on n: if f, f' satisfy the two equations, then they agree at 0; and from $f(\bar{x}, n) = f'(\bar{x}, n)$ the successor equation gives

$$f(\bar{x}, n+1) = h(\bar{x}, n, f(\bar{x}, n)) = h(\bar{x}, n, f'(\bar{x}, n)) = f'(\bar{x}, n+1).$$

This uniqueness is useful later, because arithmetic will represent a primitive-recursive computation by a finite sequence and we must know that the represented output is determined by the input.

□

Lemma 8.3 (Primitive-recursive bounded search)**.** *If $R(\bar{x}, y)$ is a primitive-recursive relation, then the bounded minimization function*

$$\mu y < z\, R(\bar{x}, y)$$

(with a fixed default value when no witness exists) is primitive recursive. Consequently bounded existential and universal quantification preserve primitive recursiveness.

Proof. Let $R(\bar{x}, y)$ be a primitive-recursive predicate and let $b(\bar{x})$ be a primitive-recursive bound. Define the bounded-search function

$$\mu_{y<b(\bar{x})} R(\bar{x}, y) = \begin{cases} \min\{y < b(\bar{x}) : R(\bar{x}, y)\}, & \text{if such } y \text{ exists,} \\ b(\bar{x}), & \text{otherwise.} \end{cases}$$

Construct it by primitive recursion on $n \le b(\bar{x})$. Let $s(\bar{x}, 0) = b(\bar{x})$, and at stage $n+1$ replace the stored value by n exactly when no smaller witness has been stored and $R(\bar{x}, n)$ holds. The tests "$s = b$", "$n < b$", and R are primitive recursive, and case distinction on a primitive-recursive predicate is primitive recursive. Hence s is primitive recursive. Evaluating it at the primitive-recursive bound $b(\bar{x})$ gives the required bounded minimum. The essential point is that the search range is bounded in advance; unrestricted minimization would leave the primitive-recursive class.

Point specific to this result. Bounded search is total because the search interval is finite. A primitive recursion can carry the least successful index found so far, so no unbounded minimization principle is being introduced. □

Proposition 8.4 (Primitive-recursive pairing and projection functions)**.** *There are primitive-recursive functions $\langle x, y\rangle, \pi_0, \pi_1$ such that*

$$\pi_0(\langle x, y\rangle) = x, \qquad \pi_1(\langle x, y\rangle) = y$$

for all $x, y \in \mathbb{N}$.

Proof. Use Cantor's pairing polynomial

$$\langle x, y\rangle = \frac{(x+y)(x+y+1)}{2} + y.$$

Addition and multiplication are primitive recursive. The numerator is always even, and integer division by 2 can be implemented by bounded search for the unique $q \le (x+y)(x+y+1)$ with $2q$ equal to that numerator. Thus pairing is primitive recursive. To recover (x, y) from z, first find the unique $w \le z$ such that

$$\frac{w(w+1)}{2} \le z < \frac{(w+1)(w+2)}{2}.$$

This is bounded search. Put

$$y = z - \frac{w(w+1)}{2}, \qquad x = w - y.$$

Truncated subtraction and the preceding bounded searches are primitive recursive, so both projections $\pi_0(z) = x$ and $\pi_1(z) = y$ are primitive recursive. Direct substitution verifies

$$\pi_0(\langle x, y\rangle) = x, \qquad \pi_1(\langle x, y\rangle) = y.$$

These identities, rather than the particular polynomial, are what later coding arguments use.

Point specific to this result. For pairing, it is not enough that a bijection $\mathbb{N}^2 \to \mathbb{N}$ exists; both encoding and decoding must be primitive recursive. The inverse coordinates are found by searches bounded by the code itself, preserving primitive recursiveness. □

Worked example

Worked Example 8.5 (A bounded search)**.** If $R(x, y)$ is primitive recursive, define $m(x, z)$ to be the least $y < z$ with $R(x, y)$, or z if there is none. A recursion scans $0, 1, \ldots, z-1$ while storing the first successful index. Because the search is bounded in advance, the construction remains primitive recursive.

A useful warning

Unbounded minimization is not primitive recursion. Later c.e. predicates may use an existentially quantified computation witness, but that is conceptually different from a bounded primitive-recursive search.

Looking ahead

The next chapter, *Coding finite objects*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 9

Coding finite objects

Definitions and setup

Formal definitions used in this chapter

Gödel coding of finite data. Fix primitive-recursive predicates/functions

$$\mathrm{Seq}(s), \qquad \ell(s), \qquad (s)_i$$

such that $\mathrm{Seq}(s)$ means that s codes a finite sequence, $\ell(s)$ is its length, and

$$0 \le i < \ell(s) \Longrightarrow (s)_i \in \omega$$

is its i-th entry. A coding of finite labelled trees is a sequence coding of the finite node set together with primitive-recursive parent and label functions. "Effective coding" in this volume always means that encoding, decoding, and the local well-formedness predicates are primitive recursive.

A Gödel-style code for a finite object must support effective recognition and component extraction. We use a primitive-recursive coding of finite sequences and then encode strings, trees, and derivations as tagged sequences. The length function, the i-th component function, concatenation, and bounded modification are required to be primitive recursive. The precise numerical values of the codes are irrelevant; the uniform primitive-recursive operations on them are the load-bearing content.

How this chapter fits together

Gödel coding needs more than a pairing function: formulas and proofs are variable-length finite objects. We therefore encode finite sequences with primitive-recursive access to their length and coordinates. Once sequence manipulation is available, finite strings, trees, substitutions, and derivations can all be treated arithmetically.

Theorem 9.1 (Primitive-recursive coding and decoding of finite sequences)**.** *There is a primitive-recursive coding of finite sequences by natural numbers together with primitive-recursive functions* $\mathrm{lh}(s)$ *and* $(s)_i$ *such that every finite sequence* $\langle a_0, \dots, a_{n-1} \rangle$ *has a code* s *with* $\mathrm{lh}(s) = n$ *and* $(s)_i = a_i$ *for* $i < n$*.*

How to read the symbols: sequence codes are data structures

A natural number is being used as a serialized finite list. The coding theorem must supply three operations analogous to a programming interface: create a code, recover its length, and read its i-th entry. Later

arithmetization never relies on the visual form of the code; it relies only on the fact that these operations are primitive recursive and hence representable in weak arithmetic.

Proof. Fix a standard β-coding of finite sequences. One convenient form is to code a sequence by a pair $s = \langle b, c\rangle$ and define

$$\beta(b, c, i) = \mathrm{rem}(b, 1 + (i + 1)c).$$

The remainder function is primitive recursive because the quotient is bounded by b. For every finite sequence $a_0, \ldots, a_{n-1}$, choose c divisible by sufficiently many small integers and large enough that the moduli

$$m_i = 1 + (i + 1)c$$

are pairwise coprime and exceed a_i. The Chinese remainder theorem gives b with $b \equiv a_i \pmod{m_i}$. Hence $\beta(b, c, i) = a_i$ for $i < n$. Define a sequence code to include its length, for example

$$\mathrm{Seq}(s) \iff s = \langle n, \langle b, c\rangle\rangle.$$

Then $\mathrm{lh}(s) = n$ and $(s)_i = \beta(b, c, i)$ for $i < n$. Pairing, projections, remainder, and bounded comparison are primitive recursive, so Seq, length, and component extraction are primitive recursive. Thus finite sequences can be quantified over inside arithmetic using single natural-number codes.

Point specific to this result. The sequence theorem requires a uniform component function: from one code, an index i, and the recorded length, arithmetic can recover the i-th entry. This uniformity is what later lets arithmetic inspect a proof line by line. □

Proposition 9.2 (Primitive-recursive coding of finite trees and strings)**.** *Finite labelled strings and finite rooted trees over a fixed finite or computably coded alphabet admit Gödel codings for which the predicates "s codes such a string/tree" and the basic constructor, predecessor, and label operations are primitive recursive.*

Proof. A finite string over a finite alphabet is a finite sequence of symbol codes, so the sequence coding theorem already gives primitive-recursive coding and decoding. A finite rooted labelled tree can be coded by a finite sequence

$$\langle \langle \ell_i, p_i\rangle : i < n\rangle,$$

where ℓ_i is the label of node i and $p_i < i$ is the parent index, with a distinguished root convention. The predicate saying that such a code is a tree is bounded: check for every $i < n$ that the parent index has the required range, that the root is unique, and that the local arities agree with the labels. Bounded universal quantification over a primitive-recursive predicate is primitive recursive. Consequently "t codes a finite labelled tree", "node i has label a", and "j is a child of i" are primitive-recursive relations. Syntax trees will later be a special case in which the labels are logical symbols and the arity test enforces the grammar.

Point specific to this result. Trees and strings are reduced to finite sequences by storing node tags, parent data, and labels. Local well-formedness is then a bounded property of the code, so later syntactic parse trees can be checked arithmetically. □

Lemma 9.3 (Bounded sequence manipulation is primitive recursive)**.** *For the chosen sequence coding, the operations of taking length, extracting a component, concatenating two coded sequences, replacing a bounded component, and checking a bounded local property of all components are primitive recursive.*

Proof. Let s code $\langle a_0, \ldots, a_{n-1}\rangle$. Component extraction $(s)_i$ and length $\mathrm{lh}(s)$ are primitive recursive. Hence bounded operations such as

$$\sum_{i<n}(s)_i, \qquad \prod_{i<n}(1 + (s)_i), \qquad \max_{i<n}(s)_i$$

are defined by primitive recursion on $i < n$. Concatenation is obtained by constructing a new code t of length $n + m$ satisfying

$$(t)_i = \begin{cases} (s)_i, & i < n, \\ (r)_{i-n}, & n \le i < n + m. \end{cases}$$

The component function is primitive recursive and the finite-sequence coding theorem supplies a code s with the required length and components; choosing the least such code is a bounded search once an explicit elementary bound on the coding parameters is fixed. Similarly, replacing the j-th component by u, taking an initial segment, and testing membership among the components are primitive recursive. Thus later proof- and syntax-manipulation routines can operate on coded finite lists without leaving $\mathcal{PR}$.

Point specific to this result. Operations such as taking an initial segment, replacing one entry, and concatenating two coded sequences use only bounded component extraction and reconstruction. The bounds are explicit functions of the original codes. □

Proposition 9.4 (Coding finite derivations by natural numbers)**.** *Fix a recursive formal proof calculus. There is a primitive-recursive relation* $\mathrm{Deriv}(p, q)$ *saying that* p *codes a finite derivation whose last formula has Gödel number* q*.*

Proof. Fix numerical codes for formulas, axiom instances, and inference-rule tags. Code a finite derivation

$$\varphi_0, \dots, \varphi_{n-1}$$

as a finite sequence d of line records. A line record contains the formula code and enough bounded auxiliary data to certify its justification: an axiom tag, or the indices of earlier premises used by an inference rule. Define

$$\mathrm{LocOK}(d, i)$$

to mean that line $i < \mathrm{lh}(d)$ is either a permitted axiom or follows from earlier lines according to one of the finitely many rules. Formula recognition and checking an axiom schema are primitive recursive once syntax is coded; the referenced indices are bounded by i. Hence LocOK is primitive recursive. Then

$$\mathrm{Deriv}(d) \iff \mathrm{Seq}(d) \wedge \forall i < \mathrm{lh}(d)\ \mathrm{LocOK}(d, i)$$

(with the stray ‘a‘ omitted in the actual formula) is a bounded universal condition and therefore primitive recursive. Thus finite formal derivations admit effective numerical coding whose local correctness can be checked primitive recursively.

Point specific to this result. A derivation code stores a finite formula sequence together with enough local justification data to check each line. This makes proof verification a finite arithmetic computation rather than a meta-level appeal to the existence of a derivation. □

Worked example

Worked Example 9.5 (Encoding a short sequence)**.** A sequence code is useful only if arithmetic can recover its entries. For a sequence $(3, 1, 4)$, the chosen β-style code packages parameters b, c so that $\beta(b, c, 0) = 3$, $\beta(b, c, 1) = 1$, and $\beta(b, c, 2) = 4$. The numerical value of the code is unimportant; what matters is that the relation “the i-th entry is a” is primitive recursive.

A useful warning

Do not confuse existence of an external coding scheme with representability of its decoding operations inside arithmetic. Both levels are needed later.

Looking ahead

The next chapter, *Recursive and computably enumerable relations*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 10

Recursive and computably enumerable relations

Definitions and setup

Formal definitions used in this chapter

Recursive relation. A relation whose characteristic function is total computable.
Computably enumerable relation. A relation recognized by a partial computable procedure, equivalently one of the form $\exists y\, C(\bar{x}, y)$ with computable/primitive-recursive computation predicate C.

A relation is recursive when there is an algorithm deciding both positive and negative instances. It is computably enumerable (c.e.) when positive instances can be enumerated, equivalently when membership is the existence of a finite halting computation. A partial recursive function may fail to halt; primitive-recursive functions are total. These distinctions matter because a formal theory may have a c.e. axiom set even when axiom membership is not decidable. Later proof certificates will carry witnesses for c.e. axiomhood.

How this chapter fits together

The distinction between decidable and merely enumerable relations is central to incompleteness. A recursive relation has a terminating decision procedure; a c.e. relation has a terminating positive search when the answer is yes. Finite computation histories turn the latter into existential statements over primitive-recursive predicates, which is exactly the form needed for c.e. theories.

Theorem 10.1 (Closure properties of recursive relations)**.** *The class of recursive relations is closed under complement, finite conjunction/disjunction, substitution of recursive functions, and bounded quantification.*

Proof. A relation $R \subseteq \mathbb{N}^k$ is recursive when its characteristic function χ_R is total recursive. If R, S are recursive, then Boolean combinations are recursive because

$$\chi_{\neg R} = 1 - \chi_R, \quad \chi_{R \cap S} = \chi_R \chi_S, \quad \chi_{R \cup S} = \operatorname{sg}(\chi_R + \chi_S),$$

where $\operatorname{sg}(0) = 0$ and $\operatorname{sg}(n+1) = 1$. For bounded existential quantification define

$$Q(\bar{x}) \iff \exists y < b(\bar{x})\, R(\bar{x}, y).$$

A terminating algorithm checks $y = 0, 1, \ldots, b(\bar{x}) - 1$, so Q is recursive whenever b and R are recursive. Bounded universal quantification follows by complement. Substitution of total recursive functions into

a recursive relation is recursive by composition. These closure facts are algorithmic: every construction displayed above terminates on every input, which is the defining difference from merely c.e. relations.

Point specific to this result. For recursive relations, both positive and negative membership terminate. To decide the complement, run the same terminating decision procedure and reverse its yes/no output; and conjunction/disjunction are decided by combining the terminating decision procedures. □

Theorem 10.2 (Closure properties of computably enumerable relations)**.** *The class of computably enumerable relations is closed under finite conjunction/disjunction, existential quantification, and computable substitution. A relation is recursive exactly when both it and its complement are computably enumerable.*

Proof. A relation $R(\bar{x})$ is computably enumerable if some partial computable search halts exactly on the tuples in R, equivalently if there is a recursive relation $C(\bar{x}, s)$ such that

$$R(\bar{x}) \iff \exists s\, C(\bar{x}, s).$$

If R and S are c.e., dovetail their searches. For the union, halt when either search succeeds; for the intersection, run both searches in parallel and halt once both have succeeded. Thus $R \cup S$ and $R \cap S$ are c.e. If f is total computable, then

$$Q(\bar{x}) \iff R(f(\bar{x}))$$

is c.e. by first computing $f(\bar{x})$ and then running the enumerator for R. Existential projection preserves c.e.-ness: for $\exists y\, R(\bar{x}, y)$, dovetail the searches for $R(\bar{x}, 0), R(\bar{x}, 1), \ldots$. Complement is deliberately absent from this list: a c.e. relation need not have c.e. complement.

Point specific to this result. For c.e. relations one cannot in general wait for negative evidence. Union is handled by dovetailing enumerations, intersection by waiting until an item has appeared in both, and existential projection by dovetailing over witnesses. □

Theorem 10.3 (Normal-form characterization of c.e. relations in the form needed later)**.** *A relation $R(\bar{x})$ is computably enumerable if and only if there is a primitive-recursive relation $C(\bar{x}, y)$ such that*

$$R(\bar{x}) \iff \exists y\, C(\bar{x}, y).$$

The witness y may be taken to code a finite accepting computation.

Proof. Let $R(\bar{x})$ be c.e. Choose a machine M halting exactly when $R(\bar{x})$ holds. A finite computation history of M is a finite sequence of configurations. Because each machine transition is local and effective, there is a primitive-recursive predicate

$$C_M(\bar{x}, s)$$

saying that s codes a valid halting computation of M on input $\bar{x}$. Hence

$$R(\bar{x}) \iff \exists s\, C_M(\bar{x}, s).$$

Conversely, if C is primitive recursive, an algorithm enumerates $s = 0, 1, 2, \ldots$ and halts at the first s satisfying $C(\bar{x}, s)$. Therefore every relation of the displayed existential primitive-recursive form is c.e. This normal form is the exact bridge needed later: the existential variable s will become a numerical witness that a c.e. axiom has appeared or that a finite computation/proof certificate is correct.

□

Proposition 10.4 (Effective enumeration of finite proof searches)**.** *For every c.e. theory T and bound b, the finite search through all proof certificates of code at most b is effective; equivalently, the relation "some T-proof of φ has code $\leq b$" is recursive uniformly in b and $\ulcorner\varphi\urcorner$.*

Proof. Enumerate all finite sequences of natural numbers effectively, for example in increasing code order. For each candidate d, run the primitive-recursive predicate $\mathrm{Deriv}(d)$. If it holds, output the code of the last formula of d, together with d if a proof witness is desired. Because every genuine finite derivation has some natural-number code, it eventually appears in this enumeration. Thus the relation

$$\mathrm{Thm}(q) \iff \exists d\,[\mathrm{Deriv}(d) \wedge \mathrm{Last}(d) = q]$$

is c.e. More generally, if the nonlogical axiom set is itself c.e., attach to each axiom line an enumeration-stage witness; local proof checking then remains primitive recursive. The enumeration is therefore effective even though membership in the theorem set need not be decidable.

Point specific to this result. Because candidate proof codes are natural numbers and local proof correctness is decidable once the relevant certificate data are supplied, one can enumerate all successful finite proofs by scanning codes in increasing order. □

Worked example

Worked Example 10.5 (Dovetailing two searches)**.** If both R and its complement are c.e., run the two enumerations in alternating steps. Exactly one must eventually exhibit the input, so the combined procedure halts with the correct yes/no answer. Thus a relation is recursive precisely when it and its complement are c.e. in the setting used here.

A useful warning

An enumerator need not tell us that a sentence will never appear. That asymmetry is why c.e. theorem sets can be effectively listed without being decidable.

Looking ahead

The next chapter, *Weak arithmetic*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 11

Weak arithmetic

Definitions and setup

Formal definitions used in this chapter

Robinson arithmetic Q. The theory in the language $0, S, +, \cdot$ with axioms

$$Sx \neq 0, \qquad Sx = Sy \to x = y, \qquad x \neq 0 \to \exists y\,(x = Sy),$$

$$x + 0 = x, \qquad x + Sy = S(x + y),$$

$$x \cdot 0 = 0, \qquad x \cdot Sy = x \cdot y + x.$$

It has no induction schema.
Σ_1 **formula.** A formula equivalent to $\exists \bar{y}\, \delta$ with bounded Δ_0 matrix δ.

Robinson arithmetic Q is a finitely axiomatized first-order theory of $0, S, +, \times$ containing basic defining equations but no induction scheme. A bounded (Δ_0) formula has all quantifiers explicitly bounded by terms. A Σ_1 formula begins with existential number quantifiers followed by a bounded matrix. The key fact used later is instance-wise: true finite primitive-recursive calculations can be verified on numerals in a weak theory, and true Σ_1 statements have concrete witnesses.

How this chapter fits together

Robinson arithmetic Q is intentionally weak: it has enough arithmetic to verify each concrete finite computation but lacks induction as a general axiom scheme. This makes it a useful baseline for incompleteness. The key fact is instance-wise Σ_1-completeness: a true existential statement with a bounded computable matrix has concrete witnesses whose calculation Q can verify.

Proposition 11.1 (Basic arithmetic derivations in Robinson arithmetic Q)**.** *Robinson arithmetic Q proves every true closed instance of the defining equations for successor, addition, and multiplication on numerals; in particular, for concrete $m, n \in \mathbb{N}$, Q proves the correct numeral equations for $\bar{m} + \bar{n}$, $\bar{m} \cdot \bar{n}$, and inequality of distinct numerals.*

Proof. Robinson arithmetic Q has the equations

$$x + 0 = x, \qquad x + S(y) = S(x + y),$$

$$x \cdot 0 = 0, \qquad x \cdot S(y) = x \cdot y + x,$$

together with the usual successor axioms. Fix standard numerals $\bar{n} = S^n 0$. For concrete m, n, repeated use of the recursion equations yields finite derivations

$$Q \vdash \bar{m} + \bar{n} = \overline{m+n}, \qquad Q \vdash \bar{m} \cdot \bar{n} = \overline{mn}.$$

The proof is an external induction on the standard integer n, not an induction carried out inside Q. For example,

$$\bar{m} + \overline{n+1} = \bar{m} + S(\bar{n}) = S(\bar{m} + \bar{n}) = S(\overline{m+n}) = \overline{m+n+1}.$$

Likewise the successor axioms prove distinct standard numerals unequal. These finite numeral calculations are the only arithmetic strength needed for many later correctness-on-standard-codes arguments.

Point specific to this result. The claim concerns concrete closed calculations, not induction in Q. Each numeral computation is reduced finitely by the defining equations for successor, addition, and multiplication. □

Lemma 11.2 (Numeralwise correctness of Q for bounded primitive-recursive calculations)**.** *Let $R(\bar{x})$ be a bounded relation represented by a Δ_0 formula $\rho_R(\bar{x})$, and let $\bar{n} = (n_1, \ldots, n_k) \in \omega^k$. If $R(\bar{n})$ is true, then*

$$Q \vdash \rho_R(\bar{n}_1, \ldots, \bar{n}_k),$$

and if $R(\bar{n})$ is false, then

$$Q \vdash \neg\rho_R(\bar{n}_1, \ldots, \bar{n}_k).$$

Thus Q correctly verifies every closed bounded computation occurring in the coding developed here.

Proof. Let $R(\bar{x})$ be a bounded relation obtained from addition, multiplication, order, and bounded quantifiers. Fix a standard tuple $\bar{n}$. Every bounded quantifier in $R(\bar{n})$ ranges over a finite standard set. Evaluate the atomic arithmetic expressions using the numeral calculations already proved in Q. Then eliminate bounded quantifiers by a finite case analysis. Thus, if $R(\bar{n})$ is true in $\mathbb{N}$, one obtains a finite derivation

$$Q \vdash R(\overline{\mathbf{n}}),$$

and if it is false,

$$Q \vdash \neg R(\overline{\mathbf{n}}).$$

No induction schema is invoked inside Q: the metatheoretic evaluation supplies a finite proof for this particular tuple. This "numeralwise correctness" is exactly what is later used when the primitive-recursive proof predicate is evaluated at an actual standard natural-number code of a proof.

Point specific to this result. Numeralwise correctness is external-to-internal: for each actual primitive-recursive computation on fixed numerals, one constructs a finite Q-derivation of its correct value. The theorem does not assert a single universally quantified induction principle inside Q. □

Theorem 11.3 (Sigma-1 completeness of Q in the required form)**.** *If σ is a true Σ_1 sentence of arithmetic, then $Q \vdash \sigma$. More generally, every true Σ_1 formula is provable in Q after substituting numerals for its free variables.*

Commented symbolic trace: why true Σ_1 sentences are provable in Q

A true Σ_1 sentence has the form $\exists\bar{x}\,\delta(\bar{x})$ with bounded δ. Externally choose the actual natural-number witnesses $\bar{n}$. The bounded matrix $\delta(\bar{n})$ reduces to a finite arithmetic computation, and Q proves that concrete computation. Only then do we use existential introduction inside the formal proof. Thus the proof moves from an external witness to an internal derivation; it does not assume that Q can search for the witness by induction.

Proof. Let σ be a true Σ_1 sentence. Put it in the form

$$\sigma \equiv \exists x_1 \cdots \exists x_k \, \delta(x_1, \ldots, x_k),$$

where δ is bounded. Since $\mathbb{N} \models \sigma$, choose standard witnesses $n_1, \ldots, n_k$ with

$$\mathbb{N} \models \delta(n_1, \ldots, n_k).$$

By numeralwise correctness,

$$Q \vdash \delta(\bar{n}_1, \ldots, \bar{n}_k).$$

Apply existential introduction successively to obtain

$$Q \vdash \exists x_1 \cdots \exists x_k \, \delta(x_1, \ldots, x_k),$$

which is $Q \vdash \sigma$. The proof is one-way: truth of a Σ_1 sentence yields provability in Q. We have not claimed that every sentence provable in an arbitrary extension of Q is true; that would be a soundness assumption on the extension, not Σ_1-completeness of Q.

Point specific to this result. A true Σ_1 sentence comes with actual numerical witnesses. After substituting their numerals, the bounded matrix is a finite computation that Q can verify; existential introduction then proves the original sentence. □

Proposition 11.4 (Comparison of Q, elementary arithmetic fragments, and PA for later metamathematics)**.** *For the metamathematical arguments of Volumes II–III, Q suffices for numeralwise verification and Σ_1-completeness, while induction principles used to reason uniformly about coded proofs are explicitly assigned to stronger fragments such as $I\Sigma_1$ or PA. No later theorem may use an induction scheme not present in its stated base theory.*

Proof. The theories are separated by the induction they provide. Robinson arithmetic Q has no induction schema; it suffices for verifying individual standard computations. A bounded-arithmetic fragment adds induction only for a specified syntactic class, while $I\Sigma_1$ contains induction for all Σ_1 formulas. Peano arithmetic contains induction for every first-order arithmetic formula. Consequently

$$Q \subseteq I\Sigma_1 \subseteq \mathrm{PA}$$

for the standard presentations. Later arguments are tagged accordingly: construction and numeralwise verification of Prf_T need only weak arithmetic; formalized closure of proofs and the Hilbert–Bernays–Löb derivability conditions are most cleanly carried out in $I\Sigma_1$ or PA. The point of this proposition is therefore not an abstract inclusion alone, but a dependency rule: whenever an internal induction is used, the proof must explicitly move to a theory containing the corresponding induction scheme.

Point specific to this result. The comparison isolates which later arguments need only numeralwise computation in Q and which require induction available in stronger arithmetic. This prevents the incompleteness proofs from silently borrowing PA-strength at coding steps where Q suffices. □

Worked example

Worked Example 11.5 (Verifying a concrete computation in Q)**.** To prove a true sentence such as $\overline{2}+\overline{3}=\overline{5}$, repeatedly use the defining equations for addition until the closed term reduces to the expected numeral. A true Σ_1 sentence first supplies actual numerical witnesses; the bounded matrix is then reduced by the same kind of finite calculation.

A useful warning

Σ_1-completeness of Q is not full soundness, completeness, or induction. It is a one-way theorem about true Σ_1 sentences.

Looking ahead

The next chapter, *Representability*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 12

Representability

Definitions and setup

Formal definitions used in this chapter

Representability of a function. A formula $F(\bar{x}, y)$ represents f when true numeral instances are provable and the output is provably unique.
Representability of a relation. A formula represents R when positive and, for strong representability, negative numeral instances are provable correctly.

A numerical function f is represented in a theory T by a formula $F(\bar{x}, y)$ when, for each standard input, the theory proves the correct output instance and proves uniqueness in the strength required by later arguments. A relation is represented by a formula whose numeral instances are provable when true and whose negations are provable when false in the strong form. Representability is a theorem about formal arithmetic, stronger than the external fact that a function is computable.

How this chapter fits together

Representability is the bridge from external computation to internal arithmetic. A primitive-recursive computation is encoded by a finite sequence satisfying local transition equations; arithmetic can then assert the existence of such a sequence. This is why recursion on natural numbers becomes an existential arithmetic formula about a coded history.

Theorem 12.1 (Representability of primitive-recursive functions)**.** *For every primitive-recursive function $f : \mathbb{N}^k \to \mathbb{N}$ there is an arithmetic formula $F_f(\bar{x}, y)$ such that, for every standard tuple $\mathbf{n} = (n_1, \ldots, n_k)$ and $m = f(\mathbf{n})$,*

$$Q \vdash \forall y \left(F_f(\overline{\mathbf{n}}, y) \leftrightarrow y = \bar{m} \right).$$

Thus the graph formula proves both the correct output and uniqueness of that output for each standard input tuple.

Proof. We induct on the generation of the primitive-recursive function f.

For the zero, successor, and projection functions use respectively

$$y = 0, \qquad y = Sx_1, \qquad y = x_i.$$

For each standard input tuple, Q proves the corresponding closed equation and proves that no different numeral can satisfy it.

Suppose

$$f(\bar{x}) = h(g_1(\bar{x}), \ldots, g_r(\bar{x}))$$

and graph formulas $G_i(\bar{x}, z_i)$ and $H(\bar{z}, y)$ already strongly represent the component functions. Put

$$F_f(\bar{x}, y) := \exists z_1 \cdots z_r \left(\bigwedge_{i=1}^{r} G_i(\bar{x}, z_i) \wedge H(\bar{z}, y) \right).$$

Fix a standard tuple $\mathbf{n}$. The induction hypotheses identify each intermediate variable z_i with the unique numeral $\overline{g_i(\mathbf{n})}$, after which the hypothesis for H identifies y with $\overline{f(\mathbf{n})}$. Hence Q proves the required biconditional for this input.

For primitive recursion, with

$$f(\bar{x}, 0) = g(\bar{x}), \qquad f(\bar{x}, n+1) = h(\bar{x}, n, f(\bar{x}, n)),$$

let $F_f(\bar{x}, n, y)$ assert the existence of a coded sequence s of length $n+1$ whose first entry satisfies the graph formula for g, whose successive entries satisfy the graph formula for h, and whose last entry is y. Sequence coding and bounded component extraction are primitive recursive, so this is an arithmetic formula. For a fixed standard n, the recursion equations can be unrolled through exactly n steps inside Q. The induction hypotheses at those finitely many steps prove both existence of the actual computation sequence and uniqueness of every entry. Therefore, if $m = f(\mathbf{n}, n)$,

$$Q \vdash \forall y \big(F_f(\overline{\mathbf{n}}, \bar{n}, y) \leftrightarrow y = \bar{m} \big).$$

This is strong numeralwise representability. No uniform induction principle inside Q is being claimed; for each standard input, only a finite derivation obtained by unrolling the corresponding finite computation is required. □

Theorem 12.2 (Representability of primitive-recursive relations)**.** *For every primitive-recursive relation $R \subseteq \mathbb{N}^k$ there is an arithmetic formula $\rho_R(\bar{x})$ such that for every $\bar{n}$,*

$$R(\bar{n}) \Rightarrow Q \vdash \rho_R(\overline{\mathbf{n}}), \qquad \neg R(\bar{n}) \Rightarrow Q \vdash \neg \rho_R(\overline{\mathbf{n}}).$$

Proof. Let $R(\bar{x})$ be primitive recursive. Its characteristic function χ_R is primitive recursive, so by the preceding theorem there is a formula $C_R(\bar{x}, y)$ representing the graph of χ_R. Define

$$\rho_R(\bar{x}) := C_R(\bar{x}, \bar{1}).$$

If $R(\bar{n})$ holds in $\mathbb{N}$, then $\chi_R(\bar{n}) = 1$, and representability gives

$$Q \vdash \rho_R(\overline{\mathbf{n}}).$$

If $R(\bar{n})$ fails, then $\chi_R(\bar{n}) = 0$; uniqueness of the represented output together with $\bar{0} \neq \bar{1}$ gives

$$Q \vdash \neg \rho_R(\overline{\mathbf{n}}).$$

Thus primitive-recursive relations are strongly numeralwise representable. This two-sided standard-instance correctness is what lets arithmetic later verify that a particular natural number is, or is not, a correct proof code.

Point specific to this result. For a primitive-recursive relation, represent its characteristic function and assert that its value is 1. Numeralwise correctness then gives proofs of true instances and, in the strong form used here, proofs of negated false instances. □

Theorem 12.3 (Representability of recursive relations)**.** *Every recursive relation is strongly representable in Q in the same numeralwise sense as a primitive-recursive relation.*

Proof. If R is recursive, its characteristic function is total recursive. Choose a terminating computation model for that function. The relation

$$C(\bar{x}, y, s)$$

that says "s codes a halting computation on input $\bar{x}$ with output y" is primitive recursive. Therefore

$$F_R(\bar{x}, y) := \exists s\, C(\bar{x}, y, s)$$

represents the graph of the characteristic function. Determinism of the computation gives uniqueness of the output, and termination guarantees that for every standard input some computation witness exists. Define

$$\rho_R(\bar{x}) := F_R(\bar{x}, 1).$$

Then true and false standard instances are proved exactly as for primitive-recursive relations. The difference is that the existential computation witness s need not be bounded by a primitive-recursive function of the input; the representing formula is therefore generally more complex than the primitive-recursive case.

Point specific to this result. A recursive relation has terminating positive/negative computation. Representing its decision procedure therefore transfers both outcomes into arithmetic, which is stronger than merely representing successful enumerations. □

Proposition 12.4 (Weak representability of c.e. relations)**.** *If $R(\bar{x})$ is computably enumerable, then there is a Σ_1 formula $\rho_R(\bar{x})$ such that*

$$R(\bar{n}) \Rightarrow Q \vdash \rho_R(\overline{\mathbf{n}}),$$

and $\mathbb{N} \models \rho_R(\overline{\mathbf{n}})$ exactly when $R(\bar{n})$.

Proof. Let $R(\bar{x})$ be c.e. By the c.e. normal form there is a primitive-recursive relation $C(\bar{x}, s)$ with

$$R(\bar{x}) \iff \exists s\, C(\bar{x}, s).$$

Let $\gamma_C(\bar{x}, s)$ strongly represent C. Put

$$\rho_R(\bar{x}) := \exists s\, \gamma_C(\bar{x}, s).$$

If $R(\bar{n})$ is true, choose an actual witness m. Then

$$Q \vdash \gamma_C(\overline{\mathbf{n}}, \bar{m}),$$

so $Q \vdash \rho_R(\overline{\mathbf{n}})$. This is the positive, or weak, representability direction. If $R(\bar{n})$ is false, there is no single finite witness to its failure; proving $\neg\rho_R(\overline{\mathbf{n}})$ may require much stronger information. Hence c.e. relations are represented in the one-sided sense required for theoremhood and provability predicates.

Point specific to this result. For a c.e. relation $R(\bar{x}) \iff \exists y\, C(\bar{x}, y)$, represent the primitive-recursive computation predicate C and existentially quantify the witness. Positive instances are provable; no corresponding theorem for negative instances is claimed. □

Worked example

Worked Example 12.5 (Representing addition)**.** The graph of addition is represented by the formula expressing the recursive equations $x + 0 = x$ and $x + S(y) = S(x + y)$. For a general primitive-recursive function, the same idea is packaged into a coded computation sequence: the first entry is the initial value, each later entry satisfies the recursive transition, and the last entry is the output.

A useful warning

Representability is a property of a formula relative to a theory; it is stronger than merely saying that the intended function is computable in the metatheory.

Looking ahead

The next chapter, *Gödel numbering of syntax*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 13

Gödel numbering of syntax

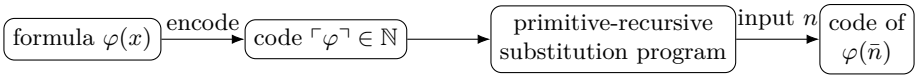

Figure 13.1: Arithmetization separates an expression from its numerical code. The arithmetic theory manipulates the codes.

Definitions and setup

Formal definitions used in this chapter

Gödel numbering. A fixed effective injection from finite syntactic objects to $\mathbb{N}$ with primitive-recursive constructor and destructor operations.
Diagonal function. $d(n)$ is the code obtained by substituting the numeral $\bar{n}$ into the distinguished free-variable position of the formula coded by n.

A Gödel numbering is an injective coding of syntactic objects by natural numbers with primitive-recursive formation and manipulation predicates. We write $\ulcorner\varphi\urcorner$ for the code of φ and $\overline{n}$ for the numeral naming n inside arithmetic. The substitution function $\mathrm{Sub}(m, n)$ is numerical: given the code of a formula and a number, it returns the code obtained by substituting the corresponding numeral. This numerical function is what makes diagonalization formalizable.

How this chapter fits together

With sequence coding in place, syntax itself becomes arithmetic data. Formation, free-variable occurrence, substitution, and diagonalization must be shown primitive recursive, not merely described as mechanical. The diagonal function will later feed a formula its own code, so this chapter carries the technical burden behind self-reference.

Theorem 13.1 (Primitive-recursive coding of terms and formulas)**.** *For the fixed Gödel numbering, the predicates "n codes a term", "n codes a formula", and the functions extracting immediate constituents, complexity, and outermost constructor are primitive recursive.*

Proof. Assign distinct numerical tags to variables, function symbols, relation symbols, connectives, and quantifiers. Define codes recursively, for example

$$\ulcorner f(t_1, \ldots, t_k)\urcorner = \langle \mathrm{tag}(f), \ulcorner t_1\urcorner, \ldots, \ulcorner t_k\urcorner \rangle,$$

and similarly for formulas. Because pairing and finite-sequence coding are primitive recursive, forming the code of a compound expression from the codes of its immediate constituents is primitive recursive. Recognition proceeds by course-of-values recursion on the code: inspect the outer tag, decode the finite list of constituents, and recursively verify that each constituent has the required syntactic category and arity. Hence

$$\mathrm{Term}(n), \qquad \mathrm{Form}(n)$$

are primitive-recursive predicates. The exact numerical coding is conventional; later arguments depend only on these primitive-recursive formation and decoding operations, not on a particular choice of tags.

Point specific to this result. The constructors are tagged so arithmetic can distinguish variables, function applications, atomic formulas, connectives, and quantifiers. Course-of-values recursion on the code then recognizes well-formed expressions. □

Theorem 13.2 (Primitive-recursive free-variable, substitution, and diagonalization functions)**.** *For the fixed Gödel numbering, free-variable occurrence, capture-avoiding substitution on codes, and the diagonal function*

$$d(n) = \mathrm{Sub}(n, v_0, \overline{n})$$

are primitive recursive on their natural domains.

How to read the symbols: codes versus the expressions they encode

If $n = \ulcorner \varphi \urcorner$, then n is a natural number and φ is a formula. The substitution function operates on the number n, decodes just enough syntax to locate the relevant variable occurrences, and returns another number coding the substituted formula. The diagonal function is a special case of this numerical program. This distinction between an expression and its code is the type discipline needed for Gödel's self-reference construction.

Proof. Work by recursion on the syntax code. The predicate $\mathrm{Free}(v, n)$ is computed from the outer constructor: at an atomic formula it is the corresponding free-variable test on terms; Boolean connectives take unions of the free-variable sets; and

$$\mathrm{Free}(v, \ulcorner \forall w\, \varphi \urcorner) \iff v \neq w \wedge \mathrm{Free}(v, \ulcorner \varphi \urcorner).$$

For substitution, recursively descend through the code. At a quantifier $\forall w$, if w would capture a free variable of the substituted term, choose the least fresh variable code and first rename the bound variable. Freshness is a bounded syntactic check. Thus

$$\mathrm{Sub}(n, v, m)$$

– the code obtained by substituting the numeral/term coded by m for free occurrences of variable v in formula code n – is primitive recursive. The diagonal function is the special case

$$d(n) = \mathrm{Sub}(n, v_0, \ulcorner \bar{n} \urcorner),$$

so it too is primitive recursive.

Point specific to this result. Free-variable occurrence and substitution descend recursively through the parse code, with a bound-variable renaming clause for capture avoidance. Diagonalization is then the special substitution that inserts the numeral of the input code into its designated free-variable position. □

Proposition 13.3 (Primitive-recursive coding of axioms for recursive theories)**.** *If T is recursively axiomatized, the predicate $\mathrm{Ax}_T(n)$ saying that n is the Gödel code of a nonlogical axiom of T may be chosen recursive; hence its use in checking finite proof codes is effective.*

Proof. For a recursively axiomatized theory T, fix a primitive-recursive predicate $\mathrm{Ax}_T(n)$ deciding whether n is the code of a nonlogical axiom, together with the already primitive-recursive predicates for logical axiom instances. Then

$$\mathrm{Axiom}_T(n) \iff \mathrm{LogAx}(n) \vee \mathrm{Ax}_T(n)$$

is primitive recursive. If the chosen theory is only c.e., replace the decision predicate by a primitive-recursive enumeration relation $E_T(n, s)$ satisfying

$$n \text{ is an axiom of } T \iff \exists s\, E_T(n, s).$$

A proof certificate can then carry the witness s on each nonlogical axiom line. This distinction is essential: it keeps local proof checking primitive recursive even when bare membership in the axiom set is not decidable.

Point specific to this result. For a recursive axiom set, the axiom predicate can be decided directly from the formula code. The later c.e. case is deliberately different: it records an enumeration witness in the proof certificate instead of assuming decidable axiomhood. □

Worked example

Worked Example 13.4 (Computing a diagonal code)**.** Suppose n codes a one-variable formula $\psi(x)$. The diagonal function first constructs the numeral $\bar{n}$, then applies the already coded capture-avoiding substitution operation to replace the distinguished free variable by $\bar{n}$. Its output is the code of $\psi(\bar{n})$. Every component of this pipeline is primitive recursive.

A useful warning

A Gödel number has no intrinsic mathematical meaning. Meaning enters through the primitive-recursive decoding predicates that recognize well-formed terms, formulas, and substitutions.

Looking ahead

The next chapter, *Proof and provability predicates*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 14

Proof and provability predicates

Definitions and setup

Formal definitions used in this chapter

Proof predicate. $\mathrm{Prf}_T(p, q)$ means that p codes a correct finite T-proof whose final formula has code q.
Provability predicate. $\mathrm{Prov}_T(q) :\equiv \exists p\, \mathrm{Prf}_T(p, q)$.

Fix an effectively axiomatized theory T. $\mathrm{Prf}_T(p, q)$ is the arithmetic relation saying that p codes a correct finite T-proof whose final formula has code q. The provability predicate is $\mathrm{Prov}_T(q) \equiv \exists p\, \mathrm{Prf}_T(p, q)$. Externally, $T \vdash \varphi$ means an actual finite proof exists; internally, $\mathrm{Prov}_T(\ulcorner \varphi \urcorner)$ is an arithmetic sentence. The distinction between these two levels drives all of Volume III.

How this chapter fits together

A c.e. axiom set requires slightly more care than a recursive one: a proof line claiming to be an axiom carries a finite witness that the axiom enumerator produced it. With those witnesses included in the proof certificate, checking a purported proof becomes primitive recursive. Provability is then the existential statement that some code passes this check.

Theorem 14.1 (Effective proof predicate for c.e. theories)**.** *If T is computably enumerable, there is a primitive-recursive certificate relation $\mathrm{Prf}_T(p, q)$ such that*

$$\mathbb{N} \models \mathrm{Prf}_T(p, q)$$

if and only if p codes a correct finite T-derivation whose last formula has Gödel number q.

How to read the symbols: what a proof code certifies

The predicate $\mathrm{Prf}_T(p, q)$ should be read as a verifier, not as an oracle. Its input p is a proposed finite proof certificate and q is the code of the claimed final formula. The verifier checks each line locally: axiom instance, witnessed theory axiom, or valid inference from earlier lines. Because each check is finite and primitive recursive, arithmetic can represent the verifier itself.

Proof. Let T be c.e. and choose a primitive-recursive enumeration relation $E_T(a, s)$ for its axioms. A certified proof code p is a finite sequence of line records. For each line i, the record contains a formula code and a justification. Define $\mathrm{LineOK}_T(p, i)$ by a finite disjunction:

(logical axiom)	$\mathrm{LogAx}((p)_i)$,
(theory axiom)	$\exists s \le p\, E_T((p)_i, s)$ with the stored witness s,
(rule)	the stored earlier indices satisfy the primitive-recursive rule check.

All quantifiers involved in inspecting the finite proof record are bounded by the code or its length. Hence LineOK_T is primitive recursive, and so is

$$\mathrm{Prf}_T(p, q) \iff \mathrm{Seq}(p) \wedge \mathrm{lh}(p) > 0 \wedge (p)_{\mathrm{lh}(p)-1} = q \wedge \forall i < \mathrm{lh}(p)\ \mathrm{LineOK}_T(p, i).$$

Thus even for a c.e. theory, proof correctness has a primitive-recursive certificate relation.

Point specific to this result. The certificate attached to each theory-axiom line is the critical device: it turns c.e. axiom membership into a finitely checkable claim. Once every line has finitely checkable justification data, the whole proof predicate is primitive recursive. □

Theorem 14.2 (Arithmetical representability of $\mathrm{Prf}_T(x, y)$)**.** *The primitive-recursive relation* $\mathrm{Prf}_T(p, q)$ *has an arithmetic representing formula, again denoted* $\mathrm{Prf}_T(x, y)$*, for which true and false numeral instances are correctly proved in the chosen weak arithmetic.*

Proof. The numerical relation $\mathrm{Prf}_T(p, q)$ constructed above is primitive recursive. By the representability theorem for primitive-recursive relations, there is an arithmetic formula, which we denote by the same symbol,

$$\mathrm{Prf}_T(x, y),$$

with the following standard-instance correctness: for every $m, n \in \mathbb{N}$,

$$\mathrm{Prf}_T(m, n) \Rightarrow Q \vdash \mathrm{Prf}_T(\bar{m}, \bar{n}),$$

$$\neg\mathrm{Prf}_T(m, n) \Rightarrow Q \vdash \neg\mathrm{Prf}_T(\bar{m}, \bar{n}).$$

The first argument codes a finite proof certificate; the second codes its concluding formula. This is stronger than merely saying "proofhood is definable": later incompleteness arguments use the first displayed implication for an actual proof code and, in Rosser's argument, the second implication for finitely many smaller nonproof codes.

Point specific to this result. Having shown the external certificate relation primitive recursive, apply the representability theorem to obtain an arithmetic formula $\mathrm{Prf}_T(p, q)$. This is the precise bridge from finite metamathematical proofs to formulas manipulated inside arithmetic. □

Theorem 14.3 (Definability and basic complexity of $\mathrm{Prov}_T(y)$)**.** *Define*

$$\mathrm{Prov}_T(y) \ \equiv\ \exists x\, \mathrm{Prf}_T(x, y).$$

Then Prov_T *is a* Σ_1 *formula representing theoremhood of* T*: for every sentence* φ*,*

$$T \vdash \varphi \quad\Longleftrightarrow\quad \mathbb{N} \models \mathrm{Prov}_T(\ulcorner\varphi\urcorner).$$

Proof. Define the canonical provability predicate by existentially quantifying the represented proof relation:

$$\mathrm{Prov}_T(y) := \exists x\, \mathrm{Prf}_T(x, y).$$

Because Prf_T can be chosen Δ_0 (or primitive-recursive representable in an equivalent bounded form) in the fixed arithmetization, Prov_T is a Σ_1 formula. For every sentence φ, if $T \vdash \varphi$, choose an actual standard natural-number code of a proof n. Then representability gives

$$Q \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner\varphi\urcorner),$$

and hence, by existential introduction,

$$Q \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner).$$

If $Q \subseteq T$, the same provability assertion is therefore derivable in T. This is the external-to-internal bridge used in the first derivability condition; no converse $\mathrm{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi$ is asserted here.

Point specific to this result. Define $\mathrm{Prov}_T(q) \equiv \exists p\, \mathrm{Prf}_T(p, q)$. Since the matrix is primitive recursive and hence representable by a bounded/low-complexity predicate in the chosen coding, provability is Σ_1 in the required presentation. □

Worked example

Worked Example 14.4 (A proof certificate)**.** For a three-line derivation, the code contains the three formula codes together with tags recording whether each line is a logical axiom, a witnessed theory axiom, or follows by a rule from earlier line numbers. Proof checking inspects only finite local data. Consequently $\mathrm{Prf}_T(p, q)$ is arithmetically tame even when membership in the axiom set is only c.e.

A useful warning

The formula $\mathrm{Prov}_T(q) \equiv \exists p\, \mathrm{Prf}_T(p, q)$ expresses existence of a proof code; it does not itself assert that the proved sentence is true.

End-of-volume perspective

Syntax and formal proofs have now become arithmetical objects. The next volume will exploit that internal coding to construct sentences that talk, indirectly but precisely, about their own provability.

Part III

Volume III: Incompleteness, Provability, and Truth

Preface

Self-reference enters only after proof checking and substitution have been arithmetized. The volume keeps consistency, soundness, and truth sharply separated.

Position in the series

Volume III is the first major endpoint of the series: Gödel incompleteness, Löb's theorem, Gödel's second theorem, and Tarski undefinability. The remainder of the series shifts from arithmetic metamathematics to set-theoretic independence.

Sources and further reading

The main proof families are Gödel's incompleteness argument [4], Rosser's consistency-only refinement [5], and Löb's theorem [6].

Notation for this volume

G_T and R_T denote the canonical Gödel and Rosser sentences for T. $\mathrm{Con}(T)$ denotes the canonical arithmetical statement $\neg \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner)$.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 15

Self-reference

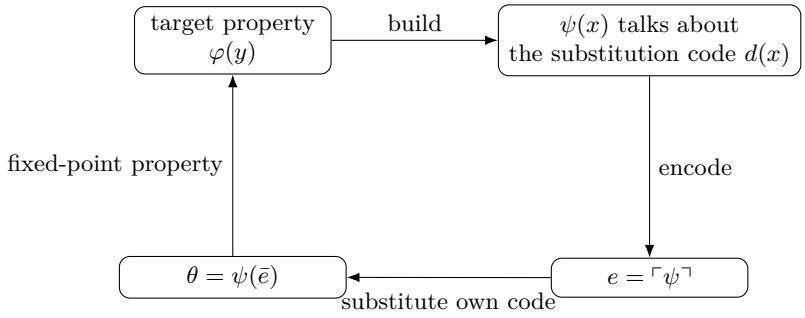

Figure 15.1: The diagonal construction is a feedback loop on codes, not a sentence literally containing itself. The bottom arrow performs the self-substitution; the left arrow records the property thereby obtained.

Definitions and setup

Formal definitions used in this chapter

Fixed point. A sentence θ is a fixed point of $\varphi(y)$ over T if $T \vdash \theta \leftrightarrow \varphi(\ulcorner\theta\urcorner)$.
Object-language code. $\ulcorner\theta\urcorner$ denotes the natural number coding the finite syntactic expression θ; it is not the sentence itself.

Self-reference is implemented syntactically, not by an English paradox. A fixed point for a one-variable formula $\varphi(x)$ is a sentence θ such that the base theory proves $\theta \leftrightarrow \varphi(\ulcorner\theta\urcorner)$. The construction uses the represented substitution function. We reserve the word diagonalization for this coding mechanism and do not assume that arbitrary semantic predicates can be named inside arithmetic.

How this chapter fits together

The diagonal lemma is best understood as a composition theorem for syntax. A formula does not literally contain its own infinitely long description; instead, arithmetic represents the finite substitution function sending a formula code to the code obtained after inserting a numeral. Feeding the code of a carefully chosen formula into that function creates the fixed point.

Lemma 15.1 (Substitution/diagonal function lemma)**.** *The diagonal substitution function*

$$d(n) = \mathrm{Sub}(n, v_0, \mathrm{Num}(n))$$

is primitive recursive. Hence there is a graph formula $D(x, y)$ such that for every standard n,

$$Q \vdash \forall y \left(D(\bar{n}, y) \leftrightarrow y = \overline{d(n)}\right).$$

The universal biconditional, not merely correctness on pairs of standard numerals, is the form used in the diagonal lemma.

Proof.

Formal derivation spine

Formal spine: represent the numerical diagonal function strongly.

$$d(n) = \mathrm{Sub}(n, v_0, \mathrm{Num}(n)), \qquad Q \vdash \forall y\, [D(\bar{n}, y) \leftrightarrow y = \overline{d(n)}].$$

Volume II proves that the code-level substitution function

$$\mathrm{Sub}(e, i, m)$$

is primitive recursive, as is the numeral-code function $n \mapsto \mathrm{Num}(n)$. Closure under composition therefore gives

$$d(n) := \mathrm{Sub}(n, 0, \mathrm{Num}(n))$$

as a primitive-recursive numerical function.

Apply strong representability of primitive-recursive functions. There is an arithmetic formula $D(x, y)$ such that, for each standard n,

$$Q \vdash \forall y \left(D(\bar{n}, y) \leftrightarrow y = \overline{d(n)}\right). \tag{1}$$

Equation (1) says more than $Q \vdash D(\bar{n}, \overline{d(n)})$: it excludes nonstandard alternative witnesses for y when the first argument is the fixed numeral $\bar{n}$. That uniqueness is exactly what permits an existential quantifier over $D(\bar{n}, y)$ to be replaced, inside Q, by substitution of the single numeral $\overline{d(n)}$. □

Theorem 15.2 (Fixed-point (diagonal) lemma)**.** *Let T be any theory in the language of arithmetic with $Q \subseteq T$. For every arithmetic formula $\varphi(y)$ with at most the displayed free variable, there exists an arithmetic sentence θ such that*

$$T \vdash \theta \leftrightarrow \varphi(\ulcorner \theta \urcorner).$$

The construction is effective from a Gödel code of φ.

Uses: the strong representability of the diagonal substitution function and the preceding diagonal-graph lemma.

Commented symbolic trace: the diagonal lemma as a typed program

Given a one-variable formula $\varphi(y)$, form a new formula $\psi(x)$ that says: "apply the substitution program to x, obtaining code y, and then $\varphi(y)$ holds." Now feed ψ's own code $e = \ulcorner \psi \urcorner$ into ψ. The numerical identity $d(e) = \ulcorner \psi(\bar{e}) \urcorner$ is what turns ordinary substitution into self-reference. Nothing in the construction asks a formula to literally contain itself as a substring.

Proof.

Formal derivation spine

Formal spine: feed a formula its own code.

$$\psi(x) := \exists y(D(x, y) \wedge \varphi(y)), \quad e := \ulcorner \psi \urcorner, \quad \theta := \psi(\bar{e}), \quad d(e) = \ulcorner \theta \urcorner.$$

Let $D(x, y)$ be the strongly representing graph formula from the preceding lemma. Given $\varphi(y)$, define

$$\psi(x) := \exists y \big(D(x, y) \wedge \varphi(y)\big).$$

Let $e := \ulcorner \psi \urcorner$ and put

$$\theta := \psi(\bar{e}).$$

By the definition of the numerical diagonal function,

$$d(e) = \ulcorner \psi(\bar{e}) \urcorner = \ulcorner \theta \urcorner. \tag{2}$$

Strong representability gives

$$Q \vdash \forall y \big(D(\bar{e}, y) \leftrightarrow y = \overline{d(e)}\big). \tag{3}$$

Using (3) in the definition of θ, first-order logic proves in Q

$$\theta \leftrightarrow \varphi(\overline{d(e)}). \tag{4}$$

The two numerals in (2) are literally the same numeral in the chosen coding, so (4) is

$$Q \vdash \theta \leftrightarrow \varphi(\ulcorner \theta \urcorner).$$

Since $Q \subseteq T$, the same biconditional is provable in T. Every step occurs at a specified level: $d(e)$ is an external numerical computation, (3) is its internal arithmetic representation, and the final displayed formula is a theorem of T. □

Remark 15.3 (Source note)**.** The fixed-point mechanism is the standard arithmetized diagonal argument underlying Gödelś 1931 paper; compare [4].

Proposition 15.4 (Uniform diagonal lemma with parameters)**.** *For every formula $\varphi(y, \bar{z})$ there is a formula $\theta(\bar{z})$ such that*

$$T \vdash \forall \bar{z}\, [\theta(\bar{z}) \leftrightarrow \varphi(\ulcorner \theta \urcorner, \bar{z})].$$

Only the code variable is diagonalized; the parameters $\bar{z}$ remain free.

Proof.

Formal derivation spine

Formal spine: diagonalize only the code variable.

$$\psi(x, \bar{z}) := \exists u(D(x, u) \wedge \varphi(u, \bar{z})), \quad T \vdash \forall \bar{z}\, [\theta(\bar{z}) \leftrightarrow \varphi(\ulcorner \theta \urcorner, \bar{z})].$$

Keep the parameter tuple $\bar{z}$ free and diagonalize only the code variable. Define

$$\psi(x, \bar{z}) := \exists u \big(D(x, u) \wedge \varphi(u, \bar{z})\big), \qquad e := \ulcorner \psi \urcorner,$$

and set

$$\theta(\bar{z}) := \psi(\bar{e}, \bar{z}).$$

The numerical substitution identity is independent of any values assigned to $\bar{z}$:

$$d(e) = \ulcorner \theta \urcorner. \tag{1}$$

Strong representability gives

$$Q \vdash \forall u \big(D(\bar{e}, u) \leftrightarrow u = \overline{d(e)}\big). \tag{2}$$

Substituting (2) into the defining formula for θ, while leaving $\bar{z}$ free, yields

$$Q \vdash \forall \bar{z} \big(\theta(\bar{z}) \leftrightarrow \varphi(\overline{d(e)}, \bar{z})\big).$$

Now use (1) to identify the displayed numeral with $\ulcorner \theta \urcorner$. Because $Q \subseteq T$,

$$T \vdash \forall \bar{z} \big(\theta(\bar{z}) \leftrightarrow \varphi(\ulcorner \theta \urcorner, \bar{z})\big).$$

The code is the code of the open formula $\theta(\bar{z})$; it does not vary with the parameter values. □

Worked Theorem 15.5 (Explicit construction of a self-referential arithmetic sentence)**.** *Given a concrete arithmetic predicate $\varphi(y)$, the diagonal construction effectively produces a specific sentence θ and a formal T-derivation of $\theta \leftrightarrow \varphi(\ulcorner\theta\urcorner)$.*

Proof.

Formal derivation spine

Formal spine for Explicit construction of a self-referential arithmetic sentence.

$$a := \ulcorner\varphi(y)\urcorner,$$

$$\psi(x) := \exists u\,(D(x,u) \wedge \varphi(u)).$$

$$m := d(e) = \mathrm{Sub}(e, 0, \mathrm{Num}(e)).$$

This is the algorithmic version of the preceding theorem. Starting from the concrete code

$$a := \ulcorner\varphi(y)\urcorner,$$

compute the code e of

$$\psi(x) := \exists u\,(D(x,u) \wedge \varphi(u)).$$

Because formula formation and substitution on codes are primitive recursive, e is effectively computable from a. Next compute

$$m := d(e) = \mathrm{Sub}(e, 0, \mathrm{Num}(e)).$$

Decode m; by construction it is the code of the closed sentence

$$\theta := \psi(\bar{e}).$$

The represented computation gives

$$T \vdash D(\bar{e}, \bar{m}),$$

so unfolding the definition of ψ yields

$$T \vdash \theta \leftrightarrow \varphi(\bar{m}).$$

Finally $m = \ulcorner\theta\urcorner$, hence

$$T \vdash \theta \leftrightarrow \varphi(\ulcorner\theta\urcorner).$$

Every step is an effective operation on finite codes. *Specialization to this result.* To obtain an explicit self-referential sentence, choose a concrete one-variable target predicate and follow the fixed-point construction literally. The point is that no quotation device is added to the object language: every reference to syntax is mediated by numerals and the represented substitution relation.

Detailed formal verification. The coding argument is carried out on natural numbers. Every syntactic constructor has a primitive-recursive code operation, so substitution is obtained by course-of-values recursion on the parse code. The crucial closure statement is

$$\text{primitive recursive syntax operations} \Longrightarrow \text{representable numerical graph in } Q.$$

Whenever a formula code is transformed, the proof identifies the corresponding numerical function first and only then replaces its graph by an arithmetic formula. This prevents the common type error of treating a formula as if it were its Gödel number. □

Worked example

Worked Example 15.6 (A schematic fixed point)**.** Given a one-variable formula $\varphi(y)$, let $\psi(x)$ say that φ holds of the result of diagonalizing x. Substitute the numeral for $\ulcorner\psi\urcorner$ into ψ itself. The resulting sentence θ is provably equivalent to $\varphi(\ulcorner\theta\urcorner)$. The apparent circularity is replaced by an ordinary finite substitution calculation.

A useful warning

The fixed-point lemma is syntactic. No truth predicate and no semantic liar sentence is used in its construction.

Looking ahead

The next chapter, *Gödel's first theorem*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 16

Gödel's first theorem

Definitions and setup

Formal definitions used in this chapter

Gödel sentence. G_T is a fixed point for $\neg\mathrm{Prov}_T(y)$.
ω-consistency. T is ω-consistent if there is no formula $\varphi(x)$ such that $T \vdash \exists x\varphi(x)$ and $T \vdash \neg\varphi(\bar{n})$ for every $n \in \mathbb{N}$.

For an effectively axiomatized theory T, the canonical Gödel sentence G_T is a fixed point of $\neg\mathrm{Prov}_T(x)$. Consistency means there is no sentence whose proof and negated proof both exist. ω-consistency is stronger: T cannot prove an existential statement while refuting every numeral instance of its matrix. The original Gödel proof uses ordinary consistency for one direction and ω-consistency for the other; the next chapter replaces the latter with Rosser's argument.

How this chapter fits together

Apply diagonalization to non-provability to obtain G_T. The two halves of the classical argument have different hypotheses. Consistency alone rules out a proof of G_T; Gödel's original argument uses the stronger ω-consistency assumption to rule out a proof of $\neg G_T$. Rosser's refinement in the next chapter removes that asymmetry.

Proposition 16.1 (Construction of the canonical Gödel sentence G_T)**.** *Let T be c.e. and extend Q. There is a sentence G_T satisfying*

$$T \vdash G_T \leftrightarrow \neg\mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

How to read the symbols: what G_T says and where it says it

Inside arithmetic, G_T is an ordinary sentence. The displayed equivalence

$$T \vdash G_T \leftrightarrow \neg\mathrm{Prov}_T(\ulcorner G_T \urcorner)$$

is a theorem of T about the numeral coding G_T. Outside the theory, we interpret Prov_T as the canonical proof predicate. Keep those levels separate: the fixed-point theorem gives the internal equivalence; consistency is an external hypothesis used later to infer unprovability.

Proof.

Formal derivation spine

Formal spine for Construction of the canonical Gödel sentence G_T.

$$T \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

$$T \vdash \varphi, \qquad \mathrm{Prov}_T(\ulcorner \varphi \urcorner), \qquad \mathbb{N} \models \varphi.$$

Apply the diagonal lemma to $\varphi(y) \equiv \neg \mathrm{Prov}_T(y)$. We obtain a sentence G_T with

$$T \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

This equivalence is the precise formal content of the informal phrase "G_T says that it is not provable". The construction depends only on the effective proof predicate and diagonal substitution, both established earlier.

Further explanation. The target predicate is chosen to be $\neg \mathrm{Prov}_T(y)$, not an informal English sentence. The diagonal lemma then supplies a sentence whose own code occupies the argument place y. Consequently every later use of G_T rests on the formal equivalence displayed above. In particular, saying that G_T is 'true' requires an external hypothesis about the actual proof relation on standard natural numbers; that semantic conclusion is not part of the fixed-point construction itself. □

Theorem 16.2 (Consistency implies T does not prove G_T)**.** *Let T be a consistent c.e. extension of Q, and let G_T be its canonical Gödel sentence. Then*

$$T \nvdash G_T.$$

Proof.

Formal derivation spine

Formal spine: consistency excludes a proof of the Gödel sentence.

$$T \vdash G_T \;\&\; n \text{ codes that proof} \Longrightarrow T \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner G_T \urcorner) \Longrightarrow T \vdash \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

But the fixed point gives

$$T \vdash G_T \to \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner),$$

so $T \vdash G_T$ would make T inconsistent.

Assume T is consistent and $T \vdash G_T$. Let n be the actual code of such a proof. Representability of the proof relation yields $T \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner G_T \urcorner)$, hence $T \vdash \mathrm{Prov}_T(\ulcorner G_T \urcorner)$. On the other hand the fixed-point equivalence and $T \vdash G_T$ yield $T \vdash \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner)$. Thus T proves a contradiction, contrary to consistency. Therefore $T \nvdash G_T$.

Point specific to this result. This half uses only ordinary consistency: an actual proof of G_T can be arithmetized into a proof of $\mathrm{Prov}_T(\ulcorner G_T \urcorner)$, while the fixed-point equivalence gives its negation. No soundness assumption is used.

Further explanation. Notice the two levels in the contradiction. The natural number n is an external, actual proof code. Representability lets the theory verify the finite fact that n is a proof, producing the internal sentence $\mathrm{Prov}_T(\ulcorner G_T \urcorner)$. The fixed point converts the assumed internal theorem G_T into the negation of that same provability assertion. Consistency is therefore enough because the argument starts from an actual derivation, not from an assumption that every theorem of T is true. □

Theorem 16.3 (Omega-consistency implies T does not prove not-G_T)**.** *Let T be a c.e. extension of Q that is ω-consistent. For its canonical Gödel sentence,*

$$T \nvdash \neg G_T.$$

Commented symbolic trace: where ω-consistency enters

Assuming $T \vdash \neg G_T$ gives one internal existential statement: $T \vdash \exists p\, \mathrm{Prf}_T(p, \ulcorner G_T \urcorner)$. Ordinary consistency separately tells us that no standard natural number actually codes a proof of G_T; numeralwise correctness therefore gives $T \vdash \neg \mathrm{Prf}_T(\bar{n}, \ulcorner G_T \urcorner)$ for every standard n. ω-consistency forbids exactly this pattern: proving an existential while refuting every standard numeral instance.

Proof.

Formal derivation spine

Formal spine for Omega-consistency implies T does not prove not-G_T.

$$T \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

$$T \vdash \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

$$T \vdash \exists p\, \mathrm{Prf}_T(p, \ulcorner G_T \urcorner).$$

Assume that T is ω-consistent and, toward a contradiction, that $T \vdash \neg G_T$. From the fixed-point equivalence

$$T \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

Because we are assuming $T \vdash \neg G_T$, the left side of this equivalence is false in the theory; propositional reasoning inside T therefore yields the negation of the right-hand side's negation, namely

$$T \vdash \mathrm{Prov}_T(\ulcorner G_T \urcorner).$$

Now unfold the definition of the provability predicate: $\mathrm{Prov}_T(q)$ is the existential statement saying that some number codes a T-proof of the formula with code q. Thus the previous line is exactly

$$T \vdash \exists p\, \mathrm{Prf}_T(p, \ulcorner G_T \urcorner).$$

On the other hand, ordinary consistency already implies $T \nvdash G_T$. Hence no standard natural number is in fact the code of a T-proof of G_T. For each standard n, the primitive-recursive proof predicate correctly recognizes this finite fact, so numeralwise representability gives

$$T \vdash \neg \mathrm{Prf}_T(\bar{n}, \ulcorner G_T \urcorner).$$

Thus T proves an existential statement $\exists p\, \mathrm{Prf}_T(p, \ulcorner G_T \urcorner)$ while proving the negation of every standard numeral instance. This is exactly the configuration forbidden by ω-consistency. Therefore $T \nvdash \neg G_T$.

Why the stronger hypothesis enters. The first half of Gödel's argument turns an actual proof into an internal provability assertion and therefore needs only consistency. The present half must rule out a theory that proves "some proof code exists" while rejecting each standard candidate separately; ω-consistency is used precisely at that final step. □

Theorem 16.4 (Gödel's original first incompleteness theorem)**.** *Every effectively axiomatized ω-consistent theory $T \supseteq Q$ is incomplete: there is a sentence G_T for which*

$$T \nvdash G_T \qquad \text{and} \qquad T \nvdash \neg G_T.$$

Uses: the fixed-point lemma, representability of the proof predicate, and the two preceding unprovability arguments.

Proof.

Formal derivation spine

Formal spine: combine the two asymmetric halves of Gödel I.

$$\omega\text{-Con}(T) \Longrightarrow \text{Con}(T),$$

$$\text{Con}(T) \Longrightarrow T \nvdash G_T, \qquad \omega\text{-Con}(T) \Longrightarrow T \nvdash \neg G_T.$$

Therefore

$$T \nvdash G_T \quad \& \quad T \nvdash \neg G_T.$$

Combine the preceding two propositions. Under consistency, G_T is not provable; under ω-consistency, its negation is not provable. Therefore an effectively axiomatized theory satisfying the stated arithmetical strength and ω-consistency is incomplete. The proof records exactly where the stronger hypothesis enters: only the refutation half of Gödel's original sentence, not the unprovability half.

Further explanation. The hypotheses should be read asymmetrically. Ordinary consistency already excludes $T \vdash G_T$. The stronger ω-consistency assumption is used only to exclude $T \vdash \neg G_T$, because that case produces an existential proof assertion together with the negation of each standard numeral instance. Rosser's refinement will replace this second argument, which is why modern formulations of the first incompleteness theorem usually require only ordinary consistency for a suitably strengthened self-referential sentence. □

Remark 16.5 (Source note)**.** This is the hypothesis structure of Gödelś original first incompleteness argument: consistency for the unprovability of G_T, and the stronger ω-consistency assumption for the unprovability of $\neg G_T$; see [4].

Worked example

Worked Example 16.6 (What a proof of the Gödel sentence would cause)**.** If n were an actual T-proof of G_T, representability would let T verify $\text{Prf}_T(\bar{n}, \ulcorner G_T \urcorner)$, hence $\text{Prov}_T(\ulcorner G_T \urcorner)$. But the fixed-point equivalence turns G_T itself into $\neg\text{Prov}_T(\ulcorner G_T \urcorner)$. Thus a proof of G_T would manufacture an outright contradiction inside T.

A useful warning

Do not replace ω-consistency by ordinary consistency in Gödel's original refutation argument. That strengthening is exactly what Rosser's proof is designed to avoid.

Looking ahead

The next chapter, *Rosser and essential undecidability*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 17

Rosser and essential undecidability

Definitions and setup

Formal definitions used in this chapter

Rosser sentence. A self-referential sentence comparing the least proof of itself with proofs of its negation.
Essential undecidability. A theory T is essentially undecidable if every consistent recursively axiomatized extension of T has an undecidable theorem set.

Rosser's idea compares the numerical sizes of a proof of a sentence and a proof of its negation. Because "there is a proof with code at most n" is bounded, the weak arithmetic can verify the finite comparison needed in the least-proof argument. This lets ordinary consistency replace ω-consistency. Essential undecidability means that every consistent c.e. extension of a given weak theory is incomplete.

How this chapter fits together

Rosser compares proof codes for a sentence and its negation. The sentence is engineered so that any proof on one side predicts a shorter proof on the other. Minimal proof codes then turn ordinary consistency into the required contradiction. Essential undecidability follows because every consistent c.e. extension of Q inherits this obstruction.

Lemma 17.1 (Rosser proof-comparison predicate)**.** *There is a primitive-recursive relation* $\mathrm{RPrf}_T(p, q)$ *comparing proof codes so that the statement "p is a proof of the sentence coded by q, and no code $r \leq p$ is a proof of its negation" is arithmetically representable.*

Proof.

Formal derivation spine

Formal spine for Rosser proof-comparison predicate.

$$\mathrm{RPrf}_T(p, q) :\Longleftrightarrow \mathrm{Prf}_T(p, q) \wedge \forall r \leq p\, \neg \mathrm{Prf}_T(r, \mathrm{Neg}(q)).$$

$$\mathrm{RPrf}_T \in \mathrm{PR}.$$

$$\mathrm{Prf}_T(\bar{q}, \ulcorner \sigma \urcorner) \quad \text{or} \quad \neg \mathrm{Prf}_T(\bar{q}, \ulcorner \sigma \urcorner) \qquad (q < p).$$

Let $\mathrm{Neg}(q)$ be the primitive-recursive function sending the code of a sentence to the code of its negation. Define

$$\mathrm{RPrf}_T(p,q) :\Longleftrightarrow \mathrm{Prf}_T(p,q) \wedge \forall r \le p\, \neg \mathrm{Prf}_T(r, \mathrm{Neg}(q)).$$

The first conjunct is primitive recursive by Volume II. The function Neg is primitive recursive because it is one syntactic constructor, and bounded universal quantification preserves primitive recursiveness. Therefore

$$\mathrm{RPrf}_T \in \mathrm{PR}.$$

Representability provides an arithmetic formula, again written $\mathrm{RPrf}_T(p,q)$, that correctly decides every numeral instance in Q. The relation does not yet assert a self-referential sentence; it merely formalizes the finite comparison "p proves the sentence coded by q, and no proof of its negation has code at most p." The next result diagonalizes this comparison predicate. *Specialization to this result.* The Rosser comparison relation is bounded in the competing proof code: given a candidate proof of one side, arithmetic checks whether a smaller proof of the negation exists. Because the search is bounded by the candidate code, weak arithmetic can verify each concrete comparison needed in the minimal-proof argument. □

Proposition 17.2 (Construction of the Rosser sentence R_T)**.** *For every c.e. theory $T \supseteq Q$ there is a Rosser sentence R_T such that*

$$T \vdash R_T \leftrightarrow \forall p\big(\mathrm{Prf}_T(p, \ulcorner R_T \urcorner) \to \exists q \le p\, \mathrm{Prf}_T(q, \ulcorner \neg R_T \urcorner)\big).$$

Proof.

Formal derivation spine

Formal spine for Construction of the Rosser sentence R_T.

$$\rho(y) := \forall p\Big(\mathrm{Prf}_T(p,y) \to \exists q \le p\, \mathrm{Prf}_T(q, \mathrm{Neg}(y))\Big).$$

$$T \vdash R_T \leftrightarrow \rho(\ulcorner R_T \urcorner).$$

$$T \vdash R_T \leftrightarrow \forall p\Big(\mathrm{Prf}_T(p, \ulcorner R_T \urcorner) \to \exists q \le p\, \mathrm{Prf}_T(q, \ulcorner \neg R_T \urcorner)\Big).$$

Use the proof-comparison relation to form the one-variable formula

$$\rho(y) := \forall p\Big(\mathrm{Prf}_T(p,y) \to \exists q \le p\, \mathrm{Prf}_T(q, \mathrm{Neg}(y))\Big).$$

By the fixed-point lemma there is a sentence R_T such that

$$T \vdash R_T \leftrightarrow \rho(\ulcorner R_T \urcorner).$$

Expanding ρ, this is

$$T \vdash R_T \leftrightarrow \forall p\Big(\mathrm{Prf}_T(p, \ulcorner R_T \urcorner) \to \exists q \le p\, \mathrm{Prf}_T(q, \ulcorner \neg R_T \urcorner)\Big).$$

This biconditional is the defining property of the Rosser sentence. The construction uses only effective proof coding and diagonalization; consistency is not needed to construct R_T. Consistency enters only in the following theorem, where least proof codes are compared. *Specialization to this result.* Diagonalize the Rosser comparison predicate rather than ordinary non-provability. The resulting sentence says that every proof of itself is accompanied by a no-larger proof of its negation, which is precisely the asymmetry needed for the least-proof contradiction. □

Theorem 17.3 (Rosser incompleteness theorem from ordinary consistency)**.** *If $T \supseteq Q$ is c.e. and consistent, then its Rosser sentence is undecidable in T:*

$$T \nvdash R_T \qquad \text{and} \qquad T \nvdash \neg R_T.$$

Proof.

Formal derivation spine

Formal spine: the two least-proof contradictions. For the Rosser sentence R,

$$T \vdash R \leftrightarrow \forall p\big(\mathrm{Prf}_T(p, \ulcorner R \urcorner) \to \exists q \le p\, \mathrm{Prf}_T(q, \ulcorner \neg R \urcorner)\big). \tag{R}$$

A least proof of R produces a no-larger proof of $\neg R$; a least proof of $\neg R$, together with the finite verification of all smaller codes, makes T prove the right-hand side of (R).

Assume first that $T \vdash R$, and let m be the least code of a T-proof of R. Since m is an actual proof code, numeralwise representability gives

$$T \vdash \mathrm{Prf}_T(\bar{m}, \ulcorner R \urcorner). \tag{1}$$

From (R) and (1), T proves

$$\exists q \le \bar{m}\, \mathrm{Prf}_T(q, \ulcorner \neg R \urcorner). \tag{2}$$

The bounded formula in (2) is true only if some standard $q \le m$ is an actual proof code of $\neg R$: bounded quantifiers below a standard numeral range over finitely many standard candidates, and the proof predicate is correct on each such candidate. Thus T would have proofs of both R and $\neg R$, contrary to consistency. Hence $T \nvdash R$.

Now assume that $T \vdash \neg R$, and let n be the least code of such a proof. For every standard $m < n$, consistency implies that m is not a proof of R. Correctness of the primitive-recursive proof predicate yields, for each of the finitely many $m < n$,

$$Q \vdash \neg \mathrm{Prf}_T(\bar{m}, \ulcorner R \urcorner).$$

Combining these finitely many derivations gives

$$Q \vdash \forall m < \bar{n}\, \neg \mathrm{Prf}_T(m, \ulcorner R \urcorner). \tag{3}$$

Because n is an actual proof of $\neg R$,

$$Q \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner \neg R \urcorner). \tag{4}$$

We now verify the right-hand side of (R) inside Q. Fix p. If $p < \bar{n}$, (3) makes the antecedent $\mathrm{Prf}_T(p, \ulcorner R \urcorner)$ false. If $\bar{n} \le p$, the witness $q = \bar{n}$, together with (4), satisfies the consequent. Since Q proves the bounded order dichotomy

$$p < \bar{n} \ \vee \ \bar{n} \le p,$$

it follows that Q, hence T, proves the Rosser condition in (R). Therefore $T \vdash R$, contradicting $T \vdash \neg R$ and consistency. Thus $T \nvdash \neg R$. Ordinary consistency suffices in both directions; no soundness or ω-consistency assumption is used. □

Remark 17.4 (Source note)**.** Rosserś refinement replaces the ω-consistency half by a bounded comparison of proof codes, reducing the hypothesis to ordinary consistency; see [5].

Theorem 17.5 (Essential undecidability of Q and its effective consequences)**.** *Every consistent c.e. extension $T \supseteq Q$ is incomplete and has undecidable theorem set. Consequently Q is essentially undecidable: no consistent recursive theory extending Q is complete.*

Proof.

Formal derivation spine

Formal spine: essential undecidability. For every consistent c.e. $S \supseteq Q$, Rosser gives

$$S \nvdash R_S \quad \& \quad S \nvdash \neg R_S,$$

so S is incomplete. If the theorem set of such an S were recursive, a recursive greedy completion would produce a consistent complete c.e. extension $S^* \supseteq Q$, contradicting Rosser again.

Let $S \supseteq Q$ be consistent and computably enumerable. Rosser's theorem, applied to S, produces a sentence R_S with

$$S \nvdash R_S, \qquad S \nvdash \neg R_S.$$

Thus every consistent c.e. extension of Q is incomplete.

It remains to prove the stronger decision-theoretic assertion. Suppose, toward a contradiction, that the theorem set

$$\mathrm{Thm}(S) := \{\ulcorner \sigma \urcorner : S \vdash \sigma\}$$

is recursive. Enumerate all sentences as $\varphi_0, \varphi_1, \ldots$, and construct finite sets F_n recursively. Put $F_0 = \varnothing$. Having constructed F_n, decide whether

$$S \cup F_n \cup \{\varphi_n\}$$

is consistent. Since F_n is finite, the Deduction Theorem gives the exact test

$$S \cup F_n \cup \{\varphi_n\} \text{ inconsistent} \iff S \vdash \neg\big(\bigwedge F_n \wedge \varphi_n\big).$$

The right-hand side is decidable because $\mathrm{Thm}(S)$ is recursive. If the extension is consistent, set $F_{n+1} = F_n \cup \{\varphi_n\}$; otherwise set $F_{n+1} = F_n \cup \{\neg\varphi_n\}$. The second choice is consistent because the first choice was detected inconsistent and $S \cup F_n$ is consistent.

Let

$$S^* := S \cup \bigcup_{n<\omega} F_n.$$

Every finite subset of S^* lies in some consistent stage, so S^* is consistent. By construction, for every sentence φ_n, exactly one of $\varphi_n, \neg\varphi_n$ is added, so S^* is complete. The construction is recursive from the assumed decision procedure for $\mathrm{Thm}(S)$; hence S^* is a computably enumerable extension of Q. Rosser's theorem applied to S^* says that S^* is incomplete, a contradiction. Therefore $\mathrm{Thm}(S)$ is undecidable.

Consequently no consistent recursive theory extending Q can be complete, and indeed no consistent c.e. extension of Q has a decidable theorem set. This is the essential-undecidability property used later. □

Worked example

Worked Example 17.6 (The least-proof trick)**.** Assume a Rosser sentence R has a least proof code m. Its defining property yields a proof of $\neg R$ with code no larger than m, contradicting consistency. Conversely, if $\neg R$ has least proof code n, the finitely many smaller candidate proofs can be checked one by one inside weak arithmetic, forcing the Rosser condition and hence R.

A useful warning

The argument uses the arithmetic ordering of proof codes, not an assumption that proofs arrive in chronological order from some enumerator.

Looking ahead

The next chapter, *Consistency and soundness notions*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 18

Consistency and soundness notions

Definitions and setup

Formal definitions used in this chapter

Soundness. Every theorem of T is true in $\mathbb{N}$.
Σ_1-soundness / 1-consistency. No false Σ_1 sentence is provable.
Consistency. $T \nvdash 0 = 1$.

Consistency, 1-consistency, ω-consistency, Σ_1-soundness, and full soundness are different hypotheses. Soundness refers to truth in the standard model; consistency is purely syntactic. The implications among them are one-way in general. We use the weakest condition sufficient for each theorem and state explicitly when an external truth claim about the Gödel sentence requires more than mere consistency.

How this chapter fits together

Several nearby notions are often conflated. Consistency forbids proving a contradiction. 1-consistency and Σ_1-soundness control false existential arithmetic claims. ω-consistency rules out proving an existential while refuting every numeral instance. Full soundness says every theorem is true in the intended model. The implication diagram matters because different incompleteness arguments use different points on it.

Proposition 18.1 (Implication chain among soundness, Sigma-1 soundness, 1-consistency, omega-consistency, and consistency)**.** *For extensions of Q, the following implications hold:*

$$\textit{sound} \Rightarrow \omega\textit{-consistent} \Rightarrow 1\textit{-consistent} \Longleftrightarrow \Sigma_1\textit{-sound} \Rightarrow \textit{consistent}.$$

Here 1-consistency means that no false Σ_1 sentence is provable.

Proof.

Formal derivation spine

Formal spine for Implication chain among soundness, Sigma-1 soundness, 1-consistency, omega-consistency, and consistency.

$$\mathrm{Con}(T) \iff T \nvdash 0 = 1,$$

$$T \vdash \sigma \ \wedge \ \sigma \in \Sigma_1 \Longrightarrow \mathbb{N} \models \sigma.$$

We use the following precise definitions. A theory is sound if every theorem is true in the standard model; it is Σ_1-sound if this is required only for Σ_1-sentences. It is 1-consistent if it never proves a false existential

statement whose matrix is decidable in the required weak arithmetic sense, and it is ω-consistent if there is no formula $\varphi(x)$ such that the theory proves $\exists x\, \varphi(x)$ while proving $\neg\varphi(\bar{n})$ for every numeral $\bar{n}$. Full soundness plainly implies Σ_1-soundness. A false Σ_1-sentence has no standard witness, so Σ_1-soundness implies 1-consistency. If a theory were ω-consistent but failed 1-consistency, the false existential assertion together with numeralwise correctness of its decidable matrix would give exactly the forbidden ω-inconsistency pattern. Finally, 1-consistency implies ordinary consistency, since an inconsistent theory proves every sentence, including a false Σ_1-sentence. The reverse implications are not used without additional hypotheses; later examples such as consistent unsound extensions show why the distinctions matter. □

Proposition 18.2 (Strictness/counterexample analysis for the consistency notions)**.** *Let $T \supseteq I\Sigma_1$ be c.e. and consistent. Then*

$$T^* := T + \neg\mathrm{Con}(T)$$

is consistent but not Σ_1-sound: consistency follows because otherwise $T \vdash \mathrm{Con}(T)$, contrary to Gödel II, while $\neg\mathrm{Con}(T)$ is a false Σ_1-sentence in $\mathbb{N}$. Consequently ordinary consistency does not imply Σ_1-soundness (equivalently, 1-consistency in the terminology fixed here). The other nonreverse implications discussed below are separated by similarly explicit examples.

Proof.

Formal derivation spine

Formal spine for Strictness/counterexample analysis for the consistency notions.

$$T \nvdash \mathrm{Con}(T).$$

$$T \vdash \mathrm{Con}(T),$$

$$T^* := T + \neg\mathrm{Con}(T)$$

The nonreverse implication from consistency to Σ_1-soundness has a concrete witness. Let $T \supseteq I\Sigma_1$ be consistent and c.e. Gödel II gives

$$T \nvdash \mathrm{Con}(T).$$

If $T + \neg\mathrm{Con}(T)$ were inconsistent, the deduction theorem would yield

$$T \vdash \mathrm{Con}(T),$$

a contradiction. Hence

$$T^* := T + \neg\mathrm{Con}(T)$$

is consistent. But because T is actually consistent, the standard model satisfies

$$\mathbb{N} \models \mathrm{Con}(T),$$

so

$$\mathbb{N} \models \neg\big(\neg\mathrm{Con}(T)\big).$$

The sentence $\neg\mathrm{Con}(T) \equiv \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner)$ is Σ_1, and T^* proves it. Therefore T^* is not Σ_1-sound. This explicitly separates ordinary consistency from 1-consistency/Σ_1-soundness. *Point specific to this result.* The implication diagram should not be read as a chain of equivalences. Consistent but unsound extensions supply the basic counterexamples, while separating 1- from ω-consistency requires attention to which existential matrices are decidable in the base theory. □

Theorem 18.3 (External truth of canonical Gödel/Rosser sentences under the precise hypotheses)**.** *Let $T \supseteq Q$ be a consistent computably enumerable theory, and use the canonical proof predicate fixed in Volume II. Then the canonical Gödel sentence G_T and the canonical Rosser sentence R_T are both true in the standard model:*

$$\mathbb{N} \models G_T \wedge R_T.$$

These are external semantic conclusions and are not assertions that either sentence is provable in T.

Proof.

Formal derivation spine

Formal spine: external absence of proof codes plus a Q-provable fixed point.

$$T \nvdash G_T \Longrightarrow \mathbb{N} \models \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner),$$

$$Q \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner) \Longrightarrow \mathbb{N} \models G_T.$$

By consistency and the first half of Gödel's theorem,

$$T \nvdash G_T.$$

Therefore no standard natural number codes a T-proof of G_T, so correctness of the canonical proof predicate on standard numbers gives

$$\mathbb{N} \models \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner). \tag{1}$$

The fixed-point biconditional was proved already in Q, not merely in the possibly unsound extension T:

$$Q \vdash G_T \leftrightarrow \neg \mathrm{Prov}_T(\ulcorner G_T \urcorner). \tag{2}$$

Every axiom of Q is true in $\mathbb{N}$, so soundness applied to (2), together with (1), yields

$$\mathbb{N} \models G_T.$$

This explicitly avoids the invalid inference that an arbitrary theorem of a consistent theory must be true in $\mathbb{N}$.

For the Rosser sentence, consistency and Rosser's theorem give

$$T \nvdash R_T \quad \text{and} \quad T \nvdash \neg R_T.$$

Hence for every standard p, $\mathbb{N} \models \neg \mathrm{Prf}_T(p, \ulcorner R_T \urcorner)$. The right-hand side of the Rosser fixed-point formula

$$\forall p \Big(\mathrm{Prf}_T(p, \ulcorner R_T \urcorner) \to \exists q \le p \, \mathrm{Prf}_T(q, \ulcorner \neg R_T \urcorner) \Big)$$

is therefore true in $\mathbb{N}$. Its biconditional with R_T is again provable in Q, hence true in $\mathbb{N}$. Consequently $\mathbb{N} \models R_T$. □

Proposition 18.4 (Why syntactic consistency and semantic truth are distinct notions)**.** *For a theory T, syntactic consistency is the assertion that no contradiction is derivable in T, whereas truth of a sentence φ is relative to a structure such as $\mathbb{N}$. In particular, a consistent theory may contain false sentences, and a true sentence may be unprovable in a consistent incomplete theory.*

Proof.

Formal derivation spine

Formal spine for Why syntactic consistency and semantic truth are distinct notions.

$$\mathrm{Con}(T) \Longleftrightarrow \neg \exists p \, \mathrm{Prf}_T(p, \ulcorner 0 = 1 \urcorner).$$

$$\mathbb{N} \models \varphi.$$

$$T \nvdash \neg \varphi$$

Syntactic consistency is a property of the derivability relation:

$$\mathrm{Con}(T) \Longleftrightarrow \neg \exists p \, \mathrm{Prf}_T(p, \ulcorner 0 = 1 \urcorner).$$

Semantic truth is evaluated in a structure:

$$\mathbb{N} \models \varphi.$$

The two predicates have different arguments and live at different levels. In particular, from

$$T \nvdash \neg\varphi$$

one cannot infer $\mathbb{N} \models \varphi$ without an additional semantic assumption. Conversely, $\mathbb{N} \models \varphi$ need not imply $T \vdash \varphi$ once T is incomplete. Gödel's sentence gives the canonical example under the hypotheses established above:

$$\mathrm{Con}(T) \Longrightarrow \mathbb{N} \models G_T \quad \text{and} \quad T \nvdash G_T.$$

Thus consistency, truth, and provability are formally distinct notions whose relationships require separate theorems. *Point specific to this result.* Consistency says only that no contradiction is derivable in the formal system. Truth refers to the standard structure (or another specified model). Incompleteness exploits precisely the fact that these notions need not coincide for an effectively axiomatized theory. □

Worked example

Worked Example 18.5 (A consistent but unsound extension)**.** Let T be a consistent computably enumerable theory extending Q for which the canonical provability predicate satisfies the hypotheses of Gödel's second incompleteness theorem. Then $T + \neg\mathrm{Con}(T)$ is consistent: if it were inconsistent, the Deduction Theorem would give $T \vdash \mathrm{Con}(T)$, contradicting Gödel II. Since consistency of T means that no standard natural number codes a T-proof of $0 = 1$, the standard model satisfies $\mathrm{Con}(T)$; hence $\neg\mathrm{Con}(T)$ is false in $\mathbb{N}$. Thus the extension is a concrete consistent but arithmetically unsound theory.

A useful warning

Whenever a proof says "therefore the sentence is true," identify which soundness hypothesis licenses the move. Unprovability alone is not truth.

Looking ahead

The next chapter, *Formalized provability*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 19

Formalized provability

Definitions and setup

Formal definitions used in this chapter

Derivability conditions. D1 internalizes actual proofs; D2 internalizes modus ponens; D3 internalizes the fact that a proof code itself has a provable existence assertion. All three refer to the canonical Σ_1 predicate Prov_T.

The Hilbert–Bernays–Löb derivability conditions are internal closure properties of the chosen provability predicate. D1 internalizes actual theoremhood, D2 internalizes modus ponens, and D3 says that provability is itself provably provable. They are proved from primitive-recursive transformations of proof codes. These are intensional properties: an arithmetical predicate with the same standard extension need not satisfy them.

How this chapter fits together

The Hilbert–Bernays–Löb conditions are not abstract assumptions here; they are consequences of explicit primitive-recursive transformations of proof codes. One transformation records a known proof, another combines two proofs by modus ponens, and a third turns a proof into a proof that such a proof exists. These internal closure properties are the engine of Löb's theorem and Gödel II.

Theorem 19.1 (First Hilbert–Bernays–Löb derivability condition)**.** *Let $T \supseteq I\Sigma_1$ be c.e. and let $\mathrm{Prf}_T(p, q)$ be the canonical primitive-recursive proof predicate fixed in Volume II. For every sentence φ,*

$$T \vdash \varphi \quad \Longrightarrow \quad T \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner), \qquad \mathrm{Prov}_T(q) := \exists p\, \mathrm{Prf}_T(p, q).$$

This is derivability condition D1.

How to read the symbols: D1, D2, and D3 are three different programs on proof codes

D1 turns an externally known proof of A into an internal proof that A is provable. D2 represents the primitive-recursive program that concatenates a proof of $A \to B$ with a proof of A and appends modus ponens. D3 represents the program that takes a proof of A and produces a proof of the existential sentence asserting that such a proof exists. Remembering the underlying proof-code programs makes the three formulas much less mysterious.

Proof.

Formal derivation spine

Formal spine: D1 inserts one concrete proof code. If n is the code of an actual T-proof of φ, then

$$T \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner\varphi\urcorner) \quad \Longrightarrow \quad T \vdash \exists p\, \mathrm{Prf}_T(p, \ulcorner\varphi\urcorner).$$

Assume externally that $T \vdash \varphi$. A formal proof is a finite object, so choose its actual natural-number code n. The relation $\mathrm{Prf}_T(p, q)$ was chosen to represent the primitive-recursive proof-checking relation. Consequently the true closed computation "n is a T-proof of φ" is verified inside the arithmetical base:

$$T \vdash \mathrm{Prf}_T(\bar{n}, \ulcorner\varphi\urcorner). \tag{1}$$

Existential introduction applied to (1) gives

$$T \vdash \exists p\, \mathrm{Prf}_T(p, \ulcorner\varphi\urcorner),$$

which is exactly

$$T \vdash \mathrm{Prov}_T(\ulcorner\varphi\urcorner).$$

The implication in D1 is metamathematical: it begins with an externally known derivation and uses its concrete code. D1 does not assert the internal formula $\varphi \to \mathrm{Prov}_T(\ulcorner\varphi\urcorner)$ for arbitrary φ. □

Theorem 19.2 (Second derivability condition)**.** *For all sentences φ, ψ, the canonical provability predicate satisfies*

$$T \vdash \mathrm{Prov}_T(\ulcorner\varphi \to \psi\urcorner) \to \big(\mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \mathrm{Prov}_T(\ulcorner\psi\urcorner)\big).$$

This is derivability condition (D2).

Proof.

Formal derivation spine

Formal spine: D2 represents the modus-ponens program. There is a primitive-recursive function c such that

$$\mathrm{Prf}_T(p, \ulcorner\varphi \to \psi\urcorner) \wedge \mathrm{Prf}_T(q, \ulcorner\varphi\urcorner) \to \mathrm{Prf}_T(c(p, q), \ulcorner\psi\urcorner). \tag{1}$$

Existentially quantify the two input proof codes.

Given coded proofs p of $\varphi \to \psi$ and q of φ, form a new coded proof by concatenating the two proof sequences and appending one modus-ponens line with conclusion ψ. Sequence concatenation, index shifting, and the addition of the last inference line are primitive recursive. Hence the code-transformer $c(p, q)$ is primitive recursive.

The proof checker verifies the transformed sequence uniformly, so the arithmetical base proves (1). Now reason inside T. Suppose

$$\mathrm{Prov}_T(\ulcorner\varphi \to \psi\urcorner) \quad \text{and} \quad \mathrm{Prov}_T(\ulcorner\varphi\urcorner).$$

Unfolding Prov_T, choose proof-code witnesses p, q. Formula (1) gives

$$\mathrm{Prf}_T(c(p, q), \ulcorner\psi\urcorner),$$

and existential introduction yields $\mathrm{Prov}_T(\ulcorner\psi\urcorner)$. Discharging the two assumptions by propositional logic gives

$$T \vdash \mathrm{Prov}_T(\ulcorner\varphi \to \psi\urcorner) \to \big(\mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \mathrm{Prov}_T(\ulcorner\psi\urcorner)\big).$$

This is D2. Its content is precisely the internal representability of one syntactic operation: combining two proofs by modus ponens. □

Theorem 19.3 (Third derivability condition)**.** *For every sentence φ, the canonical provability predicate satisfies*

$$T \vdash \mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \mathrm{Prov}_T(\ulcorner\mathrm{Prov}_T(\ulcorner\varphi\urcorner)\urcorner).$$

This is derivability condition (D3).

Proof.

Formal derivation spine

Formal spine: D3 verifies a proof certificate inside another proof. There is a primitive-recursive function v such that

$$\mathrm{Prf}_T(p, \ulcorner\varphi\urcorner) \to \mathrm{Prf}_T\big(v(p), \ulcorner\mathrm{Prov}_T(\ulcorner\varphi\urcorner)\urcorner\big). \tag{1}$$

Existentially quantify p.

Fix a code p of a T-proof of φ. Because proof checking is primitive recursive, there is an effective finite derivation in the base arithmetic of the closed statement

$$\mathrm{Prf}_T(\bar{p}, \ulcorner\varphi\urcorner).$$

Append existential introduction to obtain

$$\exists r\, \mathrm{Prf}_T(r, \ulcorner\varphi\urcorner),$$

namely $\mathrm{Prov}_T(\ulcorner\varphi\urcorner)$. Coding this uniformly defines a primitive-recursive transformation $v(p)$. Representability of the transformation gives (1) inside T.

Now suppose $\mathrm{Prov}_T(\ulcorner\varphi\urcorner)$, so some p satisfies $\mathrm{Prf}_T(p, \ulcorner\varphi\urcorner)$. Formula (1) supplies a proof code $v(p)$ for the provability sentence. Therefore

$$T \vdash \mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \mathrm{Prov}_T\big(\ulcorner\mathrm{Prov}_T(\ulcorner\varphi\urcorner)\urcorner\big).$$

Unlike D1, D3 is wholly internal: the antecedent itself asserts the existence of a proof code, and the primitive-recursive transformer converts any such witness into a proof of that existential assertion. □

Lemma 19.4 (Formalized fixed-point argument for provability predicates)**.** *Let $T \supseteq I\Sigma_1$ have the canonical provability predicate satisfying D1–D3, and suppose*

$$T \vdash \theta \leftrightarrow \chi(\ulcorner\theta\urcorner).$$

Then the equivalence can be internalized at the provability level:

$$T \vdash \mathrm{Prov}_T(\ulcorner\theta\urcorner) \leftrightarrow \mathrm{Prov}_T(\ulcorner\chi(\ulcorner\theta\urcorner)\urcorner).$$

When χ itself contains Prov_T, D2 and D3 may therefore be applied wholly inside T to the fixed-point equivalence.

Proof.

Formal derivation spine

Formal spine for Formalized fixed-point argument for provability predicates.

$$T \vdash \theta \leftrightarrow \chi(\ulcorner\theta\urcorner). \tag{F1}$$

$$T \vdash \mathrm{Prov}_T(\ulcorner\theta \to \chi(\ulcorner\theta\urcorner)\urcorner)$$

$$T \vdash \mathrm{Prov}_T(\ulcorner\chi(\ulcorner\theta\urcorner) \to \theta\urcorner).$$

Let

$$T \vdash \theta \leftrightarrow \chi(\ulcorner\theta\urcorner). \tag{F1}$$

Split (F1) into the two implications. By D1,

$$T \vdash \mathrm{Prov}_T(\ulcorner\theta \to \chi(\ulcorner\theta\urcorner)\urcorner)$$

and

$$T \vdash \mathrm{Prov}_T(\ulcorner\chi(\ulcorner\theta\urcorner) \to \theta\urcorner).$$

Applying D2 to these theorems gives

$$T \vdash \mathrm{Prov}_T(\ulcorner\theta\urcorner) \to \mathrm{Prov}_T(\ulcorner\chi(\ulcorner\theta\urcorner)\urcorner) \tag{F2}$$

and the reverse implication. Hence

$$T \vdash \mathrm{Prov}_T(\ulcorner\theta\urcorner) \leftrightarrow \mathrm{Prov}_T(\ulcorner\chi(\ulcorner\theta\urcorner)\urcorner).$$

If χ itself contains Prov_T, D3 may now be applied to the provability terms appearing in (F2). This is the precise sense in which the fixed-point calculation has been internalized. *Further explanation.* The formalized fixed-point step must itself be carried out inside the arithmetic theory. One first proves that the syntactic transformation sending a code to the code of its diagonal instance is primitive recursive and hence represented. The theory can then reason about the code transformation and derive the fixed-point biconditional, rather than relying on an external meta-level substitution. This distinction is essential in the proof of Löb's theorem, where provability assertions about the fixed point are manipulated internally. □

Worked example

Worked Example 19.5 (Combining proof codes)**.** Given a code p for a proof of $A \to B$ and a code q for a proof of A, there is a primitive-recursive function $m(p, q)$ coding the concatenated derivation ending with B by modus ponens. Representing m inside arithmetic yields the second derivability condition for Prov_T.

A useful warning

The derivability conditions apply to the chosen intensional proof predicate. An extensionally equivalent formula for theoremhood need not satisfy them.

Looking ahead

The next chapter, *Löb and second incompleteness*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 20

Löb and second incompleteness

Definitions and setup

Formal definitions used in this chapter

Canonical consistency sentence. $\mathrm{Con}(T) := \neg \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner)$.
Löb premise. A sentence φ satisfies the Löb premise when $T \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi$.

Löb's theorem concerns sentences A for which the theory proves $\mathrm{Prov}(\ulcorner A \urcorner) \to A$. The second incompleteness theorem is obtained by applying the same derivability machinery to the canonical consistency sentence $\mathrm{Con}(T) = \neg \mathrm{Prov}_T(\ulcorner \bot \urcorner)$. We fix one canonical proof predicate for the whole argument; changing the coding by an intensionally unnatural definition can change which internal consistency sentence is obtained.

How this chapter fits together

Löb's theorem turns an internal reflection implication $\mathrm{Prov}_T(\ulcorner A \urcorner) \to A$ into a proof of A. Gödel II is the special case obtained by encoding consistency as the absence of a proof of contradiction. The proof is a formalized self-reference calculation, so every use of Prov_T must remain inside the theory until the final external consistency argument.

Theorem 20.1 (Löb's theorem)**.** *Let T be a c.e. extension of $I\Sigma_1$ whose canonical provability predicate satisfies D1–D3. For every sentence φ, if*

$$T \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi,$$

then $T \vdash \varphi$.

Uses: the diagonal lemma and derivability conditions D1–D3.

Commented symbolic trace: Löb's theorem without skipping the bookkeeping

Track three sentences: the target A, the fixed point B, and the provability statements about them. The fixed point is chosen so that $B \leftrightarrow (\mathrm{Prov}(B) \to A)$. D2 moves implications through Prov; D3 supplies $\mathrm{Prov}(B) \to \mathrm{Prov}(\mathrm{Prov}(B))$; the assumed reflection implication $\mathrm{Prov}(A) \to A$ removes the last provability operator. Only after these internal steps do we use D1 to turn the resulting proof of B into $\mathrm{Prov}(B)$. Reading the proof as a pipeline prevents the nested Prov's from becoming visual noise.

Proof.

Formal derivation spine

Formal spine for Löb's theorem.

$$T \vdash B \leftrightarrow (\mathrm{Prov}(\ulcorner B \urcorner) \to A).$$

$$T \vdash B \leftrightarrow (\mathrm{Prov}_T(\ulcorner B \urcorner) \to \varphi). \tag{L1}$$

$$T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner) \to \mathrm{Prov}_T(\ulcorner \mathrm{Prov}_T(\ulcorner B \urcorner) \to \varphi \urcorner). \tag{L2}$$

Proof map. Keep the two levels visible. The diagonal lemma produces B inside arithmetic. The derivability conditions then allow T to reason about proofs of the implications involving B. The final use of the first derivability condition occurs only after $T \vdash B$ has been obtained externally from the preceding derivation.

Let $T \vdash \mathrm{Prov}(\ulcorner A \urcorner) \to A$. By the diagonal lemma choose B with

$$T \vdash B \leftrightarrow (\mathrm{Prov}(\ulcorner B \urcorner) \to A).$$

Using the derivability conditions, formalize the implication $B \to (\mathrm{Prov}(B) \to A)$ to obtain $\mathrm{Prov}(B) \to \mathrm{Prov}(\mathrm{Prov}(B) \to A)$, and hence $\mathrm{Prov}(B) \to (\mathrm{Prov}(\mathrm{Prov}(B)) \to \mathrm{Prov}(A))$. The third derivability condition gives $\mathrm{Prov}(B) \to \mathrm{Prov}(\mathrm{Prov}(B))$, so $T \vdash \mathrm{Prov}(B) \to \mathrm{Prov}(A)$. Combining the assumed $\mathrm{Prov}(A) \to A$ yields $T \vdash \mathrm{Prov}(B) \to A$, hence $T \vdash B$ by the fixed point. The first derivability condition gives $T \vdash \mathrm{Prov}(B)$, and therefore $T \vdash A$.

Detailed formal verification. For completeness, write the Löb calculation as a derivation inside T. Let B satisfy

$$T \vdash B \leftrightarrow (\mathrm{Prov}_T(\ulcorner B \urcorner) \to \varphi). \tag{L1}$$

By D1 applied to the forward implication in (L1), and then D2,

$$T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner) \to \mathrm{Prov}_T(\ulcorner \mathrm{Prov}_T(\ulcorner B \urcorner) \to \varphi \urcorner). \tag{L2}$$

D2 again yields

$$T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner) \to \big(\mathrm{Prov}_T(\ulcorner \mathrm{Prov}_T(\ulcorner B \urcorner) \urcorner) \to \mathrm{Prov}_T(\ulcorner \varphi \urcorner)\big). \tag{L3}$$

D3 supplies the middle premise, hence

$$T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner) \to \mathrm{Prov}_T(\ulcorner \varphi \urcorner). \tag{L4}$$

Using the hypothesis $T \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi$, we obtain

$$T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner) \to \varphi.$$

By (L1), $T \vdash B$; D1 gives $T \vdash \mathrm{Prov}_T(\ulcorner B \urcorner)$, and modus ponens finally gives $T \vdash \varphi$. Every displayed implication is internal to T. □

Remark 20.2 (Source note)**.** This proof follows the standard derivability-condition proof of Löbś theorem; compare the original note [6].

Theorem 20.3 (Gödel's second incompleteness theorem for the flagship theory class)**.** *Let T be a consistent c.e. extension of $I\Sigma_1$ with the canonical provability predicate. Writing*

$$\mathrm{Con}(T) := \neg \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner),$$

one has

$$T \nvdash \mathrm{Con}(T).$$

Uses: the formalized provability calculus, Löb's theorem, and the consistency hypothesis on the chosen theory.

How to read the symbols: Gödel II is about one fixed consistency sentence

Here $\mathrm{Con}(T)$ abbreviates $\neg\mathrm{Prov}_T(\ulcorner\bot\urcorner)$ for the canonical provability predicate already constructed. The theorem is not the vague statement that "a theory cannot know it is consistent." It says that, under the stated effectiveness and strength hypotheses, a consistent T does not prove this particular arithmetical sentence expressing absence of a coded T-proof of contradiction.

Proof.

Formal derivation spine

Formal spine: derivability conditions turn consistency into the Gödel sentence. Let $P := \mathrm{Prov}_T(\ulcorner G_T \urcorner)$ and $C := \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner)$. Then

$$T \vdash P \to \mathrm{Prov}_T(\ulcorner \neg P \urcorner), \qquad T \vdash P \to \mathrm{Prov}_T(\ulcorner P \urcorner),$$

so D2 gives $T \vdash P \to C$. Hence

$$T \vdash \neg C \to \neg P \to G_T.$$

Let $G = G_T$ and abbreviate

$$P := \mathrm{Prov}_T(\ulcorner G \urcorner), \qquad C := \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner).$$

The fixed-point equivalence yields

$$T \vdash G \leftrightarrow \neg P. \tag{1}$$

In particular, $T \vdash G \to \neg P$. By D1, T proves the provability of this implication; D2 then internalizes modus ponens and gives

$$T \vdash P \to \mathrm{Prov}_T(\ulcorner \neg P \urcorner). \tag{2}$$

D3 gives

$$T \vdash P \to \mathrm{Prov}_T(\ulcorner P \urcorner). \tag{3}$$

The propositional tautology $P \to (\neg P \to 0 = 1)$ has a T-proof. Applying D1 and D2 twice shows

$$T \vdash \mathrm{Prov}_T(\ulcorner P \urcorner) \to \big(\mathrm{Prov}_T(\ulcorner \neg P \urcorner) \to C\big). \tag{4}$$

Combining (2), (3), and (4), we obtain

$$T \vdash P \to C. \tag{5}$$

Classical contraposition inside T gives

$$T \vdash \neg C \to \neg P. \tag{6}$$

By the reverse direction of (1), $T \vdash \neg P \to G$, and therefore

$$T \vdash \mathrm{Con}(T) \to G, \qquad \mathrm{Con}(T) := \neg\mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner). \tag{7}$$

If the consistent theory T proved $\mathrm{Con}(T)$, then (7) would give $T \vdash G$, contradicting the already proved theorem that consistency implies $T \nvdash G$. Hence $T \nvdash \mathrm{Con}(T)$. Every internal step is licensed by D1–D3 and propositional logic; no semantic soundness assumption is hidden in the argument. □

Remark 20.4 (Source note)**.** The second incompleteness theorem is tied to the canonical intensional provability predicate developed in this volume; compare [4, 6].

Corollary 20.5 (PA does not prove Con(PA), assuming consistency)**.** *If PA is consistent, then*

$$\mathrm{PA} \nvdash \mathrm{Con}(\mathrm{PA}).$$

Proof.

Formal derivation spine

Formal spine: specialize the preceding theorem to PA.

$$PA \supseteq I\Sigma_1, \qquad PA \text{ is c.e.}, \qquad \mathrm{Prf}_{PA} \text{ satisfies D1–D3.}$$

Peano arithmetic is a computably enumerable extension of $I\Sigma_1$, and the canonical proof predicate fixed in Volume II satisfies the three derivability conditions proved above. Therefore all hypotheses of Gödel's second incompleteness theorem apply with $T = PA$. If PA is consistent, the conclusion is

$$PA \nvdash \mathrm{Con}(PA), \qquad \mathrm{Con}(PA) := \neg \mathrm{Prov}_{PA}(\ulcorner 0 = 1 \urcorner).$$

This corollary does not assert that $\mathrm{Con}(PA)$ is false. Under the external assumption that PA is consistent, $\mathrm{Con}(PA)$ is true in the standard model; the theorem says only that the consistent theory cannot prove its own canonical consistency sentence. □

Corollary 20.6 (ZF/ZFC analogues with the chosen canonical proof predicate)**.** *Let T be ZF or ZFC in a fixed recursive first-order proof calculus. Code finite T-derivations by natural numbers and let Prov_T be the resulting canonical arithmetical provability predicate, interpreted in T through its standard internal natural-number structure ω^T. The primitive-recursive proof transformations establishing D1–D3 are formalizable in the arithmetic proved by T. Therefore, if T is consistent,*

$$T \nvdash \mathrm{Con}(T), \qquad \mathrm{Con}(T) := \neg \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner).$$

Proof.

Formal derivation spine

Formal spine: set theories arithmetize their own finite proofs.

$$\mathrm{Prf}_T(p, q) \text{ is represented in the internal copy of arithmetic,}$$

and D1–D3 give the same derivation $\mathrm{Con}(T) \to G_T$ as above.

Fix a recursive proof calculus for $T \in \{\mathrm{ZF}, \mathrm{ZFC}\}$. Proof checking is primitive recursive, so the code relation $\mathrm{Prf}_T(p, q)$ and the formula

$$\mathrm{Prov}_T(q) := \exists p \, \mathrm{Prf}_T(p, q)$$

are arithmetic formulas. ZF proves that its internal ω, with the usual recursive operations, satisfies enough arithmetic to verify primitive-recursive computations and the proof-code transformations used in D1–D3. Thus, under the standard interpretation of arithmetic in set theory, T proves the three derivability conditions for its own canonical provability predicate.

The diagonal lemma is also formalized in this internal arithmetic. The proof of the flagship second incompleteness theorem therefore applies inside T to the sentence

$$\mathrm{Con}(T) \equiv \neg \mathrm{Prov}_T(\ulcorner 0 = 1 \urcorner).$$

If $T \vdash \mathrm{Con}(T)$, the internal Gödel-II calculation yields $T \vdash 0 = 1$. Hence, externally assuming consistency of T,

$$T \nvdash \mathrm{Con}(T).$$

The assertion is about this fixed intensional proof predicate. It is not a claim about every arithmetical formula having the same standard extension as Prov_T. □

Worked example

Worked Example 20.7 (Why proving consistency would be too much)**.** Suppose a consistent T satisfying the derivability conditions proved $\mathrm{Con}(T)$. The formalized incompleteness argument proves inside T that $\mathrm{Con}(T) \to G_T$; hence T would prove G_T. First incompleteness says a consistent T cannot do so. The contradiction is external, but the implication from consistency to the Gödel sentence is internal.

A useful warning

Gödel II is not the assertion that no one can prove $\mathrm{Con}(T)$. A stronger metatheory may prove it; the theorem says that, under the stated hypotheses, T cannot prove its own canonical consistency statement.

Looking ahead

The next chapter, *Truth, reflection, and iterated consistency*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 21

Truth, reflection, and iterated consistency

Definitions and setup

Formal definitions used in this chapter

Truth predicate. A formula $\mathrm{Tr}(x)$ would define arithmetical truth if it held of exactly the Gödel numbers of true arithmetic sentences.
Local reflection. The schema $\mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$ for a specified class of formulas.

Tarski's theorem concerns a single arithmetical predicate purporting to define truth for all arithmetic sentences. A local reflection principle, by contrast, is a schema of implications $\mathrm{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$ for a restricted family. Iterated consistency theories add particular metamathematical sentences as new axioms. These notions are related by diagonalization but are not interchangeable.

How this chapter fits together

Tarski's theorem blocks a single arithmetic predicate that correctly captures truth for all arithmetic sentences. Local reflection avoids this by treating restricted instances or schemes rather than a global internal truth definition. Adding $\mathrm{Con}(T)$ then illustrates a systematic way of strengthening a theory, while the text carefully stops before pretending that arbitrary transfinite progressions require no additional machinery.

Theorem 21.1 (Tarski undefinability of truth)**.** *There is no arithmetic formula* $\mathrm{Tr}(x)$ *such that for every arithmetic sentence* σ*,*

$$\mathbb{N} \models \mathrm{Tr}(\ulcorner\sigma\urcorner) \leftrightarrow \sigma.$$

Thus arithmetical truth is not arithmetically definable in the standard model.

Proof.

Formal derivation spine

Formal spine: diagonalize against the proposed truth predicate.

$$L \leftrightarrow \neg\,\mathrm{Tr}(\ulcorner L\urcorner), \qquad \mathrm{Tr}(\ulcorner L\urcorner) \leftrightarrow L \quad\Longrightarrow\quad L \leftrightarrow \neg L.$$

Suppose an arithmetic formula $\mathrm{Tr}(x)$ satisfies the full Tarski biconditional for every arithmetic sentence σ:

$$\mathbb{N} \models \mathrm{Tr}(\ulcorner\sigma\urcorner) \leftrightarrow \sigma. \tag{1}$$

Apply the diagonal lemma to the formula $\neg \operatorname{Tr}(x)$. There is a sentence L such that the weak arithmetic used for diagonalization proves

$$L \leftrightarrow \neg \operatorname{Tr}(\ulcorner L \urcorner). \tag{2}$$

Since that weak arithmetic is true in $\mathbb{N}$, (2) holds in $\mathbb{N}$. The instance of (1) for the sentence L gives

$$\mathbb{N} \models \operatorname{Tr}(\ulcorner L \urcorner) \leftrightarrow L. \tag{3}$$

Combining (2) and (3) in the standard two-valued semantics yields

$$\mathbb{N} \models L \leftrightarrow \neg L,$$

which is impossible. Therefore no arithmetic formula defines the set of Gödel codes of all true arithmetic sentences.

The theorem does not prohibit partial truth predicates for bounded formula complexity, nor the satisfaction relation for a fixed set-sized structure developed in Volume V. Those predicates satisfy biconditionals only for a restricted syntactic class; diagonalization produces a sentence outside the simultaneously covered class. □

Remark 21.2 (Source note)**.** The undefinability argument is the diagonal core of Tarskiś theorem on truth for formalized languages; compare [7].

Proposition 21.3 (Local reflection implies the corresponding instances of soundness)**.** *Let S be a metatheory in which Prf_T is represented and which verifies the D1 proof-certification property*

$$T \vdash \psi \quad \Longrightarrow \quad S \vdash \operatorname{Prov}_T(\ulcorner \psi \urcorner).$$

Let Γ be a class of sentences. If for every $\varphi \in \Gamma$,

$$S \vdash \operatorname{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi,$$

then for every $\varphi \in \Gamma$,

$$T \vdash \varphi \quad \Longrightarrow \quad S \vdash \varphi.$$

Proof.

Formal derivation spine

Fix $\varphi \in \Gamma$. The two hypotheses give the implication chain

$$T \vdash \varphi \Longrightarrow S \vdash \operatorname{Prov}_T(\ulcorner \varphi \urcorner) \Longrightarrow S \vdash \varphi.$$

The first arrow is D1 as verified in S; the second is modus ponens with the local reflection instance.

Let $\varphi \in \Gamma$ and assume externally that $T \vdash \varphi$. By the stated proof-certification property for S,

$$S \vdash \operatorname{Prov}_T(\ulcorner \varphi \urcorner).$$

By hypothesis, S also proves the corresponding reflection instance

$$S \vdash \operatorname{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi.$$

One application of modus ponens in S therefore yields

$$S \vdash \varphi.$$

Since φ was arbitrary in Γ, this proves the stated external Γ-soundness conclusion relative to the metatheory S.

The distinction between the three levels is essential. The assumption $T \vdash \varphi$ is made in the surrounding metatheory. The formula $\operatorname{Prov}_T(\ulcorner \varphi \urcorner)$ is an arithmetical sentence inside S. The implication $\operatorname{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi$ is one member of a reflection schema, not a single global truth predicate. Löb's theorem explains why a

consistent theory T satisfying the Hilbert–Bernays–Löb conditions cannot prove all of its own unrestricted reflection instances: if

$$T \vdash \mathrm{Prov}_T(\ulcorner \varphi \urcorner) \to \varphi,$$

then Löb gives $T \vdash \varphi$. Thus local reflection is used only with the explicit metatheoretic hypotheses displayed in the statement above. □

Theorem 21.4 (Strict strengthening by adjoining a consistency statement)**.** *If T is consistent and satisfies the hypotheses of Gödel's second incompleteness theorem, then*

$$T \nvdash \mathrm{Con}(T).$$

Consequently $T + \mathrm{Con}(T)$ is a proper deductive extension of T: it proves $\mathrm{Con}(T)$, which T does not. This properness assertion does not by itself imply that the extension is consistent. If, in addition, the standard model satisfies every axiom of T, then $\mathbb{N} \models T + \mathrm{Con}(T)$, and the extension is consistent.

Proof.

Formal derivation spine

Formal spine: nonderivability already gives a proper extension.

$$T \nvdash \mathrm{Con}(T), \qquad T + \mathrm{Con}(T) \vdash \mathrm{Con}(T),$$

so

$$\mathrm{Thm}(T) \subsetneq \mathrm{Thm}(T + \mathrm{Con}(T)).$$

Gödel's second incompleteness theorem gives

$$T \nvdash \mathrm{Con}(T). \tag{1}$$

Every theorem of T remains a theorem after the new axiom is added, so

$$\mathrm{Thm}(T) \subseteq \mathrm{Thm}(T + \mathrm{Con}(T)).$$

The extension proves $\mathrm{Con}(T)$ by the added axiom, while (1) says that T does not. Hence the inclusion is strict. This conclusion is valid whether or not the extension is consistent; an inconsistent extension is still deductively stronger.

Consistency is a separate question. If $\mathbb{N} \models T$, then no standard number codes a T-proof of contradiction, so

$$\mathbb{N} \models \mathrm{Con}(T).$$

Thus $\mathbb{N} \models T + \mathrm{Con}(T)$, and soundness implies that the extension is consistent. Without such a model or soundness hypothesis, consistency of T alone need not imply consistency of $T + \mathrm{Con}(T)$: a consistent theory can prove the negation of its own canonical consistency sentence. □

Proposition 21.5 (Finite and transfinite-looking iterated consistency schemes: what is proved and what is only previewed)**.** *Define $T_0 = T$ and $T_{n+1} = T_n + \mathrm{Con}(T_n)$. Conditional on consistency of each T_n, the inclusions $T_n \subsetneq T_{n+1}$ are strict. No claim about transfinite progressions is made without an additional ordinal-notation and recursion apparatus.*

Proof.

Formal derivation spine

Formal spine for Finite and transfinite-looking iterated consistency schemes: what is proved and what is only previewed.

$$T_0 := T, \qquad T_{n+1} := T_n + \mathrm{Con}(T_n).$$

$$T_n \nvdash \mathrm{Con}(T_n),$$

$$T_{n+1} \vdash \mathrm{Con}(T_n).$$

Define

$$T_0 := T, \qquad T_{n+1} := T_n + \mathrm{Con}(T_n).$$

Assume inductively that each T_n is consistent and satisfies the Gödel-II hypotheses. Then

$$T_n \nvdash \mathrm{Con}(T_n),$$

while

$$T_{n+1} \vdash \mathrm{Con}(T_n).$$

Hence

$$T_n \subsetneq T_{n+1} \qquad (n < \omega).$$

This proves only the finite progression indexed by ordinary natural numbers. A transfinite progression would require, in addition, a system of ordinal notations O, an effective rule for limit stages, and a definition such as

$$T_\lambda := \bigcup_{\beta <_O \lambda} T_\beta.$$

None of that machinery has been introduced here, so no transfinite theorem is claimed. *Point specific to this result.* The finite progression $T_{n+1} = T_n + \mathrm{Con}(T_n)$ is justified stage by stage. Extending this construction through arbitrary ordinals would require a separate effective ordinal-notation system, so no such transfinite result is being smuggled into the present statement. □

Worked example

Worked Example 21.6 (Truth versus a partial satisfaction predicate)**.** For each fixed formula-complexity class, arithmetic can define a satisfaction predicate for formulas in that class. Tarski's argument rules out one arithmetic predicate satisfying all truth biconditionals at once. The difference explains why bounded truth definitions are legitimate tools while a global internal truth predicate is impossible.

A useful warning

A scheme consisting of many formulas is not automatically equivalent to one formula asserting all of them. This distinction is crucial for reflection and truth.

End-of-volume perspective

The incompleteness results delimit formal proof without undermining ordinary mathematical reasoning: stronger theories can prove consistency statements unavailable to weaker ones, but every consistent computably enumerable theory satisfying the hypotheses of Gödel's second incompleteness theorem fails to prove its own canonical consistency sentence.

Part IV

Volume IV: ZF Set Theory and the Transfinite

Preface

Set theory is developed without silently assuming Choice. Ordinal recursion, rank, Hartogs' theorem, and collapse are the infrastructure for both constructibility and forcing.

Position in the series

Volume IV builds the ZF and transfinite infrastructure shared by constructibility and forcing: rank, ordinals, cardinal machinery without hidden Choice, and Mostowski collapse.

Sources and further reading

The set-theoretic development follows standard ZF practice; for broader context and comparison of conventions see [13, 14].

Notation for this volume

Ordinals are Greek letters; V_α is the cumulative hierarchy; $\aleph_\alpha$ and $\beth_\alpha$ are kept distinct unless a theorem identifies them.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 22

ZF axioms and elementary constructions

Definitions and setup

Formal definitions used in this chapter

ZF. ZF is the first-order theory of $\in$ consisting of Extensionality, Empty Set, Pairing, Union, Power Set, Infinity, the Separation scheme, the Replacement scheme, and Foundation.
Function. A set f is a function iff

$$\forall x\,\forall y\,\forall z\,\big((\langle x,y\rangle \in f \wedge \langle x,z\rangle \in f) \to y = z\big).$$

Its domain is $\mathrm{dom}(f) = \{x : \exists y\,\langle x,y\rangle \in f\}$, and $f(x) = y$ abbreviates $\langle x,y\rangle \in f$.

ZF consists of Extensionality, Empty Set (derivable in common presentations), Pairing, Union, Power Set, Infinity, Separation, Replacement, Foundation, and the usual logical framework. Ordered pairs are Kuratowski pairs. Relations are sets of ordered pairs, and functions are functional relations. Each construction explicitly identifies the axiom that supplies the required set. Choice is not part of ZF and is never imported silently.

How this chapter fits together

The axioms of ZF are now used as construction principles. Ordered pairs, products, relations, and functions must be built as sets; Cantor–Bernstein compares injections without Choice; Cantor's diagonal theorem proves that no power set is exhausted by a list; and transitive closure prepares for Foundation and rank. These are the set-theoretic primitives used throughout the rest of the series.

Theorem 22.1 (Construction of ordered pairs, Cartesian products, relations, and functions in ZF)**.** *ZF proves the existence of the Kuratowski ordered pair $\langle x,y\rangle = \{\{x\},\{x,y\}\}$, Cartesian products $A \times B$, relations $R \subseteq A \times B$, and functions represented as functional relations; each construction is a set whenever its parameters are sets.*

Proof. Pairing gives $\{x\}$ and $\{x,y\}$, hence the Kuratowski pair

$$\langle x,y\rangle := \{\{x\},\{x,y\}\}.$$

Extensionality proves $\langle x, y\rangle = \langle x', y'\rangle$ iff $x = x'$ and $y = y'$. For sets A, B, Replacement applied to each $a \in A$ gives the set $\{\langle a, b\rangle : b \in B\}$; another Replacement followed by Union yields a set containing all such pairs. Separation extracts

$$A \times B = \{z : \exists a \in A\, \exists b \in B\, z = \langle a, b\rangle\}.$$

A relation from A to B is simply a subset of $A \times B$, hence a set by Power Set and Separation. A function is a relation $f \subseteq A \times B$ satisfying

$$\forall a \in A\, \exists! b \in B\, \langle a, b\rangle \in f.$$

Thus functions, domains, ranges, restrictions, inverses of injections, and compositions are all ordinary sets definable from the parameters by the ZF axioms.

□

Theorem 22.2 (Cantor–Bernstein theorem)**.** *If there are injections $f : A \to B$ and $g : B \to A$, then there is a bijection $h : A \to B$.*

Proof. Let $f : A \to B$ and $g : B \to A$ be injections. Define

$$A_0 = A \setminus g[B], \qquad A_{n+1} = g[f[A_n]], \qquad C = \bigcup_{n<\omega} A_n.$$

Set

$$h(a) = \begin{cases} f(a), & a \in C, \\ g^{-1}(a), & a \notin C. \end{cases}$$

The second branch is defined because if $a \notin C$, then in particular $a \notin A_0$, so $a \in g[B]$; injectivity of g gives a unique preimage. The two images are disjoint. Indeed, if $a \in C$ and $a' \notin C$ with $f(a) = g^{-1}(a')$, then $a' = g(f(a)) \in C$, contradiction. Hence h is injective. For surjectivity, take $b \in B$. If $g(b) \notin C$, then $h(g(b)) = b$. If $g(b) \in C$, it cannot lie in A_0; thus $g(b) = g(f(a))$ for some $a \in C$, and injectivity of g gives $b = f(a) = h(a)$. Therefore $h : A \to B$ is a bijection.

□

Theorem 22.3 (Cantor's theorem)**.** *For every set X, no function $f : X \to \mathcal{P}(X)$ is surjective. Hence*

$$|X| < |\mathcal{P}(X)|$$

in the sense that X injects into $\mathcal{P}(X)$ but the two sets are not equipotent.

Proof. The map $x \mapsto \{x\}$ injects X into $\mathcal{P}(X)$. Suppose, toward contradiction, that $f : X \to \mathcal{P}(X)$ is surjective. Separation forms the diagonal subset

$$D = \{x \in X : x \notin f(x)\}.$$

Since $D \in \mathcal{P}(X)$, surjectivity gives $d \in X$ with $f(d) = D$. Then

$$d \in D \iff d \notin f(d) \iff d \notin D,$$

a contradiction. Hence no map from X onto $\mathcal{P}(X)$ exists. In particular there is no bijection between X and its power set. No form of Choice has been used: the argument requires only Separation and elementary logic once $\mathcal{P}(X)$ exists.

□

Proposition 22.4 (Transitive closure exists)**.** *For every set x there exists a least transitive set* $\mathrm{tc}(x)$ *containing x, namely the union of the finite iterates of the union operation starting from x.*

Proof. Define iterated unions by recursion on ω:

$$T_0 = x, \qquad T_{n+1} = \bigcup T_n.$$

Replacement collects $\{T_n : n < \omega\}$, and Union forms

$$\mathrm{tc}(x) = x \cup \bigcup_{n<\omega} T_n.$$

If $y \in z \in \mathrm{tc}(x)$, then either $z \in x$, in which case $y \in T_1$, or $z \in T_n$ for some n, in which case $y \in T_{n+1}$. Thus $\mathrm{tc}(x)$ is transitive. If M is any transitive set containing x, induction on n gives $T_n \subseteq M$; therefore $\mathrm{tc}(x) \subseteq M$. Hence this is the least transitive set containing x.

□

Worked example

Worked Example 22.5 (Cantor diagonalization)**.** Given any proposed map $f : X \to \mathcal{P}(X)$, form $D = \{x \in X : x \notin f(x)\}$. If $D = f(d)$ for some d, then $d \in D$ iff $d \notin D$. Thus D is a concrete subset omitted by the range of f. Notice that no appeal to Choice occurs.

A useful warning

When working in ZF, never replace "there is an injection" by a cardinal inequality that presupposes every set is well-orderable.

Looking ahead

The next chapter, *Natural numbers and recursion*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 23

Natural numbers and recursion

Definitions and setup

Formal definitions used in this chapter

Inductive set. A set containing $0 = \varnothing$ and closed under $n \mapsto n \cup \{n\}$.
ω**.** The least inductive set; its elements are the von Neumann natural numbers.

A set is inductive when it contains $0 = \varnothing$ and is closed under successor $x \mapsto x \cup \{x\}$. ω is the least inductive set. The von Neumann natural n is the set of all smaller naturals. Recursion on ω defines arithmetic operations and finite sequences. Later coding arguments may identify these set-theoretic naturals with the domain intended by arithmetic, but the constructions are logically distinct.

How this chapter fits together

Infinity gives an inductive set, but ω is obtained by intersecting all inductive subsets and proving minimality. That minimality supplies induction; finite approximation supplies recursion. Arithmetic operations are then defined internally on von Neumann naturals, making later references to arithmetic inside set theory precise.

Theorem 23.1 (Existence and minimality of omega)**.** *ZF proves that there is a unique least inductive set ω; equivalently, ω is the intersection of all inductive subsets of any set supplied by Infinity.*

Proof. Infinity supplies an inductive set I, meaning $0 = \varnothing \in I$ and $n \in I \Rightarrow n \cup \{n\} \in I$. By Separation form

$$\omega = \{x \in I : \forall J \subseteq I\,[J \text{ inductive} \Rightarrow x \in J]\}.$$

Equivalently, ω is the intersection of all inductive subsets of I. It is itself inductive: every inductive $J \subseteq I$ contains 0, and if all such J contain n, then they contain $n \cup \{n\}$. By definition $\omega \subseteq J$ for every inductive J. Thus ω is the least inductive set. Its elements are the von Neumann finite ordinals, since the successor operation is $S(n) = n \cup \{n\}$. This minimality immediately yields induction on ω: any subset of ω containing 0 and closed under successor must equal ω.

Point specific to this result. Minimality is obtained by intersecting all inductive subsets of one set supplied by Infinity. This both defines ω as a set and makes induction a consequence of its least-inductive-set property. □

Theorem 23.2 (Recursion theorem on omega)**.** *Let A be a set, $a \in A$, and $F : A \to A$. There is a unique function $f : \omega \to A$ satisfying*

$$f(0) = a, \qquad f(n+1) = F(f(n)).$$

The theorem extends uniformly to recursions whose next value depends on n and the preceding value.

Proof. Let A be a set, $a \in A$, and $F : A \to A$. A finite approximation of length $n+1$ is a function $s : n+1 \to A$ satisfying

$$s(0) = a, \qquad s(k+1) = F(s(k)) \quad (k < n).$$

Induction on n proves existence and uniqueness of such an approximation: extend the unique approximation of length $n+1$ by assigning $F(s(n))$ at $n+1$. Replacement collects the approximations, and their union is a function $f : \omega \to A$ because uniqueness makes them compatible. It satisfies

$$f(0) = a, \qquad f(n+1) = F(f(n)).$$

If g satisfies the same equations, induction on n gives $f(n) = g(n)$ for every n, so $f = g$. The same argument handles recursions whose successor value depends on both n and the previous value.

Point specific to this result. Finite partial recursion graphs are unique on overlapping domains; their compatible union is therefore a function on all of ω. Replacement is used only to collect the finite approximations. □

Proposition 23.3 (Peano arithmetic operations inside set theory)**.** *There are unique operations $+, \cdot, {}^{(-)} : \omega \times \omega \to \omega$ satisfying*

$$m + 0 = m, \qquad m + (n+1) = S(m+n),$$
$$m \cdot 0 = 0, \qquad m \cdot (n+1) = m \cdot n + m,$$
$$m^0 = 1, \qquad m^{n+1} = m^n \cdot m.$$

ZF proves, by induction on ω, associativity and commutativity of $+$ and $\cdot$, distributivity of multiplication over addition, and the laws $m^0 = 1$ and $m^{n+k} = m^n m^k$.

Proof. Using recursion on ω, define

$$m + 0 = m, \qquad m + (n+1) = S(m+n),$$

and then

$$m \cdot 0 = 0, \qquad m \cdot (n+1) = m \cdot n + m.$$

Exponentiation is defined similarly by $m^0 = 1$ and $m^{n+1} = m^n \cdot m$. The recursion theorem guarantees that these are functions $\omega^2 \to \omega$. The familiar arithmetic laws follow by induction. For example, associativity of addition is proved by induction on k:

$$(m+n) + 0 = m + n = m + (n+0),$$

and if $(m+n) + k = m + (n+k)$, then

$$(m+n) + (k+1) = S((m+n)+k) = S(m+(n+k)) = m + (n + (k+1)).$$

Thus the set-theoretic naturals carry the standard Peano arithmetic operations constructed entirely inside ZF.

Point specific to this result. Addition and multiplication are instances of recursion on ω: $m + 0 = m$, $m + S(n) = S(m+n)$, and similarly for multiplication. Their familiar algebraic laws are then proved by induction rather than assumed from external arithmetic. □

Proposition 23.4 (Finite sequences and elementary number systems in ZF)**.** *ZF constructs the set $A^{<\omega} = \bigcup_{n<\omega} A^n$ of finite sequences over any set A, and the standard constructions of $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ may be carried out as sets from ω using quotients and power sets.*

Proof. A finite sequence from a set A is a function $s : n \to A$ for some $n \in \omega$. Since functions are sets and $n \times A$ is a set, all such sequences of fixed length form a subset of $\mathcal{P}(n \times A)$; Replacement over $n \in \omega$ and Union collect $A^{<\omega}$. Integers may be constructed as equivalence classes of pairs $(m, n) \in \omega^2$ under

$$(m, n) \sim (m', n') \iff m + n' = m' + n,$$

and rationals as equivalence classes of integer pairs with nonzero denominator. Reals can then be coded by Dedekind cuts or Cauchy sequences of rationals. The quotient sets exist by Replacement/Separation from the relevant power sets. The details matter chiefly to establish that the standard number systems are definable sets in ZF and that finite syntactic sequences used later are ordinary set-theoretic functions.

□

Worked example

Worked Example 23.5 (The first von Neumann naturals)**.** Starting with $0 = \varnothing$ and successor $S(n) = n \cup \{n\}$, we obtain $1 = \{0\}$, $2 = \{0, 1\}$, and $3 = \{0, 1, 2\}$. Hence $m < n$ is literally $m \in n$. Recursive addition recovers the familiar finite calculations while remaining a set-theoretic function.

A useful warning

The existence of ω uses Infinity; the recursion theorem then uses Replacement-style collection of finite approximations. These are distinct logical steps.

Looking ahead

The next chapter, *Foundation and rank*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 24

Foundation and rank

Definitions and setup

Formal definitions used in this chapter

Transitive set. M is transitive if $x \in y \in M \Rightarrow x \in M$.
Rank. $\operatorname{rank}(x) = \sup\{\operatorname{rank}(y) + 1 : y \in x\}$.

Foundation says every nonempty set has an $\in$-minimal member. It yields $\in$-induction and supports well-founded recursion. The rank of x is the least ordinal above the ranks of all members of x. The cumulative hierarchy is defined by $V_0 = \varnothing$, $V_{\alpha+1} = \mathcal{P}(V_\alpha)$, and unions at limits. Rank locates every set in this hierarchy.

How this chapter fits together

Foundation turns membership into a well-founded relation and thereby licenses induction and recursion on sets. Rank assigns an ordinal measuring how far a set lies above the empty set; the cumulative hierarchy V_α then organizes the universe by those ranks. Later inner-model and forcing arguments repeatedly use these rank bounds to keep constructions set-sized.

Theorem 24.1 (Membership induction from Foundation)**.** *If a formula $\varphi(x)$ satisfies*

$$\forall x \big[(\forall y \in x\, \varphi(y)) \to \varphi(x)\big],$$

then Foundation proves $\forall x\, \varphi(x)$.

Proof. Assume a formula $P(x)$ satisfies

$$\forall x\, [\, (\forall y \in x\, P(y)) \Rightarrow P(x)\,].$$

Suppose P fails somewhere and form the nonempty set of counterexamples inside the transitive closure of one counterexample. By Foundation, it has an $\in$-minimal member a. Every $y \in a$ is then not a counterexample, so $P(y)$ holds for all $y \in a$. The displayed hypothesis yields $P(a)$, contradiction. Therefore $P(x)$ holds for every set x. This is membership induction. The role of Foundation is exactly the least-counterexample step: it prevents an infinite descending membership chain from avoiding a minimal failure.

Point specific to this result. If a counterexample existed, Foundation would select an $\in$-minimal counterexample. All of its members would satisfy the property, so the induction hypothesis would force the counterexample itself to satisfy it, contradiction. □

Theorem 24.2 (Well-founded recursion)**.** *Let R be a well-founded relation on a set A, let $\bar{p}$ be parameters, and let $\Phi(a, h, y, \bar{p})$ be a formula such that for every $a \in A$ and every function h with domain $\{b \in A : bRa\}$, there is a unique set y satisfying $\Phi(a, h, y, \bar{p})$. Then there is a unique function G with domain A such that*

$$\Phi\big(a, G \restriction \{b \in A : bRa\}, G(a), \bar{p}\big)$$

holds for every $a \in A$.

Proof.

Formal derivation spine

Formal spine: local solutions agree on overlaps. For $a \in A$, put $P_a := \{b \in A : bRa\}$. A solution satisfies

$$\Phi(a, G \restriction P_a, G(a), \bar{p}).$$

If G, H are solutions, well-founded induction gives

$$G \restriction P_a = H \restriction P_a \Longrightarrow G(a) = H(a).$$

For each $a \in A$, let C_a be the R-predecessor closure of a. Using Foundation on the set of finite partial functions on C_a, construct a local function G_a satisfying the recursion equation at every point of C_a. The existence hypothesis on Φ supplies the next value once all strict predecessor values have been assigned.

We first prove compatibility. If G_a and G_c are local solutions and $x \in C_a \cap C_c$, assume inductively that they agree on all R-predecessors of x. Then

$$G_a \restriction P_x = G_c \restriction P_x.$$

Both values $G_a(x)$ and $G_c(x)$ satisfy the same formula

$$\Phi(x, G_a \restriction P_x, y, \bar{p}),$$

so uniqueness of y gives $G_a(x) = G_c(x)$. Well-founded induction therefore proves agreement on every overlap.

The union

$$G := \bigcup_{a \in A} G_a$$

is consequently a function. Every $a \in A$ belongs to its own predecessor closure, so $\mathrm{dom}(G) = A$, and the local recursion equation gives the required formula at each a. If H is any other global solution, the same well-founded induction proves $H(a) = G(a)$ for every a. Thus the recursive function exists and is unique. □

Proposition 24.3 (Every set has a rank)**.** *There is a unique rank function satisfying*

$$\mathrm{rank}(x) = \sup\{\mathrm{rank}(y) + 1 : y \in x\}.$$

For every set x, $x \subseteq V_{\mathrm{rank}(x)}$ and $x \in V_{\mathrm{rank}(x)+1}$.

Proof. Define rank by membership recursion:

$$\mathrm{rank}(x) = \sup\{\mathrm{rank}(y) + 1 : y \in x\}.$$

The predecessor set x is a set, so Replacement collects the ordinals $\mathrm{rank}(y) + 1$; their union is an ordinal and hence the displayed supremum exists. Membership induction proves that every $y \in x$ has smaller rank. If V_α is the cumulative hierarchy, induction on the defining equation gives

$$y \in x \Rightarrow y \in V_{\mathrm{rank}(x)},$$

so

$$x \subseteq V_{\mathrm{rank}(x)} \quad \text{and hence} \quad x \in V_{\mathrm{rank}(x)+1}.$$

Thus every set has a well-defined ordinal rank measuring the first stage of the cumulative hierarchy at which it appears.

Point specific to this result. Rank is the recursion $\operatorname{rank}(x) = \sup\{\operatorname{rank}(y) + 1 : y \in x\}$. The supremum is an ordinal, and induction immediately gives $y \in x \Rightarrow \operatorname{rank}(y) < \operatorname{rank}(x)$.

Further explanation. The rank formula is not merely a definition by a proper-class supremum. For a fixed set x, Replacement first collects the set of ordinals $\{\operatorname{rank}(y) + 1 : y \in x\}$; its supremum is therefore an ordinal and a set. The induction also proves the useful equivalence $y \in x \Rightarrow \operatorname{rank}(y) < \operatorname{rank}(x)$, which is why rank can be used as a well-founded measure in later recursive constructions and forcing-name arguments. □

Theorem 24.4 (Cumulative hierarchy characterization V = union_alpha V_alpha)**.** *Define $V_0 = \varnothing$, $V_{\alpha+1} = \mathcal{P}(V_\alpha)$, and $V_\lambda = \bigcup_{\beta<\lambda} V_\beta$. Then every set belongs to some V_α, so*

$$V = \bigcup_{\alpha \in \mathrm{Ord}} V_\alpha.$$

Proof. Define the cumulative hierarchy by transfinite recursion:

$$V_0 = \varnothing, \qquad V_{\alpha+1} = \mathcal{P}(V_\alpha), \qquad V_\lambda = \bigcup_{\beta<\lambda} V_\beta \quad (\lambda \text{ limit}).$$

Induction on α gives monotonicity $V_\alpha \subseteq V_\beta$ for $\alpha < \beta$ and transitivity of every V_α. By the rank theorem, each set x belongs to $V_{\operatorname{rank}(x)+1}$. Hence

$$V = \bigcup_{\alpha \in \mathrm{Ord}} V_\alpha$$

in the class sense: every set lies in some rank. Conversely every element of each V_α is a set by the recursive construction, so the union contains nothing outside the universe of sets. The equation is therefore an exhaustive stratification of the universe by rank.

Point specific to this result. Once rank exists, $x \subseteq V_{\operatorname{rank}(x)}$, hence $x \in V_{\operatorname{rank}(x)+1}$. Thus every set appears at a definite stage of the cumulative hierarchy, proving the union characterization. □

Worked example

Worked Example 24.5 (Computing small ranks)**.** $\operatorname{rank}(\varnothing) = 0$, $\operatorname{rank}(\{\varnothing\}) = 1$, and $\operatorname{rank}(\{\varnothing, \{\varnothing\}\}) = 2$. More generally $x \in V_{\alpha+1}$ exactly when every member of x lies in V_α. The recursion formula $\operatorname{rank}(x) = \sup_{y \in x}(\operatorname{rank}(y) + 1)$ captures this pattern.

A useful warning

A rank bound is not a cardinality bound. A low-rank set can be very large, and later arguments must not interchange these two notions.

Looking ahead

The next chapter, *Ordinals*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 25

Ordinals

Definitions and setup

Formal definitions used in this chapter

Ordinal. A transitive set well ordered by $\in$.
Order type. The unique ordinal isomorphic to a given well-order.

An ordinal is a transitive set well ordered by membership. Every element of an ordinal is an ordinal, and two ordinals are comparable by membership. A well-order has a unique ordinal order type. Transfinite induction and recursion are the ordinal analogues of ordinary induction/recursion, with an explicit limit stage. We use von Neumann ordinals, so $\alpha < \beta$ is literally $\alpha \in \beta$.

How this chapter fits together

Ordinals are transitive sets well-ordered by membership. Their trichotomy gives a canonical comparison; every well-order has a unique ordinal order type; and transfinite induction/recursion extend ordinary induction beyond ω. The proofs are foundational for L, whose stages are indexed by all ordinals.

Theorem 25.1 (Trichotomy of ordinals)**.** *For any ordinals α, β, exactly one of*

$$\alpha \in \beta, \qquad \alpha = \beta, \qquad \beta \in \alpha$$

holds.

Proof. An ordinal is a transitive set well ordered by $\in$. Let α, β be ordinals. Their intersection $\gamma = \alpha \cap \beta$ is an initial segment of each. If $\gamma = \alpha$, then $\alpha \subseteq \beta$, whence either $\alpha = \beta$ or $\alpha \in \beta$ because initial segments of an ordinal are precisely its elements. Symmetrically, if $\gamma = \beta$, then $\beta = \alpha$ or $\beta \in \alpha$. It is impossible for γ to be a proper initial segment of both: then $\gamma \in \alpha \cap \beta = \gamma$, contradicting Foundation/irreflexivity of ordinal membership. Therefore exactly one of

$$\alpha \in \beta, \qquad \alpha = \beta, \qquad \beta \in \alpha$$

holds. This is trichotomy for ordinals.

Point specific to this result. Because ordinals are transitive sets well ordered by membership, their intersection is a common initial segment. The first point of disagreement, if any, places one ordinal as an element of the other, giving trichotomy. □

Theorem 25.2 (Every well-order is uniquely isomorphic to an ordinal)**.** *Every well-ordered set $(X, <)$ is isomorphic to a unique ordinal α.*

Proof. Let $(X, <)$ be a well-ordered set. For each $x \in X$, define recursively

$$\pi(x) = \{\pi(y) : y < x\}.$$

Well-founded recursion on $<$ gives π. Membership induction on the initial segments shows every $\pi(x)$ is an ordinal, and if $y < x$ then $\pi(y) \in \pi(x)$. Conversely, every member of $\pi(x)$ has the form $\pi(y)$ for some $y < x$, so π is an order isomorphism of X onto the transitive set

$$\alpha = \{\pi(x) : x \in X\}.$$

That transitive set is an ordinal. If $\rho : X \to \beta$ is another isomorphism onto an ordinal, induction on x gives

$$\rho(x) = \{\rho(y) : y < x\} = \{\pi(y) : y < x\} = \pi(x).$$

Hence $\rho = \pi$ and $\beta = \alpha$. The order type is therefore unique.

Point specific to this result. Map each element of a well-order to the order type already assigned to its strict initial segment. Well-founded recursion gives the map, and the image is a transitive set of transitive sets, hence an ordinal; uniqueness follows from preservation of initial segments. □

Theorem 25.3 (Transfinite induction)**.** *If a class property P satisfies*

$$\forall \alpha \, [\, (\forall \beta < \alpha \, P(\beta)) \to P(\alpha) \,],$$

then $P(\alpha)$ holds for every ordinal α.

Proof. Let $P(\alpha)$ be a property of ordinals and assume

$$\forall \alpha \, [\, (\forall \beta < \alpha \, P(\beta)) \Rightarrow P(\alpha) \,].$$

If some ordinal failed P, the class of counterexample ordinals would have a least member α, because any nonempty set of candidate counterexamples has a least ordinal and one may start from a chosen counterexample and separate its predecessors. By minimality, $P(\beta)$ holds for every $\beta < \alpha$. The hypothesis then gives $P(\alpha)$, contradiction. Thus every ordinal satisfies P. This is transfinite induction. It is membership induction specialized to ordinals, where $\beta < \alpha$ means $\beta \in \alpha$.

Point specific to this result. The proof is the least-counterexample argument made possible by the well-ordering of every ordinal. If a counterexample existed, its least member would have all predecessors satisfying the property and therefore satisfy it as well. □

Theorem 25.4 (Transfinite recursion)**.** *Let $\Phi(s, y, \bar{p})$ be a formula such that ZF proves: whenever s is a function whose domain is an ordinal, there is a unique set y with $\Phi(s, y, \bar{p})$. Then there is a unique class function G with domain* Ord *such that, for every ordinal α,*

$$\Phi(G \restriction \alpha, G(\alpha), \bar{p}).$$

Equivalently, for a definable functional F, $G(\alpha) = F(G \restriction \alpha)$.

Proof.

Formal derivation spine

Formal spine: compatible set-sized initial solutions. For every ordinal θ, there is a unique function G_θ with domain θ such that

$$\Phi(G_\theta \restriction \alpha, G_\theta(\alpha), \bar{p}) \quad (\alpha < \theta).$$

Uniqueness gives

$$\theta < \eta \Longrightarrow G_\eta \restriction \theta = G_\theta.$$

Apply well-founded recursion to the membership relation on each ordinal θ. The hypothesis on Φ produces a unique set at every stage, so there is a unique set function G_θ on θ satisfying the displayed recursion.

If $\theta < \eta$, both G_θ and $G_\eta \restriction \theta$ solve the same recursion on θ. The uniqueness part of well-founded recursion gives

$$G_\eta \restriction \theta = G_\theta. \tag{1}$$

Define the class relation G by

$$G(\alpha) := G_{\alpha+1}(\alpha). \tag{2}$$

Compatibility (1) makes (2) independent of the chosen initial segment larger than α. Moreover,

$$G \restriction \alpha = G_\alpha,$$

so the defining property of $G_{\alpha+1}$ gives

$$\Phi(G \restriction \alpha, G(\alpha), \bar{p}).$$

Finally, if H is another class solution, transfinite induction on α shows $H \restriction \alpha = G \restriction \alpha$; uniqueness in the hypothesis on Φ then gives $H(\alpha) = G(\alpha)$. This proves uniqueness as a class function. The theorem is a schema indexed by the defining formula Φ, not an appeal to an undefined operation on proper classes. □

Worked example

Worked Example 25.5 (Order type of a finite well-order)**.** For a three-element chain $a < b < c$, map $a \mapsto 0$, $b \mapsto 1$, and $c \mapsto 2$. The image is the ordinal $3 = \{0, 1, 2\}$. The general proof assigns to each element the order type of its initial segment and uses well-founded recursion.

A useful warning

Ordinal comparison is about order type, not cardinality. Distinct ordinals such as ω and $\omega + 1$ can have the same cardinality.

Looking ahead

The next chapter, *Ordinal arithmetic*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 26

Ordinal arithmetic

Definitions and setup

Formal definitions used in this chapter

Ordinal addition. For ordinals α, β, define by recursion on β

$$\alpha + 0 = \alpha, \qquad \alpha + (\beta + 1) = (\alpha + \beta) \cup \{\alpha + \beta\},$$

$$\alpha + \lambda = \sup_{\beta<\lambda}(\alpha + \beta) \quad (\lambda \text{ limit}).$$

Ordinal multiplication.

$$\alpha \cdot 0 = 0, \qquad \alpha \cdot (\beta + 1) = \alpha \cdot \beta + \alpha, \qquad \alpha \cdot \lambda = \sup_{\beta<\lambda} \alpha \cdot \beta.$$

Ordinal exponentiation.

$$\alpha^0 = 1, \qquad \alpha^{\beta+1} = \alpha^\beta \cdot \alpha, \qquad \alpha^\lambda = \sup_{\beta<\lambda} \alpha^\beta$$

for nonzero limit λ.

Ordinal addition, multiplication, and exponentiation are defined by transfinite recursion on the right argument. They encode concatenation and lexicographic constructions of well-orders, not cardinal size. Consequently they need not commute. Suprema at limit stages are part of the definitions, so every algebraic law must be proved by transfinite induction rather than inferred from integer arithmetic.

How this chapter fits together

Ordinal operations are defined by transfinite recursion on the right argument. They record concatenation and lexicographic order types, so they are generally not commutative. Working concrete examples before proving algebraic identities helps prevent cardinal-arithmetic intuitions from being imported incorrectly.

Theorem 26.1 (Recursive construction and laws of ordinal addition)**.** *Ordinal addition is the unique operation satisfying*

$$\alpha + 0 = \alpha, \quad \alpha + (\beta + 1) = (\alpha + \beta) + 1, \quad \alpha + \lambda = \sup_{\beta<\lambda}(\alpha + \beta)$$

for limit λ; it is associative and strictly increasing in the right argument, but need not be commutative.

Commented symbolic trace: ordinal recursion at successor and limit stages

The three clauses play different roles. $\alpha + 0 = \alpha$ initializes the recursion. $\alpha + (\beta + 1) = (\alpha + \beta) + 1$ appends one new point. At a limit λ, there is no immediate predecessor, so $\alpha + \lambda$ is defined as the supremum of all earlier values $\alpha + \beta$ for $\beta < \lambda$. Whenever a later proof uses transfinite induction on the right argument, check these three cases separately rather than treating "limit" as a larger successor case.

Proof. For each fixed ordinal α, transfinite recursion on β defines ordinal addition by

$$\alpha + 0 = \alpha, \qquad \alpha + (\beta + 1) = S(\alpha + \beta), \qquad \alpha + \lambda = \sup_{\beta<\lambda}(\alpha + \beta)$$

for limit λ. The supremum is the union of a set of ordinals and therefore an ordinal. Uniqueness follows from transfinite recursion. Associativity is proved by induction on the third argument γ. At successors,

$$(\alpha + \beta) + (\gamma + 1) = S((\alpha + \beta) + \gamma) = S(\alpha + (\beta + \gamma)) = \alpha + (\beta + (\gamma + 1)),$$

and at limits both sides are the supremum of the corresponding earlier values. The definition is order-sensitive; in general $\alpha + \beta \neq \beta + \alpha$.

Point specific to this result. Addition is recursive in the right argument and models concatenation of well-orders. The limit clause takes the supremum of earlier sums, which explains both associativity and the failure of commutativity.

Further explanation. Associativity is proved by transfinite induction on the final argument γ. The successor case uses the recursive clause twice, while the limit case uses continuity: $\alpha + (\beta + \lambda) = \sup_{\gamma<\lambda} \alpha + (\beta + \gamma)$. The same recursion shows monotonicity in the right argument. What fails is commutativity, as the worked examples below make explicit; the theorem therefore records the algebraic laws that genuinely follow from order-type concatenation rather than importing laws from integer arithmetic. □

Theorem 26.2 (Recursive construction and laws of ordinal multiplication)**.** *Ordinal multiplication is the unique operation satisfying*

$$\alpha \cdot 0 = 0, \quad \alpha(\beta + 1) = \alpha\beta + \alpha, \quad \alpha\lambda = \sup_{\beta<\lambda} \alpha\beta$$

for limit λ. It is associative and left-distributive over ordinal addition.

Proof. Fix α. Define multiplication by transfinite recursion on the right factor:

$$\alpha \cdot 0 = 0, \qquad \alpha \cdot (\beta + 1) = \alpha \cdot \beta + \alpha, \qquad \alpha \cdot \lambda = \sup_{\beta<\lambda} \alpha \cdot \beta.$$

Every stage is an ordinal because addition and suprema of sets of ordinals are already available. Transfinite induction proves associativity and left distributivity over ordinal addition. For example, at a successor stage,

$$\alpha(\beta + (\gamma + 1)) = \alpha((\beta + \gamma) + 1) = \alpha(\beta + \gamma) + \alpha = (\alpha\beta + \alpha\gamma) + \alpha = \alpha\beta + \alpha(\gamma + 1).$$

Limit stages follow by taking suprema. Right distributivity and commutativity generally fail, reflecting the concatenation interpretation of ordinal multiplication.

Point specific to this result. Multiplication iterates ordinal addition on the right. Its limit clause is a supremum of earlier products, so left distributivity follows by induction, while reversing the order can change the resulting order type.

Further explanation. The distributive law valid for ordinal multiplication is $\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$, proved by transfinite induction on γ. For associativity, transfinite induction on the rightmost factor gives the successor identity by the recursive multiplication clause and the limit identity by continuity of both sides under suprema. At a limit λ, the definition $\alpha\lambda = \sup_{\beta<\lambda} \alpha\beta$ is the crucial continuity clause. Left distributivity generally fails, so the proof deliberately keeps track of which argument is being recursively varied. □

Theorem 26.3 (Recursive construction and laws of ordinal exponentiation)**.** *Ordinal exponentiation is the unique operation satisfying*

$$\alpha^0 = 1, \quad \alpha^{\beta+1} = \alpha^\beta \alpha, \quad \alpha^\lambda = \sup_{\beta<\lambda} \alpha^\beta$$

for nonzero limit λ, with the standard conventions for bases $0, 1$.

Proof. For ordinals α, β, define exponentiation by recursion on β:

$$\alpha^0 = 1, \qquad \alpha^{\beta+1} = \alpha^\beta \cdot \alpha, \qquad \alpha^\lambda = \sup_{\beta<\lambda} \alpha^\beta$$

for nonzero limit λ (with the usual convention for 0^0). Apply the transfinite-recursion theorem to the displayed successor and limit clauses; it yields a unique function of the exponent, because multiplication and suprema of sets of ordinals are already defined. Induction proves the exponent law

$$\alpha^{\beta+\gamma} = \alpha^\beta \cdot \alpha^\gamma$$

with the convention that the recursion is on the exponent and multiplication is ordinal multiplication. At a successor, use the recursive equation and associativity of multiplication; at a limit, both sides are obtained as the supremum of the corresponding earlier powers. Again these are ordinal, not cardinal, powers and encode order type.

Point specific to this result. Exponentiation iterates multiplication with $\alpha^0 = 1$ and takes suprema at limits. Transfinite induction establishes its recursion laws and monotonicity in the exponent under the stated hypotheses.

Further explanation. The recursion on the exponent gives $\alpha^0 = 1$, $\alpha^{\beta+1} = \alpha^\beta \alpha$, and $\alpha^\lambda = \sup_{\beta<\lambda} \alpha^\beta$ at limits. Transfinite induction then proves the exponent laws in their correct ordinal form, for example $\alpha^{\beta+\gamma} = \alpha^\beta \alpha^\gamma$. Limit continuity is again essential, and it is the mechanism behind fixed points such as epsilon numbers. □

Worked Theorem 26.4 (Canonical calculations with omega, omega+1, 1+omega, omega^2, and epsilon-style fixed-point preview)**.** *The recursive definitions give*

$$1 + \omega = \omega < \omega + 1, \qquad 2 \cdot \omega = \omega < \omega \cdot 2,$$

and $\omega^2 = \sup_{n<\omega} \omega \cdot n$. The normal function $\alpha \mapsto \omega^\alpha$ has arbitrarily large fixed points, the least being ε_0.

Proof. The recursive definitions give concrete calculations. First,

$$1 + \omega = \sup_{n<\omega} (1 + n) = \omega,$$

whereas $\omega + 1 = S(\omega) > \omega$. Thus addition is not commutative. Using the multiplication limit clause,

$$2 \cdot \omega = \sup_{n<\omega} 2n = \omega, \qquad \omega \cdot 2 = \omega + \omega > \omega.$$

For exponentiation,

$$\omega^2 = \sup_{n<\omega} \omega \cdot n,$$

which is the order type of ω many consecutive blocks of type ω. More generally $\omega^\omega = \sup_{n<\omega} \omega^n$. These computations illustrate the governing principle: ordinal operations remember the order in which blocks are concatenated, so cardinal equalities such as $|1 + \omega| = |\omega + 1|$ do not imply equality of ordinals.

□

Worked example

Worked Example 26.5 (Noncommutativity in one line)**.** $1 + \omega = \sup_{n<\omega}(1+n) = \omega$, while $\omega + 1$ has a greatest element and is strictly larger than ω. Likewise $2 \cdot \omega = \omega$, whereas $\omega \cdot 2$ consists of two consecutive copies of ω.

A useful warning

Only the stated distributive and associative laws hold. In particular, left and right distributivity are not interchangeable for ordinal multiplication.

Looking ahead

The next chapter, *Cardinals without assuming choice*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 27

Cardinals without assuming choice

Definitions and setup

Formal definitions used in this chapter

Cardinal. An initial ordinal representing the size of a well-orderable set.
Hartogs number. $h(X)$ is the least ordinal that does not inject into X.
Beth numbers. $\beth_0 = \aleph_0$, $\beth_{\alpha+1} = 2^{\beth_\alpha}$, with suprema at limits.

Without Choice, not every set is known to be well orderable. A well-orderable cardinal is represented by its least equinumerous ordinal, an initial ordinal. Hartogs' theorem assigns to every set an ordinal too large to inject into it, without using Choice. The aleph hierarchy enumerates infinite initial ordinals; the beth hierarchy iterates power set sizes. Equalities between alephs and beths require additional hypotheses such as GCH.

How this chapter fits together

Without Choice, arbitrary sets need not be comparable by well-orderable cardinal representatives. Hartogs' lemma nevertheless assigns to every set an ordinal too large to inject into it. This is enough to build the aleph hierarchy of well-orderable cardinals; the beth hierarchy separately follows repeated power sets. Keeping the two hierarchies distinct prepares the precise formulation of GCH in L.

Theorem 27.1 (Hartogs lemma)**.** *For every set X there exists an ordinal $h(X)$ that does not inject into X. The least such ordinal is the Hartogs number of X.*

How to read the symbols: Hartogs' theorem avoids Choice

The theorem does not choose a well-ordering of X. It looks at the set of all well-orderings that already exist on subsets of X, takes their order types, and then forms an ordinal larger than every type realized this way. If that larger ordinal injected into X, its image would itself be a well-ordered subset of X of the forbidden type. The construction therefore manufactures a non-injectable ordinal without well-ordering X globally.

Proof. Let X be a set. The set of all binary relations on subsets of X is contained in $\mathcal{P}(X \times X)$. By Separation, collect those relations that well order their fields. Replacement sends each such well-order to its unique ordinal order type; let A be the resulting set of ordinals. Put

$$h(X) = \sup A + 1.$$

If there were an injection $i : h(X) \to X$, the image $i[h(X)] \subseteq X$ could be well ordered by transporting the ordinal order on $h(X)$. That would produce a well-order of a subset of X having order type $h(X)$, so $h(X) \in A$. But $h(X) > \alpha$ for every $\alpha \in A$, contradiction. Hence no injection $h(X) \to X$ exists. This is Hartogs' theorem and requires no Choice.

□

Proposition 27.2 (Initial ordinals and alephs)**.** *A cardinal is an initial ordinal, i.e. an ordinal not equipotent with any smaller ordinal. Every well-orderable set is equipotent with a unique initial ordinal; the infinite initial ordinals are denoted $\aleph_\alpha$.*

Proof. A cardinal is an ordinal κ not bijective with any smaller ordinal. For every well-orderable set X, the set of ordinals bijective with X is nonempty, and ordinal well-ordering gives a least such ordinal; this is $|X|$. Starting from ω, Hartogs' theorem ensures that above every initial ordinal κ there is an ordinal not injectible into κ, hence there is a least cardinal strictly larger than κ, denoted κ^+. Define

$$\aleph_0 = \omega, \qquad \aleph_{\alpha+1} = \aleph_\alpha^+, \qquad \aleph_\lambda = \sup_{\beta<\lambda} \aleph_\beta.$$

The last supremum is an initial ordinal: if it were bijective with a smaller ordinal, that ordinal would bound all earlier $\aleph_\beta$, contradicting cofinality of the supremum. These are the well-ordered infinite cardinals, defined in ZF without asserting that every set is well orderable.

Point specific to this result. An initial ordinal is the least ordinal in its equinumerosity class. Hartogs guarantees that, for every well-orderable cardinal, there is a strictly larger ordinal and hence a next initial ordinal. □

Theorem 27.3 (Aleph recursion)**.** *There is a unique class function $\alpha \mapsto \aleph_\alpha$ such that $\aleph_0 = \omega$, $\aleph_{\alpha+1}$ is the least cardinal strictly above $\aleph_\alpha$, and $\aleph_\lambda = \sup_{\beta<\lambda} \aleph_\beta$ for limit λ.*

Proof. Apply the transfinite recursion theorem to the functional

$$F(s) = \begin{cases} \omega, & \operatorname{dom}(s) = 0, \\ (s(\beta))^+, & \operatorname{dom}(s) = \beta + 1, \\ \sup_{\gamma<\lambda} s(\gamma), & \operatorname{dom}(s) = \lambda \text{ limit}. \end{cases}$$

This yields a unique class function $\alpha \mapsto \aleph_\alpha$. Transfinite induction proves strict increase: at successors this is the definition of cardinal successor; at a limit λ, every $\aleph_\beta$ with $\beta < \lambda$ is below the supremum. It also proves continuity at limits,

$$\aleph_\lambda = \bigcup_{\beta<\lambda} \aleph_\beta.$$

Thus the aleph hierarchy is not merely notation: it is a transfinite recursively defined enumeration of all infinite initial ordinals.

Point specific to this result. Set $\aleph_0 = \omega$, take the next initial ordinal at successors, and take the least initial ordinal above all previous stages at limits. Transfinite recursion makes this definition uniform over all ordinals. □

Proposition 27.4 (Beth hierarchy and the relation to power sets)**.** *In ZF define sets B_α by transfinite recursion:*

$$B_0 = \omega, \qquad B_{\alpha+1} = \mathcal{P}(B_\alpha), \qquad B_\lambda = \bigcup_{\beta<\lambda} (\{\beta\} \times B_\beta)$$

for nonzero limit λ. These sets exist without Choice. If AC holds, each B_α is well orderable; writing $\beth_\alpha := |B_\alpha|$, one obtains

$$\beth_0 = \aleph_0, \qquad \beth_{\alpha+1} = 2^{\beth_\alpha}, \qquad \beth_\lambda = \sup_{\beta<\lambda} \beth_\beta.$$

Without AC the sets B_α still form a canonical power-set hierarchy, but their sizes need not be represented by initial ordinals.

Proof.

Formal derivation spine

Formal spine: separate the ZF set hierarchy from its AC cardinal sequence.

$$B_0 = \omega, \quad B_{\alpha+1} = \mathcal{P}(B_\alpha), \quad B_\lambda = \coprod_{\beta<\lambda} B_\beta.$$

Under AC,

$$|B_{\alpha+1}| = 2^{|B_\alpha|}, \qquad |B_\lambda| = \sup_{\beta<\lambda} |B_\beta|.$$

Transfinite recursion constructs the sets B_α. At a successor stage, the Power Set axiom gives $B_{\alpha+1}$. At a limit stage, Replacement collects the tagged products $\{\beta\} \times B_\beta$, and Union forms their disjoint union. No well-ordering is used, so the set-valued hierarchy exists in bare ZF.

Assume AC and put $\beth_\alpha := |B_\alpha|$. The successor equation follows directly from the definition of cardinal exponentiation:

$$\beth_{\alpha+1} = |\mathcal{P}(B_\alpha)| = 2^{\beth_\alpha}. \tag{1}$$

Let λ be a nonzero limit ordinal and set

$$\kappa := \sup_{\beta<\lambda} \beth_\beta.$$

The tagged union contains a copy of every B_β, so

$$\kappa \le |B_\lambda|. \tag{2}$$

For the reverse inequality, choose injections $i_\beta : B_\beta \to \kappa$. Then

$$(\beta, x) \longmapsto (\beta, i_\beta(x))$$

embeds B_λ into $\lambda \times \kappa$. The strictly increasing beth sequence gives $|\lambda| \le \kappa$, so it remains to justify $|\kappa \times \kappa| = \kappa$.

Here is the required AC cardinal calculation. If some infinite initial ordinal failed $\mu^2 = \mu$, let μ be the least one. Well-order $\mu \times \mu$ first by $\max\{\xi, \eta\}$ and then lexicographically. Every proper initial segment is contained in $(\gamma + 1)^2$ for some $\gamma < \mu$, and by minimality its cardinality is $< \mu$. If the entire order had cardinality $> \mu$, it would have a proper initial segment of order type μ, contradicting the preceding bound. Hence $\mu^2 \le \mu$, while the diagonal injection gives the reverse inequality. Therefore $\mu^2 = \mu$ for every infinite cardinal μ.

Applying this to κ,

$$|B_\lambda| \le |\lambda \times \kappa| \le |\kappa \times \kappa| = \kappa. \tag{3}$$

Equations (2)–(3) give

$$\beth_\lambda = |B_\lambda| = \sup_{\beta<\lambda} \beth_\beta.$$

Without AC, the sets B_α still exist, but a power set can fail to be well orderable; the symbols $\beth_\alpha$ as initial-ordinal cardinalities are therefore not introduced in that setting. □

Worked example

Worked Example 27.5 (Hartogs for a finite set)**.** If X has three elements, every well-order on a subset has order type at most 3, while the ordinal 4 cannot inject into X. Hartogs' theorem generalizes this elementary observation without first well-ordering X itself.

A useful warning

In bare ZF, do not write $|X| = \aleph_\alpha$ for an arbitrary set X unless its well-orderability has been established.

Looking ahead

The next chapter, *Well-founded relations and collapse*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 28

Well-founded relations and collapse

Definitions and setup

Formal definitions used in this chapter

Well-founded relation. Every nonempty subset of its field has an R-minimal member.
Extensional relation. Distinct points have distinct predecessor sets.
Collapse. The recursively defined map $\pi(a) = \{\pi(b) : bRa\}$.

A binary relation R on A is well founded when every nonempty subset of A has an R-minimal member. It is extensional when distinct points have distinct predecessor sets. The Mostowski collapse sends a recursively to the set of collapses of its predecessors. Well-foundedness guarantees recursion; extensionality guarantees injectivity; the image is transitive.

How this chapter fits together

Mostowski collapse replaces an abstract well-founded extensional relation by ordinary membership on a transitive set. The collapsing map is defined recursively from predecessor images. This theorem is the main device by which elementary hulls later become transitive models and, in L, actual constructible levels.

Theorem 28.1 (Mostowski collapse theorem)**.** *If R is a well-founded extensional relation on a set A, then there is a unique transitive set M and a unique isomorphism $\pi : (A, R) \cong (M, \in)$, given recursively by*

$$\pi(a) = \{\pi(b) : bRa\}.$$

Proof. Let (A, R) be well founded and extensional. Define by well-founded recursion

$$\pi(a) = \{\pi(b) : bRa\}.$$

Let $M = \pi[A]$. If $x = \pi(b) \in \pi(a)$, then $x \in M$, so M is transitive. Moreover

$$bRa \iff \pi(b) \in \pi(a).$$

It remains to prove injectivity. Suppose $\pi(a) = \pi(a')$. For each bRa, $\pi(b) \in \pi(a')$, so $\pi(b) = \pi(b')$ for some $b'Ra'$. By induction on the well-founded predecessor relation, equality of collapse values implies $b = b'$. Thus a and a' have the same R-predecessors; extensionality yields $a = a'$. Hence π is an isomorphism

$$(A, R) \cong (M, \in),$$

called the Mostowski collapse.

Point specific to this result. The collapse map is forced recursively by $\pi(a) = \{\pi(b) : bRa\}$. Extensionality is exactly the hypothesis that makes this map injective; well-foundedness is exactly what makes the recursion legitimate. □

Corollary 28.2 (Uniqueness of the transitive collapse)**.** *If $\pi : (A, R) \cong (M, \in)$ and $\pi' : (A, R) \cong (M', \in)$ are collapses onto transitive sets, then $M = M'$ and $\pi = \pi'$.*

Proof. Suppose $\pi : (A, R) \to (M, \in)$ and $\rho : (A, R) \to (N, \in)$ are two transitive collapses. We prove $\pi(a) = \rho(a)$ for every $a \in A$ by well-founded induction. Assume equality holds for all bRa. Then

$$\pi(a) = \{\pi(b) : bRa\} = \{\rho(b) : bRa\} = \rho(a).$$

Therefore $\pi = \rho$, and consequently $M = \pi[A] = \rho[A] = N$. Thus both the collapse map and the resulting transitive set are unique. The proof uses only the recursive equation characterizing a collapse; once that equation is established, no additional appeal to extensionality is needed at the uniqueness stage.

Point specific to this result. Any isomorphism into a transitive membership structure must satisfy the same recursion $a \mapsto \{\pi(b) : bRa\}$. Uniqueness of well-founded recursion therefore gives uniqueness of both the collapse map and its transitive range. □

Proposition 28.3 (Absoluteness of collapse data in the basic transitive setting)**.** *Let M be a transitive model of a finite fragment of ZF sufficient for well-founded recursion, and suppose $(A, R) \in M$. Assume that R is externally well founded and extensional on A. Then the collapse map π^M constructed inside M is the same set function as the ambient Mostowski collapse π; in particular, M and the ambient universe obtain the same transitive collapse target.*

Proof.

Formal derivation spine

Formal spine: uniqueness compares the two recursions.

$$\pi(a) = \{\pi(b) : bRa\}, \qquad \pi^M(a) = \{\pi^M(b) : bRa\}.$$

External well-founded induction yields $\pi^M(a) = \pi(a)$ for all $a \in A$.

Because M is transitive and $A, R \in M$, the externally interpreted sets A and R are the same objects inside and outside M. The assumed fragment of ZF lets M carry out well-founded recursion and obtain a function π^M satisfying

$$\pi^M(a) = \{\pi^M(b) : bRa\} \quad (a \in A). \tag{1}$$

The ambient Mostowski theorem gives a function π satisfying

$$\pi(a) = \{\pi(b) : bRa\}. \tag{2}$$

We compare them by external well-founded induction on R. Suppose $\pi^M(b) = \pi(b)$ for every bRa. Transitivity of M makes the predecessor set $\{b \in A : bRa\}$ the same in both computations, so (1) and (2) give

$$\pi^M(a) = \{\pi^M(b) : bRa\} = \{\pi(b) : bRa\} = \pi(a).$$

Thus $\pi^M = \pi$ on A. Their ranges are consequently identical. The external well-foundedness hypothesis is explicit: a transitive model can contain a relation it incorrectly regards as well founded only when the ambient relation is actually ill founded, and the present comparison does not cover that situation. □

Worked Theorem 28.4 (Explicit collapse of finite and countable coded well-founded relations)**.** *Let (A, R) be a finite or countable coded well-founded extensional relation. Define its rank by*

$$\mathrm{rk}_R(a) = \sup\{\mathrm{rk}_R(b) + 1 : bRa\}.$$

Then the collapse map is computed in rank order by

$$\pi(a) = \{\pi(b) : bRa\};$$

every value is determined at stage $\mathrm{rk}_R(a) + 1$*, even when the total relation has transfinite countable rank.*

Proof. Consider the finite extensional relation on $A = \{a, b, c\}$ given by

$$aRb, \qquad aRc, \qquad bRc.$$

Its ranks are

$$\mathrm{rk}_R(a) = 0, \qquad \mathrm{rk}_R(b) = 1, \qquad \mathrm{rk}_R(c) = 2.$$

The recursive collapse is therefore computed in that order:

$$\pi(a) = \varnothing,$$

$$\pi(b) = \{\pi(a)\} = \{\varnothing\},$$

$$\pi(c) = \{\pi(a), \pi(b)\} = \{\varnothing, \{\varnothing\}\}.$$

Thus (A, R) collapses to the transitive set $3 = \{0, 1, 2\}$, with a, b, c corresponding to $0, 1, 2$.

For a countable coded relation the same calculation may require transfinite countable stages. Let

$$A_\alpha := \{a \in A : \mathrm{rk}_R(a) < \alpha\}.$$

At stage $\alpha + 1$, define $\pi(a)$ for every a of rank α from values already defined on A_α. At a limit stage take the union of the previous partial functions. Well-foundedness ensures every a has an ordinal rank, so each individual value is assigned at a definite stage; no claim is made that the entire countable computation finishes after finitely many stages. □

Worked example

Worked Example 28.5 (Collapsing a finite relation)**.** Let aRb and suppose a has no predecessors. Then $\pi(a) = \varnothing$ and $\pi(b) = \{\varnothing\}$. If a third point c has predecessors a, b, then $\pi(c) = \{\varnothing, \{\varnothing\}\}$. The abstract diagram has become a transitive set with membership reproducing R.

A useful warning

Extensionality is essential for injectivity of the collapse. Two distinct points with exactly the same predecessor set would otherwise collapse to the same set.

End-of-volume perspective

The transfinite machinery now supplies the ambient language for inner models: ranks control size of constructions, ordinals index recursion, and collapse turns abstract well-founded structures into transitive sets.

Part V

Volume V: Choice, Models, Satisfaction, and Absoluteness

Preface

This volume develops the model-theoretic set theory needed later. Internal versus external viewpoints are kept explicit, especially for cardinality and satisfaction.

Position in the series

Volume V develops satisfaction, absoluteness, reflection, and elementary hulls. These tools feed directly into the construction and analysis of L, and they also clarify the model-theoretic viewpoint used in forcing.

Sources and further reading

The chapters on satisfaction, absoluteness, reflection, and elementary submodels prepare the exact model-theoretic infrastructure needed by constructibility and forcing; see [13, 14].

Notation for this volume

$M \models \varphi$ always refers to the satisfaction relation of the set-sized structure M. $M \preccurlyeq N$ denotes elementary substructure.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 29

Axiom of Choice and equivalents

Definitions and setup

Formal definitions used in this chapter

Axiom of Choice (AC). For every set A of nonempty sets there is a function f with

$$\operatorname{dom}(f) = A, \qquad \forall x \in A\ f(x) \in x.$$

Zorn property. If $(P, \leq)$ is a poset and every chain $C \subseteq P$ has an upper bound in P, then

$$\exists m \in P\ \forall p \in P\,(m \leq p \to p = m).$$

Well-ordering principle. For every set X there is a relation $<\subseteq X \times X$ such that every nonempty $Y \subseteq X$ has a $<$-least element.

The Axiom of Choice says every family of nonempty sets has a choice function. The well-ordering theorem says every set admits a well-order. Zorn's lemma says a partially ordered set in which every chain has an upper bound has a maximal element. Hausdorff's maximal principle says every chain is contained in a maximal chain. In ZF these are nontrivial equivalent principles, and the proofs must not use one of them before its implication from AC has been established.

How this chapter fits together

Choice is introduced through several equivalent maximality principles. The equivalences are useful because later arguments naturally ask for different forms: a choice function, a well-order, a maximal partial object, or a maximal chain. The proofs also provide a template for identifying exactly where Choice enters cardinal arithmetic.

Theorem 29.1 (AC implies the well-ordering theorem)**.** *Assuming the Axiom of Choice, every set X admits a well-ordering.*

Proof. Let X be a set and assume AC. By Hartogs' theorem there is an ordinal $h(X)$ for which no injection $h(X) \to X$ exists. Apply AC to the family of nonempty subsets of X and fix a choice function

$$c : \mathcal{P}(X) \setminus \{\varnothing\} \longrightarrow X, \qquad c(A) \in A.$$

We recursively choose new points of X. As long as earlier choices have not exhausted X, put

$$x_\alpha = c(X \setminus \{x_\beta : \beta < \alpha\})\,.$$

Transfinite recursion makes this definition legitimate at every stage at which the displayed complement is nonempty. The construction cannot continue through all $\alpha < h(X)$: if it did, the map $\alpha \mapsto x_\alpha$ would be an injection $h(X) \to X$, contradicting the defining property of the Hartogs ordinal. Hence there is a least ordinal $\gamma < h(X)$ such that

$$X = \{x_\beta : \beta < \gamma\}.$$

The map $e : \gamma \to X$, $e(\beta) = x_\beta$, is therefore bijective. Transport the ordinal order on γ to X:

$$x_\alpha <_X x_\beta \quad \Longleftrightarrow \quad \alpha < \beta.$$

Every nonempty $Y \subseteq X$ corresponds under e^{-1} to a nonempty subset of the ordinal γ, hence has a least index. Its image is the $<_X$-least element of Y. Thus $<_X$ is a well-order of X. Notice where Choice enters: it supplies one coherent selector c that can be used at every transfinite stage; Hartogs prevents the recursion from choosing distinct points forever. □

Theorem 29.2 (Well-ordering theorem implies AC)**.** *If every set can be well ordered, then every family $\mathcal{A}$ of nonempty sets has a choice function f with $f(A) \in A$ for every $A \in \mathcal{A}$.*

Proof. Assume every set can be well ordered. Let $\mathcal{A}$ be a set of nonempty sets. We need a single function f with domain $\mathcal{A}$ and $f(A) \in A$ for all $A \in \mathcal{A}$.

Form the union

$$U := \bigcup \mathcal{A}.$$

By hypothesis, choose a well-order $<$ of U. Every $A \in \mathcal{A}$ is a nonempty subset of U, so it has a unique $<$-least element. Define

$$f(A) := \min_{<} A.$$

The graph of f is the set

$$\{\langle A, x\rangle \in \mathcal{A} \times U : x \in A \ \wedge \ \forall y \in A\,(x \leq y)\},$$

which exists by Separation. Uniqueness of least elements makes this graph functional, and the nonemptiness of each A makes its domain all of $\mathcal{A}$. Hence f is a choice function.

No additional use of Choice is hidden in the construction: the only global selection is the assumed well-order of U. Once that order is fixed, the value $f(A)$ is defined uniquely by a first-order property rather than selected afresh for each A. □

Theorem 29.3 (AC is equivalent to Zorn's lemma)**.** *Over ZF, the Axiom of Choice and Zorn's Lemma are equivalent. Zorn's Lemma states: if every chain in a partially ordered set P has an upper bound in P, then P has a maximal element.*

Proof. We prove both implications over ZF.

AC ⇒ Zorn. By the preceding theorem, AC well-orders every set. Let $(P, \leq)$ be a poset in which every chain has an upper bound. Well-order P as $\langle p_\alpha : \alpha < \theta\rangle$. By transfinite recursion build a chain C: at stage α, add p_α exactly when it is comparable with every element already chosen. The resulting C is a maximal chain: any $p \in P$ comparable with all of C would have been added when its index was considered. By hypothesis, C has an upper bound $u \in P$. If u were not maximal in P, choose $v > u$. Since u bounds C, the element v is comparable with every member of C, so $C \cup \{v\}$ would be a strictly larger chain, contradiction. Thus u is maximal.

Zorn ⇒ AC. Let $\mathcal{A}$ be a family of nonempty sets. Let P be the set of partial choice functions:

$$P = \{f : \mathrm{dom}(f) \subseteq \mathcal{A}, \ \forall A \in \mathrm{dom}(f) \ f(A) \in A\},$$

ordered by inclusion. If $C \subseteq P$ is a chain, then $\bigcup C$ is a function because members of a chain agree on overlaps, and it is again a partial choice function. Hence every chain has an upper bound. Zorn gives a maximal $f \in P$. If some $A \in \mathcal{A} \setminus \mathrm{dom}(f)$ remained, choose one $a \in A$ (one existential choice, not AC) and extend f by $f(A) = a$, contradicting maximality. Therefore $\mathrm{dom}(f) = \mathcal{A}$, so f is a choice function. □

Theorem 29.4 (AC is equivalent to the Hausdorff maximal principle)**.** *Over ZF, the Axiom of Choice is equivalent to the Hausdorff maximal principle: every chain in a partially ordered set is contained in a maximal chain.*

Proof. Again we prove the two directions over ZF.

Assume AC. By the equivalence just proved, Zorn's lemma is available. Given a poset $(P, \leq)$ and a chain $C_0 \subseteq P$, let

$$\mathcal{C} = \{C \subseteq P : C \text{ is a chain and } C_0 \subseteq C\},$$

ordered by inclusion. A chain $\mathcal{D} \subseteq \mathcal{C}$ has upper bound $\bigcup \mathcal{D}$: any two of its elements lie together in some larger member of the inclusion-chain, so they are comparable in P. Zorn therefore gives a maximal element of $\mathcal{C}$, namely a maximal chain containing C_0. Taking $C_0 = \varnothing$ yields Hausdorff's maximal principle.

Conversely assume Hausdorff's maximal principle. To recover Zorn, let $(P, \leq)$ be a poset in which every chain has an upper bound. Choose a maximal chain $C \subseteq P$. Let u be an upper bound of C. If u were not maximal in P, there would exist $v \in P$ with $u < v$. Every $c \in C$ satisfies $c \leq u < v$, hence $C \cup \{v\}$ is a strictly larger chain, contradiction. Thus u is maximal and Zorn's lemma holds. The previous theorem then gives AC. Hence AC and Hausdorff's maximal principle are equivalent. □

Worked example

Worked Example 29.5 (Choosing by a well-order)**.** If a family $\mathcal{A}$ of nonempty sets has union U and U is well-ordered, define $f(A)$ to be the least element of A for each $A \in \mathcal{A}$. This one-line construction proves the easy direction from the well-ordering theorem to AC.

A useful warning

Zorn's lemma requires every chain in the chosen poset to have an upper bound in that poset. Verifying this closure condition is part of the proof, not an automatic formality.

Looking ahead

The next chapter, *Cardinal arithmetic with choice*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 30

Cardinal arithmetic with choice

Definitions and setup

Formal definitions used in this chapter

Cofinality. $\mathrm{cf}(\alpha)$ is the least order type of an unbounded subset of a limit ordinal α.
Cardinal sum/product. Sizes of disjoint union and Cartesian product after cardinal representatives are chosen.

Under AC every set can be well ordered, so every cardinal is represented by an initial ordinal and arbitrary cardinals are comparable. Cofinality $\mathrm{cf}(\alpha)$ is the least order type of an unbounded subset of an ordinal α. König's theorem is a diagonal inequality between sums and products of cardinals and is the basic obstruction controlling exponentiation. Choice assumptions are carried explicitly into every cardinal-arithmetic statement.

How this chapter fits together

Once every set can be well-ordered, cardinals may be represented by initial ordinals and compared linearly. Infinite-cardinal addition and multiplication simplify dramatically, but exponentiation and cofinality retain genuine structure. König's theorem supplies a diagonal obstruction that will later control cardinal exponentiation.

Proposition 30.1 (Cardinal comparability under AC)**.** *Assuming AC, for any sets A, B, either $|A| \le |B|$ or $|B| \le |A|$. Equivalently, the cardinal numbers are linearly ordered by injection.*

Proof. Assume AC and let A, B be arbitrary sets. By the well-ordering theorem choose well-orders of A and B. By the ordinal representation theorem from Volume IV, there are unique ordinals α, β and bijections

$$e_A : \alpha \xrightarrow{\cong} A, \qquad e_B : \beta \xrightarrow{\cong} B.$$

Replace α, β, if necessary, by the least ordinals equipotent with A, B; these are their initial-ordinal representatives, so $|A| = \alpha$ and $|B| = \beta$ as cardinals.

Ordinal trichotomy gives exactly one of

$$\alpha \in \beta, \qquad \alpha = \beta, \qquad \beta \in \alpha.$$

If $\alpha \le \beta$, the inclusion $i : \alpha \hookrightarrow \beta$ is injective. Transport it through the two bijections:

$$A \xrightarrow{e_A^{-1}} \alpha \xrightarrow{i} \beta \xrightarrow{e_B} B.$$

This is an injection $A \hookrightarrow B$, so $|A| \leq |B|$. The case $\beta \leq \alpha$ is symmetric. Thus any two cardinals are comparable.

The proof shows exactly why Choice is needed: ordinal trichotomy compares only well-orderable representatives. AC is what ensures every set has such a representative. Without AC, arbitrary cardinals need not be linearly ordered by injection. □

Theorem 30.2 (Arithmetic of infinite cardinals under AC)**.** *Assume AC and let κ be an infinite cardinal. Then*

$$\kappa + \kappa = \kappa \cdot \kappa = \kappa.$$

Equivalently, if X is any infinite set, then $X \sqcup X$ and $X \times X$ are equipotent with X.

Proof. By AC it suffices to prove the assertion for an infinite initial ordinal κ. The map into either summand proves $\kappa \leq \kappa + \kappa$, and $\alpha \mapsto (\alpha, 0)$ proves $\kappa \leq |\kappa \times \kappa|$. The substantive point is an injection in the opposite direction.

We now prove by transfinite induction that for every infinite initial ordinal κ,

$$|\kappa \times \kappa| = \kappa.$$

Here is the induction mechanism. Well-order $\kappa \times \kappa$ by first comparing $\max\{\alpha, \beta\}$, and within each level use a fixed lexicographic tie-break. For $\gamma < \kappa$, the initial segment below the level γ is contained in

$$(\gamma + 1) \times (\gamma + 1).$$

Its cardinal is strictly below κ: if $\lambda = |\gamma + 1| < \kappa$, the induction hypothesis at the smaller infinite cardinal λ gives $\lambda^2 = \lambda < \kappa$ (the finite case is trivial). Hence every proper initial segment of this well-order has cardinality $< \kappa$. Recursively assign to each pair the least unused ordinal below κ. At each stage fewer than κ ordinals have been used, so one remains. This yields an injection $\kappa \times \kappa \hookrightarrow \kappa$. Cantor–Bernstein gives $|\kappa \times \kappa| = \kappa$.

Since $\kappa \sqcup \kappa$ injects into $\kappa \times \{0, 1\} \subseteq \kappa \times \kappa$ and κ injects into the disjoint union, Cantor–Bernstein also gives $\kappa + \kappa = \kappa$. Transporting the bijections along a well-ordering of an arbitrary infinite set X gives the set formulation. □

Proposition 30.3 (Cofinality basics)**.** *For a limit ordinal α, $\mathrm{cf}(\alpha)$ is the least order type of an unbounded subset of α. It is a cardinal, $\mathrm{cf}(\alpha) \leq |\alpha|$, and $\mathrm{cf}(\alpha)$ is regular.*

Proof. Let α be a limit ordinal. A subset $C \subseteq \alpha$ is *cofinal* if

$$\forall \beta < \alpha \; \exists \gamma \in C \; (\beta \leq \gamma).$$

Define $\mathrm{cf}(\alpha)$ to be the least order type of a cofinal subset of α. Equivalently, it is the least ordinal κ for which there exists a cofinal map $f : \kappa \to \alpha$.

First, the least such ordinal is a cardinal. If $\kappa = \mathrm{cf}(\alpha)$ were equipotent with a smaller ordinal $\lambda < \kappa$, compose a bijection $\lambda \to \kappa$ with a cofinal map $\kappa \to \alpha$; this would give a cofinal map from λ, contradicting minimality. Thus κ is an initial ordinal.

Second, $\mathrm{cf}(\alpha) \leq |\alpha|$: the identity map on α is cofinal, so the least size of a cofinal domain cannot exceed the cardinality of α.

Finally, $\kappa = \mathrm{cf}(\alpha)$ is regular. Suppose $\mathrm{cf}(\kappa) = \lambda < \kappa$. Choose cofinal maps

$$g : \lambda \to \kappa, \qquad f : \kappa \to \alpha.$$

Replacing f by the increasing map $\xi \mapsto \sup f[\xi + 1]$ if needed, we may assume it is increasing and cofinal. Then $f \circ g : \lambda \to \alpha$ is cofinal: given $\beta < \alpha$, choose $\xi < \kappa$ with $\beta \leq f(\xi)$, then choose $\eta < \lambda$ with $\xi \leq g(\eta)$, so $\beta \leq f(g(\eta))$. This contradicts minimality of κ. Hence $\mathrm{cf}(\kappa) = \kappa$. □

Theorem 30.4 (König's theorem and its cardinal-exponent corollaries)**.** *Assume AC. If $\kappa_i < \lambda_i$ for every $i \in I$, then*

$$\sum_{i\in I} \kappa_i < \prod_{i\in I} \lambda_i.$$

Consequently, for every infinite cardinal κ,

$$\kappa^{\mathrm{cf}(\kappa)} > \kappa.$$

Proof. Assume $\kappa_i < \lambda_i$ for each $i \in I$. Realize the cardinal sum as a disjoint union

$$S = \bigsqcup_{i\in I} K_i, \qquad |K_i| = \kappa_i,$$

and the product as $P = \prod_{i\in I} L_i$, with $|L_i| = \lambda_i$. Suppose toward contradiction that there is a surjection

$$F : S \twoheadrightarrow P.$$

For each i, look only at the i-th block $K_i \times \{i\} \subseteq S$. The set of i-th coordinates occurring among its images is

$$R_i := \{F(x, i)(i) : x \in K_i\} \subseteq L_i.$$

Because $|R_i| \le \kappa_i < \lambda_i = |L_i|$, choose $y_i \in L_i \setminus R_i$. AC is available in this volume, so the choices y_i form a point $y = (y_i)_{i\in I} \in P$. If $F(x, i) = y$, then its i-th coordinate would be $y_i \in R_i$, contradiction. Hence F is not surjective. In cardinal notation,

$$\sum_{i\in I} \kappa_i < \prod_{i\in I} \lambda_i.$$

For the exponentiation corollary, let $\lambda = \mathrm{cf}(\kappa)$ and choose an increasing cofinal sequence $\langle \kappa_i : i < \lambda\rangle$ of cardinals below κ. Put $\lambda_i = \kappa_i^+ \le \kappa$. Cofinality gives

$$\sum_{i<\lambda} \kappa_i = \kappa.$$

König's inequality yields

$$\kappa < \prod_{i<\lambda} \kappa_i^+ \le \prod_{i<\lambda} \kappa = \kappa^\lambda = \kappa^{\mathrm{cf}(\kappa)}.$$

This is the claimed strict inequality. □

Worked example

Worked Example 30.5 (A cofinality calculation)**.** The natural numbers are cofinal in ω, so $\mathrm{cf}(\omega) = \omega$. By contrast, any finite subset of ω is bounded. For a successor ordinal $\alpha + 1$, the singleton $\{\alpha\}$ is cofinal, so its cofinality is 1 as an ordered set even though cardinal cofinality conventions focus on infinite initial ordinals.

A useful warning

The identity $\kappa + \kappa = \kappa$ for infinite cardinals is a theorem using well-orderability; it is not a theorem about arbitrary infinite sets in ZF.

Looking ahead

The next chapter, *Satisfaction for set-sized structures*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 31

Satisfaction for set-sized structures

Definitions and setup

Formal definitions used in this chapter

Set-sized satisfaction. For a set M, $\mathrm{Sat}(M, \ulcorner\varphi\urcorner, a)$ is the recursively defined truth relation for the structure $(M, \in)$.
Relativization. φ^M restricts every formerly unbounded quantifier of φ to M.

For a set-sized structure M, the satisfaction relation is itself a set relation on formula codes and finite assignments. It is defined recursively on formula complexity. Tarski's later undefinability theorem forbids an arithmetical truth predicate for the entire standard arithmetic universe; there is no conflict because here the structure is a fixed set and satisfaction is constructed externally in set theory.

How this chapter fits together

Set theory can define satisfaction for any set-sized structure because formulas and assignments form sets and the recursion on formula complexity is set-sized. This is the technical basis for definability, relativization, reflection, and L. It does not contradict Tarski: no full truth predicate for the entire universe is being defined inside that universe.

Theorem 31.1 (Recursive construction of satisfaction for set-sized first-order structures)**.** *For every set-sized $\mathcal{L}$-structure M, there is a unique relation $\mathrm{Sat}_M(\ulcorner\varphi\urcorner, a)$ between formula codes and assignments satisfying the Tarski recursion for atomic formulas, Boolean connectives, and quantifiers.*

How to read the symbols: satisfaction is recursive evaluation

Think of $\mathrm{Sat}_M(\ulcorner\varphi\urcorner, s)$ as an interpreter whose inputs are a formula code and an assignment. Atomic formulas call the structure's interpretations of function and relation symbols; Boolean connectives recursively call the interpreter on subformulas; $\exists x$ loops over the set-sized domain. Tarski's later undefinability theorem does not conflict with this construction because the interpreter is built externally for one fixed set-sized structure.

Proof. Fix a set-sized $\mathcal{L}$-structure M. Let $\mathrm{Form}_{\mathcal{L}}$ be the set of formula codes and let Asn_M be the set of finite assignments into M. We define

$$\mathrm{Sat}_M \subseteq \mathrm{Form}_{\mathcal{L}} \times \mathrm{Asn}_M$$

by recursion on formula complexity.

For an atomic equality $t = u$, set

$$\mathrm{Sat}_M(\ulcorner t = u \urcorner, s) \iff t^M[s] = u^M[s].$$

For an atomic relation $R(t_1, \ldots, t_n)$, use membership of the tuple of term values in R^M. For Boolean connectives impose exactly the truth-table clauses, for example

$$\mathrm{Sat}_M(\ulcorner \neg\varphi \urcorner, s) \iff \neg\, \mathrm{Sat}_M(\ulcorner \varphi \urcorner, s),$$

$$\mathrm{Sat}_M(\ulcorner \varphi \wedge \psi \urcorner, s) \iff \mathrm{Sat}_M(\ulcorner \varphi \urcorner, s) \wedge \mathrm{Sat}_M(\ulcorner \psi \urcorner, s).$$

For existential quantification,

$$\mathrm{Sat}_M(\ulcorner \exists x\, \varphi \urcorner, s) \iff \exists a \in M\ \ \mathrm{Sat}_M(\ulcorner \varphi \urcorner, s[x \mapsto a]).$$

Every recursive call is made on a proper subformula, so the recursion is well founded. Since the formula codes, assignments, and each earlier stage are sets, Replacement and Separation collect the recursive stages into a set relation.

Uniqueness is proved by induction on formula complexity. If S and S' satisfy the same Tarski clauses, they agree on atomic formulas. Assuming agreement for the immediate subformulas, each Boolean or quantifier clause determines the next truth value uniquely. Hence $S = S'$. This gives the unique satisfaction relation Sat_M. □

Theorem 31.2 (Definability of the satisfaction predicate over the ambient set theory)**.** *There is a first-order formula of set theory* $\mathrm{Sat}(M, n, a)$ *that, for every set-sized coded structure* M*, formula code* $n = \ulcorner \varphi \urcorner$*, and assignment* a*, expresses* $M, a \models \varphi$*.*

Proof. The preceding construction was uniform: none of its clauses depended on a special feature of one particular structure beyond the coded interpretations of its symbols. We now internalize that recursion in the language of set theory.

Code a structure M, a formula n, and an assignment a as sets. A *satisfaction computation up to complexity* k is a set S whose domain consists of formula/assignment pairs of complexity at most k and which satisfies the atomic, Boolean, and quantifier clauses from the previous theorem. Each clause is first-order expressible in set theory because it quantifies only over the coded structure, the finite syntax tree, assignments, and elements of the set-sized domain.

Define

$$\mathrm{Sat}(M, n, a)$$

to mean: there exists a satisfaction computation S up to the complexity of n such that $(n, a) \in S$. By the uniqueness part of the recursive-construction theorem, two such computations agree on their common domain, so this definition does not depend on which witness S is chosen. Conversely the set recursion constructs such an S, so the formula has exactly the intended extension.

Therefore, for every coded set-sized structure M, every formula φ, and every assignment a,

$$V \models \mathrm{Sat}(M, \ulcorner \varphi \urcorner, a) \quad \iff \quad M, a \models \varphi.$$

The point is uniform definability in the ambient set theory, not a truth predicate for the whole universe. The domain being evaluated is the fixed set M, which is why there is no conflict with Tarski's undefinability theorem. □

Lemma 31.3 (Relativization lemma)**.** *Let* M *be a class/set and let* φ *be a formula of set theory. Its relativization* φ^M *is obtained by replacing* $\exists x$ *by* $\exists x \in M$ *and* $\forall x$ *by* $\forall x \in M$*. For transitive set models with the inherited membership relation,*

$$(M, \in) \models \varphi(\bar{a}) \iff V \models \varphi^M(\bar{a}).$$

Proof. We prove the equivalence by induction on the construction of φ. Atomic formulas are unchanged by relativization, and because M is transitive the inherited membership and equality relations agree with the ambient ones on parameters from M. Thus the assertion is true for atomic formulas.

The Boolean steps follow immediately from the induction hypothesis. For example,

$$(M,\in) \models \neg\psi(\bar a) \iff \neg\big((M,\in)\models\psi(\bar a)\big) \iff V \models \neg\psi^M(\bar a).$$

The quantifier step is the substantive one. Suppose $\varphi(\bar y) = \exists x\,\psi(x,\bar y)$. By the semantics of the structure $(M,\in)$,

$$(M,\in)\models \exists x\,\psi(x,\bar a) \iff \exists b\in M\ (M,\in)\models\psi(b,\bar a).$$

By the induction hypothesis this is equivalent to

$$\exists b\in M\ V\models\psi^M(b,\bar a),$$

which is exactly

$$V\models (\exists x\in M)\,\psi^M(x,\bar a) = (\exists x\,\psi)^M(\bar a).$$

Universal quantifiers are analogous or follow by negation. Hence

$$(M,\in)\models\varphi(\bar a) \iff V\models\varphi^M(\bar a)$$

for all parameters $\bar a\in M$. □

Proposition 31.4 (Parameter bookkeeping for satisfaction and definability)**.** *The satisfaction relation is uniform in the finite tuple of parameters: if $\varphi(x,\bar y)$ is fixed, then the set it defines over M from parameters $\bar a\in M$ is determined by the code $\ulcorner\varphi\urcorner$ and a finite coded tuple $\bar a$.*

Proof. Fix a formula $\varphi(x,y_0,\dots,y_{n-1})$. If $\bar a=(a_0,\dots,a_{n-1})\in M^n$, the subset of M defined by φ with parameters $\bar a$ is

$$X_{\varphi,\bar a} := \{x\in M : M\models\varphi(x,\bar a)\}.$$

Using the uniform satisfaction formula from the preceding theorem, this becomes

$$X_{\varphi,\bar a} = \{x\in M : \mathrm{Sat}(M,\ulcorner\varphi\urcorner,\langle x,a_0,\dots,a_{n-1}\rangle)\}.$$

Thus the defining data consist of the natural-number code $\ulcorner\varphi\urcorner$ and the finite tuple $\bar a$. Finite tuples are themselves uniformly coded as sets, so the relation

$$(e,\bar a,x)\longmapsto \mathrm{Sat}(M,e,\langle x,\bar a\rangle)$$

is a single first-order relation of the ambient set theory.

This bookkeeping matters later in the construction of $\mathrm{Def}(M)$. There are only set-many formula codes and set-many finite tuples from a set M:

$$\omega\times M^{<\omega}$$

is a set. Replacement can therefore map each pair $(e,\bar a)$ to the corresponding definable subset. Nothing like a proper-class collection of parameters is being hidden in the phrase "definable with parameters." □

Worked example

Worked Example 31.5 (Relativizing a quantifier)**.** If M is a set and φ is $\forall x\,\exists y\,(x\in y)$, its relativization is $\forall x\in M\,\exists y\in M\,(x\in y)$. The satisfaction recursion interprets the latter by ranging only over elements of M. This simple change is the core of every later "$M\models\varphi$" calculation.

A useful warning

A satisfaction predicate for a set-sized model is an object of the surrounding universe. Do not turn it into a global truth predicate for V.

Looking ahead

The next chapter, *Absoluteness*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 32

Absoluteness

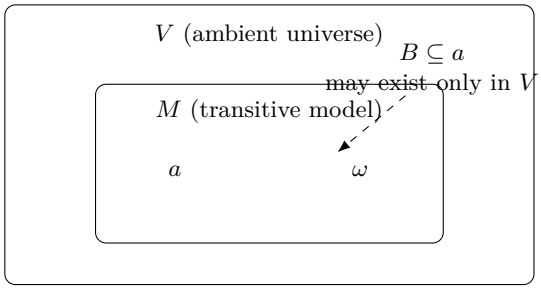

Figure 32.1: Bounded formulas are absolute because a transitive model sees the same members of an existing set. Power sets and cardinalities can differ because the ambient universe may contain additional subsets or functions.

Definitions and setup

Formal definitions used in this chapter

Absolute formula. φ is absolute between $M \subseteq N$ for parameters in M when $M \models \varphi(\bar{a}) \iff N \models \varphi(\bar{a})$.
Δ_0 **formula.** Every quantifier is bounded, of the form $\forall x \in y$ or $\exists x \in y$.

A statement is absolute between models when it has the same truth value in both. For transitive models, bounded quantifiers over an existing set range over the same elements, which is why Δ_0 formulas are absolute. Power Set and cardinality are typical nonabsolute notions because a smaller model may omit subsets or functions that exist externally. Every later absoluteness claim names its model hypotheses.

How this chapter fits together

Absoluteness asks whether an assertion has the same truth value in a transitive model and the ambient universe. Bounded formulas are absolute because their quantifiers range over elements of already present sets. Cardinality and power set are not: the smaller model may simply lack functions or subsets that exist externally.

Theorem 32.1 (Delta-0 absoluteness for transitive models)**.** *If $M \subseteq N$ are transitive sets/classes and $\bar{a} \in M$, then every Δ_0 formula φ is absolute:*

$$M \models \varphi(\bar{a}) \iff N \models \varphi(\bar{a}).$$

Proof. Let $M \subseteq N$ be transitive and let $\bar{a} \in M$. We prove by induction on Δ_0 formulas that

$$M \models \varphi(\bar{a}) \iff N \models \varphi(\bar{a}).$$

Atomic equality and membership are absolute because both structures use the ambient equality and membership relations. Boolean connectives preserve equivalence by the induction hypothesis.

For the bounded existential step, suppose

$$\varphi(\bar{a}) \equiv \exists x \in b\, \psi(x, \bar{a}), \qquad b \in M.$$

Transitivity of M gives $b \subseteq M$; since $M \subseteq N$, both structures quantify over exactly the same actual elements of b. Therefore

$$M \models \exists x \in b\, \psi(x, \bar{a})$$

iff there is $x \in b$ with $M \models \psi(x, \bar{a})$, iff by induction there is $x \in b$ with $N \models \psi(x, \bar{a})$, iff

$$N \models \exists x \in b\, \psi(x, \bar{a}).$$

Bounded universal quantifiers are identical or follow by negation. Since every quantifier in a Δ_0 formula is bounded, this completes the induction.

The proof also explains the boundary of the theorem. An unbounded quantifier may range over objects in $N \setminus M$, so the same argument does not apply to arbitrary formulas. Later non-absoluteness examples exploit exactly that difference. □

Proposition 32.2 (Absoluteness of ordinals for transitive models)**.** *If M is transitive and $\alpha \in M$, then $M \models$ "α is an ordinal" if and only if α is an ordinal in the ambient universe; moreover the order relation on such ordinals is the same.*

Proof. Use the standard set-theoretic definition

$$\mathrm{Ord}(\alpha) \iff \alpha \text{ is transitive and } \in \text{ linearly orders } \alpha.$$

Every quantifier in this definition is bounded by α, so $\mathrm{Ord}(\alpha)$ is Δ_0. If M is transitive and $\alpha \in M$, the Δ_0-absoluteness theorem gives

$$M \models \mathrm{Ord}(\alpha) \iff V \models \mathrm{Ord}(\alpha).$$

Thus M and the ambient universe agree about whether α is an ordinal.

If $\alpha, \beta \in M$ are ordinals, their order is given by membership:

$$\alpha < \beta \quad \iff \quad \alpha \in \beta.$$

Membership is literally the same relation inside a transitive submodel and outside it. Hence

$$M \models \alpha < \beta \iff \alpha \in \beta \iff V \models \alpha < \beta.$$

In particular, ordinal comparison and all bounded statements about initial segments of ordinals are absolute between the two settings. What may fail to be absolute is whether a given ordinal is a *cardinal*, because cardinality asks for the existence or nonexistence of functions, and an inner model can omit functions that exist outside it. □

Proposition 32.3 (Absoluteness of omega and elementary finite constructions in the required setting)**.** *Let M be a transitive model of ZF and assume $\omega \in M$. Then*

$$\omega^M = \omega.$$

Moreover, for all $m, n \in \omega$, successor, addition, and multiplication computed in M agree with the ambient operations, and every finite sequence of natural numbers belonging to M has the same domain, length, and entries in M and in the ambient universe.

Proof. Because $\omega \in M$ and M is transitive, every $n \in \omega$ belongs to M. The ambient ω is an inductive set in M, so by minimality of ω^M,

$$\omega^M \subseteq \omega.$$

Conversely ω^M is inductive in M. The assertions $0 \in \omega^M$ and $n \in \omega^M \Rightarrow n+1 \in \omega^M$ are absolute for the finite ordinals, so external induction yields $n \in \omega^M$ for every $n < \omega$. Hence

$$\omega \subseteq \omega^M,$$

and therefore $\omega^M = \omega$.

Addition and multiplication on ω are characterized by their recursion equations. For example,

$$m + 0 = m, \qquad m + (n+1) = (m+n) + 1.$$

The same equations hold in M, and induction on the ambient natural number n shows that the value computed in M equals the ambient value. The multiplication proof is identical after using the already established absoluteness of addition.

A finite sequence $s = \langle s(0), \dots, s(k-1) \rangle$ is coded by finitely many applications of pairing, union, and the natural-number operations. These constructions are absolute in a transitive model satisfying the indicated elementary ZF axioms. Hence M and V agree about the length k, the domain k, and every entry $s(i)$. This is the finite-syntax absoluteness used later. □

Worked Theorem 32.4 (Failure of absoluteness for cardinality and power set)**.** *Power set is not absolute even between transitive sets. With von Neumann naturals $0 = \varnothing$, $1 = \{0\}$, $2 = \{0, 1\}$, put*

$$M = \{0, 1, 2\}, \qquad N = M \cup \{\{1\}\}, \qquad x = 2.$$

Then $M \subsetneq N$ are transitive and

$$\mathcal{P}(x) \cap M \subsetneq \mathcal{P}(x) \cap N.$$

More generally, for transitive models $M \subseteq N$, cardinality statements can also fail to be absolute when a witnessing injection, surjection, or bijection exists in N but not in M.

Proof. Let

$$0 = \varnothing, \qquad 1 = \{0\}, \qquad 2 = \{0, 1\},$$

and define

$$M = \{0, 1, 2\}, \qquad N = M \cup \{\{1\}\}.$$

Both M and N are transitive: every element of an element of either set is again in that set. Put $x = 2 = \{0, 1\}$. The actual power set is

$$\mathcal{P}(x) = \{\varnothing, \{0\}, \{1\}, \{0, 1\}\} = \{0, 1, \{1\}, 2\}.$$

Hence

$$\mathcal{P}(x) \cap M = \{0, 1, 2\}, \qquad \mathcal{P}(x) \cap N = \{0, 1, \{1\}, 2\},$$

so

$$\mathcal{P}(x) \cap M \subsetneq \mathcal{P}(x) \cap N.$$

This is a completely explicit failure of Power-Set absoluteness.

Cardinality fails for the same structural reason but requires an infinite setting. If $M \subseteq N$ are transitive models, $x, \kappa \in M$, and a bijection $f : x \to \kappa$ belongs to $N \setminus M$, then N may satisfy $|x| = \kappa$ while M does not, simply because M's quantifier "there exists a bijection" ranges only over functions in M. Thus transitivity makes bounded membership facts absolute, not existence claims about arbitrary subsets or functions. □

Worked example

Worked Example 32.5 (A missing subset)**.** If M is a transitive model and $a \in M$, then every element of a is in M; bounded quantifiers over a therefore see the same domain. But $\mathcal{P}^M(a) = \mathcal{P}(a) \cap M$ can be strictly smaller than the ambient $\mathcal{P}(a)$. "All subsets" is consequently not absolute.

A useful warning

Transitivity alone does not make arbitrary formulas absolute. Unbounded quantifiers may range over many more sets in the ambient universe.

Looking ahead

The next chapter, *Reflection*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 33

Reflection

Definitions and setup

Formal definitions used in this chapter

Reflection for Φ. An ordinal α reflects a finite formula set Φ when V_α and V agree on every formula in Φ for parameters from V_α.

Reflection is a finite-fragment theorem: given finitely many formulas and parameters, some rank V_α agrees with the universe on those formulas. The proof closes the formula set under subformulas and then ensures existential witnesses appear below a common rank. It does not produce a definable truth predicate and it does not assert that a single V_α satisfies every axiom of ZF.

How this chapter fits together

Reflection says that for every fixed finite collection Φ of formulas and every finite parameter tuple $\bar{a}$, there is an ordinal α with $\bar{a} \in V_\alpha$ such that every $\varphi \in \Phi$ is absolute between V_α and V on parameters from V_α. The construction closes under witnesses for existential subformulas and then takes a limit stage. It is a local approximation theorem, not a claim that one set-sized V_α is elementary in the entire universe for every formula.

Lemma 33.1 (Finite-fragment closure construction for formulas)**.** *Given finitely many formulas Φ and a rank α_0, one can construct an increasing ω-sequence of ordinals whose limit α is closed under the witness requirements for every existential subformula occurring in Φ.*

Proof. Let Φ be finite. First replace it by the finite set Φ^* of all subformulas of formulas in Φ; reflection for Φ^* implies reflection for Φ. Fix α_0.

Suppose α_n has been chosen. For every existential formula

$$\exists x\, \psi(x, \bar{y}) \in \Phi^*$$

and every parameter tuple $\bar{a} \in V_{\alpha_n}$, if $V \models \exists x\, \psi(x, \bar{a})$, choose one witness $b_{\psi,\bar{a}}$ and its rank. There are set-many relevant tuples and only finitely many formulas, so Replacement collects the witness ranks into a set. Choose $\alpha_{n+1} > \alpha_n$ such that every selected witness lies in $V_{\alpha_{n+1}}$.

Set

$$\alpha = \sup_{n<\omega} \alpha_n.$$

Then $V_\alpha = \bigcup_{n<\omega} V_{\alpha_n}$. The crucial closure property is:

$$\bar{a} \in V_\alpha \ \wedge \ V \models \exists x\, \psi(x, \bar{a}) \Longrightarrow \exists b \in V_\alpha \ V \models \psi(b, \bar{a})$$

for every existential subformula in Φ^*. Indeed $\bar{a}$ already belongs to some V_{α_n}, and the witness selected at stage $n+1$ lies below α. This witness-closure is exactly the ingredient needed for the existential step of the Reflection Theorem. □

Theorem 33.2 (Reflection theorem in the form used later)**.** *For every finite set Φ of formulas of set theory and every set of parameters A, there is an ordinal α with $A \subseteq V_\alpha$ such that for every $\varphi \in \Phi$ and $\bar{a} \in V_\alpha$,*

$$V \models \varphi(\bar{a}) \iff V_\alpha \models \varphi(\bar{a}).$$

Commented symbolic trace: reflection is finite dependency closure

Fix finitely many formulas. Close that finite list under subformulas so every recursive truth check that may be needed is already present. Then build a sequence of ranks large enough to contain witnesses whenever one of those existential formulas becomes true. At the limit of the sequence, induction on formula complexity works because the required witnesses were deliberately placed below the chosen rank. Reflection is therefore a controlled finite closure construction, not a global truth predicate for the universe.

Proof. Let Φ^* be the finite set of subformulas of the given finite family Φ. Choose α_0 with $A \subseteq V_{\alpha_0}$, and apply the preceding closure construction to obtain $\alpha = \sup_n \alpha_n$ satisfying the witness-closure property for every existential formula in Φ^*.

We prove by induction on formula complexity that for every $\psi \in \Phi^*$ and every parameter tuple $\bar{a} \in V_\alpha$,

$$V \models \psi(\bar{a}) \iff V_\alpha \models \psi(\bar{a}).$$

Atomic formulas are absolute because V_α is transitive. Boolean connectives follow from the induction hypothesis.

For $\psi(\bar{y}) = \exists x\, \theta(x, \bar{y})$, suppose first that $V_\alpha \models \psi(\bar{a})$. Then some $b \in V_\alpha$ satisfies $V_\alpha \models \theta(b, \bar{a})$; by induction $V \models \theta(b, \bar{a})$, hence $V \models \psi(\bar{a})$. Conversely, if $V \models \psi(\bar{a})$, witness-closure supplies $b \in V_\alpha$ with $V \models \theta(b, \bar{a})$. The induction hypothesis gives $V_\alpha \models \theta(b, \bar{a})$, so $V_\alpha \models \psi(\bar{a})$.

Thus every formula in Φ is absolute between V and V_α on parameters from V_α. The prescribed set A is present because $A \subseteq V_{\alpha_0} \subseteq V_\alpha$. □

Proposition 33.3 (Reflection with parameters)**.** *In the Reflection Theorem, the reflecting ordinal α may be chosen above the ranks of any prescribed finite set of parameters, so the displayed equivalences hold uniformly for those parameters.*

Proof. The parameter version is obtained by changing only the starting rank in the closure construction. Let P be the prescribed finite set of parameters. Choose an ordinal α_0 satisfying

$$\forall p \in P \quad \operatorname{rank}(p) < \alpha_0.$$

Equivalently, $P \subseteq V_{\alpha_0}$. Now run the finite-fragment witness-closure recursion from α_0, obtaining $\alpha \geq \alpha_0$.

The proof of reflection itself is unchanged: induction on the finite subformula closure gives

$$V \models \varphi(\bar{a}) \iff V_\alpha \models \varphi(\bar{a})$$

for every formula under consideration and every $\bar{a} \in V_\alpha$. In particular all tuples formed from the prescribed parameters are allowed because those parameters lie in V_α.

The point is not a second reflection theorem but control of the reflecting level. Given any preliminary rank bound β, we may require

$$\alpha_0 > \max\bigl(\beta, \sup_{p\in P}(\operatorname{rank}(p)+1)\bigr).$$

The resulting reflecting α then satisfies $\alpha > \beta$ and contains all parameters. This strengthened form is what later constructions use when they need reflection above already chosen objects or ranks. □

Corollary 33.4 (Set-sized approximations to finite fragments of ZF)**.** *For every finite fragment $T_0 \subseteq$ ZF, there are arbitrarily large ordinals α such that $V_\alpha \models T_0$.*

Proof. Let $T_0 \subseteq$ ZF be finite, and let β be any ordinal. Let Φ consist of the finitely many sentences in T_0, together with all subformulas needed by the Reflection Theorem. Apply reflection with the prescribed parameter β (or simply demand the starting rank be above β). We obtain $\alpha > \beta$ such that for every $\sigma \in T_0$,

$$V \models \sigma \iff V_\alpha \models \sigma.$$

Because each σ is an axiom of ZF and we are reasoning in ZF, $V \models \sigma$. Therefore

$$V_\alpha \models \sigma \qquad \text{for every } \sigma \in T_0,$$

so $V_\alpha \models T_0$.

Since the initial bound β was arbitrary, such reflecting ordinals occur above every ordinal. Thus there are arbitrarily large set-sized structures $(V_\alpha, \in)$ satisfying any fixed finite fragment of ZF. This does *not* assert that some V_α satisfies all of ZF simultaneously; that stronger conclusion would not follow from finite reflection and would conflict with the metamathematical distinctions developed later. □

Worked example

Worked Example 33.5 (Reflecting one existential formula)**.** Suppose the finite formula list contains $\exists y\, \psi(x, y)$. Whenever x occurs in the current rank and the universe has a witness y, choose some later rank containing such a witness. Repeating this closure for the finitely many subformulas and taking the supremum of an ω-sequence ensures the existential step of the truth induction.

A useful warning

Reflection is always for a specified finite set of formulas (or a scheme proved one formula at a time); treating it as a global truth predicate would be illegitimate.

Looking ahead

The next chapter, *Elementary hulls*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 34

Elementary hulls

Definitions and setup

Formal definitions used in this chapter

Skolem function. For a formula $\exists y\, \varphi(y, \bar{x})$ in a structure M, a Skolem function is a function $F_\varphi : M^{|\bar{x}|} \to M$ satisfying

$$M \models \exists y\, \varphi(y, \bar{a}) \Longrightarrow M \models \varphi(F_\varphi(\bar{a}), \bar{a}).$$

Skolem hull. Given $A \subseteq M$, define

$$H_0 = A, \qquad H_{n+1} = H_n \cup \{F_i(\bar{a}) : i < \omega,\ \bar{a} \in H_n^{<\omega}\},$$

and

$$\mathrm{Hull}^M(A) := \bigcup_{n<\omega} H_n.$$

This is the least subset of M containing A and closed under the chosen language functions and Skolem functions.

A Skolem function chooses a witness for an existential formula whenever one exists. Closing a parameter set under all Skolem functions yields a Skolem hull. Tarski–Vaught turns the witness property into elementarity. When the ambient relation is well founded and extensional, Mostowski collapse produces a transitive copy. External countability of the hull is a metatheoretic fact and need not be recognized internally.

How this chapter fits together

Skolem functions turn existential truth into explicit closure operations. Closing a small parameter set under those functions produces a small elementary substructure by Tarski–Vaught. If the relation is well founded and extensional, Mostowski collapse then gives a transitive copy. This two-step hull-and-collapse pattern reappears in Condensation.

Proposition 34.1 (Skolem-function expansion)**.** *Assume the ambient metatheory has Choice. Let M be an $\mathcal{L}$-structure. There is an expansion M^{Sk} obtained by adding, for each formula $\exists x\, \varphi(x, \bar{y})$, a function symbol f_φ such that for every tuple $\bar{a} \in M$,*

$$M \models \exists x\, \varphi(x, \bar{a}) \implies M^{\mathrm{Sk}} \models \varphi(f_\varphi(\bar{a}), \bar{a}).$$

If the existential statement is false, the value of $f_\varphi(\bar{a})$ may be chosen arbitrarily.

Proof. Assume Choice in the ambient metatheory. For each existential $\mathcal{L}$-formula $\exists x\,\varphi(x,\bar{y})$, add a new function symbol f_φ of arity $|\bar{y}|$. We must interpret it on every tuple $\bar{a} \in M^{|\bar{y}|}$.

For a fixed φ and tuple $\bar{a}$, let

$$W_{\varphi,\bar{a}} := \{b \in M : M \models \varphi(b,\bar{a})\}.$$

If this set is nonempty, the desired Skolem property requires choosing one element of it. The family of all nonempty $W_{\varphi,\bar{a}}$ is a set: there are only set-many formulas and set-many finite tuples from M. Ambient AC therefore supplies a simultaneous choice function. Define $f_\varphi^M(\bar{a})$ to be the chosen witness when $W_{\varphi,\bar{a}} \neq \varnothing$, and choose any fixed default element of M otherwise (the empty-structure edge case is handled separately by the usual convention).

Then by construction,

$$M \models \exists x\,\varphi(x,\bar{a}) \Longrightarrow M^{\mathrm{Sk}} \models \varphi(f_\varphi(\bar{a}),\bar{a}).$$

The reduct of M^{Sk} to the original language is exactly M. Thus the expansion changes no old-language truth; it merely names witnesses uniformly. If a definable well-order of M is already available, one can choose least witnesses and avoid using ambient AC; the theorem states the general metatheoretic version explicitly. □

Theorem 34.2 (Skolem hull construction)**.** *For a structure M with Skolem functions and $A \subseteq M$, the closure* $\mathrm{Hull}^M(A)$ *under the basic and Skolem functions is an elementary substructure of M, with cardinality at most* $\max(|A|, |\mathcal{L}|, \aleph_0)$.

Proof. Let M be expanded by Skolem functions and let $A \subseteq M$. Define an increasing sequence

$$H_0 = A, \qquad H_{n+1} = H_n \cup \{f^M(\bar{a}) : f \text{ is a basic or Skolem function, } \bar{a} \in H_n^{<\omega}\},$$

and set

$$H = \bigcup_{n<\omega} H_n.$$

Then H is closed under every function symbol of the expanded language, so it is a substructure.

To prove elementarity, use the Tarski–Vaught criterion. Suppose

$$M \models \exists x\,\varphi(x,\bar{a}) \qquad (\bar{a} \in H).$$

All entries of $\bar{a}$ lie in some common H_n. The corresponding Skolem function is one of the closure operations, hence

$$b = f_\varphi^M(\bar{a}) \in H_{n+1} \subseteq H,$$

and by the Skolem property $M \models \varphi(b,\bar{a})$. Thus every existential formula true in M with parameters from H has a witness in H, so $H \preccurlyeq M$.

For the cardinal bound, let

$$\mu = \max(|A|, |\mathcal{L}|, \aleph_0).$$

The Skolemized language has at most $\max(|\mathcal{L}|, \aleph_0)$ function symbols. If $|H_n| \leq \mu$, then there are at most $\mu^{<\omega} = \mu$ finite tuples and at most μ function applications, so $|H_{n+1}| \leq \mu$. By induction and a countable union, $|H| \leq \mu$. This H is the Skolem hull $\mathrm{Hull}^M(A)$. □

Theorem 34.3 (Countable elementary submodel lemma)**.** *If M is a structure in a countable language and $A \subseteq M$ is countable, then there is a countable elementary substructure $N \preccurlyeq M$ containing A.*

Proof. Let M be a structure in a countable language $\mathcal{L}$, and let $A \subseteq M$ be countable. By the Skolem-function expansion theorem, expand M to a Skolemized structure M^{Sk}. Because there are only countably many $\mathcal{L}$-formulas, the expanded language is still countable.

Form the Skolem hull

$$H = \mathrm{Hull}^{M^{\mathrm{Sk}}}(A).$$

The cardinal estimate from the preceding theorem gives

$$|H| \le \max(|A|, |\mathcal{L}|, \aleph_0) = \aleph_0.$$

Thus H is countable. Tarski–Vaught gives

$$H \preccurlyeq M^{\mathrm{Sk}}.$$

Now take reducts to the original language. Elementarity is preserved under reduct: for every original-language formula φ and tuple $\bar{a} \in H$,

$$H \models \varphi(\bar{a}) \iff M^{\mathrm{Sk}} \models \varphi(\bar{a}) \iff M \models \varphi(\bar{a}).$$

Therefore the original-language reduct $N = H \restriction \mathcal{L}$ is a countable elementary substructure of M containing A.

The theorem is external: "countable" refers to cardinality in the surrounding universe. This distinction becomes essential in the later discussion of Skolem's paradox. □

Theorem 34.4 (Collapse of an elementary hull to a transitive model)**.** *If $X \prec (M, \in)$ and $\in$ is well founded and extensional on X, then the Mostowski collapse $\pi : X \cong \bar{X}$ maps X isomorphically onto a transitive set $\bar{X}$, and $\bar{X}$ satisfies exactly the first-order sentences satisfied by X.*

Proof. Assume $X \prec (M, \in)$ and that $\in$ restricted to X is well founded and extensional. By the Mostowski Collapse Theorem there is a unique transitive set $\bar{X}$ and a unique isomorphism

$$\pi : (X, \in) \xrightarrow{\cong} (\bar{X}, \in)$$

satisfying the recursion

$$\pi(x) = \{\pi(y) : y \in X \ \wedge \ y \in x\}.$$

The collapse theorem supplies transitivity of $\bar{X}$ and preservation of the membership relation:

$$y \in x \text{ in } X \quad \iff \quad \pi(y) \in \pi(x).$$

Every isomorphism preserves first-order satisfaction. Thus, by induction on formulas, for every formula φ and tuple $\bar{a} \in X$,

$$X \models \varphi(\bar{a}) \iff \bar{X} \models \varphi(\pi(\bar{a})).$$

In particular X and $\bar{X}$ satisfy exactly the same sentences. Since $X \prec M$, we also have for sentences σ

$$M \models \sigma \iff X \models \sigma \iff \bar{X} \models \sigma.$$

The last equivalence is restricted to the theory seen by the elementary substructure; the collapse does not assert that $\bar{X}$ contains all objects of M. Its role is to replace a possibly nontransitive elementary hull by a transitive isomorphic copy in which membership has its usual set-theoretic meaning. □

Worked example

Worked Example 34.5 (Building a hull in stages)**.** Start with a countable set A. Let H_1 contain all values of the countably many Skolem functions on finite tuples from A; then repeat with H_1, and set $H = \bigcup_n H_n$. Each stage is countable, so H is countable. Every existential formula true with parameters in H has its Skolem witness in H, hence $H \preccurlyeq M$.

A useful warning

An elementary submodel need not be transitive. The collapse is a separate theorem, and it can move parameters unless they are known to be fixed.

Looking ahead

The next chapter, *Inner models and the Skolem paradox*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 35

Inner models and the Skolem paradox

Definitions and setup

Formal definitions used in this chapter

Inner model. A class M is an inner model of ZF iff

$$M \text{ is transitive}, \qquad \mathrm{Ord} \subseteq M, \qquad (M, \in) \models \mathrm{ZF}.$$

Internal cardinality. For $X, Y \in M$, write

$$M \models |X| \leq |Y|$$

iff there exists $f \in M$ such that $M \models$ "$f : X \to Y$ is injective". Equality of internal cardinalities is defined using a bijection belonging to M. The ambient universe may contain functions not belonging to M, so internal and external cardinal comparisons can differ.

An inner model is a transitive proper class containing all ordinals and satisfying the relevant set-theoretic axioms internally. Internal power sets and cardinalities are computed using only subsets/functions belonging to that class. The Skolem paradox is the paradigm for this internal/external distinction. Relative consistency arguments later use inner models, but the existence of an inner model inside each model must be proved rather than inferred from consistency alone.

How this chapter fits together

An inner model is transitive and contains all ordinals, but those two facts alone do not prove every axiom of ZF inside it. The hard closure axioms must be checked. The Skolem paradox then illustrates why internal and external cardinality can disagree: functions witnessing countability externally may simply be absent internally.

Theorem 35.1 (Inner-model criterion)**.** *Let M be a transitive class containing all ordinals. To prove that M is an inner model of a theory T, it is enough to verify internally in $(M, \in)$ every axiom of T; transitivity alone automatically handles only the absolute parts and does not by itself give Separation, Replacement, Power Set, or Choice.*

Proof. Recall the definition: an inner model is a transitive class M containing every ordinal and satisfying the specified theory in the structure $(M, \in)$. Therefore, once transitivity and inclusion of all ordinals have

been established, the remaining task is exactly to prove

$$(M, \in) \models \sigma \qquad \text{for every axiom } \sigma \in T.$$

If this is done, M is an inner model of T by definition.

What requires emphasis is what transitivity does *not* prove. Extensionality and many bounded statements are inherited because membership between elements of M is the ambient membership relation. But Separation asks that for $a, \bar{p} \in M$, the subset

$$\{x \in a : M \models \varphi(x, \bar{p})\}$$

belong to M. Replacement asks for the image of a set under an internally definable functional relation to lie in M. Power Set asks for the set

$$\mathcal{P}^M(a) = \{x \in M : x \subseteq a\}$$

to be an element of M, not merely to be a class externally. Choice asks for choice functions that themselves belong to M. None of these closure properties follows from transitivity alone.

Thus the criterion is deliberately axiom-by-axiom. Later, when $M = L$ or a forcing/symmetric extension is proposed, the proof must supply separate owners for precisely these nonautomatic axioms. □

Proposition 35.2 (Internal versus external finiteness/countability/cardinality)**.** *Let M be transitive and suppose $\omega^M = \omega$. For $x, \kappa \in M$, the internal assertions*

$$M \models \text{“}x \text{ is at most countable}'' \quad \text{and} \quad M \models \text{“}|x| = \kappa''$$

mean respectively that M contains an injection $x \to \omega$ and that M contains a bijection $x \to \kappa$. Therefore these assertions may differ from the ambient ones precisely because the required witnessing functions may exist outside M but not inside M.

Proof. Assume M is transitive and $\omega^M = \omega$. Satisfaction of cardinality assertions is governed by which functions belong to M. For instance,

$$M \models \text{“}x \text{ is at most countable}''$$

means

$$\exists f \in M \, \big(M \models \text{“}f : x \hookrightarrow \omega \text{ is injective}''\big).$$

Because M is transitive, if such an $f \in M$ exists then it is genuinely an injection in the ambient universe. Hence internal countability implies external countability.

The converse can fail. An external injection $g : x \to \omega$ need not be an element of M, so it need not witness the existential quantifier evaluated inside M. Likewise

$$M \models \text{“}|x| = \kappa''$$

means that some bijection $f : x \to \kappa$ belongs to M. An outer model can contain a new bijection and thereby change the cardinality it assigns to the same old set.

Finiteness is better behaved for a transitive model $M \models \mathrm{ZF}$ with $\omega^M = \omega$. If M contains a bijection between x and some $n < \omega$, then that same finite graph is an external bijection, so x is externally finite. Conversely, if externally $x = \{a_0, \dots, a_{n-1}\}$ with $x \in M$, transitivity gives $a_i \in M$ for every $i < n$; repeated Pairing and Union construct the finite tuple $\langle a_0, \dots, a_{n-1} \rangle \in M$, and Separation constructs the graph $\{\langle i, a_i \rangle : i < n\} \in M$. Hence M also sees x as finite. The conceptual lesson is that infinite cardinality is an existential statement about functions, so its absoluteness depends on whether the relevant functions are available internally. □

Worked Theorem 35.3 (Full resolution of Skolem's paradox)**.** *Let M be a set-sized model of ZF such that $M \models$ “$\mathbb{R}$ is uncountable”. If $N \preccurlyeq M$ is externally countable, then $N \models$ “$\mathbb{R}$ is uncountable” although the set $\mathbb{R}^N$ is externally countable. There is no contradiction: an external enumeration of $\mathbb{R}^N$ need not belong to N.*

Proof. Let M be a set-sized model of ZF in a countable language and suppose

$$M \models \text{“}\mathbb{R} \text{ is uncountable.”}$$

By Downward Löwenheim–Skolem there is, externally, a countable elementary submodel

$$N \preccurlyeq M.$$

Elementarity transfers the sentence about the reals:

$$N \models \text{“}\mathbb{R} \text{ is uncountable.”}$$

Nevertheless N itself is a countable set in the ambient universe. Therefore every subset of N, in particular the externally viewed collection

$$\mathbb{R}^N := \{x \in N : N \models \text{“}x \text{ is a real”}\},$$

is externally countable. Let

$$e : \omega \twoheadrightarrow \mathbb{R}^N$$

be an external enumeration.

Why does this not contradict the sentence true in N? Because

$$N \models \text{“}\mathbb{R} \text{ is uncountable”}$$

means that N contains no function which N regards as a surjection from its ω onto its reals. The external enumeration e need not belong to N. Indeed, if $e \in N$, absoluteness of ω and the elementary interpretation of $\mathbb{R}^N$ would make e an internal enumeration, contradicting the displayed sentence.

Thus “countable” is model-relative because the quantifier over possible enumerating functions is model-relative. The paradox disappears once the location of the alleged enumeration is stated explicitly. □

Proposition 35.4 (Model-theoretic formulation of relative consistency and its limitations)**.** *If a first-order theory T is syntactically consistent, completeness yields a (possibly ill-founded and nontransitive) model of T. This model-existence statement must not be confused with the stronger assertion that T has a transitive model.*

Proof. Let T be a first-order theory. Syntactic consistency means

$$T \nvdash \bot.$$

By the Completeness Theorem,

$$T \nvdash \bot \quad \Longleftrightarrow \quad \exists M \, (M \models T).$$

Thus an external proof of $\mathrm{Con}(T)$ can be converted into an external existence statement for some first-order model of T.

Nothing in completeness says that the model is transitive, well founded, or a submodel of the ambient set-theoretic universe. If T is a set theory, the membership relation E^M of a model supplied by completeness can be externally ill founded, and its “natural numbers” or “ordinals” can be nonstandard. The assertion

$$\exists M \, (M \text{ is transitive and } M \models T)$$

is therefore strictly stronger than the bare model-existence conclusion of first-order completeness.

This distinction is essential for forcing. The pedagogical construction of a generic extension over a countable transitive model is extremely useful for understanding names and the Truth Lemma, but from $\mathrm{Con}(T)$ alone one may not infer the existence of such a transitive model. The later Boolean-valued/finite-fragment consistency-transfer argument is included precisely to bridge that metatheoretic gap without making the invalid inference. □

Worked example

Worked Example 35.5 (Skolem's paradox in one diagram)**.** Let M be externally countable and suppose $M \models$ "$\mathbb{R}$ is uncountable". Externally enumerate the elements of $\mathbb{R}^M$. That enumeration is not required to belong to M. Internally, uncountability asserts that no bijection $\omega^M \to \mathbb{R}^M$ exists among the functions that M contains, so there is no contradiction.

A useful warning

Whenever a model says "there exists a function," the quantified function must itself be an element of the model. This is the key to all internal/external cardinality distinctions.

End-of-volume perspective

Satisfaction, absoluteness, reflection, hulls, and collapse together form the model-theoretic toolkit used twice ahead: first to analyze L, and then to compare a forcing ground model with its extensions.

Part VI

Volume VI: The Constructible Universe

Preface

The constructible universe is built as an inner model. The difficult points are Replacement/Collection, Condensation, and the cardinal localization needed for GCH.

Position in the series

Volume VI gives the positive consistency branch: the constructible universe L satisfies ZF, Choice, and GCH. Condensation is the main structural theorem behind the cardinal calculation.

Sources and further reading

The constructible universe and the consistency of Choice and GCH originate in Gödel's monograph [8]. The treatment here also follows the modern condensation-and-cardinality organization of [13, 14].

Notation for this volume

$L_0 = \varnothing$, $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, and $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$. $<_L$ denotes the canonical definable well-order.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 36

Definability and Def(M)

Definitions and setup

Formal definitions used in this chapter

Definable subset. $X \subseteq M$ is definable over $(M, \in)$ from parameters if $X = \{x \in M : M \models \varphi(x, \bar{a})\}$.
$\mathrm{Def}(M)$**.** The set of all subsets of M definable over M from finitely many parameters in M.

For a set-sized structure M, $\mathrm{Def}(M)$ is the set of subsets of M definable over M using finitely many parameters from M. A definition is coded by a formula code and a finite parameter tuple. The satisfaction relation from Volume V turns this description into a genuine set-theoretic construction. Uniform coding is needed so the entire L-hierarchy can later be defined by one formula.

How this chapter fits together

Constructibility begins with a set-sized operation: $\mathrm{Def}(M)$, the subsets of M definable over M using parameters from M. Satisfaction for set-sized structures ensures this collection is itself a set. Uniform coding of formulas and parameters is important because the same definition will be iterated through all ordinals.

Theorem 36.1 (Coding definable subsets of a set-sized structure)**.** *For a set-sized structure M, every definable subset of M has the form*

$$X_{\varphi,\bar{a}} = \{x \in M : M \models \varphi(x, \bar{a})\}$$

for a formula φ and finite parameter tuple $\bar{a} \in M^{<\omega}$; the pair $(\ulcorner\varphi\urcorner, \bar{a})$ is a set-sized code for $X_{\varphi,\bar{a}}$.

Proof.

Formal derivation spine

Formal spine for Coding definable subsets of a set-sized structure.

$$X = X_{e,\bar{a}} := \{x \in M : \mathrm{Sat}(M, e, x, \bar{a})\}.$$

$$(e, \bar{a}) \longmapsto X_{e,\bar{a}}$$

$$\omega \times M^{<\omega}.$$

Fix a set-sized structure M. A subset $X \subseteq M$ is definable with parameters precisely when there are a formula code e and a finite tuple $\bar{a} \in M^{<\omega}$ such that

$$X = X_{e,\bar{a}} := \{x \in M : \mathrm{Sat}(M, e, x, \bar{a})\}.$$

The map

$$(e, \bar{a}) \longmapsto X_{e,\bar{a}}$$

is definable because the satisfaction relation for the fixed set-sized structure M is definable. This provides an explicit coding of every definable subset by finite syntactic data plus finitely many parameters. The present proposition is only the coding statement; the next theorem uses the fact that the code space is itself a set to prove that $\mathrm{Def}(M)$ is a set. *Point specific to this result.* A definable subset is determined by a formula code together with finitely many parameters. Since both the syntax codes and the finite parameter tuples form sets, Replacement turns this coding into a set-sized family of subsets. □

Proposition 36.2 (Def(M) is a set)**.** *For every set M,*

$$\mathrm{Def}(M) := \{X \subseteq M : \exists \varphi\, \exists \bar{a} \in M^{<\omega}\, X = \{x \in M : (M, \in) \models \varphi(x, \bar{a})\}\}$$

is a set.

Proof.

Formal derivation spine

Formal spine for Def(M) is a set.

$$C_M := \omega \times M^{<\omega},$$

$$(e, \bar{a}) \mapsto X_{e,\bar{a}}$$

$$\mathrm{Def}(M) = \{X \in Y : X \subseteq M \text{ and } X = X_{e,\bar{a}} \text{ for some } (e, \bar{a}) \in C_M\}.$$

The set of possible codes is

$$C_M := \omega \times M^{<\omega},$$

which is a set. By the preceding coding proposition, every member of $\mathrm{Def}(M)$ is equal to $X_{e,\bar{a}}$ for some $(e, \bar{a}) \in C_M$. Replacement applied to the definable map

$$(e, \bar{a}) \mapsto X_{e,\bar{a}}$$

produces a set Y containing all these values. Separation then defines

$$\mathrm{Def}(M) = \{X \in Y : X \subseteq M \text{ and } X = X_{e,\bar{a}} \text{ for some } (e, \bar{a}) \in C_M\}.$$

Hence $\mathrm{Def}(M)$ is a set. The argument never invokes a power set of a proper class or an unrestricted set of formulas. *Point specific to this result.* The previous coding gives a surjection from a set of formula/parameter codes onto $\mathrm{Def}(M)$. Replacement followed by Separation therefore establishes sethood without appealing to an unrestricted collection of definitions. □

Lemma 36.3 (Uniform coding of definable subsets)**.** *There is a uniform coding relation* $\mathrm{Defines}(M, e, \bar{a}, X)$ *such that e is a formula code, $\bar{a} \in M^{<\omega}$, and $X = \{x \in M : \mathrm{Sat}(M, e, (x, \bar{a}))\}$. Thus the definition of* $\mathrm{Def}(M)$ *is uniform in M.*

Proof.

Formal derivation spine

Formal spine for Uniform coding of definable subsets.

$$\mathrm{Defines}(M, e, \bar{a}, X)$$

$$X \subseteq M \ \wedge\ \forall x \in M \left(x \in X \leftrightarrow \mathrm{Sat}(M, e, x, \bar{a})\right).$$

$$X \in \mathrm{Def}(M) \iff \exists e \in \omega\, \exists \bar{a} \in M^{<\omega}\ \mathrm{Defines}(M, e, \bar{a}, X).$$

Let e range over Gödel codes of formulas and let $\bar{a} \in M^{<\omega}$. Define

$$\mathrm{Defines}(M, e, \bar{a}, X)$$

to mean

$$X \subseteq M \ \wedge \ \forall x \in M \, \big(x \in X \leftrightarrow \mathrm{Sat}(M, e, x, \bar{a})\big).$$

All quantifiers in this definition range over sets, and Sat is the uniform satisfaction predicate from Volume V. Consequently the same first-order formula Defines works for every set M; only the parameters change. In particular,

$$X \in \mathrm{Def}(M) \iff \exists e \in \omega \, \exists \bar{a} \in M^{<\omega} \ \mathrm{Defines}(M, e, \bar{a}, X).$$

This uniformity, rather than mere sethood, is what later makes the hierarchy $\alpha \mapsto L_\alpha$ first-order definable. *Point specific to this result.* Uniformity means one satisfaction formula handles every formula code and every finite parameter tuple. This is what later allows the operation $M \mapsto \mathrm{Def}(M)$ to be iterated definably through the ordinals. □

Proposition 36.4 (Absoluteness properties of Def in the transitive setting)**.** *Let M be a transitive set. Let N_0, N_1 be transitive models of a fixed finite fragment of ZF sufficient to code finite syntax, define satisfaction for the set-sized structure $(M, \in)$, and form $\mathrm{Def}(M)$. If $M \in N_0 \cap N_1$ and both models contain the standard finite formula codes, then*

$$\mathrm{Sat}^{N_0}(M, e, \bar{a}) \iff \mathrm{Sat}^{N_1}(M, e, \bar{a})$$

for every formula code e and parameter tuple $\bar{a} \in M^{<\omega}$. Consequently

$$\mathrm{Def}(M)^{N_0} = \mathrm{Def}(M)^{N_1}.$$

Proof.

Formal derivation spine

Formal spine: satisfaction for one fixed set structure is absolute. For every formula φ and assignment $s \in M^{<\omega}$,

$$N_0 \models \mathrm{Sat}(M, \ulcorner \varphi \urcorner, s) \iff N_1 \models \mathrm{Sat}(M, \ulcorner \varphi \urcorner, s).$$

Induct on the complexity of φ. Atomic equality and membership are absolute because M, N_0, N_1 are transitive and all three use the ambient membership relation on the same set M. The Boolean clauses follow directly from the induction hypothesis. For the existential step,

$$(M, \in) \models \exists x \, \psi(x, \bar{a})$$

means that there is a witness $b \in M$ satisfying $\psi(b, \bar{a})$. Both ambient models quantify over exactly the same set M, and the induction hypothesis gives the same truth value for each candidate b. Hence they agree on the existential formula.

Thus N_0 and N_1 compute the same satisfaction relation for $(M, \in)$. For each standard formula code e and parameter tuple $\bar{a}$, they therefore define the same actual subset

$$X_{e,\bar{a}} := \{x \in M : \mathrm{Sat}(M, e, x, \bar{a})\}.$$

The set of codes $\omega \times M^{<\omega}$ is also the same in the two transitive models. It follows that both collect exactly the same family $\{X_{e,\bar{a}}\}$, namely the same $\mathrm{Def}(M)$. The theorem concerns a fixed set-sized structure; it makes no claim that a truth predicate for an ambient universe is absolute or definable. □

Worked example

Worked Example 36.5 (A definable subset of a finite structure)**.** If $M = \{0, 1, 2\}$ with the inherited order, the subset $\{0\}$ is definable as the set of elements with no predecessor, while $\{0, 1\}$ is definable as the elements below the parameter 2. $\mathrm{Def}(M)$ collects subsets obtainable in this way; in the constructible hierarchy the same idea is applied at every stage.

A useful warning

Definability is always relative to a structure and allowed parameters. “Definable in the universe” and “definable over L_α” are different notions.

Looking ahead

The next chapter, *The L-hierarchy*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 37

The L-hierarchy

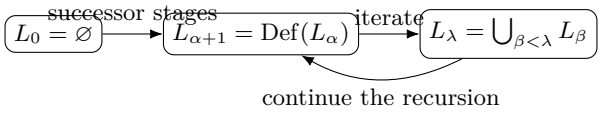

Figure 37.1: The constructible hierarchy grows by definability at successor stages and by union at limit stages.

Definitions and setup

Formal definitions used in this chapter

Constructible hierarchy. $L_0 = \varnothing$, $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, and $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$.
Constructible universe. $L = \bigcup_{\alpha\in\mathrm{Ord}} L_\alpha$.

The constructible hierarchy replaces the power-set successor step of V by definability: $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, with unions at limits. L is the union of all levels. Every constructible set therefore has a first stage of appearance and a finite definition from earlier parameters. This finiteness is the source of the canonical well-order and of the cardinal bounds used in GCH.

How this chapter fits together

The hierarchy is defined by $L_0 = \varnothing$, $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, and unions at limits. Transfinite induction simultaneously proves transitivity and the presence of ordinals. The construction is deliberately parsimonious: only subsets definable from earlier-stage parameters are admitted at successor stages.

Theorem 37.1 (Recursive construction of L_α)**.** *There is a unique hierarchy $\langle L_\alpha : \alpha \in \mathrm{Ord}\rangle$ satisfying*

$$L_0 = \varnothing, \qquad L_{\alpha+1} = \mathrm{Def}(L_\alpha), \qquad L_\lambda = \bigcup_{\beta<\lambda} L_\beta \quad (\lambda \textit{ limit}).$$

How to read the symbols: the constructible hierarchy is a recursive data pipeline

At stage α we have the database L_α. The successor operation $\mathrm{Def}(L_\alpha)$ keeps only subsets definable over that database using finitely many parameters already present. At a limit stage we take the union of all

previous databases. Thus L grows by definability rather than by taking every subset, which is why it can be much smaller than the ambient universe.

Proof.

Formal derivation spine

Formal spine for Recursive construction of L_α.

$$L_0 = \varnothing, \qquad L_{\alpha+1} = F(L_\alpha), \qquad L_\lambda = \bigcup_{\beta<\lambda} L_\beta \quad (\lambda \text{ limit}).$$

$$L_{\alpha+1} = \mathrm{Def}(L_\alpha),$$
$$L_\lambda = \bigcup_{\beta<\lambda} L_\beta.$$

Define the successor-stage operation on sets by $F(M) = \mathrm{Def}(M)$. We seek a class function $\alpha \mapsto L_\alpha$ satisfying

$$L_0 = \varnothing, \qquad L_{\alpha+1} = F(L_\alpha), \qquad L_\lambda = \bigcup_{\beta<\lambda} L_\beta \quad (\lambda \text{ limit}).$$

The operation F is definable because satisfaction for the set-sized structure $(M, \in)$ was constructed earlier, and $\mathrm{Def}(M)$ is a set. The transfinite recursion theorem therefore gives a unique function on every ordinal γ whose values satisfy these clauses below γ. Uniqueness on overlapping initial segments makes these set-sized recursions coherent, so they determine the class hierarchy $\langle L_\alpha : \alpha \in \mathrm{Ord}\rangle$. Nothing about transitivity is needed for existence of the recursion; that is the content of the next proposition.

Stage discipline. At a successor stage we add only subsets definable over the immediately preceding level with parameters from that level. At a limit stage we add no new definable subsets at all; we simply take the union of the earlier levels. Keeping these two clauses separate is essential in later induction arguments. □

Proposition 37.2 (Every L_α is transitive)**.** *For every ordinal α, the set L_α is transitive:*

$$x \in y \in L_\alpha \quad \Longrightarrow \quad x \in L_\alpha.$$

Proof.

Formal derivation spine

Formal spine: transitivity of each constructible level. Induct on α. At a successor stage,

$$y \in L_{\alpha+1} = \mathrm{Def}(L_\alpha) \Longrightarrow y \subseteq L_\alpha,$$

so $x \in y \Rightarrow x \in L_\alpha \subseteq L_{\alpha+1}$. At a limit,

$$L_\lambda = \bigcup_{\beta<\lambda} L_\beta$$

is a union of an increasing chain of transitive sets.

We prove by transfinite induction on α that L_α is transitive. The claim is trivial for $L_0 = \varnothing$. Suppose L_α is transitive and let $x \in L_{\alpha+1} = \mathrm{Def}(L_\alpha)$. By definition, x is a subset of L_α. Hence every $y \in x$ belongs to L_α, and therefore to $L_{\alpha+1}$ because the hierarchy is increasing. Thus $L_{\alpha+1}$ is transitive. If λ is a limit ordinal and $x \in L_\lambda = \bigcup_{\beta<\lambda} L_\beta$, choose $\beta < \lambda$ with $x \in L_\beta$. By the induction hypothesis $x \subseteq L_\beta \subseteq L_\lambda$, so L_λ is transitive.

It remains only to justify the monotonicity used at the successor step. By induction, every $a \in L_\alpha$ is a subset of L_α; moreover the set a is definable over L_α from the parameter a by the formula $z \in a$. Hence $a \in \mathrm{Def}(L_\alpha) = L_{\alpha+1}$. Therefore $L_\alpha \subseteq L_{\alpha+1}$, completing the induction.

What is special here. The successor argument uses both ingredients in the definition of $\mathrm{Def}(L_\alpha)$: its members are subsets of L_α, and parameters from L_α ensure that the hierarchy itself is increasing. □

Proposition 37.3 (alpha is a subset of L_α)**.** *For every ordinal α,*

$$\alpha \subseteq L_\alpha.$$

In particular, each $\beta < \alpha$ belongs to L_α.

Proof.

Formal derivation spine

Formal spine: ordinals occur by their own stage. The induction target is

$$\forall \alpha\ \alpha \subseteq L_\alpha.$$

For $\beta < \alpha$, the induction hypothesis makes β a definable subset of the earlier level containing all smaller ordinals, hence

$$\beta \in L_{\beta+1} \subseteq L_\alpha.$$

Thus every member of α lies in L_α.

We prove simultaneously by transfinite induction that $L_\alpha \cap \mathrm{Ord} = \alpha$, which implies $\alpha \subseteq L_\alpha$. The claim is trivial at 0. At a limit λ, $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$, so the ordinal members are exactly $\bigcup_{\beta<\lambda} \beta = \lambda$. At a successor $\beta + 1$, every ordinal below β already lies in L_β. The set β itself is the subset of L_β consisting of its ordinal elements, a subset definable over L_β by the first-order formula "x is an ordinal". Hence $\beta \in \mathrm{Def}(L_\beta) = L_{\beta+1}$. No ordinal $\geq \beta + 1$ can belong to $L_{\beta+1}$, since every new element is a subset of L_β and transitivity would otherwise force too many ordinals below it into L_β. Thus the induction closes. *Reading the induction.* The simultaneous statement $L_\alpha \cap \mathrm{Ord} = \alpha$ contains both inclusions needed at the successor stage. The inclusion $\beta \subseteq L_\beta$ lets us talk about the ordinal elements already present, while the reverse inclusion says there is no unexpected ordinal in the level. Defining β as "the set of ordinal elements of L_β" therefore creates exactly the next ordinal and no larger one. This is why the proof is stronger than merely showing $\alpha \subseteq L_\alpha$: the equality with the ordinal trace controls the induction. □

Theorem 37.4 (L contains every ordinal and is transitive)**.** *The class*

$$L := \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha$$

is transitive and contains every ordinal.

Proof.

Formal derivation spine

Formal spine: pass from the levels to the class L. Since every ordinal α satisfies $\alpha \subseteq L_\alpha$,

$$\alpha \in L_{\alpha+1} \subseteq L.$$

If $y \in L$, choose α with $y \in L_\alpha$; transitivity of L_α gives

$$x \in y \Longrightarrow x \in L_\alpha \subseteq L.$$

Hence L contains all ordinals and is transitive.

Define $L_0 = \varnothing$, $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, and $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$. Transfinite recursion gives the hierarchy. By induction each L_α is transitive: a definable subset of L_α is a subset of L_α, while elements of L_α already lie at earlier stages. A simultaneous induction shows $\alpha \subseteq L_\alpha$, so every ordinal belongs to a later constructible level. Thus $L = \bigcup_\alpha L_\alpha$ is a transitive proper class containing all ordinals.

Point specific to this result. A simultaneous induction shows $\alpha \subseteq L_\alpha$ and hence $\alpha \in L_{\alpha+1}$. Taking the union over all stages therefore puts every ordinal into L, while stage transitivity makes L transitive.

Further explanation. The two assertions are proved simultaneously but play different roles. Transitivity guarantees that membership among constructible sets is inherited from the ambient universe. Containing every ordinal ensures that L has the same ordinal scale on which ranks, cardinals, and later condensation arguments are measured. At successor stages the new elements are subsets of the preceding transitive level; at limits the union of an increasing chain of transitive levels is transitive. The inclusion $\alpha \subseteq L_\alpha$ then puts every ordinal into L. □

Worked example

Worked Example 37.5 (The first levels)**.** $L_0 = \varnothing$. Since the only subset of the empty set is empty, $L_1 = \{\varnothing\}$. Subsequent finite stages build the hereditarily finite sets. By stage ω, all hereditarily finite sets have appeared; later stages begin adding infinite definable subsets.

A useful warning

Do not identify $L_{\alpha+1}$ with the full power set $\mathcal{P}(L_\alpha)$. It contains only the subsets definable over L_α from parameters in that level.

Looking ahead

The next chapter, *Uniform definability and absoluteness of L*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 38

Uniform definability and absoluteness of L

Definitions and setup

Formal definitions used in this chapter

Uniform level definition. A single set-theoretic formula $\mathrm{Lev}(x, \alpha)$ such that for every ordinal α,

$$V \models \mathrm{Lev}(x, \alpha) \iff x \in L_\alpha.$$

Absoluteness of the constructible levels. If $M, N \models \mathrm{ZF}$ are transitive and $\alpha \in M \cap N$ is an ordinal, then

$$(L_\alpha)^M = (L_\alpha)^N = L_\alpha.$$

The equality is external and concerns the common ambient sets constructed at stage α.

A statement is absolute between models when it has the same truth value in both. For transitive models, bounded quantifiers over an existing set range over the same elements, which is why Δ_0 formulas are absolute. Power Set and cardinality are typical nonabsolute notions because a smaller model may omit subsets or functions that exist externally. Every later absoluteness claim names its model hypotheses.

How this chapter fits together

To use L as an inner model, the hierarchy must be definable uniformly rather than by a separate meta-definition at each ordinal. Uniform satisfaction for set-sized levels permits a single formula to express membership in L_α. Transitive models then agree on the construction in the required range, yielding $L^L = L$.

Theorem 38.1 (Uniform definability of the L-hierarchy)**.** *There is a first-order formula* $\mathrm{Lev}(x, \alpha)$ *of set theory such that for every ordinal* α*,*

$$V \models \mathrm{Lev}(x, \alpha) \quad \iff \quad x \in L_\alpha.$$

Thus the hierarchy is uniformly definable from its ordinal index.

Proof.

Formal derivation spine

Formal spine for Uniform definability of the L-hierarchy.

$$F(\alpha, x) \quad \text{“}x = L_\alpha\text{”}.$$

$$F(0, \varnothing), \qquad F(\alpha+1, v) \leftrightarrow \exists u\,(F(\alpha, u) \wedge \mathrm{DefStage}(u, v)),$$

$$\mathrm{Lev}(x, \alpha) \iff x \in L_\alpha.$$

Proof map. The construction of L is recursive on the ordinal parameter, but the recursion can be coded uniformly because satisfaction for each set-sized structure is already uniformly definable.

Fix the formula coding for first-order set theory developed in Volume V. There is a formula $\mathrm{DefStage}(u, v)$ saying that v is exactly the family of subsets of u definable over $(u, \in)$ from parameters in u: the formula quantifies over a formula code and a finite parameter tuple, and invokes the set-sized satisfaction predicate. By the transfinite-recursion theorem, the relation

$$F(\alpha, x) \quad \text{“}x = L_\alpha\text{”}.$$

Here F is the graph of the stage-construction: for each ordinal input α, it should pick out exactly one set x. The recursive clauses that specify this graph are

$$F(0, \varnothing), \qquad F(\alpha+1, v) \leftrightarrow \exists u\,(F(\alpha, u) \wedge \mathrm{DefStage}(u, v)),$$

and the corresponding union clause at limit ordinals. The general theorem on definable transfinite recursion converts these clauses into one formula $F(\alpha, x)$ of set theory. Consequently the relation $x \in L_\alpha$ is uniformly first-order definable from α, rather than requiring a new definition at each stage. □

Theorem 38.2 (Absoluteness of the L-construction between suitable transitive models)**.** *Let M and N be transitive models of ZF with $M \subseteq N$, and let $\alpha \in M \cap \mathrm{Ord}$. Then both models compute the same constructible level at α:*

$$(L_\alpha)^M = (L_\alpha)^N = L_\alpha.$$

In particular, $L_\alpha \subseteq M$, and the internally computed $(L_\alpha)^M$ is exactly the ambient constructible level L_α.

Proof.

Formal derivation spine

Formal spine for Absoluteness of the L-construction between suitable transitive models.

$$\mathrm{Lev}(x, \alpha) \iff x \in L_\alpha.$$

Let $M \subseteq N$ be transitive models satisfying the fragment of ZF needed for the L-construction, and suppose that the ordinal α under discussion belongs to both. We prove by induction on $\beta \leq \alpha$ that the two models compute the same constructible level whenever all parameters required by the comparison are present.

At stage 0, both constructions give the empty set by definition, so the two hierarchies agree without using an induction hypothesis. At a successor stage, assume the two models agree on L_β. The structure $(L_\beta, \in)$ is then literally the same set-sized structure in both models. Satisfaction for a fixed set-sized structure is absolute between transitive models for the recursive definition established in Volume V; hence the same formula/parameter pairs define the same subsets of L_β. Thus both models obtain the same $\mathrm{Def}(L_\beta) = L_{\beta+1}$. At a limit stage both take the union of the same earlier levels. This proves the desired absoluteness by transfinite induction. The restriction to suitable transitive models is essential: without transitivity, the purported earlier levels and their satisfaction relations need not be computed correctly. □

Proposition 38.3 (Definability of x in L)**.** *There is a parameter-free formula $\mathrm{Constr}(x)$ such that*

$$V \models \mathrm{Constr}(x) \quad \iff \quad x \in L.$$

One may take $\mathrm{Constr}(x) \equiv \exists \alpha(\alpha \text{ is ordinal} \wedge x \in L_\alpha)$ using the uniform level formula.

Proof.

Formal derivation spine

Formal spine for Definability of x in L.

$$x \in L \iff \exists\alpha \big(\alpha \text{ is an ordinal } \wedge\ \exists y\,(F(\alpha, y) \wedge x \in y)\big).$$

$$\mathrm{Lev}(x, \alpha) \iff x \in L_\alpha.$$

By the uniform-definability theorem there is a formula $F(\alpha, y)$ expressing $y = L_\alpha$. Therefore

$$x \in L \iff \exists\alpha \big(\alpha \text{ is an ordinal } \wedge\ \exists y\,(F(\alpha, y) \wedge x \in y)\big).$$

Every symbol on the right is first-order definable in the language of set theory, so membership in the constructible universe is a first-order definable class relation.

Conversely, if $x \in L$, by definition $x \in L_\alpha$ for some ordinal α, so the displayed formula holds. If the formula holds, the witness y is the uniquely determined L_α, hence x is constructible. The theorem asserts definability of the class L; it does not assert the existence of a set collecting all constructible sets. *Reading the displayed definition.* The outer existential quantifier chooses a stage index α. The next existential quantifier chooses the unique set y that the uniform stage formula identifies as L_α. The final atomic clause $x \in y$ says that x has appeared by that stage. Thus a proper-class definition is reduced to quantification over ordinary sets and ordinals. □

Theorem 38.4 (L computed inside L is L)**.** *If the constructible hierarchy is formed internally to L, then for every ordinal α,*

$$(L_\alpha)^L = L_\alpha,$$

and hence $L^L = L$.

Proof.

Formal derivation spine

Formal spine for L computed inside L is L.

$$L_\alpha^L = L_\alpha$$

$$L^L = \bigcup_{\alpha \in \mathrm{Ord}^L} L_\alpha^L = \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha = L.$$

$$\mathrm{Lev}(x, \alpha) \iff x \in L_\alpha.$$

The class L is transitive and contains every ordinal. Compute the constructible hierarchy internally in L, writing L_α^L for the level that L itself constructs. We prove

$$L_\alpha^L = L_\alpha$$

for every ordinal α. At the base stage both sides are empty. At a limit stage, both sides are unions of the same previously identified levels; transitivity and agreement on the relevant ordinals make those unions equal. For the successor step, the induction hypothesis identifies the underlying set-sized structure L_α in the internal and external computations. The absoluteness theorem for the L-construction then gives the same definable subsets and hence the same next level.

Taking the union over all ordinals,

$$L^L = \bigcup_{\alpha \in \mathrm{Ord}^L} L_\alpha^L = \bigcup_{\alpha \in \mathrm{Ord}} L_\alpha = L.$$

Thus constructibility is idempotent: once one passes to L, repeating the construction adds nothing new. □

Worked example

Worked Example 38.5 (Why uniformity matters)**.** Suppose a transitive model M contains an ordinal α and has computed every earlier L_β correctly. Because the formula defining $\mathrm{Def}(L_\beta)$ is uniform, both M and the ambient universe apply the same set-sized satisfaction recursion at the successor step. The induction can therefore continue through α.

A useful warning

Absoluteness of the construction requires transitivity and the relevant ordinals/parameters. It is not an unrestricted statement about arbitrary nonstandard models.

Looking ahead

The next chapter, *ZF axioms in L, part I*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 39

ZF axioms in L, part I

Definitions and setup

Formal definitions used in this chapter

Internal power set. $\mathcal{P}^L(a) = \mathcal{P}(a) \cap L$.
Truth in L. For parameters $\bar{a} \in L$,

$$L \models \varphi(\bar{a}) \quad \Longleftrightarrow \quad V \models \varphi^L(\bar{a}),$$

where φ^L is obtained by restricting every unbounded quantifier of φ to L.

ZF consists of Extensionality, Empty Set (derivable in common presentations), Pairing, Union, Power Set, Infinity, Separation, Replacement, Foundation, and the usual logical framework. Ordered pairs are Kuratowski pairs. Relations are sets of ordered pairs, and functions are functional relations. Each construction explicitly identifies the axiom that supplies the required set. Choice is not part of ZF and is never imported silently.

How this chapter fits together

The easy axioms follow mostly from transitivity and closure of the hierarchy, but Separation and Power Set require genuine arguments. Separation uses definability over a sufficiently high level. Power Set is internal: L needs the set of constructible subsets of a, not every ambient subset of a. Stage bounds are therefore essential.

Theorem 39.1 (L satisfies Extensionality, Foundation, Pairing, Union, and Infinity)**.** *The structure $(L, \in)$ satisfies Extensionality, Foundation, Pairing, Union, and Infinity.*

Proof.

Formal derivation spine

Formal spine: witnesses for the elementary ZF axioms occur at bounded levels. For $x, y \in L_\alpha$,

$$\{x, y\} \in L_{\alpha+2}, \qquad \bigcup x \in L_{\alpha+2}.$$

Also $\omega \in L_{\omega+1}$. Extensionality and Foundation are inherited from the transitive class $(L, \in)$. These facts give Extensionality, Foundation, Pairing, Union, and Infinity in L.

We verify only the five axioms named in the statement.

Extensionality is inherited because L is transitive and uses the ambient membership relation. *Foundation* is likewise inherited: a nonempty constructible set has an $\in$-minimal element in the ambient universe, and transitivity keeps that witness inside L. For *Pairing*, if $x, y \in L_\alpha$, the set $\{x, y\}$ is definable over L_α from x, y, so it belongs to $L_{\alpha+1}$. For *Union*, if $x \in L_\alpha$, then $\bigcup x$ is definable over a sufficiently later level containing x and its elements, hence is constructible. Finally, every finite ordinal is constructible and ω appears at a countable stage of the hierarchy, so L contains an inductive set. Therefore L satisfies Infinity.

No use of Separation, Replacement, Power Set, or Choice is hidden in this verification; those axioms receive independent proofs below. □

Theorem 39.2 (L satisfies Separation)**.** *For every formula $\varphi(x, \bar{p})$, every $a, \bar{p} \in L$, the set*

$$\{x \in a : L \models \varphi(x, \bar{p})\}$$

belongs to L. Thus $L \models$ Separation.

Commented symbolic trace: Separation inside L

Suppose $a \in L$ and parameters $\bar{p} \in L$. First choose one level L_β containing a and $\bar{p}$ and high enough for the relevant satisfaction calculation. The desired subset is then defined *over that set-sized structure* by the formula describing membership in a together with φ. Therefore it belongs to $\mathrm{Def}(L_\beta) = L_{\beta+1}$. The proof never assumes that every ambient subset of a is constructible.

Proof.

Formal derivation spine

Formal spine for L satisfies Separation.

$$b = \{x \in a : L \models \varphi(x, \bar{p})\}.$$

$$L \models \varphi(x, \bar{p}) \quad \Longleftrightarrow \quad L_\beta \models \varphi(x, \bar{p}).$$

$$b = \{x \in L_\beta : x \in a \ \&\ L_\beta \models \varphi(x, \bar{p})\}$$

Let $a, \bar{p} \in L$ and suppose the subset required by Separation is

$$b = \{x \in a : L \models \varphi(x, \bar{p})\}.$$

Choose an ordinal γ with $a, \bar{p} \in L_\gamma$. Apply the Reflection theorem to the finite set consisting of φ and all of its subformulas, with parameters $a, \bar{p}$. It yields an ordinal $\beta > \gamma$ such that, for every $x \in a$,

$$L \models \varphi(x, \bar{p}) \quad \Longleftrightarrow \quad L_\beta \models \varphi(x, \bar{p}).$$

The right-hand condition is a definition over the set-sized structure L_β using parameters $a, \bar{p} \in L_\beta$. Hence

$$b = \{x \in L_\beta : x \in a \ \&\ L_\beta \models \varphi(x, \bar{p})\}$$

belongs to $\mathrm{Def}(L_\beta) = L_{\beta+1}$. Thus $b \in L$, proving every instance of Separation internally in L. The reflection step is the point that converts truth in the proper class L into definability over one set-sized level. *What changed between the three displayed lines.* The first line defines the desired subset using truth in the whole class L, which is not yet a set-sized definition. Reflection replaces that class-level truth test by the equivalent truth test inside one level L_β. The third line then rewrites the same subset as a definable subset of the set L_β, so the successor operation $\mathrm{Def}(L_\beta)$ can actually create it. □

Theorem 39.3 (L satisfies the internal Power Set axiom)**.** *For every $a \in L$, the internal power set*

$$\mathcal{P}^L(a) := \{x \in L : x \subseteq a\} = \mathcal{P}(a) \cap L$$

is an element of L. Therefore $L \models$ Power Set.

Proof.

Formal derivation spine

Formal spine: first form the ambient set of constructible subsets, then bound their stages.

$$C := \{x \in \mathcal{P}(a) : x \in L\}$$

is a set in the ambient universe. If α_x is the least stage with $x \in L_{\alpha_x}$, then Replacement on C gives a bound β, and

$$C = \{x \in L_\beta : x \subseteq a\} \in L_{\beta+1}.$$

Fix $a \in L$. The formula "$x \in L$" is first-order definable by the uniform definability of the constructible hierarchy. Hence ambient Separation applied to the ambient set $\mathcal{P}(a)$ forms

$$C := \mathcal{P}(a) \cap L = \{x \in L : x \subseteq a\} = \mathcal{P}^L(a). \tag{1}$$

This step is essential: only after (1) do we have an ambient set over which Replacement may be applied.

For each $x \in C$, let α_x be the least ordinal such that $x \in L_{\alpha_x}$. The least stage is definable, so ambient Replacement forms the set $\{\alpha_x : x \in C\}$. Choose an ordinal β larger than its supremum and large enough that $a \in L_\beta$. Then every member of C belongs to L_β, and conversely every $x \in L_\beta$ with $x \subseteq a$ belongs to C. Therefore

$$C = \{x \in L_\beta : x \subseteq a\}. \tag{2}$$

The right-hand side of (2) is definable over the set structure L_β from the parameter a, so it belongs to

$$\mathrm{Def}(L_\beta) = L_{\beta+1}.$$

Thus $\mathcal{P}^L(a) = C \in L$. The proof constructs the internal power set, not the full ambient power set when nonconstructible subsets of a exist. □

Lemma 39.4 (Rank/localization control for witnesses used in later axioms)**.** *For every fixed formula $\varphi(x, y, \bar{p})$ and constructible parameters $a, \bar{p}$, there is an ordinal β such that every witness y selected for $x \in a$ may be replaced by a witness lying in L_β, whenever $L \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p})$.*

Proof.

Formal derivation spine

Formal spine: bound all witness stages over one set. For each $x \in a$, define

$$\beta_x := \min\{\gamma : \exists y \in L_\gamma\ L \models \varphi(x, y, \bar{p})\}.$$

Ambient Replacement collects $\{\beta_x : x \in a\}$; with

$$\beta := \sup_{x \in a}(\beta_x + 1),$$

every $x \in a$ has a witness in L_β.

Let $a, \bar{p} \in L$ and suppose that for every $x \in a$ there exists a constructible witness y satisfying a fixed formula $\varphi(x, y, \bar{p})$ in L. For each $x \in a$, define $\rho(x)$ to be the least ordinal α for which some witness $y \in L_\alpha$ exists. The class relation defining "$\rho(x) = \alpha$" is first-order definable because membership in L, the hierarchy, and satisfaction over set-sized levels are uniformly definable.

Ambient Replacement applied to the set a therefore collects the ordinals $\rho(x)$. Their supremum is bounded by some ordinal β, and every required witness may be chosen in L_β. This is only an external stage-localization statement at this point; the next chapter uses the uniform bound to prove Collection inside L without assuming Replacement in L in advance. □

Worked example

Worked Example 39.5 (Internal power set)**.** If $a \in L$, then $\mathcal{P}^L(a) = \{x \in L : x \subseteq a\}$. An ambient subset of a that is not constructible is irrelevant to the Power Set axiom as evaluated inside L. The proof must instead show that all constructible subsets of a are collected together in some later constructible level.

A useful warning

Never prove $L \models$ Power Set by claiming $\mathcal{P}(a) \subseteq L$; that assertion is generally false when the ambient universe contains nonconstructible subsets.

Looking ahead

The next chapter, *ZF axioms in L, part II*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 40

ZF axioms in L, part II

Definitions and setup

Formal definitions used in this chapter

Collection in L. Uniformly bound witnesses for all elements of a constructible domain by one constructible set.
Replacement in L. For a functional definable relation on a constructible set, its range is constructible.

ZF consists of Extensionality, Empty Set (derivable in common presentations), Pairing, Union, Power Set, Infinity, Separation, Replacement, Foundation, and the usual logical framework. Ordered pairs are Kuratowski pairs. Relations are sets of ordered pairs, and functions are functional relations. Each construction explicitly identifies the axiom that supplies the required set. Choice is not part of ZF and is never imported silently.

How this chapter fits together

Replacement is the first serious closure test for L. Given a definable functional relation on a constructible set, witnesses may initially appear at unrelated stages. The proof first obtains a uniform ordinal bound on those witness stages; Collection then gathers candidates inside one level, and Separation extracts the actual range.

Lemma 40.1 (Definable collection bound for L)**.** *If $L \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p})$, then there is an ordinal β such that*

$$L \models \forall x \in a\, \exists y \in L_\beta\, \varphi(x, y, \bar{p}).$$

The bound β is obtained from the least constructibility stages at which witnesses occur.

Proof.

Formal derivation spine

Formal spine: the definable Collection bound. Assuming $L \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p})$, define the least witness stages β_x and let

$$\beta = \sup\{\beta_x + 1 : x \in a\}.$$

Then

$$L \models \forall x \in a\, \exists y \in L_\beta\, \varphi(x, y, \bar{p}).$$

Proof map. The key issue is uniformity. It is not enough that each $x \in a$ has some witness somewhere in L. Associate to x the least ordinal stage at which an appropriate witness appears, using the definable

constructible well-order to remove ambiguity. Ambient Replacement collects those stage ordinals and their supremum gives one level containing witnesses for all $x \in a$.

Let $a, \bar{p} \in L$ and assume $L \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p})$. Because membership in L and satisfaction by a set-sized level L_α are uniformly definable in the ambient universe, for each $x \in a$ there is a least ordinal α_x for which some $y \in L_{\alpha_x}$ satisfies the relativized witness condition. The relation $x \mapsto \alpha_x$ is definable and functional in the ambient ZF universe, so ambient Replacement produces the set $\{\alpha_x : x \in a\}$. Let β exceed its supremum and the ranks of all parameters. Then every $x \in a$ has a witness in the single set L_β. This uniform stage bound is the definable-collection bound; no choice of one witness from each fiber is required. □

Theorem 40.2 (L satisfies Collection)**.** *For every formula $\varphi(x, y, \bar{p})$, if $a, \bar{p} \in L$ and*

$$L \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p}),$$

then there is $b \in L$ such that

$$L \models \forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p}).$$

Hence $L \models$ Collection.

Uses: uniform definability of the constructible hierarchy, stage localization, and ambient Replacement for the set of least witness stages.

How to read the symbols: Collection needs a uniform stage bound

For each $x \in a$, existence gives at least one constructible witness y. The key move is not to choose a preferred witness; it is to choose the least ordinal stage at which *some* witness appears. Replacement in the ambient universe collects these ordinal stage numbers, whose supremum gives one level L_β containing a witness for every $x \in a$. Only after this common bound is obtained do we use Separation to collect a set of witnesses.

Proof.

Formal derivation spine

Formal spine for L satisfies Collection.

$$b = \{y \in L_\beta : \exists x \in a\, \varphi^L(x, y, \bar{p})\}.$$

$$\beta_x := \min\{\beta : \exists y \in L_\beta\ L \models \varphi(x, y, \bar{p})\}.$$

Proof map. With the stage bound in hand, Collection inside L is now local: all needed witnesses lie in one set L_β. A definable subset of a sufficiently later level selects the pairs satisfying the formula, and projection gives a set containing at least one witness for each element of the original domain.

Use the bound from the definable collection bound proved immediately above. There is an ordinal β such that for every $x \in a$ a witness $y \in L_\beta$ satisfies $\varphi^L(x, y, \bar{p})$. Choose a later level L_γ containing $a, \bar{p}, L_\beta$ and enough of the uniform satisfaction machinery. By Separation in L_γ, form

$$b = \{y \in L_\beta : \exists x \in a\, \varphi^L(x, y, \bar{p})\}.$$

Then $b \in L$, and L verifies that every $x \in a$ has a φ-witness in b. This is Collection. The proof is noncircular: it uses ambient Replacement only in the metatheoretic verification that the inner class L satisfies Collection, and it does not use AC in L. □

Theorem 40.3 (L satisfies Replacement)**.** *If $L \models \forall x \in a\, \exists! y\, \varphi(x, y, \bar{p})$, then the image set*

$$\{y : \exists x \in a\, \varphi(x, y, \bar{p})\}$$

belongs to L. Hence $L \models$ Replacement.

Proof.

Formal derivation spine

Formal spine for L satisfies Replacement.

$$r = \{y \in b : \exists x \in a\, \varphi(x, y, \bar{p})\}.$$

$$\beta_x := \min\{\beta : \exists y \in L_\beta\ L \models \varphi(x, y, \bar{p})\}.$$

Proof map. Replacement is derived only after Collection. For a functional relation, Collection supplies a set containing possible values; Separation selects exactly those elements that are the unique values of members of the domain. This ordering avoids circular use of Replacement in L.

Assume in L that $\varphi(x, y, \bar{p})$ is functional on a set a: for every $x \in a$ there is a unique y with $\varphi(x, y, \bar{p})$. By Collection (the Collection theorem just proved) there is $b \in L$ containing at least one witness for every $x \in a$. Separation in L then forms

$$r = \{y \in b : \exists x \in a\, \varphi(x, y, \bar{p})\}.$$

Uniqueness makes r exactly the range of the functional relation. Thus the Replacement schema holds in L. This order of proof avoids the earlier circular presentation: Collection is obtained from a uniform constructibility-stage bound, and Replacement is derived only afterward from Collection plus Separation. □

Theorem 40.4 (L satisfies ZF)**.** *The inner model $(L, \in)$ satisfies every axiom of ZF.*

Proof.

Formal derivation spine

Formal spine: assemble the separately verified ZF axioms. The preceding results establish in L:

$$\text{Ext} + \text{Found} + \text{Pair} + \text{Union} + \text{Inf},$$

$$\text{Sep} + \text{Power} + \text{Collection} + \text{Replacement}.$$

Hence every axiom in the chosen ZF axiomatization holds in $(L, \in)$, i.e.

$$L \models \text{ZF}.$$

The structural axioms Extensionality, Foundation, Pairing, Union, and Infinity were verified using transitivity and the closure of the hierarchy. Separation and the internal Power Set axiom were proved by definability and stage bounding. the Collection theorem just proved proves Collection and the following Replacement theorem derives Replacement. These are precisely the ZF axioms (in the chosen equivalent formulation with Collection/Replacement). Therefore $L \models \text{ZF}$. No appeal to the constructible well-order or to Choice is needed for this conclusion; Choice is proved only in the following chapter.

Further explanation. The conclusion is a bookkeeping theorem only after the difficult axioms have been checked separately. Extensionality, Foundation, Pairing, Union, and Infinity follow from transitivity and the level construction; Separation uses the reflecting level L_β supplied by the Reflection theorem; Power Set is the internal constructible power set; and Collection/Replacement require the uniform stage-bound argument of the preceding chapter. Since each axiom has now been verified inside L, the conjunction of those results is exactly $L \models \text{ZF}$, with no appeal yet to Choice. □

Remark 40.5 (Source note)**.** Gödelś detailed constructibility proof verifies the ZF axioms in the inner model L; the present organization separates the easy structural axioms from Separation, Power Set, Collection, and Replacement. Compare [8, 13, 14].

Worked example

Worked Example 40.6 (Bounding witness stages)**.** For each $x \in a$, let β_x be the least stage containing a witness to $\varphi(x, y)$, choosing the least witness in the canonical constructible order when needed. Ambient Replacement collects the ordinals β_x, and their supremum gives one stage L_β containing witnesses for every $x \in a$. This uniform bound is the heart of Collection in L.

A useful warning

Do not use Replacement in L while proving Replacement in L. The stage-bound argument may use Replacement in the ambient metatheory, but the internal conclusion must be obtained from already verified closure properties.

Looking ahead

The next chapter, *Canonical well-order and Choice*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 41

Canonical well-order and Choice

Definitions and setup

Formal definitions used in this chapter

Canonical constructible code. The least stage, formula code, and parameter tuple that define an element of L.
$<_L$. The lexicographic well-order induced by these least construction codes.

Each constructible set has at least one definition code consisting of a stage, a formula, and earlier parameters. Order definition codes lexicographically using the already constructed orders of earlier stages, and assign each set its least code. This gives the canonical constructible well-order $<_L$. The order is built from definability data and does not assume AC; AC is then derived by choosing the $<_L$-least element from each nonempty constructible set.

How this chapter fits together

Every constructible set has a least description at the first stage where it appears. Ordering these description codes lexicographically produces a definable global well-order of L. Choice is then obtained internally by a concrete construction: select the least element of each nonempty set. The construction also supplies the canonical choices used in Condensation and GCH.

Proposition 41.1 (Canonical well-order of each L_α)**.** *For every ordinal α, there is a well-order $<_{L,\alpha}$ of L_α definable uniformly from α, obtained by comparing the least formula/parameter codes defining each element at the first stage where it appears.*

Proof.

Formal derivation spine

Formal spine for Canonical well-order of each L_α.

$$L_{\alpha+1} = \mathrm{Def}(L_\alpha)$$

$$x <_{L,\alpha+1} y \iff \mathrm{code}_\alpha(x) <_{\mathrm{lex}} \mathrm{code}_\alpha(y).$$

$$(\alpha, e, \bar{a})$$

Assume inductively that $<_{L,\alpha}$ well-orders L_α. Every element of

$$L_{\alpha+1} = \mathrm{Def}(L_\alpha)$$

has at least one definition code $(e, \bar{a}) \in \omega \times L_\alpha^{<\omega}$. Order such codes lexicographically using the natural order on e and the induced lexicographic order on parameter tuples from $<_{L,\alpha}$. For each $x \in L_{\alpha+1}$, choose the least code defining x; define

$$x <_{L,\alpha+1} y \iff \mathrm{code}_\alpha(x) <_{\mathrm{lex}} \mathrm{code}_\alpha(y).$$

At limit stages use the least stage at which an element appears and then the previously defined order within that stage. Transfinite recursion yields a well-order of every L_α. *Point specific to this result.* For this result, the shared construction is applied only to the specific conclusion stated above; the preceding definitions determine which part of the common induction or recursion is doing the work. □

Theorem 41.2 (Definable global well-order of L)**.** *There is a parameter-free definable class relation $<_L$ that well-orders L and whose restriction to each L_α is a set well-order.*

Proof.

Formal derivation spine

Formal spine for Definable global well-order of L.

$$\alpha_x := \min\{\alpha : x \in L_{\alpha+1}\}.$$

$$x <_L y \iff (\alpha_x, c_x) <_{\mathrm{lex}} (\alpha_y, c_y).$$

$$(\alpha, e, \bar{a})$$

For $x \in L$, let

$$\alpha_x := \min\{\alpha : x \in L_{\alpha+1}\}.$$

The preceding theorem gives a canonical code c_x of x over L_{α_x}. Define the global relation

$$x <_L y \iff (\alpha_x, c_x) <_{\mathrm{lex}} (\alpha_y, c_y).$$

Uniform definability of the hierarchy, satisfaction, and the level orders implies that the relation $<_L$ is definable without parameters. Every nonempty set $A \in L$ is contained in some L_θ, and the restriction of $<_L$ to L_θ is a set well-order, so A has a least element. Hence $<_L$ is a definable global well-order of L. *Point specific to this result.* Order constructible sets first by the least stage at which they appear and then by their least definition codes within that stage. Uniform definability of the hierarchy makes this a single definable class relation on L. □

Theorem 41.3 (L satisfies AC)**.** *The definable global well-order $<_L$ yields a choice function for every constructible family of nonempty sets. Hence*

$$L \models \mathrm{AC}.$$

Proof.

Formal derivation spine

Formal spine for L satisfies AC.

$$f(x) := \min_{<_L} x \qquad (x \in A).$$

$$y = f(x) \iff y \in x \ \wedge\ \forall z \in x\, \neg(z <_L y)$$

$$L \models \forall x \in A\, [f(x) \in x],$$

Let $A \in L$ be a set of nonempty sets. Using the global well-order from the preceding theorem, define

$$f(x) := \min_{<_L} x \qquad (x \in A).$$

For each $x \in A$, existence and uniqueness of $f(x)$ follow from the fact that $<_L$ well-orders L. The defining relation

$$y = f(x) \iff y \in x \ \wedge \ \forall z \in x \, \neg(z <_L y)$$

is first-order definable in L. Replacement in L, proved earlier without Choice, collects the graph of f. Thus

$$L \models \forall x \in A \, [f(x) \in x],$$

so f is a choice function and $L \models \mathrm{AC}$. *Point specific to this result.* For any constructible family of nonempty constructible sets, the global $<_L$-well-order selects a least element from each member. Replacement, already established inside L, collects these selected elements into a choice function. □

Corollary 41.4 (Relative consistency Con(ZF) implies Con(ZFC) via L, subject to the metatheoretic transfer proved later)**.** *Once the metatheoretic consistency-transfer principle of Volume VIII is available, the inner-model theorem $L \models \mathrm{ZF} + \mathrm{AC}$ yields*

$$\mathrm{Con}(\mathrm{ZF}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC}).$$

Proof.

Formal derivation spine

Formal spine for Relative consistency Con(ZF) implies Con(ZFC) via L, subject to the metatheoretic transfer proved later.

$$(\alpha, e, \bar{a})$$

$$f(A) = \min_{<_L} A$$

Inside any model $M \models \mathrm{ZF}$, the preceding construction can be carried out internally to form its constructible class L^M. The proofs of the ZF axioms and of the canonical constructible well-order are formalizable in ZF, so M satisfies that L^M is an inner model of ZFC (and later, after the GCH chapter, of $\mathrm{ZFC} + \mathrm{GCH}$). Consequently, at the model-theoretic level, every model of ZF contains an internally defined model of ZFC. To turn this statement into the bare syntactic implication $\mathrm{Con}(\mathrm{ZF}) \Rightarrow \mathrm{Con}(\mathrm{ZFC})$ without assuming a transitive model, we use the formal model-existence/consistency-transfer machinery developed in Volume VIII. This proposition records the mathematical inner model construction while explicitly deferring only that metatheoretic conversion. □

Worked example

Worked Example 41.5 (Choosing the least constructible element)**.** Let $X \in L$ be nonempty. Because $<_L$ well-orders L, the subset X has a unique $<_L$-least element. For a family $\mathcal{A} \in L$ of nonempty sets, map $A \in \mathcal{A}$ to its least element. Replacement in L collects these values into a choice function.

A useful warning

The global relation $<_L$ is definable; it is not required to be a set containing all constructible objects, since L itself is a proper class.

Looking ahead

The next chapter, *Condensation*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 42

Condensation

Definitions and setup

Formal definitions used in this chapter

Condensation setup. For $X \prec L_\theta$, let $\pi : X \cong M$ be the Mostowski collapse. Condensation identifies the transitive collapse M with an earlier level L_β.

Condensation identifies elementary substructures of constructible levels. Start with $X \prec L_\theta$, collapse X to a transitive set M, and use uniform definability of constructibility to show $M = L_\beta$ for its ordinal height β. The collapse fixes ordinals that are contained pointwise in the hull. This identification is the key localization tool for constructible subsets of a cardinal.

How this chapter fits together

Condensation identifies the transitive collapse of a suitable elementary substructure of L_θ with an actual earlier level L_β. The proof combines three ideas already developed separately: Skolem hulls, Mostowski collapse, and uniform definability of the constructible hierarchy. The identification with L_β, rather than with an arbitrary transitive model, is the decisive step.

Lemma 42.1 (Skolem hulls inside L_theta)**.** *Let θ be an ordinal and $A \subseteq L_\theta$. Closing A under the definable Skolem functions of $(L_\theta, \in, <_L)$ produces an elementary hull $X = \mathrm{Hull}^{L_\theta}(A) \prec L_\theta$.*

Proof.

Formal derivation spine

Formal spine: construct the Skolem hull. Let $H_0 = A$ and recursively put

$$H_{n+1} = H_n \cup \{F_\varphi(\bar{a}) : \bar{a} \in H_n^{<\omega},\ F_\varphi \text{ a Skolem function of } L_\theta\}.$$

Then

$$X := \bigcup_{n<\omega} H_n$$

is closed under all Skolem functions, so the Tarski–Vaught criterion yields $X \prec L_\theta$.

Expand the structure by chosen Skolem functions for existential formulas. Beginning with the parameter set A, let $H_0 = A$ and let H_{n+1} contain the values of all basic and Skolem functions on tuples from H_n. Set $H = \bigcup_n H_n$. Cardinal arithmetic bounds its size by $\max(|A|, |L|, \aleph_0)$. The Skolem property gives the

Tarski–Vaught witness condition, hence $H \preccurlyeq M$. When the ambient relation is well founded/extensional, Mostowski collapse sends H to a transitive model, preserving the relevant first-order theory by isomorphism.

Specialization to this result. Inside L_θ, use the definable constructible well-order to choose Skolem witnesses canonically. Closing the chosen parameter set under those functions yields an elementary hull whose size is controlled by the starting parameters and language. □

Lemma 42.2 (Mostowski collapse of the L-hull)**.** *If $X \prec L_\theta$, the well-founded extensional structure $(X, \in)$ has a transitive Mostowski collapse $\pi : X \cong M$.*

Proof.

Formal derivation spine

Formal spine: collapse the elementary hull. Because $(X, \in)$ is well founded and extensional, define by well-founded recursion

$$\pi(x) = \{\pi(y) : y \in x \cap X\}.$$

Then $M := \pi[X]$ is transitive and

$$x \in_X y \iff \pi(x) \in \pi(y),$$

so $\pi : X \cong (M, \in)$.

Let R be well founded and extensional on A. Define recursively $\pi(a) = \{\pi(b) : bRa\}$. Well-founded recursion supplies a unique function. If $\pi(a) = \pi(a')$, then extensionality and induction on predecessors imply that a, a' have the same R-predecessors, hence $a = a'$; thus π is injective. Let $M = \pi[A]$. By definition, $x \in \pi(a)$ iff $x = \pi(b)$ for some bRa, so M is transitive and π is an isomorphism $(A, R) \cong (M, \in)$. Any other collapse satisfies the same recursion and hence equals π.

Point specific to this result. The Skolem hull is well founded because it is a substructure of the transitive L_θ. Mostowski collapse therefore produces a transitive set; the next Condensation argument identifies that set with an earlier constructible level. □

Theorem 42.3 (Condensation lemma for L)**.** *(Condensation.) Let θ be a limit ordinal. If $X \prec L_\theta$ and $\pi : X \cong M$ is its transitive collapse, then there is an ordinal β such that*

$$M = L_\beta.$$

Uses: Skolem hulls inside an L-level, Mostowski collapse, and absoluteness of the L-hierarchy.

Commented symbolic trace: Condensation: hull → collapse → identify

There are three constructions to keep distinct. First take an elementary hull $X \prec L_\theta$. Second apply Mostowski collapse to obtain a transitive set M isomorphic to X. Third use elementarity plus uniform definability of the L-hierarchy to prove that this transitive set is not arbitrary: it is exactly L_β for its ordinal height β. Condensation is the third identification step, not merely the existence of a collapse.

Proof.

Formal derivation spine

Formal spine for Condensation lemma for L.

$$\pi(L_\alpha \cap X) = L_{\pi(\alpha)}.$$

$$\beta := \mathrm{Ord} \cap M.$$

$$L_\theta \models \text{“}y = L''_\gamma \iff M \models \text{“}\pi(y) = L''_{\pi(\gamma)}.$$

Proof map. After collapsing $X \prec L_\theta$ to a transitive set M, the difficult point is identification. Uniform definability of the L-hierarchy is preserved by the collapse, so for each ordinal $\gamma \in M$, the structure M contains exactly the level it calls L_γ. Induction up to $\beta = M \cap \mathrm{Ord}$ yields $M = L_\beta$.

Let θ be the limit ordinal from the theorem, let $X \prec L_\theta$, and let $\pi : X \to M$ be the Mostowski collapse. Since $(L_\theta, \in)$ is well founded and extensional, M is transitive. Put $\beta = M \cap \mathrm{Ord}$. The formula uniformly defining the constructible hierarchy is absolute through the collapse: if $\alpha \in X \cap \mathrm{Ord}$, then elementarity says that L_α is the unique object satisfying that definition at α, so

$$\pi(L_\alpha \cap X) = L_{\pi(\alpha)}.$$

By induction on $\delta < \beta$, this puts every L_δ inside M, hence $L_\beta \subseteq M$. Conversely, $L_\theta \models V = L$ for all of its elements. Elementarity transfers this to X, and after collapse every $x \in M$ is constructible from ordinals below β; hence $x \in L_\beta$. Thus $M = L_\beta$. This is the Condensation Lemma in the form used below.

Detailed formal verification. Let $\pi : X \to M$ be the Mostowski collapse and write

$$\beta := \mathrm{Ord} \cap M.$$

Because $X \prec L_\theta$, every definition of a constructible level is preserved by elementarity. If $\gamma \in X \cap \mathrm{Ord}$, uniform definability of the hierarchy gives

$$L_\theta \models \text{“}y = L''_\gamma \iff M \models \text{“}\pi(y) = L''_{\pi(\gamma)}.$$

Hence

$$\pi(L_\gamma) = L_{\pi(\gamma)}. \tag{C1}$$

Every $a \in M$ is $\pi(x)$ for some $x \in X$. Choose $\gamma \in X \cap \mathrm{Ord}$ with $x \in L_\gamma$. By (C1),

$$a = \pi(x) \in L_{\pi(\gamma)} \subseteq L_\beta,$$

so $M \subseteq L_\beta$. Conversely, if $\delta < \beta$, choose $\gamma \in X \cap \mathrm{Ord}$ with $\pi(\gamma) = \delta$. Then (C1) gives $L_\delta = \pi(L_\gamma) \in M$, and transitivity of M yields $L_\delta \subseteq M$. Therefore

$$L_\beta = \bigcup_{\delta < \beta} L_\delta \subseteq M.$$

Thus $M = L_\beta$, which is the condensation conclusion. □

Remark 42.4 (Source note)**.** Condensation is used here in the modern hull-and-collapse form that makes the later cardinal localization transparent; compare [8, 13, 14].

Corollary 42.5 (Localization of constructible subsets by condensation)**.** *Let κ be an infinite cardinal of L and $A \subseteq \kappa$ with $A \in L$. Then*

$$A \in L^L_{\kappa^+},$$

where κ^+ is computed in L.

Proof.

Formal derivation spine

Formal spine: localize a constructible subset below κ^+. Choose $X \prec L_\theta$ with

$$A \in X, \qquad \kappa + 1 \subseteq X, \qquad |X|^L = \kappa.$$

Condensation gives a collapse $X \cong L_\beta$. The collapse fixes $\kappa + 1$, hence $\pi(A) = A$, while

$$|\beta|^L \leq \kappa \Longrightarrow \beta < \kappa^+.$$

Therefore $A \in L_\beta \subseteq L_{\kappa^+}$.

Let $A \in L$ be an infinite transitive set. Choose a limit ordinal $\theta > \operatorname{rank}_L(A) + \omega$ such that $A \in L_\theta$. By downward Löwenheim–Skolem choose $X \prec L_\theta$ with $A \cup \{A\} \subseteq X$ and $|X| \leq |A|$. Condensation collapses X to L_β for some β with $|\beta| \leq |X| \leq |A|$, hence $\beta < |A|^+$. Because A is transitive and every element of A lies in X, induction on rank shows that the collapse fixes A pointwise and sends A to itself. Therefore $A \in L_\beta \subseteq L_{|A|^+}$. Applying this to the transitive set $\kappa \cup \{B\}$ when $B \subseteq \kappa$ gives the early-localization statement needed for GCH.

Further explanation. In the application above, choose a limit ordinal θ with the constructible set A and all stated parameters in L_θ. Form a small elementary hull and collapse it. Condensation identifies the collapse with L_β for some smaller β, while elementarity ensures that the image of A still belongs to that level. When the hull fixes the relevant ordinal parameters, this places A itself in the desired early constructible level. □

Worked example

Worked Example 42.6 (Collapsing a hull that contains a real)**.** Let $A \subseteq \omega$ belong to some large L_θ, and take a countable elementary hull $X \prec L_\theta$ containing A. Collapse X to a transitive structure M. Since ω and its elements are fixed by the collapse, A is fixed as well. Condensation identifies $M = L_\beta$ for a countable β, showing that the real already appears at a countable constructible stage.

A useful warning

Mostowski collapse alone gives only a transitive set. Condensation additionally uses elementarity and definability of L to identify that set as a specific constructible level.

Looking ahead

The next chapter, *GCH in L*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 43

GCH in L

Definitions and setup

Formal definitions used in this chapter

Constructible successor cardinal. κ^+ is computed inside L unless explicitly annotated otherwise.
GCH in L. The assertion $L \models 2^\kappa = \kappa^+$ for every infinite L-cardinal κ.

GCH in L is a localization theorem plus a counting theorem. Condensation shows that a constructible subset of an infinite constructible cardinal κ already appears below L_{κ^+}. The number of elements of that level is at most κ^+. Cantor gives the reverse strict inequality over κ, forcing $(2^\kappa)^L = \kappa^+$. CH is written separately as the concrete $\kappa = \aleph_0$ case.

How this chapter fits together

The GCH proof has a cardinal-counting side and a localization side. First, L_α has at most $\max(\aleph_0, |\alpha|)$ elements because its members are coded by formulas and finite parameter tuples. Second, Condensation shows that every constructible subset of an infinite cardinal κ appears before κ^+. The two facts bound the power set from above, while Cantor supplies the strict lower bound.

Theorem 43.1 (Cardinality bounds for levels L_α)**.** *For every infinite ordinal α,*

$$|L_\alpha| \leq \max(\aleph_0, |\alpha|).$$

In particular, if $\alpha < \kappa^+$ for an infinite cardinal κ, then $|L_\alpha| \leq \kappa$.

Proof.

Formal derivation spine

Formal spine for Cardinality bounds for levels L_α.

$$|L_\alpha| \leq \max(\aleph_0, |\alpha|).$$

$$|L_{\beta+1}| \leq \max(\aleph_0, |L_\beta|).$$

$$|L_\lambda| \leq \max(\aleph_0, |\lambda|).$$

We prove by transfinite induction on the infinite ordinal α that

$$|L_\alpha| \leq \max(\aleph_0, |\alpha|).$$

At a successor stage, every element of $L_{\beta+1} = \mathrm{Def}(L_\beta)$ is specified by a first-order formula together with a finite tuple of parameters from L_β. There are only $\aleph_0$ formulas, and for infinite L_β there are $|L_\beta|^{<\omega} = |L_\beta|$ finite parameter tuples. Hence

$$|L_{\beta+1}| \leq \max(\aleph_0, |L_\beta|).$$

The induction hypothesis bounds the right-hand side by $\max(\aleph_0, |\beta|) \leq \max(\aleph_0, |\beta+1|)$.

If λ is a limit ordinal, then $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$. The union contains $|\lambda|$ many sets, each of size at most $\max(\aleph_0, |\lambda|)$, so

$$|L_\lambda| \leq \max(\aleph_0, |\lambda|).$$

This proves the level-size estimate and nothing stronger; the localization of subsets needed for GCH is the next, separate theorem. □

Theorem 43.2 (Constructible subsets of a cardinal appear sufficiently early)**.** *If κ is an infinite cardinal in L, then every constructible subset of κ belongs to L_{κ^+}. Consequently*

$$\mathcal{P}^L(\kappa) \subseteq L_{\kappa^+}.$$

Proof.

Formal derivation spine

Formal spine for Constructible subsets of a cardinal appear sufficiently early.

$$X \prec L_\theta$$

$$\pi(A) = \{\pi(\xi) : \xi \in A\} = A.$$

$$A \in L_{\kappa^+}.$$

Work inside L, where Choice has already been established. Let κ be an infinite cardinal of L and let $A \subseteq \kappa$ belong to L. Choose a limit ordinal $\theta > \kappa^+$ such that $A, \kappa \in L_\theta$. By the Skolem-hull theorem, choose

$$X \prec L_\theta$$

such that $A \in X$, $\kappa + 1 \subseteq X$, and $|X| = \kappa$ in L. Let $\pi : X \to M$ be the Mostowski collapse. Condensation gives $M = L_\beta$ for some β.

Because every ordinal below or equal to κ lies in X, the collapse fixes those ordinals pointwise. Since $A \subseteq \kappa$ and $A \in X$,

$$\pi(A) = \{\pi(\xi) : \xi \in A\} = A.$$

Thus $A \in L_\beta$. Moreover the ordinals of M are the collapsed ordinals of X, so $|\beta| \leq |X| = \kappa$; hence $\beta < \kappa^+$. Therefore

$$A \in L_{\kappa^+}.$$

Every constructible subset of κ therefore appears strictly before the level L_{κ^+}. □

Theorem 43.3 (CH in L)**.** *Inside L,*

$$2^{\aleph_0} = \aleph_1.$$

Thus $L \models \mathrm{CH}$.

Proof.

Formal derivation spine

Formal spine for CH in L.

$$|L_{\omega_1^L}|^L \leq \aleph_1^L,$$

$$L \models 2^{\aleph_0} = \aleph_1.$$

$$A \subseteq \kappa \ \wedge \ A \in L \Longrightarrow A \in L_{(\kappa^+)^L}.$$

Apply the preceding localization theorem with $\kappa = \omega$. Every real of L, viewed as a subset of ω, belongs to $L_{\omega_1^L}$. The level-size estimate gives

$$|L_{\omega_1^L}|^L \leq \aleph_1^L,$$

so L has at most $\aleph_1^L$ many reals. Cantor's theorem, which holds in L, gives $2^{\aleph_0} > \aleph_0$. Since $L \models \mathrm{AC}$, its cardinals are linearly well ordered and there is no cardinal strictly between $\aleph_0$ and $\aleph_1$. Hence

$$L \models 2^{\aleph_0} = \aleph_1.$$

This proves CH in L as the first concrete instance of the general GCH argument.

The superscript L matters throughout: ω_1^L is the first uncountable ordinal as computed inside L. The argument is internal to that model and does not require ω_1^L to equal the ambient universe's ω_1. This is exactly why the result is a model of CH rather than an assertion that CH holds in the ambient universe. □

Theorem 43.4 (GCH in L)**.** *For every infinite cardinal κ of L,*

$$L \models 2^\kappa = \kappa^+.$$

Hence $L \models$ GCH.

Uses: Condensation/localization, the cardinality bounds for L_α, and Cantor's theorem.

How to read the symbols: the GCH calculation is two inequalities

For an infinite constructible cardinal κ, localization shows that every constructible subset of κ appears before L_{κ^+}. The level-size estimate then gives at most κ^+ such subsets, so $2^\kappa \leq \kappa^+$ inside L. Cantor's theorem gives $2^\kappa > \kappa$. Since κ^+ is the least larger cardinal, the two inequalities force $2^\kappa = \kappa^+$.

Proof.

Formal derivation spine

Formal spine for GCH in L.

$$\mathcal{P}^L(\kappa) \subseteq L_{\kappa^+}.$$

$$\mathcal{P}^L(\kappa) \subseteq L_{(\kappa^+)^L}. \tag{GCH1}$$

$$\left|L_{(\kappa^+)^L}\right|^L \leq (\kappa^+)^L. \tag{GCH2}$$

Proof map. For arbitrary infinite constructible κ, apply the same localization to subsets of κ: every such constructible subset occurs before κ^+. The level-size estimate gives at most κ^+ many subsets; Cantor gives $2^\kappa > \kappa$. Choice in L then forces $2^\kappa = \kappa^+$.

Work inside L, where AC has already been proved, and let κ be an infinite cardinal of L. The preceding localization theorem states directly that every constructible subset $B \subseteq \kappa$ belongs to L_{κ^+}, where κ^+ is computed in L. Hence

$$\mathcal{P}^L(\kappa) \subseteq L_{\kappa^+}.$$

The level-size estimate gives $|L_{\kappa^+}|^L \leq \kappa^+$, and therefore $(2^\kappa)^L \leq \kappa^+$. Cantor's theorem holds in L, so $(2^\kappa)^L > \kappa$. Since $L \models \mathrm{AC}$, κ^+ is the least cardinal strictly larger than κ; consequently $(2^\kappa)^L \geq \kappa^+$. The two inequalities give $(2^\kappa)^L = \kappa^+$ for every infinite cardinal κ of L.

Detailed formal verification. Fix an infinite cardinal κ of L. Localization gives, internally to L,

$$\mathcal{P}^L(\kappa) \subseteq L_{(\kappa^+)^L}. \tag{GCH1}$$

The level-size induction yields

$$\left|L_{(\kappa^+)^L}\right|^L \leq (\kappa^+)^L. \tag{GCH2}$$

Combining (GCH1) and (GCH2),

$$(2^\kappa)^L = |\mathcal{P}^L(\kappa)|^L \leq (\kappa^+)^L. \tag{GCH3}$$

Cantor's theorem is valid in L, so

$$L \models \kappa < 2^\kappa. \tag{GCH4}$$

By definition, $(\kappa^+)^L$ is the least constructible cardinal strictly larger than κ. The inequalities (GCH3)–(GCH4) therefore force

$$L \models 2^\kappa = \kappa^+.$$

Notice that every cardinal symbol in this calculation is interpreted inside L; no ambient successor cardinal is silently substituted for the constructible one.

Second detailed calculation. A useful way to see the cardinal step without suppressing the model is to write all quantities with an L-superscript:

$$(2^\kappa)^L := |\mathcal{P}(\kappa) \cap L|^L.$$

Localization says that the identity map embeds this set into $L_{(\kappa^+)^L}$, hence

$$(2^\kappa)^L \leq |L_{(\kappa^+)^L}|^L.$$

The level bound is proved by induction using only constructible cardinal arithmetic, so its value at the successor stage is

$$|L_{(\kappa^+)^L}|^L \leq (\kappa^+)^L.$$

On the other hand, Cantor's diagonal map is absolute enough to run internally in L, and therefore

$$L \models \kappa < 2^\kappa.$$

There is no L-cardinal strictly between κ and $(\kappa^+)^L$. Thus the upper and lower bounds identify the same internal cardinal. This last sentence is where successor-cardinal minimality is used; it should not be replaced by an ambient statement about κ^+. □

Remark 43.5 (Source note)**.** Gödelś constructible universe satisfies the generalized continuum hypothesis. The proof here factors the argument into level-size bounds plus Condensation/localization; compare [8, 13, 14].

Worked example

Worked Example 43.6 (CH as the prototype)**.** For a constructible real $A \subseteq \omega$, choose a limit ordinal $\theta > \omega_1^L$ with $A \in L_\theta$, and take a countable elementary hull of L_θ containing A. Condensation places A in some L_β with $\beta < \omega_1$. There are only $\aleph_1$ such stages and each has size at most $\aleph_0$, so L has at most $\aleph_1$ reals. Cantor gives more than $\aleph_0$, hence exactly $\aleph_1$.

A useful warning

The upper bound must be established inside L with correct constructible cardinals. Ambient cardinal arithmetic cannot simply be substituted for internal cardinal arithmetic.

End-of-volume perspective

The constructible universe provides the positive consistency side: inside L, both Choice and GCH hold. The remaining task is to construct alternative universes in which selected statements fail without destroying ZF or ZFC.

Part VII

Volume VII: Forcing I: Generic Extensions and the Forcing Theorem

Preface

Forcing is first developed as a theorem about generic extensions. Atomic forcing, definability, and the Truth Lemma are all proved rather than compressed into a slogan.

Position in the series

Volume VII develops forcing as a theorem about generic extensions. It owns names, the forcing relation, the Truth Lemma, and the proof that ordinary generic extensions preserve ZFC.

Sources and further reading

The forcing theorem is presented in modern name-and-truth-lemma form. Cohen's original method is represented historically by [9, 10, 11]; systematic modern treatments include [13, 14].

Notation for this volume

The order convention is that $q \leq p$ means q is stronger. τ^G is name valuation, $\check{x}$ a ground-model name, and $p \Vdash \varphi$ the forcing relation.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 44

Posets, dense sets, and generics

Definitions and setup

Formal definitions used in this chapter

Forcing order. $q \leq p$ means q contains at least as much information as p; $1_{\mathbb{P}}$ is the largest, weakest condition.
Dense set. $D \subseteq \mathbb{P}$ is dense if $\forall p \exists q \leq p\, (q \in D)$.
Filter. A set $G \subseteq \mathbb{P}$ is a filter when it is upward closed toward weaker conditions,

$$p \in G \ \wedge \ p \leq q \Longrightarrow q \in G,$$

and directed toward stronger conditions,

$$p, q \in G \Longrightarrow \exists r \in G\ (r \leq p, q).$$

Generic filter. If M is the ground model, G is M-generic when it meets every set $D \in M$ that M regards as a dense subset of $\mathbb{P}$.

A forcing notion is a preorder/partial order $(\mathbb{P}, \leq)$ with a largest (weakest) condition $1_{\mathbb{P}}$, and $q \leq p$ means that q carries at least as much information. If a poset is initially given without a largest condition, we adjoin one; this does not change its generic extensions. Two conditions are compatible when they have a common strengthening. A set D is dense if every condition has a strengthening in D. A filter is directed and upward closed in the weaker-condition direction. An M-generic filter meets every dense subset of $\mathbb{P}$ that belongs to M.

How this chapter fits together

Forcing starts with an order convention: stronger conditions carry more information. Dense sets encode requirements that can always be met by extending a condition. Over an externally countable model, Rasiowa–Sikorski recursively meets all dense sets from that model, producing a generic filter. This countable-model construction is pedagogically central but later separated from the formal relative-consistency argument.

Proposition 44.1 (Equivalent forcing-order conventions and separativity basics)**.** *A forcing notion is a preorder $(\mathbb{P}, \leq, 1_{\mathbb{P}})$ with a largest condition, where $q \leq p$ means that q is stronger than p. Define*

$$p \leq^* q \quad \Longleftrightarrow \quad \forall r \leq p\ \exists s\, (s \leq r \text{ and } s \leq q).$$

Then $p \equiv q \iff (p \leq^* q \wedge q \leq^* p)$ is an equivalence relation, and the quotient ordered by $[p] \leq_{\text{sep}} [q] \iff p \leq^* q$ is separative and densely forcing equivalent to $\mathbb{P}$.

Proof.

Formal derivation spine

Formal spine: the separative preorder and its quotient.

$$p \leq^* q \iff \forall r \leq p\, \exists s \leq r\, (s \leq q), \qquad p \equiv q \iff p \leq^* q \;\&\; q \leq^* p.$$

The quotient order is $[p] \leq_{\mathrm{sep}} [q] \iff p \leq^* q$.

We verify the order-theoretic points rather than treating the quotient as notation. Reflexivity of $\leq^*$ follows by taking $s = r$. For transitivity, assume $p \leq^* q$ and $q \leq^* t$, and fix $r \leq p$. Choose $s \leq r, q$ from the first assumption. Since $s \leq q$, the second assumption gives $u \leq s, t$. Then $u \leq r, t$, so $p \leq^* t$.

Consequently $\equiv$ is reflexive and symmetric, and transitivity follows from transitivity of $\leq^*$. If $p \equiv p'$, $q \equiv q'$, and $p \leq^* q$, then

$$p' \leq^* p \leq^* q \leq^* q',$$

so the quotient order is well defined. Antisymmetry holds by the definition of the equivalence classes.

To prove separativity, suppose $[p] \not\leq_{\mathrm{sep}} [q]$. Then $p \not\leq^* q$, hence there is $r \leq p$ for which no $s \leq r, q$ exists. Thus $r \perp q$, and $[r] \leq_{\mathrm{sep}} [p]$ is incompatible with $[q]$. This is exactly the separativity condition.

Finally the quotient map $i(p) = [p]$ preserves order and incompatibility, and it is onto, hence dense. If H is generic for the quotient, then $i^{-1}[H]$ is a filter meeting every dense subset of $\mathbb{P}$; conversely the upward closure of $i[G]$ is quotient-generic whenever G is $\mathbb{P}$-generic. These translations preserve the recursive valuation of translated names. Therefore passing to the separative quotient changes neither the generic extension nor forcing truth. □

Theorem 44.2 (Rasiowa–Sikorski lemma)**.** *Let $\langle D_n : n < \omega \rangle$ be dense subsets of a poset $\mathbb{P}$, and let $p_0 \in \mathbb{P}$. There is a filter $G \subseteq \mathbb{P}$ with $p_0 \in G$ and $G \cap D_n \neq \varnothing$ for every n.*

Proof.

Formal derivation spine

Formal spine: descend through the dense sets and generate a filter.

$$p_{n+1} \leq p_n, \qquad p_{n+1} \in D_n, \qquad G := \{q : \exists n\, (p_n \leq q)\}.$$

Starting with the prescribed condition p_0, recursively choose

$$p_{n+1} \leq p_n \quad \text{with} \quad p_{n+1} \in D_n,$$

using density of D_n. Define

$$G := \{q \in \mathbb{P} : \exists n\, (p_n \leq q)\}.$$

Since $p_0 \leq p_0$, the initial condition lies in G. If $q \in G$ and $q \leq r$, choose n with $p_n \leq q$. Transitivity gives $p_n \leq r$, so $r \in G$; thus G is upward closed toward weaker conditions. If $q, r \in G$, choose m, n with $p_m \leq q$ and $p_n \leq r$. The descending sequence satisfies

$$p_{\max(m,n)} \leq q, r,$$

and this common strengthening belongs to G. Hence G is directed toward stronger conditions. Finally,

$$p_{n+1} \in G \cap D_n$$

for every n. The lemma uses only countable choice along the displayed sequence; model-theoretic countability enters later when the dense subsets of a countable model are enumerated. □

Theorem 44.3 (Existence of an M-generic filter over an externally countable model)**.** *If M is externally countable, $\mathbb{P} \in M$, and $M \models$ "$\mathbb{P}$ is a forcing notion", then for every $p \in \mathbb{P}^M$ there exists, in the ambient universe, an M-generic filter $G \subseteq \mathbb{P}^M$ with $p \in G$.*

Proof.

Formal derivation spine

Formal spine for Existence of an M-generic filter over an externally countable model.

$$\{D \in M : M \models \text{“}D \subseteq \mathbb{P} \text{ is dense}''\}$$

$$G \text{ is } M\text{-generic} \iff \forall D \in M\,[D \subseteq \mathbb{P} \text{ dense} \Rightarrow G \cap D \neq \varnothing].$$

Assume M is externally countable and $\mathbb{P} \in M$. Externally, the family

$$\{D \in M : M \models \text{“}D \subseteq \mathbb{P} \text{ is dense}''\}$$

is countable, because it is a subset of the countable set M. Enumerate it as $D_0, D_1, \ldots$. Applying the Rasiowa–Sikorski construction to this enumeration produces a filter $G \subseteq \mathbb{P}$ meeting every D_n, hence every dense subset of $\mathbb{P}$ that belongs to M. Therefore G is M-generic.

The word “externally” is essential. Nothing in the argument asserts that M itself contains an enumeration of all its dense subsets, and the existence of such a countable transitive model is not inferred merely from Con(ZFC). *Why the displayed family is countable externally.* Every dense set used in the construction is itself an element of M. Since the surrounding universe sees M as countable, it can enumerate those elements of M that happen to satisfy the internal formula “dense subset of $\mathbb{P}$”. The enumeration is a metatheoretic tool and need not be an element of M. □

Proposition 44.4 (Genericity as meeting all ground-model dense sets)**.** *A filter $G \subseteq \mathbb{P}^M$ is M-generic exactly when*

$$\forall D \in M\,[M \models \text{“}D \subseteq \mathbb{P} \text{ is dense”} \Rightarrow G \cap D \neq \varnothing].$$

Proof.

Formal derivation spine

Formal spine for Genericity as meeting all ground-model dense sets.

$$G \cap D \neq \varnothing$$

$$G \text{ is } M\text{-generic} \iff \forall D \in M\,[D \subseteq \mathbb{P} \text{ dense} \Rightarrow G \cap D \neq \varnothing].$$

This is the intrinsic characterization used throughout the volume. A filter $G \subseteq \mathbb{P}$ is M-generic exactly when

$$G \cap D \neq \varnothing$$

for every dense $D \subseteq \mathbb{P}$ with $D \in M$. If “dense” is replaced by “predense”, the condition is equivalent: from a predense set take its downward closure, which is dense, and conversely every dense set is predense.

The formulation explains why genericity depends on the chosen ground model. A filter may meet all dense sets coded in one model while failing to meet a dense set appearing in a larger model. Later forcing arguments always specify the ground model before speaking of a generic filter. □

Worked Theorem 44.5 (Construction of a generic filter for a toy forcing)**.** *For the forcing of finite partial functions $p : \omega \rightharpoonup 2$, meeting the dense sets $D_n = \{p : n \in \operatorname{dom} p\}$ produces a filter G whose union $g = \bigcup G$ is a total element of 2^ω; adding further dense requirements yields an explicit toy generic construction.*

Proof.

Formal derivation spine

Formal spine for Construction of a generic filter for a toy forcing.

$$D_n = \{p : n \in \operatorname{dom} p\}.$$

$$E_x = \{p : \exists n \in \operatorname{dom} p\ (p(n) \neq x(n))\}.$$

$$G \text{ is } M\text{-generic} \iff \forall D \in M\, [D \subseteq \mathbb{P} \text{ dense} \Rightarrow G \cap D \neq \varnothing].$$

Take $\mathbb{P} = \mathrm{Fn}(\omega, 2, <\omega)$, ordered by reverse inclusion. For each n, let

$$D_n = \{p : n \in \operatorname{dom} p\}.$$

Each D_n is dense, because any finite condition can be extended by assigning a value to a previously undecided coordinate. Meeting all D_n's makes the union of the filter a total binary sequence.

To see how genuinely generic requirements enter, fix a ground-model real $x \in 2^\omega$ and let

$$E_x = \{p : \exists n \in \operatorname{dom} p\ (p(n) \neq x(n))\}.$$

This set is dense: extend any condition at a fresh coordinate with the opposite bit. A filter meeting every E_x for ground-model x has a union different from every old real. Thus the toy forcing already exhibits the central mechanism of Cohen forcing. □

Worked example

Worked Example 44.6 (Meeting bit-deciding dense sets)**.** For finite partial functions $p : \omega \to 2$, let $D_n = \{p : n \in \mathrm{dom}(p)\}$. Each D_n is dense because any condition can be extended by assigning a value to n. A filter meeting every D_n has a union that is a total binary sequence.

A useful warning

The statement "there exists an M-generic filter" is made externally when M is countable. It is not a theorem that an arbitrary model contains a filter generic over itself.

Looking ahead

The next chapter, *Names and valuation*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 45

Names and valuation

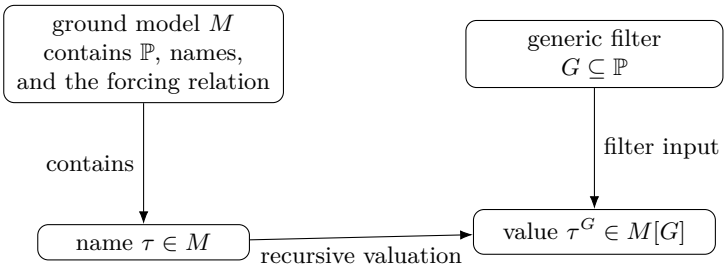

Figure 45.1: Names and the forcing relation are ground-model data. The generic filter is the additional input that turns the ground-model name τ into the extension object τ^G.

Definitions and setup

Formal definitions used in this chapter

Forcing name. A well-founded set of pairs $\langle \sigma, p \rangle$ with σ a lower-rank name and $p \in \mathbb{P}$.
Valuation. $\tau^G = \{\sigma^G : \exists p \in G\, (\sigma, p) \in \tau\}$.
Generic extension. $M[G] = \{\tau^G : \tau \in M\}$.

A forcing name is built recursively from pairs of lower-rank names and conditions. Its interpretation τ^G includes the interpretations of constituents whose attached conditions lie in the generic filter. Check names represent ground-model sets; $\dot{G}$ represents the generic filter. The generic extension $M[G]$ is the class of interpretations of names belonging to M.

How this chapter fits together

Names are well-founded recipes for objects that will appear after a generic filter is chosen. Their ranks ensure that valuation is a legitimate recursion. Check names recover ground-model objects, while $\dot{G}$ names the generic filter itself. This chapter explains how a universe is enlarged without creating new ordinals.

Theorem 45.1 (Recursive rank and construction of forcing names)**.** *Define the hierarchy of $\mathbb{P}$-names by*

$$V_0^{\mathbb{P}} = \varnothing, \qquad V_\alpha^{\mathbb{P}} = \{\tau : \tau \subseteq \bigcup_{\beta<\alpha} (V_\beta^{\mathbb{P}} \times \mathbb{P})\}.$$

Every name has a least rank $\operatorname{rank}(\tau)$*, and all names form* $V^{\mathbb{P}} = \bigcup_\alpha V_\alpha^{\mathbb{P}}$.

Proof.

Formal derivation spine

Formal spine for Recursive rank and construction of forcing names.

$$V_\alpha^{\mathbb{P}} = \big\{\tau : \tau \text{ is a set of pairs } \langle \sigma, p\rangle \text{ with } \sigma \in V_\beta^{\mathbb{P}} \text{ for some } \beta < \alpha,\ p \in \mathbb{P}\big\}.$$

$$\tau^G = \{\sigma^G : \exists p \in G\, \langle \sigma, p\rangle \in \tau\}.$$

Define the hierarchy of names by transfinite recursion:

$$V_\alpha^{\mathbb{P}} = \big\{\tau : \tau \text{ is a set of pairs } \langle \sigma, p\rangle \text{ with } \sigma \in V_\beta^{\mathbb{P}} \text{ for some } \beta < \alpha,\ p \in \mathbb{P}\big\}.$$

A name has rank one larger than the supremum of the ranks of names occurring as first coordinates of its pairs. Foundation makes this rank assignment well founded.

The point of the rank is not cosmetic. Every later recursive definition–valuation, atomic forcing, and induction on names–descends to constituent names of smaller rank. Thus the name hierarchy supplies the well-founded measure that prevents circularity in the forcing definitions.

A convenient convention is to permit a name to occur at every higher stage once it has appeared. Then the hierarchy is increasing, and rank may be taken as the least stage containing the name. None of the later arguments depends on this presentational choice; what matters is only that constituents have strictly smaller rank. □

Theorem 45.2 (Well-founded valuation tau -> tau^G)**.** *For a filter* $G \subseteq \mathbb{P}$*, valuation is the unique rank-recursive map*

$$\tau^G = \{\sigma^G : \exists p \in G\, (\langle \sigma, p\rangle \in \tau)\}.$$

How to read the symbols: a forcing name is code; valuation executes the code

A name τ is a ground-model set of pairs $\langle \sigma, p\rangle$. It is not yet an extension object. Once a generic filter G is given, valuation keeps exactly those pairs whose condition lies in G and recursively evaluates their lower-rank names. Thus $\tau \mapsto \tau^G$ is analogous to executing a recursively structured program with G as an external input.

Proof.

Formal derivation spine

Formal spine for Well-founded valuation tau -> tau^G.

$$\tau^G = \{\sigma^G : \exists p \in G\, (\langle \sigma, p\rangle \in \tau)\}.$$

$$\tau^G = \{\sigma^G : \exists p \in G\, \langle \sigma, p\rangle \in \tau\}.$$

For a filter $G \subseteq \mathbb{P}$, define valuation by recursion on name rank:

$$\tau^G = \{\sigma^G : \exists p \in G\, (\langle \sigma, p\rangle \in \tau)\}.$$

Every σ occurring on the right has smaller rank than τ, so the recursive value has already been defined. The well-founded recursion theorem therefore gives existence and uniqueness of the valuation map.

Induction on rank also shows that valuation ignores pairs whose conditions are not in G, and that two names can have the same value even when they are syntactically different. This is why equality between forcing names cannot be defined as literal equality of names and must instead receive its own forcing relation. *Reading the recursive clause.* The condition $\langle \sigma, p\rangle \in \tau$ is ground-model syntax. The additional test $p \in G$ decides

whether that branch of the name is active. Only after the branch is active do we use the already-defined lower-rank value σ^G. The rank decrease is the well-founded measure that prevents the definition from calling itself on the same input. □

Proposition 45.3 (Canonical ground-model names check(x))**.** *For each ground-model set x, define*

$$\check{x} = \{\langle \check{y}, 1_{\mathbb{P}} \rangle : y \in x\}.$$

Then for every generic G, $\check{x}^G = x$.

Proof.

Formal derivation spine

Formal spine for Canonical ground-model names check(x).

$$\check{x} = \{\langle \check{y}, 1_{\mathbb{P}} \rangle : y \in x\},$$

$$\check{x}^G = x.$$

$$\check{x}^G = \{\check{y}^G : y \in x\} = \{y : y \in x\} = x,$$

Define the canonical name of a ground-model set recursively by

$$\check{x} = \{\langle \check{y}, 1_{\mathbb{P}} \rangle : y \in x\},$$

where $1_{\mathbb{P}}$ is the weakest condition. We prove by induction on the rank of x that

$$\check{x}^G = x.$$

To verify the induction claim, expand the valuation definition for the specific name $\check{x}$. Every pair in $\check{x}$ carries the top condition, so every constituent survives valuation:

$$\check{x}^G = \{\check{y}^G : y \in x\} = \{y : y \in x\} = x,$$

using the induction hypothesis in the middle equality.

Thus every old set has a canonical name whose value is independent of the generic filter. This gives the literal embedding of the ground model into every generic extension.

The construction also preserves membership: $y \in x$ iff $\check{y}$ occurs in $\check{x}$ with the top condition. Thus check names reproduce the entire ground-model membership structure, not merely its underlying collection of objects. □

Proposition 45.4 (Canonical name dot(G))**.** *Define*

$$\dot{G} = \{\langle \check{p}, p \rangle : p \in \mathbb{P}\}.$$

Then for every generic filter G, $\dot{G}^G = G$.

Proof.

Formal derivation spine

Formal spine for Canonical name dot(G).

$$\dot{G} = \{\langle \check{p}, p \rangle : p \in \mathbb{P}\}.$$

$$\dot{G}^G = \{\check{p}^G : p \in G\} = \{p : p \in G\} = G,$$

$$\tau^G = \{\sigma^G : \exists p \in G\, \langle \sigma, p \rangle \in \tau\}.$$

Let

$$\dot{G} = \{\langle \check{p}, p \rangle : p \in \mathbb{P}\}.$$

When we evaluate this name, a pair survives exactly when its second coordinate p belongs to the chosen generic filter. Therefore

$$\dot{G}^G = \{\check{p}^G : p \in G\} = \{p : p \in G\} = G,$$

where the second equality uses the check-name theorem. Hence the generic filter itself is an element of the extension whenever the standard name $\dot{G}$ belongs to the ground model.

This name also clarifies the two viewpoints used in forcing: externally, G is chosen to meet the ground-model dense sets; internally to the extension, G is simply the value of the ground-model name $\dot{G}$.

Although G itself is new, the name $\dot{G}$ is old. This is a recurring forcing pattern: the ground model can contain a syntactic name for an object that exists only after a generic filter is supplied. □

Theorem 45.5 (M is contained in M[G]and M[G]is transitive)**.** *If M is transitive and G is M-generic, define*

$$M[G] = \{\tau^G : \tau \in M \text{ is a } \mathbb{P}\text{-name}\}.$$

Then $M \subseteq M[G]$, $G \in M[G]$, and $M[G]$ is transitive.

Proof.

Formal derivation spine

Formal spine for M is contained in M[G]and M[G]is transitive.

$$\tau^G = \{\sigma^G : \exists p \in G\, \langle \sigma, p\rangle \in \tau\}.$$

Every $x \in M$ belongs to $M[G]$ because $x = \check{x}^G$ and $\check{x} \in M$. Thus $M \subseteq M[G]$.

For transitivity, suppose $x \in y \in M[G]$. Choose $\tau \in M$ with $y = \tau^G$. By the definition of valuation, there are $\sigma \in M$ and $p \in G$ such that $\langle \sigma, p\rangle \in \tau$ and $x = \sigma^G$. Therefore $x \in M[G]$. So $M[G]$ is transitive.

The proof uses only that M is transitive enough for the names and their constituents to lie in M. It does not yet show that $M[G]$ satisfies any particular set-theoretic axiom; those preservation arguments come later.

The inclusion $M \subseteq M[G]$ is therefore canonical, not an arbitrary isomorphic copy. In later cardinal arguments this lets us compare the same ground-model ordinals and cardinals directly with their status in the extension. □

Worked example

Worked Example 45.6 (Evaluating a simple name)**.** If $\tau = \{\langle \check{0}, p\rangle, \langle \check{1}, q\rangle\}$, then τ^G contains 0 exactly when $p \in G$, and contains 1 exactly when $q \in G$. A name is therefore a condition-dependent recipe, not a fixed set waiting to be uncovered.

A useful warning

Two different names may have the same value in a generic extension. Forced equality is therefore an essential part of the forcing relation.

Looking ahead

The next chapter, *Atomic forcing*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 46

Atomic forcing

Definitions and setup

Formal definitions used in this chapter

Atomic forcing. The relations $p \Vdash \sigma \in \tau$ and $p \Vdash \sigma = \tau$ are defined simultaneously by recursion on name ranks using dense-below conditions.
Decision. A condition decides an atomic formula when it forces that formula or its negation.

Atomic forcing must be defined before semantic truth in the extension can be used. Membership and equality are defined simultaneously because each refers recursively to the other on lower-rank names. Monotonicity means stronger conditions preserve forced atomic facts. The atomic truth lemma is proved directly from genericity and name valuation and becomes the base case for the full Truth Lemma.

How this chapter fits together

Membership and equality of names must be defined simultaneously because each refers recursively to the other. The recursion decreases name rank, not formula complexity. Monotonicity and the atomic truth lemma then verify that the recursive definition matches actual valuation in a generic extension.

Theorem 46.1 (Simultaneous recursive definition of atomic forcing for equality and membership)**.** *There are unique atomic forcing relations, defined simultaneously by recursion on name rank, satisfying*

$$p \Vdash \sigma \in \tau \iff \{q \le p : \exists \langle \rho, r\rangle \in \tau \, (q \le r \wedge q \Vdash \sigma = \rho)\} \text{ is dense below } p,$$

$$p \Vdash \sigma \subseteq \tau \iff \forall \langle \rho, r\rangle \in \sigma \, \forall q \le p, r \, \exists s \le q \, (s \Vdash \rho \in \tau),$$

and

$$p \Vdash \sigma = \tau \iff p \Vdash \sigma \subseteq \tau \ \wedge \ p \Vdash \tau \subseteq \sigma.$$

Every recursive reference involves a constituent name of strictly lower rank.

Commented symbolic trace: why atomic forcing is defined simultaneously

Membership and equality of names refer to one another. To decide whether $p \Vdash \sigma \in \tau$, we compare σ with names occurring inside τ, so equality is needed. To decide whether $p \Vdash \sigma = \tau$, we require mutual forced membership, so membership is needed. The apparent circularity is removed by a simultaneous recursion on name ranks: every recursive call involves constituents of strictly smaller rank.

Proof.

Formal derivation spine

Formal spine: simultaneous recursion on pairs of name ranks. Order pairs of names by

$$(\sigma', \tau') \prec (\sigma, \tau) \iff \mathrm{rk}(\sigma') \leq \mathrm{rk}(\sigma), \quad \mathrm{rk}(\tau') \leq \mathrm{rk}(\tau),$$

with at least one strict inequality. This product order is well founded. Every atomic recursive call lies at a $\prec$-smaller pair.

Fix names σ, τ, and suppose the two atomic relations have already been defined for every smaller rank pair. In the membership clause, a recursive call has the form

$$q \Vdash \sigma = \rho \quad \text{with} \quad \langle \rho, r \rangle \in \tau.$$

Because ρ is a constituent of τ, $\mathrm{rk}(\rho) < \mathrm{rk}(\tau)$, so $(\sigma, \rho) \prec (\sigma, \tau)$.

For forced inclusion, a recursive call has the form

$$s \Vdash \rho \in \tau \quad \text{with} \quad \langle \rho, r \rangle \in \sigma.$$

Here $\mathrm{rk}(\rho) < \mathrm{rk}(\sigma)$, so $(\rho, \tau) \prec (\sigma, \tau)$. Thus the displayed membership and inclusion clauses depend only on already defined lower-rank cases. We then define equality at the current pair by

$$p \Vdash \sigma = \tau \iff p \Vdash \sigma \subseteq \tau \ \& \ p \Vdash \tau \subseteq \sigma.$$

Well-founded recursion on $\prec$ therefore gives existence. For uniqueness, suppose two pairs of atomic relations satisfy the same clauses and choose a $\prec$-least pair at which they differ. All recursive calls in either clause occur at smaller pairs, where the two definitions agree; hence the clauses assign the same truth value at the chosen pair, a contradiction. The atomic forcing relations are therefore uniquely determined. □

Lemma 46.2 (Monotonicity of atomic forcing)**.** *If $q \leq p$ and $p \Vdash \sigma = \tau$ (respectively $p \Vdash \sigma \in \tau$), then $q \Vdash \sigma = \tau$ (respectively $q \Vdash \sigma \in \tau$).*

Proof.

Formal derivation spine

Formal spine: strengthening preserves atomic forcing. For either atomic relation $\Phi(\sigma, \tau)$,

$$q \leq p \ \& \ p \Vdash \Phi(\sigma, \tau) \Longrightarrow q \Vdash \Phi(\sigma, \tau).$$

The dense set witnessing the clause below p remains dense after restricting to the cone below q.

We prove monotonicity simultaneously for atomic membership and equality by induction on the rank pair of the names. Assume $p \Vdash \sigma \in \tau$ and $q \leq p$. The definition says that below p the conditions witnessing an appropriate constituent ρ of τ and forcing $\sigma = \rho$ are dense. Any extension of q is also an extension of p, so the same set remains dense below q; hence $q \Vdash \sigma \in \tau$.

For equality, $p \Vdash \sigma = \tau$ means forced inclusion in both directions. Restricting from the cone below p to the smaller cone below q preserves each of those dense requirements, and the induction hypothesis applies to the lower-rank constituent equalities. Thus stronger conditions preserve atomic forcing. □

Lemma 46.3 (Atomic density lemma)**.** *For an atomic formula Φ in forcing names, let*

$$D_\Phi := \{p \in \mathbb{P} : p \Vdash \Phi \text{ or } \forall q \leq p \, (q \nVdash \Phi)\}.$$

Then D_Φ is dense. After the negation clause $p \Vdash \neg\Phi \iff \forall q \leq p \, (q \nVdash \Phi)$ is introduced, this says that conditions deciding Φ are dense.

Proof.

Formal derivation spine

Formal spine: one exhaustive dichotomy below p.

$$(\exists q \le p\ q \Vdash \Phi) \quad \vee \quad (\forall q \le p\ q \nVdash \Phi).$$

The first case supplies a forcing extension; the second puts p itself in D_Φ.

Fix $p \in \mathbb{P}$. There are two exhaustive cases.

If some $q \le p$ forces Φ, then $q \in D_\Phi$ and we have found a deciding extension.

Otherwise

$$\forall q \le p\ (q \nVdash \Phi).$$

Then $p \in D_\Phi$ by the second clause in the definition of D_Φ. Thus every condition has an extension in D_Φ, so D_Φ is dense. When general forcing is defined, the displayed universal condition is exactly the clause for $p \Vdash \neg\Phi$. The present lemma therefore establishes decision density without presupposing the full recursive forcing relation. □

Theorem 46.4 (Atomic truth lemma)**.** *If G is generic, then for names $\sigma, \tau \in M$,*

$$\sigma^G \in \tau^G \iff \exists p \in G\,(p \Vdash \sigma \in \tau),$$

and

$$\sigma^G = \tau^G \iff \exists p \in G\,(p \Vdash \sigma = \tau).$$

Proof.

Formal derivation spine

Formal spine for Atomic truth lemma.

$$\sigma^G \in \tau^G \iff \exists p \in G\ (p \Vdash \sigma \in \tau),$$

$$q \le p\ \wedge\ p \Vdash \Phi \Longrightarrow q \Vdash \Phi$$

We prove simultaneously that, for $\sigma, \tau \in M$ and an M-generic G,

$$\sigma^G \in \tau^G \iff \exists p \in G\ (p \Vdash \sigma \in \tau),$$

and likewise for equality.

For membership, if $\sigma^G \in \tau^G$, choose $\langle \rho, q\rangle \in \tau$ with $q \in G$ and $\rho^G = \sigma^G$. By the inductive equality statement some condition of G forces $\rho = \sigma$; directedness of G gives a common strengthening with q, yielding a condition in G that forces $\sigma \in \tau$. Conversely, if $p \in G$ forces membership, the dense witness set in the definition meets G; its constituent ρ has $\rho^G = \sigma^G$ by the inductive equality clause and $\rho^G \in \tau^G$ by valuation. Equality follows by applying the membership argument in both directions and ordinary extensionality of sets. *The induction has two mutually supporting components.* Membership needs the equality statement in order to replace the constituent ρ by σ; equality in turn is defined through mutual membership. Both references move to lower-rank names, so simultaneous induction is legitimate. This is the semantic counterpart of the simultaneous recursive definition of atomic forcing. □

Proposition 46.5 (Extensionality of the forcing relation on names)**.** *Forced equality is substitutive for atomic forcing: if $p \Vdash \sigma = \tau$, then below p the names σ and τ may be interchanged in atomic membership and equality formulas.*

Proof.

Formal derivation spine

Formal spine: substitution of forced-equal names. If $p \Vdash \sigma = \tau$ and $q \leq p$, then for every name ρ,

$$q \Vdash \rho \in \sigma \iff q \Vdash \rho \in \tau, \qquad q \Vdash \sigma \in \rho \iff q \Vdash \tau \in \rho,$$

and consequently

$$q \Vdash \rho = \sigma \iff q \Vdash \rho = \tau.$$

We prove simultaneously, by induction on the relevant name-rank tuple, that forced equality is an equivalence relation and is substitutive in both arguments of forced membership. Reflexivity is obtained by matching every constituent with itself; symmetry is built into the mutual-inclusion definition. For transitivity, assume $q \Vdash \sigma = \tau$ and $q \Vdash \tau = \upsilon$. A constituent of σ is densely matched with a constituent of τ, which is in turn densely matched with a constituent of υ. Directedness below q combines the two strengthenings, and the lower-rank induction hypothesis composes the two forced equalities. This proves $q \Vdash \sigma \subseteq \upsilon$; the reverse inclusion is symmetric.

Now assume $q \Vdash \rho \in \sigma$ and $q \Vdash \sigma = \tau$. Below an arbitrary $q_0 \leq q$, the first assertion supplies a stronger condition matching ρ with some constituent α of σ. Forced inclusion $\sigma \subseteq \tau$ then supplies a further strengthening forcing $\alpha \in \tau$. The lower-rank substitution hypothesis replaces α by the forced-equal name ρ, yielding a condition forcing $\rho \in \tau$. Since this can be done below every $q_0 \leq q$, the membership clause gives $q \Vdash \rho \in \tau$. The converse uses $\tau \subseteq \sigma$.

For membership in a third name, a witness for $q \Vdash \sigma \in \rho$ is a constituent α of ρ forced equal to σ. Transitivity of forced equality combines $\tau = \sigma$ with $\sigma = \alpha$, giving the same witness for $q \Vdash \tau \in \rho$. Finally, equality is mutual forced inclusion, so the two membership substitution laws give substitution in atomic equality formulas as well. □

Worked example

Worked Example 46.6 (Why equality and membership are simultaneous)**.** To decide whether $p \Vdash \sigma \in \tau$, one searches below p for a constituent ρ of τ forced equal to σ. But forced equality $\sigma = \rho$ itself says that every forced member of one is a forced member of the other. Neither notion can be defined first in isolation; the pair is handled by one well-founded recursion on ranks.

A useful warning

The informal clause "p forces membership iff some pair occurs in the name" is too weak. Compatibility, strengthening, and forced equality are needed to make the definition invariant under equivalent names.

Looking ahead

The next chapter, *General forcing relation*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 47

General forcing relation

Definitions and setup

Formal definitions used in this chapter

Forcing relation. After atomic forcing is defined, extend it by recursion on formula complexity. The load-bearing clauses are

$$p \Vdash \neg\varphi \iff \forall q \le p\ (q \nVdash \varphi),$$

$$p \Vdash (\varphi \wedge \psi) \iff (p \Vdash \varphi\ \wedge\ p \Vdash \psi),$$

and

$$p \Vdash \exists x\, \varphi(x, \bar\tau) \iff \{q \le p : \exists \sigma\ q \Vdash \varphi(\sigma, \bar\tau)\} \text{ is dense below } p.$$

Universal forcing is defined by $\forall x\, \varphi := \neg\exists x\, \neg\varphi$.

After atomic forcing is established, Boolean connectives and quantifiers are added recursively. Negation is defined by the absence of a stronger condition forcing the positive statement. A condition decides φ when it forces φ or its negation; deciding conditions are dense. The Definability Lemma asserts that $p \Vdash \varphi(\bar\tau)$ is a relation definable in the ground model from $p, \bar\tau$.

How this chapter fits together

After the atomic relation is secure, forcing extends by induction on formula complexity. Negation is formulated using all stronger conditions; quantifiers range over names; decision density guarantees that a generic filter encounters a condition settling each sentence. The Definability Lemma is crucial: forcing itself must be definable in the ground model before it can be used in ZF proofs there.

Theorem 47.1 (Recursive forcing clauses for Boolean connectives)**.** *Extend forcing recursively to Boolean connectives by*

$$p \Vdash \varphi \wedge \psi \iff (p \Vdash \varphi \wedge p \Vdash \psi), \qquad p \Vdash \neg\varphi \iff \forall q \le p\, (q \nVdash \varphi),$$

with the remaining connectives defined from these or by equivalent clauses.

Proof.

Formal derivation spine

Formal spine for Recursive forcing clauses for Boolean connectives.

$$p \Vdash (\varphi \wedge \psi) \iff p \Vdash \varphi\ \wedge\ p \Vdash \psi,$$

$$p \Vdash \neg\varphi \iff \forall q \le p\ (q \nVdash \varphi).$$

$$\varphi \to \psi := \neg(\varphi \wedge \neg\psi).$$

Once atomic forcing is fixed, define the Boolean clauses by recursion on formula complexity:

$$p \Vdash (\varphi \wedge \psi) \iff p \Vdash \varphi \ \wedge\ p \Vdash \psi,$$

$$p \Vdash \neg\varphi \iff \forall q \le p\ (q \nVdash \varphi).$$

Other propositional connectives are abbreviations, for example

$$\varphi \to \psi := \neg(\varphi \wedge \neg\psi).$$

The recursion is well founded because the forcing value of a compound formula refers only to proper subformulas. No quantifier clause is used in this theorem; those are introduced separately in the next result. *Specialization to this result.* For conjunction, force both conjuncts. For negation, require that no stronger condition force the positive formula. The clauses are chosen so that monotonicity and the Truth Lemma can be proved by induction. □

Theorem 47.2 (Forcing clauses for quantifiers)**.** *For existential formulas,*

$$p \Vdash \exists x\, \varphi(x, \bar\tau)$$

iff for every $q \le p$ there are $r \le q$ and a name σ with $r \Vdash \varphi(\sigma, \bar\tau)$. Universal forcing is defined dually via negation.

Proof.

Formal derivation spine

Formal spine for Forcing clauses for quantifiers.

$$p \Vdash \exists x\, \varphi(x, \bar\tau)$$

$$D := \{q \le p : \exists\sigma\ (q \Vdash \varphi(\sigma, \bar\tau))\}$$

$$\forall r \le p\, \exists q \le r\, \exists\sigma\ q \Vdash \varphi(\sigma, \bar\tau).$$

For an existential formula define

$$p \Vdash \exists x\, \varphi(x, \bar\tau)$$

iff the set

$$D := \{q \le p : \exists\sigma\ (q \Vdash \varphi(\sigma, \bar\tau))\}$$

is dense below p. Equivalently,

$$\forall r \le p\, \exists q \le r\, \exists\sigma\ q \Vdash \varphi(\sigma, \bar\tau).$$

Universal forcing is then defined through negation:

$$p \Vdash \forall x\, \varphi(x) \iff p \Vdash \neg\exists x\, \neg\varphi(x).$$

This density formulation is chosen so that a generic filter meeting D supplies an actual witnessing name in the Truth Lemma. *Specialization to this result.* Quantifiers range over forcing names. Existential truth is represented densely by conditions that force some named instance; after the maximum principle is established, this can be consolidated to a single witness name below the given condition. □

Lemma 47.3 (General monotonicity)**.** *For every forcing formula φ, if $q \le p$ and $p \Vdash \varphi$, then $q \Vdash \varphi$.*

Proof.

Formal derivation spine

Formal spine for General monotonicity.

$$p \Vdash \varphi \wedge \psi$$

$$q \leq p \ \wedge \ p \Vdash \varphi \Longrightarrow q \Vdash \varphi$$

$$p \Vdash \neg\varphi \iff \neg\exists q \leq p \ (q \Vdash \varphi)$$

Proceed by induction on formula complexity. The atomic case is the monotonicity theorem for atomic forcing. For conjunction,

$$p \Vdash \varphi \wedge \psi$$

implies both conjuncts are forced at p, so by induction they are forced at every $q \leq p$, hence $q \Vdash \varphi \wedge \psi$. For negation, if $p \Vdash \neg\varphi$ and $q \leq p$, then no $r \leq q$ can force φ, because such an r would also satisfy $r \leq p$. Thus $q \Vdash \neg\varphi$. For an existential formula, density below p remains density below a stronger $q \leq p$. Therefore

$$q \leq p \ \wedge \ p \Vdash \varphi \Longrightarrow q \Vdash \varphi$$

for all formulas. *Specialization to this result.* Induct on formula complexity. Strengthening preserves each atomic forcing clause by the atomic monotonicity theorem, and the Boolean/quantifier clauses are formulated so the induction passes immediately to stronger conditions. □

Theorem 47.4 (Density/decision lemma)**.** *For every formula $\varphi(\bar\tau)$ and condition p, there is $q \leq p$ such that either $q \Vdash \varphi(\bar\tau)$ or $q \Vdash \neg\varphi(\bar\tau)$. Thus the set of conditions deciding φ is dense.*

Proof.

Formal derivation spine

Formal spine for Density/decision lemma.

$$\neg\exists q \leq p \ (q \Vdash \varphi).$$

$$p \Vdash \neg\varphi.$$

$$D_\varphi := \{q : q \Vdash \varphi \text{ or } q \Vdash \neg\varphi\}$$

Fix $p \in \mathbb{P}$ and a formula φ. If some $q \leq p$ forces φ, then q already decides it. Otherwise

$$\neg\exists q \leq p \ (q \Vdash \varphi).$$

By the forcing clause for negation, this is exactly

$$p \Vdash \neg\varphi.$$

Therefore every condition has an extension deciding φ, and the set

$$D_\varphi := \{q : q \Vdash \varphi \text{ or } q \Vdash \neg\varphi\}$$

is dense. Notice that this argument uses the recursively defined negation clause; it is not an application of classical excluded middle inside the forcing extension. The density theorem is what allows genericity to turn forcing decisions into truth decisions later. *Specialization to this result.* Given p, either some extension forces φ, in which case descend to it, or no extension does, in which case p forces $\neg\varphi$. Hence conditions deciding φ are dense. □

Theorem 47.5 (Definability lemma for the forcing relation)**.** *Let $M \models \mathrm{ZF}$ be transitive and let $\mathbb{P} \in M$. For every formula $\varphi(\bar x)$ of set theory there is a first-order formula $F_\varphi(p, \bar\tau, \mathbb{P})$, constructed uniformly from φ, such that for all $p \in \mathbb{P}$ and $\mathbb{P}$-names $\bar\tau \in M$,*

$$M \models F_\varphi(p, \bar\tau, \mathbb{P}) \quad \iff \quad p \Vdash^M_{\mathbb{P}} \varphi(\bar\tau).$$

Thus the forcing relation is definable over the ground model, uniformly in the formula code.

Proof.

Formal derivation spine

Formal spine for Definability lemma for the forcing relation.

$$p \Vdash \neg\varphi \iff \neg\exists q \leq p\ (q \Vdash \varphi)$$

Proof map. Definability is proved inside the ground model by mirroring the external recursion. Atomic forcing is already a definable relation on names and conditions. Each Boolean clause is first-order definable from the previous stage, and quantification ranges over the set/class of ground-model names in the prescribed rank-bounded manner. This is what later lets Separation and Replacement in M form sets defined using $\Vdash$.

Extend forcing by recursion on formula complexity. Negation is $p \Vdash \neg\varphi$ iff no $q \leq p$ forces φ; conjunction is componentwise; existential forcing says that conditions below p densely force some instance by a name (equivalently use the maximum principle after it is proved). Monotonicity follows by induction. For every formula, the set of conditions deciding it is dense: if no extension forces φ, the current condition forces its negation; otherwise descend to one that forces φ. Because the atomic relation was defined in M and every recursive clause quantifies only over M-sets/names/conditions, induction yields a formula in M defining $p \Vdash \varphi(\bar{\tau})$. □

Worked example

Worked Example 47.6 (Forcing a negation)**.** A condition p forces $\neg\varphi$ precisely when no stronger condition forces φ. Thus if some $q \leq p$ forces φ, then p cannot force its negation. The dense set of conditions deciding φ guarantees that a generic filter cannot remain permanently undecided.

A useful warning

Existential forcing cannot be reduced naively to a single witness name until the maximum principle has been justified. Keep the dense-witness clause and the maximum principle logically ordered.

Looking ahead

The next chapter, *Truth and generic extension theorem*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 48

Truth and generic extension theorem

Definitions and setup

Formal definitions used in this chapter

Truth Lemma. Truth in $M[G]$ is equivalent to being forced by some condition in G.
Forcing Theorem. Combines definability of $\Vdash$ in M with the Truth Lemma.

The Truth Lemma states $M[G] \models \varphi(\bar{\tau}^G)$ iff some condition in G forces $\varphi(\bar{\tau})$. Together with the Definability Lemma it is the forcing theorem. The maximum principle packages locally forced existential witnesses into one name, and the mixing lemma pastes names along an antichain. These are proof tools, not additional semantic assumptions.

How this chapter fits together

The Truth Lemma is the central semantic theorem of forcing: a sentence is true in $M[G]$ exactly when some condition in G forces it. The negation and existential steps are the bottlenecks. Definability plus the Truth Lemma form the Fundamental Theorem of Forcing, after which maximum and mixing turn local decisions along antichains into single global names.

Theorem 48.1 (Truth lemma)**.** *If M is transitive, G is M-generic, and $\bar{\tau} \in M$, then*

$$M[G] \models \varphi(\bar{\tau}^G) \quad \Longleftrightarrow \quad \exists p \in G\, (p \Vdash \varphi(\bar{\tau})).$$

Uses: the atomic Truth Lemma, the recursive forcing clauses, decision density, and genericity.

How to read the symbols: the Truth Lemma is the semantic bridge

The statement

$$M[G] \models \varphi(\bar{\tau}^G) \quad \Longleftrightarrow \quad \exists p \in G\; p \Vdash \varphi(\bar{\tau})$$

connects three layers: names in M, conditions in $\mathbb{P}$, and actual objects in $M[G]$. The left side is ordinary truth in the extension. The right side is a ground-model relation about syntactic names and conditions. Genericity is what makes the two agree, because G meets the dense sets of conditions that decide the formula.

Proof.

Formal derivation spine

Formal spine for Truth lemma.

$$M[G] \models \varphi(\bar{\tau}^G) \iff \exists p \in G\, p \Vdash \varphi(\bar{\tau}).$$

$$M[G] \models \varphi \wedge \psi \iff \big(\exists p \in G\, p \Vdash \varphi\big) \wedge \big(\exists q \in G\, q \Vdash \psi\big).$$

$$r \Vdash \varphi \wedge \psi.$$

Proof map. For negation, decision density is the essential device: if the extension satisfies $\neg\varphi$, a generic condition deciding φ cannot force the positive side and therefore forces the negation. For an existential truth, first choose a witnessing extension object, represent it by a ground-model name, and then apply the induction hypothesis to that instance.

Induct on formulas. Atomic formulas were proved separately. Boolean cases use the inductive hypothesis and the fact that G is a filter. For negation, if $M[G] \models \neg\varphi$ and no condition in G forces the negation, decision density supplies a condition in G forcing φ, contradiction. For an existential truth choose a witnessing element τ^G; by induction some $p \in G$ forces the instance, and the existential forcing clause gives the result. Conversely, any condition in G forcing a formula makes it true by the same induction. Therefore

$$M[G] \models \varphi(\bar{\tau}^G) \iff \exists p \in G\, p \Vdash \varphi(\bar{\tau}).$$

Together with definability this is the forcing theorem.

Detailed formal verification. We prove the equivalence by induction on the complexity of φ. The atomic cases are exactly the Atomic Truth Lemma. For conjunction, the induction hypothesis gives

$$M[G] \models \varphi \wedge \psi \iff \big(\exists p \in G\, p \Vdash \varphi\big) \wedge \big(\exists q \in G\, q \Vdash \psi\big).$$

Directedness of G supplies $r \in G$ with $r \leq p, q$, and monotonicity gives

$$r \Vdash \varphi \wedge \psi.$$

For negation, suppose $M[G] \models \neg\varphi$. The set deciding φ is dense, so genericity gives $p \in G$ deciding it. The case $p \Vdash \varphi$ contradicts the induction hypothesis; hence

$$p \Vdash \neg\varphi.$$

For the existential step, if $M[G] \models \exists x\, \psi(x, \bar{\tau}^G)$, choose a witness $a = \sigma^G$. By induction, some $p \in G$ forces $\psi(\sigma, \bar{\tau})$, so the existential clause gives $p \Vdash \exists x\, \psi(x, \bar{\tau})$. The converse direction follows from the density clause and genericity. Thus

$$M[G] \models \varphi(\bar{\tau}^G) \iff \exists p \in G\, p \Vdash \varphi(\bar{\tau}).$$

□

Theorem 48.2 (Fundamental theorem of forcing)**.** *Let M be a countable transitive model of ZF in the ambient universe, let $\mathbb{P} \in M$, and let $p \in \mathbb{P}$. For every formula $\varphi(\bar{x})$ and names $\bar{\tau} \in M$,*

$$p \Vdash^M_{\mathbb{P}} \varphi(\bar{\tau}) \quad \iff \quad \forall G \subseteq \mathbb{P}\, \big(G \text{ is } M\text{-generic},\ p \in G \Rightarrow M[G] \models \varphi(\bar{\tau}^G)\big).$$

Together with the Definability Lemma, this is the Fundamental Theorem of Forcing in the countable-transitive-model presentation.

Proof.

Formal derivation spine

Formal spine: Definability plus the Truth Lemma.

$$p \Vdash^M \varphi(\bar{\tau}) \Longrightarrow \forall G \ni p\ M[G] \models \varphi(\bar{\tau}^G).$$

If $p \nVdash^M \varphi$, choose $q \leq p$ with $q \Vdash^M \neg\varphi$ and an M-generic $G \ni q$.

Definability was proved in the preceding chapter and the Truth Lemma in the preceding theorem. If $p \Vdash \varphi(\bar{\tau})$ and G is M-generic with $p \in G$, the forward direction of the Truth Lemma gives

$$M[G] \models \varphi(\bar{\tau}^G).$$

Conversely, suppose $p \nVdash \varphi(\bar{\tau})$. By decision density there is $q \leq p$ with

$$q \Vdash \neg\varphi(\bar{\tau}).$$

Because M is externally countable, Rasiowa–Sikorski supplies an M-generic filter G containing q, hence also containing the weaker condition p. The Truth Lemma now yields

$$M[G] \models \neg\varphi(\bar{\tau}^G).$$

Therefore it is not true that every generic extension through p satisfies φ. This proves the reverse implication. External countability is essential only for producing the counterexample generic in this semantic equivalence; the syntactic forcing relation and its Definability Lemma do not depend on that countability assumption. □

Remark 48.3 (Source note)**.** The definability and truth lemmas are the two components customarily packaged as the Forcing Theorem. The present proof uses the modern name-recursion organization; compare [11, 13, 14].

Proposition 48.4 (M and M[G]have the same ordinals)**.** *If M is transitive and G is M-generic, then*

$$\mathrm{Ord}^{M[G]} = \mathrm{Ord}^M.$$

Proof.

Formal derivation spine

Formal spine for M and M[G]have the same ordinals.

$$M \models p \Vdash \varphi(\bar{\tau}) \quad \longleftrightarrow \quad \text{a ground-model definable relation,}$$

$$M[G] \models \varphi(\bar{\tau}^G) \quad \longleftrightarrow \quad \exists p \in G\ p \Vdash \varphi(\bar{\tau}).$$

By transitivity $M[G]$ has the ordinary membership relation. Every ground-model ordinal remains an ordinal because $\check{\alpha}^G = \alpha$. Conversely, if $\gamma \in M[G]$ is an ordinal, induction on γ shows every $\beta < \gamma$ is a ground-model ordinal; the set of their ground ordinals is coded by a name in M, and the forcing rank/name-rank analysis yields a ground ordinal bounding γ. The usual minimal-counterexample argument then shows no genuinely new ordinal can appear. Thus forcing changes sets but not the class of ordinals.

Further explanation. A useful minimal-counterexample formulation is this. If a new ordinal existed in $M[G]$, choose the least one. Every smaller ordinal would then already lie in M, so the new ordinal would be a set of ground-model ordinals. The name producing it can be analyzed by rank and replaced by ground-model information coding precisely those smaller ordinals, which yields the same ordinal in M, a contradiction. Thus forcing may add new subsets of ordinals without adding new ordinals themselves. □

Theorem 48.5 (Maximum principle)**.** *Assume the ground model satisfies Choice. If*

$$p \Vdash \exists x\, \varphi(x, \bar{\tau}),$$

then there is a name σ such that

$$p \Vdash \varphi(\sigma, \bar{\tau}).$$

Proof.

Formal derivation spine

Formal spine for Maximum principle.

$$D = \{q \le p : \text{for some name } \tau,\ q \Vdash \varphi(\tau)\}.$$

$$M \models p \Vdash \varphi(\bar{\tau}) \quad \longleftrightarrow \quad \text{a ground-model definable relation,}$$

$$M[G] \models \varphi(\bar{\tau}^G) \quad \longleftrightarrow \quad \exists p \in G\ p \Vdash \varphi(\bar{\tau}).$$

Assume $p \Vdash \exists x\, \varphi(x)$. Let

$$D = \{q \le p : \text{for some name } \tau,\ q \Vdash \varphi(\tau)\}.$$

By the existential forcing clause, D is dense below p. Choose a maximal antichain $A \subseteq D$ below p, and for each $a \in A$ choose a name τ_a with $a \Vdash \varphi(\tau_a)$.

We now perform the mixing construction directly: trim each τ_a to conditions below a and take the union of the trimmed names. Call the resulting name τ. The atomic forcing induction shows that $a \Vdash \tau = \tau_a$ for every $a \in A$. Hence every $a \in A$ forces $\varphi(\tau)$. Since A is maximal below p, it is predense below p, and therefore $p \Vdash \varphi(\tau)$. Thus one name witnesses the existential statement already at p. □

Theorem 48.6 (Mixing lemma)**.** *Let A be a maximal antichain and let $\{\tau_a : a \in A\}$ be names. There is a name τ such that*

$$a \Vdash \tau = \tau_a \qquad (a \in A).$$

Proof.

Formal derivation spine

Formal spine for Mixing lemma.

$$\tau = \bigcup_{a \in A} \tau_a \restriction a.$$

$$a \Vdash \tau = \tau_a \qquad (a \in A).$$

$$M \models p \Vdash \varphi(\bar{\tau}) \quad \longleftrightarrow \quad \text{a ground-model definable relation,}$$

Let A be an antichain and let τ_a be a forcing name for each $a \in A$. For each a, replace τ_a by the equivalent name obtained by keeping only pairs whose conditions lie below a. Define

$$\tau = \bigcup_{a \in A} \tau_a \restriction a.$$

Because distinct members of A are incompatible, below a fixed a no piece coming from a different $b \in A$ can contribute to a generic valuation. Induction on constituent name rank therefore proves

$$a \Vdash \tau = \tau_a \qquad (a \in A).$$

This is the mixing lemma.

The previous maximum principle used exactly this construction. Here it is isolated as a reusable theorem: compatible local descriptions indexed by an antichain can be combined into one global forcing name. □

Worked example

Worked Example 48.7 (Using decision density in the negation step)**.** Assume $M[G] \models \neg\varphi$. If no condition in G forced $\neg\varphi$, then for each $p \in G$ there would be a stronger condition compatible with G that can force φ. The dense decision set meets G, yielding a condition in G that forces one side; the truth of $\neg\varphi$ rules out the positive side, so the generic condition must force the negation.

A useful warning

The forcing theorem has two halves: definability in the ground model and truth in the extension. Proving only the Truth Lemma is not enough for later Replacement arguments.

Looking ahead

The next chapter, *ZFC in generic extensions*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 49

ZFC in generic extensions

Definitions and setup

Formal definitions used in this chapter

Name rank and bounded names. Define

$$\operatorname{rank}_{\mathbb{P}}(\tau) = \sup\{\operatorname{rank}_{\mathbb{P}}(\sigma) + 1 : \exists p\, \langle \sigma, p\rangle \in \tau\}.$$

A name-bounding argument produces an ordinal $\theta \in M$ such that every witness or subset relevant to the given axiom has an equivalent name $\tau \in V_\theta^{\mathbb{P}} \cap M$.
Preservation of ZFC. Each axiom is verified separately in $M[G]$, rather than inferred from transitivity alone.

The statement $M[G] \models \mathrm{ZFC}$ is a list of axiom proofs. Transitivity gives Extensionality and Foundation; canonical names give Pairing, Union, and Infinity; and the Definability Lemma gives Separation. Collection and Replacement require a set-sized family of witness names. Power Set requires a bounded family of subnames representing all relevant subsets. Choice is inherited when the ground model has a well-order of the relevant name space. The hard axioms receive separate proofs.

How this chapter fits together

Preservation of ZFC is not one theorem with a one-line proof. Separation uses definability of forcing to carve a subname. Replacement/Collection requires a ground-model bound on witness names. Power Set requires a set-sized collection of names representing all subsets of a given interpreted set. Choice uses a ground-model well-order of suitable names. These constructions are treated separately.

Theorem 49.1 (Forcing preservation of the elementary ZF axioms)**.** *If $M \models \mathrm{ZF}$ is transitive, $\mathbb{P} \in M$, and $G \subseteq \mathbb{P}$ is M-generic, then $M[G]$ satisfies Extensionality, Empty Set, Foundation, Pairing, Union, and Infinity. Separation, Collection/Replacement, Power Set, and Choice are established separately in the following results.*

Proof.

Formal derivation spine

Formal spine for Forcing preservation of the elementary ZF axioms.

$$\dot{b}^G = \{x \in a : M[G] \models \varphi(x)\}.$$

We verify only the elementary ZF axioms named by this result. Extensionality and Foundation hold because $M[G]$ is a transitive class with the ambient membership relation. If $x = \sigma^G$ and $y = \tau^G$, the canonical ground-model name obtained by applying the relevant set operation to the constituent names with two constituents evaluates to $\{x, y\}$, proving Pairing. Given $x = \tau^G$, form in M the name that collects all constituents of constituents of τ; its value is $\bigcup x$, proving Union. Finally $\check{\omega}^G = \omega$, so Infinity holds.

These direct name constructions do not establish Separation, Collection/Replacement, or Power Set. Those axioms require the definability and bounding machinery developed in the forcing theorem and are proved separately below. □

Theorem 49.2 (Separation in M[G])**.** *If $M \models \mathrm{ZF}$, G is M-generic, $a = \tau^G \in M[G]$, and $\varphi(x, \bar{p})$ is a formula, then the subset*

$$\{x \in a : M[G] \models \varphi(x, \bar{p})\}$$

belongs to $M[G]$.

Proof.

Formal derivation spine

Formal spine for Separation in M[G].

$$\eta = \{\langle \sigma, q\rangle : \exists p\,(\langle \sigma, p\rangle \in \tau,\ q \le p,\ q \Vdash \varphi(\sigma, \bar{\rho}))\}.$$

$$\eta^G = \{u \in x : M[G] \models \varphi(u, \bar{a})\}.$$

$$\dot{b}^G = \{x \in a : M[G] \models \varphi(x)\}.$$

Let $x = \tau^G$ and let $\varphi(u, \bar{a})$ be a formula with parameters $\bar{a} = \bar{\rho}^G$. In the ground model define the name

$$\eta = \{\langle \sigma, q\rangle : \exists p\,(\langle \sigma, p\rangle \in \tau,\ q \le p,\ q \Vdash \varphi(\sigma, \bar{\rho}))\}.$$

The Definability Lemma makes the displayed property a ground-model definable relation, so Separation in M ensures that η is a set and a forcing name.

If $u \in \eta^G$, some condition in G forces both that u comes from τ and that $\varphi(u, \bar{a})$ holds; the Truth Lemma gives $u \in x$ and $\varphi(u, \bar{a})$. Conversely, if $u \in x$ and $M[G] \models \varphi(u, \bar{a})$, choose a name for u appearing in τ; the Truth Lemma and directedness of G give a stronger condition placing that name into η. Hence

$$\eta^G = \{u \in x : M[G] \models \varphi(u, \bar{a})\}.$$

□

Theorem 49.3 (Replacement/Collection in M[G])**.** *If $M \models \mathrm{ZF}$ and G is M-generic, then $M[G]$ satisfies Collection and Replacement: whenever a formula is total on a set $a \in M[G]$, there is a single ground-model name whose valuation contains witnesses for all elements of a.*

Uses: Definability, the Maximum Principle, the Truth Lemma, and ground-model Replacement.

Commented symbolic trace: Replacement in a generic extension

A functional formula in $M[G]$ may produce witnesses represented by many different names. The proof works in M: for each possible input constituent, use the maximum principle below a maximal antichain to choose witness names, then use Replacement in M to collect those names into one set. That ground-model set of names is the rank/size bound needed to build a single name whose valuation contains every output. The extension's Replacement axiom is therefore inherited through bounded name bookkeeping, not by appealing to truth alone.

Proof.

Formal derivation spine

Formal spine for Replacement/Collection in M[G].

$$\dot{u} = \{\langle \tau, 1\rangle : \tau \in V_\delta^M\}.$$

$$M[G] \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p}). \tag{R1}$$

$$q \Vdash \varphi(\sigma, \rho_q, \bar{\pi}) \qquad (q \in A_{\sigma,r}). \tag{R2}$$

Proof map. Replacement needs a set of possible witness names before the generic is evaluated. For each relevant domain name and condition, take a maximal antichain deciding a witness; use the maximum principle to attach witness names; then use Replacement in the ground model to collect all of this set-sized data. A mixed/bounding name evaluates to a set containing every required image.

Let $z = \dot{z}^G \in M[G]$ and suppose $M[G] \models \forall x \in z\, \exists y\, \varphi(x, y, \bar{a})$, with parameters named by $\dot{a}_i \in M$. For each constituent $\langle \rho, s\rangle \in \dot{z}$ and each $p \in \mathbb{P}$, let $f(\rho, s, p)$ be the least ordinal α such that some name $\tau \in V_\alpha^M$ satisfies $p \Vdash \varphi(\rho, \tau, \bar{a})$, when such a name exists, and put 0 otherwise. Definability of forcing makes f definable in M. Replacement in M bounds all these ordinals by one δ. Let

$$\dot{u} = \{\langle \tau, 1\rangle : \tau \in V_\delta^M\}.$$

If $x = \rho^G \in z$, choose a witness $y = \tau^G$ in the extension and a condition $p \in G$ forcing the relevant instance, by the Truth Lemma. By definition of δ an equivalent witness name can be chosen in V_δ^M, so $y \in \dot{u}^G$. Thus $M[G]$ satisfies Collection. For a functional φ, Separation from $\dot{u}^G$ gives the exact range, hence Replacement.

Detailed formal verification. Let $a = \tau^G$ and assume

$$M[G] \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p}). \tag{R1}$$

For each pair $\langle \sigma, r\rangle \in \tau$, work below r. By the Truth Lemma and the maximum principle, choose a maximal antichain $A_{\sigma,r}$ below r and names ρ_q such that

$$q \Vdash \varphi(\sigma, \rho_q, \bar{\pi}) \qquad (q \in A_{\sigma,r}). \tag{R2}$$

All indices in (R2) range over sets in M. Replacement in M therefore collects the witness names into a set $W \in M$. Form the bounding name

$$\dot{b} := \{\langle \rho, 1\rangle : \rho \in W\}.$$

If $x = \sigma^G \in a$, genericity meets the relevant antichain $A_{\sigma,r}$, so for some $q \in G$, (R2) gives a witness $\rho_q^G \in \dot{b}^G$. Hence

$$M[G] \models \forall x \in a\, \exists y \in \dot{b}^G\, \varphi(x, y, \bar{p}),$$

which is Collection. If φ is functional, Separation inside $\dot{b}^G$ extracts the exact image, giving Replacement. □

Remark 49.4 (Source note)**.** Preservation of Replacement is treated separately from the Truth Lemma because it requires a ground-model set bound on possible witness names; compare [13, 14].

Theorem 49.5 (Power Set in M[G])**.** *If $M \models$ ZF, G is M-generic, and $a \in M[G]$, then*

$$\mathcal{P}^{M[G]}(a) = \{x \in M[G] : x \subseteq a\}$$

is a set belonging to $M[G]$.

Uses: the Truth Lemma, normalization of subset names, ground-model Power Set, and Separation already proved in $M[G]$.

How to read the symbols: Power Set means internal subsets

To prove Power Set for $x = \tau^G$, we need a set in $M[G]$ containing exactly the subsets of x that exist in $M[G]$. Arbitrary names can be replaced, modulo forced equality, by names of bounded rank whose pairs use constituents from τ. Because the bounded family of candidate names is a set in M, it has a name; its valuation is the internal power set. As with L, "Power Set" is always relative to the model under discussion.

Proof.

Formal derivation spine

Formal spine: normalize every subset name into one ground-model set of candidates.

$$\mathrm{Sub}_{\dot a}(\dot y) = \{\langle \sigma, p\rangle \in \mathrm{dom}(\dot a) \times \mathbb{P} : p \Vdash \sigma \in \dot y\},$$

$$S = \mathcal{P}^M(\mathrm{dom}(\dot a) \times \mathbb{P}), \qquad \mathcal{P}^{M[G]}(a) = \{z \in \dot U^G : z \subseteq a\}.$$

Let $a = \dot a^G \in M[G]$, with $\dot a \in M$. We first normalize names for subsets of a.

Suppose $y = \dot y^G \in M[G]$ and $y \subseteq a$. Define in M

$$\mathrm{Sub}_{\dot a}(\dot y) := \{\langle \sigma, p\rangle \in \mathrm{dom}(\dot a) \times \mathbb{P} : p \Vdash \sigma \in \dot y\}. \tag{1}$$

Definability of forcing and Separation in M show that (1) is a name in M. We claim

$$\mathrm{Sub}_{\dot a}(\dot y)^G = y. \tag{2}$$

If z belongs to the left side, then $z = \sigma^G$ for some $p \in G$ with $p \Vdash \sigma \in \dot y$; the Truth Lemma gives $z \in y$. Conversely, if $z \in y \subseteq a$, choose $\sigma \in \mathrm{dom}(\dot a)$ and $r \in G$ with $z = \sigma^G$ and $\langle \sigma, r\rangle \in \dot a$. Since $M[G] \models \sigma^G \in \dot y^G$, the Truth Lemma gives $p \in G$ with $p \Vdash \sigma \in \dot y$, so z belongs to the left side of (2).

Now

$$S := \mathcal{P}^M(\mathrm{dom}(\dot a) \times \mathbb{P})$$

is a set in M. Let

$$\dot U := \{\langle \tau, 1_{\mathbb{P}}\rangle : \tau \in S\}.$$

By (2), every subset of a that belongs to $M[G]$ is an element of $\dot U^G$. Conversely $\dot U^G$ may contain values that are not subsets of a, so use the already established Separation axiom in $M[G]$ to form

$$b := \{z \in \dot U^G : z \subseteq a\}.$$

Then

$$b = \mathcal{P}^{M[G]}(a).$$

Thus the internal power set is a set in $M[G]$. The argument uses the ground-model Power Set axiom only to collect the set of normalized candidate names; it does not assume that all ambient subsets of a belong to the extension. □

Remark 49.6 (Source note)**.** Power Set likewise needs bounded normalization of subset names; transitivity of the extension alone is not enough. Compare [13, 14].

Theorem 49.7 (Choice in M[G]when M satisfies Choice)**.** *If $M \models$ ZFC and G is M-generic, then $M[G] \models$ AC. Hence ordinary set forcing over a transitive ZFC ground model produces another model of ZFC.*

Proof.

Formal derivation spine

Formal spine for Choice in M[G]when M satisfies Choice.

$$D = \{\sigma : \exists p\, (\langle \sigma, p\rangle \in \tau)\}$$

$$y <^* z \quad \Longleftrightarrow \quad r(y) \lhd r(z).$$

$$\dot{b}^G = \{x \in a : M[G] \models \varphi(x)\}.$$

Let $x = \tau^G \in M[G]$. The ground-model set

$$D = \{\sigma : \exists p\, (\langle \sigma, p\rangle \in \tau)\}$$

contains a name for every element of x. Since $M \models \mathrm{AC}$, choose in M a well-order $\lhd$ of D. In the extension, for each $y \in x$, let $r(y)$ be the $\lhd$-least $\sigma \in D$ with $\sigma^G = y$. Such a representative exists by the definition of τ^G.

Define

$$y <^* z \quad \Longleftrightarrow \quad r(y) \lhd r(z).$$

Because the representatives are chosen uniquely, $<^*$ is a well-order of x. The relation is a set in the extension by Separation/Replacement, already established there. Thus every set of $M[G]$ can be well ordered, so $M[G] \models \mathrm{AC}$. This proof uses Choice only in the ground model to well-order the set of candidate names. □

Worked example

Worked Example 49.8 (Bounding names for Replacement)**.** Suppose p forces that every member of τ has a unique φ-image. For each potential member name σ and each relevant condition, choose along a maximal antichain witness names for the image. Because τ and the antichains are sets in M, Replacement in M collects all selected names into one set. A mixed name built from them then bounds the range in $M[G]$.

A useful warning

Transitivity of $M[G]$ does not by itself prove Replacement or Power Set. Both axioms require explicit set-sized name bounds.

End-of-volume perspective

The forcing theorem turns conditions into reliable semantic information about $M[G]$. The next volume replaces dependence on an externally chosen generic with Boolean-valued semantics and then extracts formal relative-consistency conclusions.

Part VIII

Volume VIII: Forcing II: Boolean Values, Preservation, and Relative Consistency

Preface

This volume supplies the metatheoretic bridge that generic-extension pedagogy alone does not give. Boolean-valued semantics and exact preservation results are separated from countable-model intuition.

Position in the series

Volume VIII separates the metatheoretic consistency argument from countable-transitive-model intuition. Boolean-valued semantics, chain conditions, and nice names provide the transfer and preservation tools used in the final two volumes.

Sources and further reading

Boolean-valued semantics is used to separate the model-theoretic generic-extension picture from the formal relative-consistency conclusion. See [13, 14] for standard developments.

Notation for this volume

$\mathbb{B}$ is a complete Boolean algebra and $\|\varphi\| \in \mathbb{B}$ is the Boolean truth value. Ground-model cardinals carry a superscript when preservation has not yet been proved.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 50

Separative quotients and Boolean completion

Definitions and setup

Formal definitions used in this chapter

Separative quotient. Conditions with identical extension/compatibility behavior are identified.
Regular-open completion. RO($\mathbb{P}$) is the complete Boolean algebra of regular open subsets of the separative forcing.

Two forcing conditions are separatively equivalent when no extension can distinguish their compatibility behavior. The separative quotient removes redundant conditions. The regular-open algebra RO($\mathbb{P}$) is a complete Boolean algebra densely containing the separative forcing. Dense embeddings preserve genericity and forcing truth, allowing poset forcing to be translated into Boolean-valued semantics.

How this chapter fits together

Different forcing posets can encode the same extension. Passing to the separative quotient removes distinctions that no generic filter can detect, and the regular-open completion embeds the forcing densely into a complete Boolean algebra. Dense embeddings preserve generic extensions, making Boolean-valued semantics a canonical reformulation rather than a different forcing method.

Theorem 50.1 (Separative quotient theorem)**.** *For a forcing preorder $\mathbb{P}$, define $p \equiv q$ iff $p \leq^* q$ and $q \leq^* p$. Then $\mathbb{P}/\equiv$, ordered by $[p] \leq [q] \iff p \leq^* q$, is separative, and the quotient map is a dense forcing equivalence.*

Proof.

Formal derivation spine

Formal spine for Separative quotient theorem.

$$p \equiv q \quad \iff \quad p \preccurlyeq q \;\&\; q \preccurlyeq p.$$

$$p \perp q \iff i(p) \wedge i(q) = 0$$

Define $p \preccurlyeq q$ to mean that every extension of p is compatible with q, and set

$$p \equiv q \quad \iff \quad p \preccurlyeq q \;\&\; q \preccurlyeq p.$$

This is an equivalence relation. Order equivalence classes by $[p] \le_s [q]$ iff $p \preccurlyeq q$. The definition is independent of representatives because replacing either condition by an equivalent one preserves the compatibility cones.

The quotient is separative: if $[p] \not\le_s [q]$, then some extension $r \le p$ is incompatible with q, and $[r] \le_s [p]$ is incompatible with $[q]$. The quotient map has dense image because every condition has a stronger condition represented in the dense image and does not change which filters meet which dense requirements after translating through equivalence classes. Thus every forcing notion may first be replaced by a separative one without changing its forcing content. □

Theorem 50.2 (Regular-open Boolean completion of a forcing notion)**.** *For every separative forcing notion $\mathbb{P}$, the regular-open algebra $\mathrm{RO}(\mathbb{P})$ is a complete Boolean algebra and the map sending p to the regular open generated by its cone is a dense embedding $\mathbb{P} \to \mathrm{RO}(\mathbb{P})^+$.*

Proof.

Formal derivation spine

Formal spine for Regular-open Boolean completion of a forcing notion.

$$A^\perp = \{p : \text{no } q \le p \text{ is compatible with any member of } A\},$$

$$p \perp q \iff i(p) \wedge i(q) = 0$$

Assume $\mathbb{P}$ is separative. For $A \subseteq \mathbb{P}$, write

$$A^\perp = \{p : \text{no } q \le p \text{ is compatible with any member of } A\},$$

and let $\mathrm{ro}(A) = (A^\perp)^\perp$. The regular-open subsets of $\mathbb{P}$, ordered by inclusion, form a complete Boolean algebra $\mathrm{RO}(\mathbb{P})$. Arbitrary joins are obtained by regularizing unions; meets are intersections; Boolean complement is $A^\perp$.

Map a condition p to the regular open set generated by its cone $\{q : q \le p\}$. Separativity makes this map order reflecting, and every nonzero regular open set contains a cone below some condition, so its image is dense. Therefore $\mathrm{RO}(\mathbb{P})$ is a complete Boolean algebra into which $\mathbb{P}$ embeds densely–the Boolean completion of the forcing. □

Proposition 50.3 (Dense embeddings preserve generic extensions)**.** *If $i : \mathbb{P} \to \mathbb{Q}$ is a dense embedding, then generics translate in both directions and the induced translation of names yields isomorphic generic extensions.*

Proof.

Formal derivation spine

Formal spine for Dense embeddings preserve generic extensions.

$$G = \{p \in \mathbb{P} : \exists q \in H \ (q \le i(p))\}.$$

$$p \perp q \iff i(p) \wedge i(q) = 0$$

Let $i : \mathbb{P} \to \mathbb{Q}$ be a dense embedding. If $H \subseteq \mathbb{Q}$ is generic over M, define

$$G = \{p \in \mathbb{P} : \exists q \in H \ (q \le i(p))\}.$$

Density and order preservation show that G is a filter. If $D \in M$ is dense in $\mathbb{P}$, then the downward closure of $i[D]$ is dense in $\mathbb{Q}$; genericity of H therefore implies $G \cap D \neq \varnothing$. Thus G is $\mathbb{P}$-generic.

Conversely a $\mathbb{P}$-generic filter generates a unique $\mathbb{Q}$-generic filter meeting the dense image. Translate names recursively along i; induction on name rank shows that corresponding names have the same valuations. Hence the two generic extensions are identical after the canonical translation. *What the translation preserves.* Dense embedding does more than send filters to filters. The recursive translation of names commutes with valuation,

so a set constructed from a $\mathbb{P}$-name has exactly the same extension value as the corresponding $\mathbb{Q}$-name. Consequently forcing-equivalent posets may look different combinatorially while presenting the same generic universe. □

Proposition 50.4 (Forcing equivalence through Boolean completion)**.** *A forcing notion and its Boolean completion force exactly the same first-order statements after the canonical translation of names; in particular they have the same generic extensions up to the natural isomorphism.*

Proof.

Formal derivation spine

Formal spine for Forcing equivalence through Boolean completion.

$$p \Vdash_{\mathbb{P}} \varphi$$

$$p \perp q \iff i(p) \wedge i(q) = 0$$

Apply the dense-embedding theorem to the canonical dense embedding of the separative quotient of $\mathbb{P}$ into $\mathrm{RO}(\mathbb{P})$. Generics correspond in both directions, and the recursive translation of names preserves valuation. Therefore for every sentence φ and tuple of translated names,

$$p \Vdash_{\mathbb{P}} \varphi$$

holds exactly when the Boolean condition represented by p forces the translated formula in the completion.

Thus passing from a poset to its Boolean completion changes the algebraic language but not the generic extensions or the truths forced in them. This equivalence licenses the Boolean-valued semantics of the next chapter as a reformulation of the ordinary forcing developed in Volume VII. □

Worked example

Worked Example 50.5 (Two forcing-equivalent conditions)**.** If conditions p and q have exactly the same compatible extensions, no generic filter can distinguish their forcing behavior. The separative quotient identifies them. The regular-open algebra then replaces each condition by the regular open set generated by its cone, turning order-theoretic forcing into Boolean algebra.

A useful warning

Dense equivalence is about forcing behavior, not literal equality of posets. Names must be translated along the dense embedding before comparing extensions.

Looking ahead

The next chapter, *Boolean-valued semantics*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 51

Boolean-valued semantics

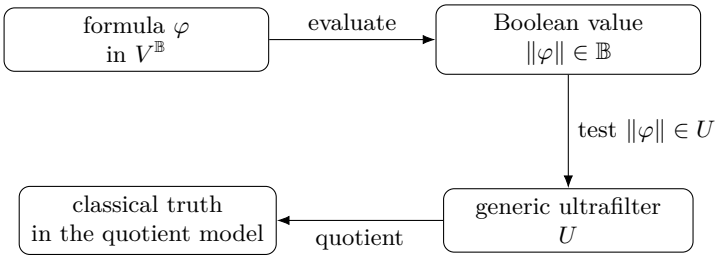

Figure 51.1: Boolean-valued semantics first assigns a Boolean element to a formula. A generic ultrafilter then converts that graded value into ordinary two-valued truth in the quotient model.

Definitions and setup

Formal definitions used in this chapter

Boolean-valued name. A function from lower-rank names to a complete Boolean algebra $\mathbb{B}$.
Boolean truth value. $\|\varphi\| \in \mathbb{B}$ is defined recursively, using Boolean operations for connectives and joins/meets for quantifiers.

A Boolean-valued universe $V^{\mathbb{B}}$ consists of names whose membership coefficients lie in a complete Boolean algebra $\mathbb{B}$. Instead of a formula being simply true or false, it receives a Boolean truth value $\|\varphi\| \in \mathbb{B}$. Equality and membership are defined simultaneously by rank recursion; connectives use Boolean operations and quantifiers use joins/meets. Truth value 1 plays the role of validity.

How this chapter fits together

A Boolean-valued universe assigns each atomic assertion a truth value in a complete Boolean algebra rather than merely 0 or 1. Equality and membership are again defined simultaneously by rank recursion; logical connectives become Boolean operations, and quantifiers become joins or meets. Ordinary forcing reappears after quotienting by a generic ultrafilter.

Theorem 51.1 (Construction of the Boolean-valued universe $V^{\mathbb{B}}$)**.** *For a complete Boolean algebra $\mathbb{B}$, define by recursion*

$$V_\alpha^{\mathbb{B}} = \{\tau : \operatorname{dom}(\tau) \subseteq \bigcup_{\beta<\alpha} V_\beta^{\mathbb{B}},\ \operatorname{ran}(\tau) \subseteq \mathbb{B}\},$$

and $V^{\mathbb{B}} = \bigcup_\alpha V_\alpha^{\mathbb{B}}$.

Proof.

Formal derivation spine

Formal spine for Construction of the Boolean-valued universe $V^{\mathbb{B}}$.

$$V_\alpha^{\mathbb{B}} = \{\tau : \tau \text{ is a function},\ \operatorname{ran}\tau \subseteq \mathbb{B},\ \operatorname{dom}\tau \subseteq \bigcup_{\beta<\alpha} V_\beta^{\mathbb{B}}\}.$$

$$\|\sigma \in \tau\| = \bigvee_{\rho\in\operatorname{dom}\tau} \tau(\rho) \wedge \|\sigma = \rho\|.$$

For a complete Boolean algebra $\mathbb{B}$, define by transfinite recursion

$$V_\alpha^{\mathbb{B}} = \{\tau : \tau \text{ is a function},\ \operatorname{ran}\tau \subseteq \mathbb{B},\ \operatorname{dom}\tau \subseteq \bigcup_{\beta<\alpha} V_\beta^{\mathbb{B}}\}.$$

A Boolean name therefore assigns to each possible lower-rank member a Boolean coefficient measuring the degree to which that member belongs. Set $V^{\mathbb{B}} = \bigcup_\alpha V_\alpha^{\mathbb{B}}$.

The recursion is set-sized at every stage, and name rank decreases through the domain relation. This makes the simultaneous definitions of Boolean membership and equality well founded. Ordinary forcing names appear as the special case in which the coefficients are Boolean values corresponding to forcing conditions.

The function presentation is equivalent to the set-of-pairs presentation used for ordinary forcing names: a pair $\langle\sigma, b\rangle$ records the coefficient $b = \tau(\sigma)$. Using functions is convenient because Boolean joins can combine multiple contributions to the same potential member into a single coefficient. □

Theorem 51.2 (Boolean truth values for equality and membership)**.** *For $\sigma, \tau \in V^{\mathbb{B}}$, define simultaneously*

$$\|\sigma \in \tau\| = \bigvee_{\rho\in\operatorname{dom}\tau} \tau(\rho) \wedge \|\sigma = \rho\|,$$

$$\|\sigma = \tau\| = \Big(\bigwedge_{\rho\in\operatorname{dom}\sigma} (\sigma(\rho) \to \|\rho \in \tau\|)\Big) \wedge \Big(\bigwedge_{\rho\in\operatorname{dom}\tau} (\tau(\rho) \to \|\rho \in \sigma\|)\Big).$$

How to read the symbols: Boolean truth values are graded truth certificates

The expression $\|\sigma \in \tau\|$ is an element of the Boolean algebra $\mathbb{B}$, not the classical number 0 or 1 in general. Its join runs over possible witnesses ρ in the domain of τ; each term combines the Boolean weight with which τ contains ρ and the Boolean degree to which $\sigma = \rho$. Classical truth is recovered after quotienting by a generic ultrafilter.

Proof.

Formal derivation spine

Formal spine: simultaneous recursion in the product of the two name ranks.

$$\|\sigma \in \tau\| = \bigvee_{\rho\in\operatorname{dom}\tau} \tau(\rho) \wedge \|\sigma = \rho\|,$$

$$\|\sigma = \tau\| = \|\sigma \subseteq \tau\| \wedge \|\tau \subseteq \sigma\|.$$

Every recursive call replaces at least one name by a constituent of strictly smaller rank.

Order pairs of names by the well-founded product relation

$$(\sigma', \tau') \prec (\sigma, \tau)$$

when

$$\operatorname{rk}(\sigma') \le \operatorname{rk}(\sigma), \qquad \operatorname{rk}(\tau') \le \operatorname{rk}(\tau),$$

with at least one strict inequality. In the membership clause, $\rho \in \operatorname{dom}(\tau)$ has $\operatorname{rk}(\rho) < \operatorname{rk}(\tau)$, so $(\sigma, \rho) \prec (\sigma, \tau)$. In the inclusion clause, a constituent ρ of σ gives a call at $(\rho, \tau) \prec (\sigma, \tau)$. Thus simultaneous recursion on $\prec$ defines

$$\|\sigma \in \tau\| = \bigvee_{\rho \in \operatorname{dom} \tau} \big(\tau(\rho) \wedge \|\sigma = \rho\|\big) \tag{1}$$

and

$$\|\sigma \subseteq \tau\| = \bigwedge_{\rho \in \operatorname{dom} \sigma} \big(\neg\sigma(\rho) \vee \|\rho \in \tau\|\big), \qquad \|\sigma = \tau\| = \|\sigma \subseteq \tau\| \wedge \|\tau \subseteq \sigma\|. \tag{2}$$

Completeness of $\mathbb{B}$ supplies the joins and meets. Uniqueness follows by taking a $\prec$-least pair where two proposed definitions differ; all right-hand recursive values then agree, so (1)–(2) assign the same value there, a contradiction.

A second induction on the same product rank proves

$$\|\sigma = \sigma\| = 1$$

and the substitutivity inequalities required by first-order equality. The use of the product relation is important: ordering only by the maximum of the two ranks would not make every equality call strictly smaller when the unchanged name has the larger rank. □

Theorem 51.3 (Boolean semantics for first-order formulas)**.** *Let $\mathbb{B}$ be a complete Boolean algebra. After the atomic Boolean values are defined, extend them recursively to all first-order formulas by*

$$\|\neg\varphi\| = \neg\|\varphi\|, \qquad \|\varphi \wedge \psi\| = \|\varphi\| \wedge \|\psi\|,$$

and

$$\|\exists x\, \varphi(x, \bar\tau)\| = \bigvee S_{\varphi,\bar\tau}, \qquad S_{\varphi,\bar\tau} = \{b \in \mathbb{B} : \exists\sigma\ (b = \|\varphi(\sigma, \bar\tau)\|)\}.$$

The set $S_{\varphi,\bar\tau} \subseteq \mathbb{B}$ exists by Separation in the metatheory, so the join exists by completeness of $\mathbb{B}$. Universal quantification is defined by

$$\|\forall x\, \varphi(x, \bar\tau)\| = \neg\|\exists x\, \neg\varphi(x, \bar\tau)\|.$$

These clauses assign a unique Boolean truth value to every first-order formula and tuple of Boolean names.

Proof.

Formal derivation spine

Formal spine for Boolean semantics for first-order formulas.

$$\|\neg\varphi\| = \neg\|\varphi\|, \qquad \|\varphi \wedge \psi\| = \|\varphi\| \wedge \|\psi\|.$$

$$S_{\varphi,\bar\tau} = \{b \in \mathbb{B} : \exists\sigma\ (\sigma \text{ is a Boolean name and } b = \|\varphi(\sigma, \bar\tau)\|)\}.$$

$$\bigvee S_{\varphi,\bar\tau}$$

Proceed by induction on formula complexity, assuming the atomic values $\|\sigma \in \tau\|$ and $\|\sigma = \tau\|$ have already been defined. For Boolean connectives use the corresponding Boolean-algebra operations. Thus

$$\|\neg\varphi\| = \neg\|\varphi\|, \qquad \|\varphi \wedge \psi\| = \|\varphi\| \wedge \|\psi\|.$$

These clauses are set-theoretic operations on elements of $\mathbb{B}$, so there is no size issue.

Now fix a formula $\exists x\, \varphi(x, \bar{\tau})$. By the induction hypothesis, for every Boolean name σ the value $\|\varphi(\sigma, \bar{\tau})\|$ is already an element of $\mathbb{B}$. Instead of writing an unexplained proper-class join, define

$$S_{\varphi,\bar{\tau}} = \{b \in \mathbb{B} : \exists \sigma \ (\sigma \text{ is a Boolean name and } b = \|\varphi(\sigma, \bar{\tau})\|)\}.$$

This is a definable subclass of the set $\mathbb{B}$; hence Separation in the ambient metatheory makes it a set. Because $\mathbb{B}$ is complete, the supremum

$$\bigvee S_{\varphi,\bar{\tau}}$$

exists in $\mathbb{B}$. We take this supremum as the Boolean value of the existential formula. Universal quantification is then defined by De Morgan duality,

$$\|\forall x\, \varphi(x, \bar{\tau})\| = \neg\|\exists x\, \neg\varphi(x, \bar{\tau})\|,$$

or equivalently as the meet of the set of attained instance values.

Uniqueness follows from the recursion on formula complexity: once the values of proper subformulas are fixed, each displayed clause determines the next value uniquely. Induction also proves invariance under Boolean-equal parameters, using the substitutivity theorem for the atomic equality relation. Finally, when $\mathbb{B} = \{0, 1\}$, joins and meets are ordinary existential and universal truth values, so the construction reduces to ordinary two-valued semantics.

The crucial size point is therefore explicit: quantification ranges over all names at the semantic level, but the values being joined live in the set $\mathbb{B}$. We take the join of the set of *attained Boolean values*, not a literal proper-class family. □

Theorem 51.4 (Boolean soundness of first-order deduction)**.** *If $\Sigma \models \varphi$ in first-order logic, then in every complete Boolean-valued structure*

$$\bigwedge_{\psi\in\Sigma} \|\psi\| \le \|\varphi\|.$$

In particular, if every premise has Boolean value 1*, then so does the conclusion.*

Proof.

Formal derivation spine

Formal spine: finite consequence gives the Boolean inequality. Choose finite $\Gamma \subseteq \Sigma$ with $\Gamma \models \varphi$, and put

$$c := \bigwedge_{\psi\in\Gamma} \|\psi\|.$$

Boolean soundness relative to c gives $c \le \|\varphi\|$; since

$$\bigwedge_{\psi\in\Sigma} \|\psi\| \le c,$$

the desired inequality follows.

By first-order compactness, $\Sigma \models \varphi$ implies that some finite set

$$\Gamma = \{\psi_1, \ldots, \psi_n\} \subseteq \Sigma$$

already satisfies $\Gamma \models \varphi$. Completeness of first-order deduction gives a formal derivation of φ from Γ. Let

$$c := \bigwedge_{i=1}^{n} \|\psi_i\|.$$

We prove by induction on that derivation that every derived formula θ satisfies $c \le \|\theta\|$. For a premise ψ_i, this is the definition of a meet. Logical and equality axioms have value 1, so $c \le 1 = \|\theta\|$. For modus ponens, the induction hypotheses give

$$c \le \|\alpha\|, \qquad c \le \|\alpha \to \beta\| = \neg\|\alpha\| \vee \|\beta\|.$$

Meeting the two inequalities and using Boolean distributivity yields $c \le \|\beta\|$. The quantifier rules follow from the join/meet definitions and the usual eigenvariable side conditions. Therefore

$$c \le \|\varphi\|. \tag{1}$$

Finally,

$$\bigwedge_{\psi\in\Sigma} \|\psi\| \le \bigwedge_{\psi\in\Gamma} \|\psi\| = c.$$

Together with (1) this proves

$$\bigwedge_{\psi\in\Sigma} \|\psi\| \le \|\varphi\|.$$

When every premise has value 1, the left side is 1, so $\|\varphi\| = 1$. Thus the proof establishes the full inequality stated in the theorem, not merely the value-one special case. □

Worked example

Worked Example 51.5 (A three-valued-looking picture)**.** If $b \in \mathbb{B}$ is neither 0 nor 1, a name may satisfy $\|\check{0} \in \tau\| = b$. The assertion is not simply true or false before a generic ultrafilter is chosen. Once an ultrafilter contains b, the quotient interprets the assertion as true; if it contains $\neg b$, it becomes false.

A useful warning

Boolean truth values live in a complete algebra. Infinite quantifiers require joins and meets, so completeness is not optional bookkeeping.

Looking ahead

The next chapter, *Boolean-valued ZFC*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 52

Boolean-valued ZFC

Definitions and setup

Formal definitions used in this chapter

Validity with value one. An axiom σ holds in $V^{\mathbb{B}}$ when $\|\sigma\| = 1_{\mathbb{B}}$.
Boolean implication. $a \to b := \neg a \vee b$.

To prove $V^{\mathbb{B}} \models \mathrm{ZFC}$ in the Boolean-valued sense, construct names witnessing every axiom with truth value 1. Separation modifies Boolean coefficients by the truth value of the defining formula. Replacement/Collection uses the mixing/maximum principle to collect a set of possible witness names. Power Set uses bounded-rank subnames. These are the same hard closure issues as in generic extensions, expressed algebraically.

How this chapter fits together

The ZF axioms are now verified with Boolean value 1. Separation modifies coefficients by the Boolean value of the defining formula; Collection/Replacement uses mixing to assemble witnesses; Power Set requires bounded-rank subnames. This mirrors the generic-extension proofs but avoids choosing an external generic filter.

Theorem 52.1 (Boolean-valued Extensionality/Foundation/Pairing/Union/Infinity)**.** *In $V^{\mathbb{B}}$, the Boolean truth value of each instance of Extensionality, Foundation, Pairing, Union, and Infinity is $1_{\mathbb{B}}$.*

Proof.

Formal derivation spine

Formal spine for Boolean-valued Extensionality/Foundation/Pairing/Union/Infinity.

$$\|\mathrm{ZF}\| = 1$$

Extensionality has Boolean value 1 by the very definition of Boolean equality. Foundation is verified by choosing a minimal-rank potential member below any nonzero Boolean value of nonemptiness; rank descent rules out an infinite membership descent. Pairing is represented by the name assigning coefficient 1 to the two prescribed names. Union is obtained by multiplying the coefficients along two successive membership steps and taking their joins. Infinity is witnessed by the canonical Boolean check-name for ω.

Each construction is a set-sized Boolean name, and direct calculation gives Boolean value 1 to the corresponding axiom. Separation, Replacement, Power Set, and Choice require additional constructions and are deliberately excluded from this elementary-axiom theorem. □

Theorem 52.2 (Boolean-valued Separation)**.** *For every name a and formula $\varphi(x, \bar{p})$, there is a name b such that*

$$\|\forall x\,(x \in b \leftrightarrow (x \in a \wedge \varphi(x, \bar{p})))\| = 1.$$

Thus Boolean-valued Separation has truth value 1*.*

Proof.

Formal derivation spine

Formal spine for Boolean-valued Separation.

$$\sigma(u) = \tau(u) \wedge \|\varphi(u, \bar{\rho})\|.$$

$$\big\|x \in \sigma\big\| = \big\|x \in \tau \wedge \varphi(x, \bar{\rho})\big\|$$

$$\|\mathrm{ZF}\| = 1$$

Let τ be a Boolean name and let $\varphi(x, \bar{\rho})$ be a formula. Define a new name σ with the same domain as τ by

$$\sigma(u) = \tau(u) \wedge \|\varphi(u, \bar{\rho})\|.$$

For each potential member u, its coefficient in σ is exactly the degree to which it belongs to τ and satisfies φ. Using the atomic membership calculation and distributivity of the complete Boolean algebra, one checks

$$\big\|x \in \sigma\big\| = \big\|x \in \tau \wedge \varphi(x, \bar{\rho})\big\|$$

up to Boolean equality of representatives. Therefore the Separation axiom has value 1. The construction is the Boolean analogue of restricting an ordinary forcing name by conditions that force the defining formula.

The equality caveat is important because distinct Boolean names can have equality value 1. The verification therefore uses the previously proved substitutivity of Boolean equality rather than treating names as literal set-theoretic representatives of extension objects. □

Theorem 52.3 (Boolean-valued Replacement/Collection)**.** *If*

$$\|\forall x \in a\, \exists y\, \varphi(x, y, \bar{p})\| = 1,$$

then there is a name b with

$$\|\forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p})\| = 1.$$

For functional φ, the corresponding Boolean-valued Replacement instance has value 1*.*

Proof.

Formal derivation spine

Formal spine for Boolean-valued Replacement/Collection.

$$\|\forall x \in \sigma\, \exists y \in u\, \varphi(x, y, \bar{\pi})\| = 1.$$

$$\|\mathrm{ZF}\| = 1$$

Proof map. Boolean Replacement repeats the witness-bounding idea with Boolean coefficients. For each potential input, maximal antichains choose names whose witness values cover Boolean value 1; mixing combines the antichain-indexed choices into single names. Metatheoretic Replacement then collects those names into a set, yielding Boolean value 1 for Collection and hence Replacement.

Assume the Boolean truth value of $\forall x \in \sigma\, \exists y\, \varphi(x, y, \bar{\pi})$ is 1. For each name ρ in the domain of σ, the Boolean value of "if $\rho \in \sigma$, then some y satisfies $\varphi(\rho, y, \bar{\pi})$" is 1. By the Boolean maximum principle, choose a name τ_ρ such that the Boolean value of the witnessing instance realizes the required supremum. Replacement in the metatheory collects the set of names $\{\tau_\rho : \rho \in \mathrm{dom}(\sigma)\}$. Form a name u containing these possible witnesses

with coefficient 1 (or the corresponding mixed coefficients when several antichain pieces are required). Boolean distributivity then gives

$$\|\forall x \in \sigma\, \exists y \in u\, \varphi(x, y, \bar{\pi})\| = 1.$$

Thus Collection has Boolean value 1. When φ is functional, Boolean Separation extracts the Boolean-valued range, proving Replacement. The set-sized collection of witness names, not an external choice over a proper class, is the essential point. □

Theorem 52.4 (Boolean-valued Power Set and Choice)**.** *For every name a there is a name b with*

$$\|\forall x\, (x \in b \leftrightarrow x \subseteq a)\| = 1.$$

If the ambient construction is carried out in ZFC, the Boolean-valued Choice axiom also has value 1*.*

Proof.

Formal derivation spine

Formal spine for Boolean-valued Power Set and Choice.

$$\|y \subseteq a\| = 1$$

$$c : \mathrm{dom}(a) \longrightarrow \mathbb{B}$$

$$\|z \in P_a\| \leq \|z \subseteq a\|.$$

We prove Power Set separately from Collection because its difficulty is not witness selection but bounding the universe of possible subset names. Fix a Boolean name a. By induction on the rank of a, every name y with

$$\|y \subseteq a\| = 1$$

is Boolean-equivalent to a *canonical subname* $\widehat{y}$ whose potential members are drawn from a fixed set of lower-rank representatives attached to $\mathrm{dom}(a)$, and whose coefficients lie in the set-sized Boolean algebra $\mathbb{B}$. Concretely, after replacing members by the chosen lower-rank representatives, $\widehat{y}$ is coded by a function

$$c : \mathrm{dom}(a) \longrightarrow \mathbb{B}$$

with each coefficient bounded by the Boolean degree to which that member can occur in a. The collection S_a of all such coefficient functions is a genuine set in the metatheory.

Form a Boolean name P_a whose domain is S_a and which assigns coefficient 1 to every canonical subname. If $z \in S_a$, its construction gives $\|z \subseteq a\| = 1$, so

$$\|z \in P_a\| \leq \|z \subseteq a\|.$$

Conversely, if $\|y \subseteq a\| = 1$, normalization supplies $z \in S_a$ with $\|y = z\| = 1$, and substitutivity of Boolean equality gives $\|y \in P_a\| = 1$. Hence

$$\|\forall y\, (y \in P_a \leftrightarrow y \subseteq a)\| = 1.$$

Assume now that the ambient metatheory is ZFC, and let u be a name forced to be a family of nonempty sets. Boolean Collection together with the maximum principle yields a set S_u of names such that, with Boolean value 1, every member of u has a witness represented by some element of S_u. Use ambient Choice to fix a well-order $<_u$ of the set S_u. For each possible member x of u, take a maximal antichain deciding the $<_u$-least witness name, and mix the locally least names along that antichain. The resulting name f satisfies, with Boolean value 1, that $f(x) \in x$ for every $x \in u$. Hence the Boolean-valued Choice axiom has value 1. This final step uses ambient Choice exactly to well-order the set S_u; the Power-Set construction itself uses only the set-sized canonical-subname bound. □

Worked example

Worked Example 52.5 (Boolean separation)**.** Given a name τ and formula $\varphi(x)$, define a new name whose coefficient at a constituent σ is $\tau(\sigma) \wedge \|\varphi(\sigma)\|$. The Boolean membership calculation then shows, with value 1, that the new name consists exactly of the members of τ satisfying φ.

A useful warning

For Replacement and Power Set, it is not enough to appeal to their truth in an eventual quotient. Their Boolean values must be shown to equal 1 by set-sized constructions in the Boolean universe.

Looking ahead

The next chapter, *Consistency transfer*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 53

Consistency transfer

Definitions and setup

Formal definitions used in this chapter

Relative consistency. $\mathrm{Con}(T) \Rightarrow \mathrm{Con}(T + \varphi)$ is a metatheoretic implication between syntactic consistency statements.
Boolean model theorem. A Boolean-valued universe assigning value 1 to the axioms supplies a consistency proof without postulating a transitive classical model.

A countable transitive ground model is a convenient picture, not the conclusion of a consistency assumption. The formal relative-consistency theorem uses Boolean-valued soundness or forcing over models of finite fragments together with first-order completeness/compactness. If a target contradiction existed, it would use finitely many axioms; the corresponding finite forcing verification would already yield an impossible contradiction in a Boolean-valued model. This is the metatheoretic bridge used by the final independence claims.

How this chapter fits together

This chapter closes the metatheoretic gap left by the pedagogical countable-transitive-model picture. Consistency alone does not supply a transitive model. Instead, Boolean-valued semantics and compactness are applied to finite fragments: if every finite part of the target theory has the appropriate Boolean-valued model, first-order completeness converts that finite satisfiability into a genuine relative-consistency statement.

Theorem 53.1 (Mixing and maximum principles in Boolean-valued form)**.** *(Mixing.) If $A \subseteq \mathbb{B}$ is a maximal antichain and τ_a are names, there is τ with $a \leq \|\tau = \tau_a\|$ for all $a \in A$. (Maximum.) For every formula $\varphi(x)$, there is a name τ such that*

$$\|\exists x\, \varphi(x)\| = \|\varphi(\tau)\|.$$

Commented symbolic trace: mixing turns local witnesses into one name

A maximal antichain A represents mutually incompatible cases. On each case $a \in A$, choose a name τ_a that works below a. The mixing construction builds one name τ whose Boolean coefficient agrees with τ_a on the part of the Boolean algebra below a. Because the antichain joins to 1, these local cases cover all possibilities. The maximum principle then says that the Boolean value of an existential statement is attained by one mixed name.

Proof.

Formal derivation spine

Formal spine for Mixing and maximum principles in Boolean-valued form.

$$a \le \|\tau = \tau_a\| \qquad (a \in A).$$

$$a \le \|\varphi(\tau_a, \bar\sigma)\|.$$

$$b \le \|\varphi(\tau, \bar\sigma)\|.$$

Let $\mathbb{B}$ be a complete Boolean algebra. The Boolean-valued mixing principle begins with a maximal antichain $A \subseteq \mathbb{B}$ and a family of names $\{\tau_a : a \in A\}$. Define a name τ by combining the coefficients of the τ_a's after multiplying them by the corresponding Boolean piece a. A direct calculation from the Boolean definitions of membership and equality gives

$$a \le \|\tau = \tau_a\| \qquad (a \in A).$$

This is the Boolean mixing lemma.

For the maximum principle, let $b = \|\exists x\, \varphi(x, \bar\sigma)\|$. By completeness of $\mathbb{B}$, choose a maximal antichain A below b such that for each $a \in A$ there is a name τ_a with

$$a \le \|\varphi(\tau_a, \bar\sigma)\|.$$

Mix the τ_a's into a single name τ. On each antichain piece a, substitution of Boolean-equal names gives $a \le \|\varphi(\tau, \bar\sigma)\|$. Taking the join over A yields

$$b \le \|\varphi(\tau, \bar\sigma)\|.$$

The reverse inequality follows from the definition of the existential Boolean value as the join of all witness values. Consequently

$$\|\exists x\, \varphi(x, \bar\sigma)\| = \|\varphi(\tau, \bar\sigma)\|,$$

so the existential Boolean value is attained by one mixed name.

Relation to ordinary forcing. The combinatorial idea is the same as in the poset maximum principle, but the conclusion is now an equality of Boolean truth values, and completeness of $\mathbb{B}$ supplies the required joins. □

Theorem 53.2 (Boolean-valued model theorem for a forced sentence)**.** *If a sentence φ has Boolean value* 1 *in a Boolean-valued model in which every axiom of a theory T has value* 1*, then every first-order consequence of $T + \varphi$ has Boolean value* 1*.*

Proof.

Formal derivation spine

Formal spine for Boolean-valued model theorem for a forced sentence.

$$\|\varphi\| = 1.$$

$$\|\bot\| = 1$$

Assume the Boolean-valued construction has assigned value 1 to every axiom of a theory T, and suppose a forcing calculation yields

$$\|\varphi\| = 1.$$

Then $V^{\mathbb{B}}$ is a Boolean-valued model of $T + \varphi$ in the precise semantic sense that every axiom of that theory has Boolean value 1. By Boolean-valued soundness, every sentence formally derivable from any finite subset of those axioms also has value 1; in particular no finite subset can have a derivation whose conclusion has Boolean value 0.

If one works over an actual generic ultrafilter, quotienting Boolean names by value 1 equality recovers the ordinary generic-extension semantics and sends every value-1 sentence to a true sentence. The theorem is therefore the semantic bridge between a top Boolean value and a model of the forced sentence. The next theorem adds the separate metatheoretic argument converting this fact into a Con-implication without assuming a transitive ground model. □

Theorem 53.3 (Formal forcing consistency transfer)**.** *Let S and U be computably enumerable theories with fixed primitive-recursive proof predicates. Suppose a formalizable Boolean-valued interpretation supplies a primitive-recursive function F such that the base arithmetic proves*

$$\forall n \big(\mathrm{Prf}_U(n, \ulcorner 0 = 1 \urcorner) \to \mathrm{Prf}_S(F(n), \ulcorner 0 = 1 \urcorner)\big). \tag{*}$$

Then the same base arithmetic proves

$$\mathrm{Con}(S) \to \mathrm{Con}(U).$$

In a forcing application, S is the ground theory and $U = S + \varphi$; Boolean value 1 for every axiom of U, Boolean soundness of the inference rules, and nontriviality $0_{\mathbb{B}} \neq 1_{\mathbb{B}}$ construct the function F.

Uses: Boolean-valued soundness, the Boolean-valued ZFC verification, and a primitive-recursive translation of contradiction proofs.

How to read the symbols: consistency transfer is a transformation of finite proof codes

The hypothesis gives a primitive-recursive program F. Its input is a code n of a target-theory contradiction proof; its output is a code $F(n)$ of a ground-theory contradiction proof. Inside arithmetic, the implication

$$\mathrm{Prf}_U(n, \ulcorner 0 = 1 \urcorner) \to \mathrm{Prf}_S(F(n), \ulcorner 0 = 1 \urcorner)$$

is then contraposed after existential quantification. No countable transitive model is inferred from $\mathrm{Con}(S)$; the argument is proof-theoretic from beginning to end.

Proof.

Formal derivation spine

Formal spine: translate contradiction proofs and contrapose.

$$\forall n\big(\mathrm{Prf}_U(n, \ulcorner 0 = 1 \urcorner) \to \mathrm{Prf}_S(F(n), \ulcorner 0 = 1 \urcorner)\big)$$

implies

$$\neg\mathrm{Con}(U) \to \neg\mathrm{Con}(S), \qquad \mathrm{Con}(S) \to \mathrm{Con}(U).$$

Assume the proof-code transformation (*). Inside the chosen weak arithmetic, suppose $\neg\mathrm{Con}(U)$. By definition,

$$\exists n\, \mathrm{Prf}_U(n, \ulcorner 0 = 1 \urcorner).$$

Choose such an n. Applying (*) gives

$$\mathrm{Prf}_S(F(n), \ulcorner 0 = 1 \urcorner),$$

hence $\neg\mathrm{Con}(S)$. We have proved

$$\neg\mathrm{Con}(U) \to \neg\mathrm{Con}(S),$$

and classical contraposition yields

$$\mathrm{Con}(S) \to \mathrm{Con}(U). \tag{1}$$

It remains to explain why a Boolean-valued interpretation supplies (*). Given a finite U-proof code n, translate every line into its Boolean-value assertion. The interpretation proves value $1_{\mathbb{B}}$ for each nonlogical

axiom of U; the Boolean soundness theorem gives primitive-recursive proof transformations for the logical axioms and inference rules. Thus the translated derivation ends with an S-proof that

$$\|0 = 1\| = 1_{\mathbb{B}}.$$

The atomic Boolean equality calculation proves

$$\|0 = 1\| = 0_{\mathbb{B}},$$

and the nontriviality axiom proves $0_{\mathbb{B}} \neq 1_{\mathbb{B}}$. Concatenating these finitely many proof transformations produces an S-proof of contradiction. Since every transformation of a finite coded proof is primitive recursive, their composition is the required function F. Formula (1) is therefore a genuine proof-theoretic relative-consistency implication and does not assume any transitive model of S. □

Remark 53.4 (Source note)**.** This theorem records the proof-theoretic consistency-transfer role of forcing/-Boolean values, rather than assuming that consistency produces a countable transitive model. Modern formal treatments may be compared in [13, 14].

Proposition 53.5 (Completeness does not supply a transitive model)**.** *For every first-order theory T, syntactic consistency implies the existence of a first-order model,*

$$\mathrm{Con}(T) \Longrightarrow \exists M\ (M \models T),$$

but the Completeness Theorem does not assert that this model is well founded or transitive. Moreover, if T is a consistent computably enumerable theory for which Gödel's second incompleteness theorem holds, then the assertion that there exists a transitive model of T with standard ω entails the stronger consistency statement $\mathrm{Con}(T + \mathrm{Con}(T))$.

Proof.

Formal derivation spine

Formal spine for completeness versus transitivity.

$$\mathrm{Con}(T) \Longrightarrow \exists M\ (M \models T),$$

where no well-foundedness or transitivity conclusion occurs. Under the additional hypotheses in the second sentence,

$$\exists M\,[M \text{ transitive},\ \omega^M = \omega,\ M \models T] \Longrightarrow \mathrm{Con}(T + \mathrm{Con}(T)).$$

The first implication is exactly first-order completeness: if T is syntactically consistent, the Henkin construction produces a structure $M \models T$. Neither the construction nor the theorem says that the membership relation of such a model is the ambient membership relation, so transitivity and well-foundedness are additional external requirements.

For the second assertion, assume that T is consistent, computably enumerable, and satisfies the hypotheses of Gödel II, and let M be a transitive model of T with $\omega^M = \omega$. Because proof checking for T is primitive recursive, transitivity together with standard ω makes the statement

$$\mathrm{Con}(T) \equiv \neg\exists p\, \mathrm{Prf}_T(p, \ulcorner 0 = 1 \urcorner)$$

absolute between M and the ambient universe. Ambient consistency of T therefore gives $M \models \mathrm{Con}(T)$, so

$$M \models T + \mathrm{Con}(T).$$

Soundness of first-order logic now yields

$$\exists M\,[M \models T + \mathrm{Con}(T)] \Longrightarrow \mathrm{Con}(T + \mathrm{Con}(T)).$$

Gödel's second incompleteness theorem says that, for such a consistent T, $T \nvdash \mathrm{Con}(T)$; equivalently, the consistency of $T + \mathrm{Con}(T)$ is not obtained merely by applying completeness to $\mathrm{Con}(T)$. This is why a countable-transitive-model presentation is a useful pedagogical framework but is not the formal relative-consistency theorem proved in the preceding result. □

Worked example

Worked Example 53.6 (Why a finite-fragment argument is enough)**.** Assume that for every finite subset $F \subseteq T + \varphi$, if $T_F \subseteq T$ is the finite set of axioms of T occurring in F, the Boolean construction gives a model of $T_F \cup (F \setminus T)$. If $T + \varphi$ were inconsistent, one formal proof of contradiction would use only finitely many axioms, producing an inconsistent finite F. This contradicts its model. Thus syntactic consistency follows without ever postulating a transitive model of all of T.

A useful warning

Do not write $\mathrm{Con}(T) \Rightarrow$ "there exists a countable transitive model of T." That implication is stronger than completeness provides and is not used here.

Looking ahead

The next chapter, *Chain conditions and closure*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 54

Chain conditions and closure

Definitions and setup

Formal definitions used in this chapter

κ-cc. Every antichain has size $< \kappa$.
$< \kappa$-closed. Every descending sequence of length $< \kappa$ has a lower bound.
$< \kappa$-distributive. Intersections of fewer than κ dense open requirements behave so that no new shorter ordinal sequences are added.

A κ-chain condition bounds antichains; ccc means every antichain is countable. Closure says descending sequences of a given length have lower bounds. Distributivity says intersections of suitably few dense open sets remain dense. These properties preserve different kinds of cardinal/cofinality information. We prove only the preservation statements later used for Cohen forcing and keep closure-based comparison results separate.

How this chapter fits together

Preservation theorems turn combinatorial properties of a forcing poset into restrictions on what the extension can change. The ccc bounds antichains and prevents ω_1 from being collapsed by a countable sequence of decisions. Closure and distributivity provide different preservation mechanisms. Only the exact variants later needed for Cohen forcing are proved.

Proposition 54.1 (Antichain calculus and kappa-chain conditions)**.** *A poset $\mathbb{P}$ satisfies the κ-chain condition if every antichain has cardinality $< \kappa$. Maximal antichains decide names locally, and the number of possible decided values of a name at one coordinate is bounded by the size of such an antichain.*

Proof.

Formal derivation spine

Formal spine: antichains bound possible decided values. If A is a maximal antichain deciding a name $\dot{x}$, define

$$S_A := \{y : \exists p \in A \ (p \Vdash \dot{x} = \check{y})\}.$$

Then

$$|S_A| \leq |A|,$$

so under the κ-cc every such local value set has size $< \kappa$.

An antichain is a set of pairwise incompatible conditions. A forcing notion is κ-cc if every antichain has cardinality $< \kappa$; ccc is the case $\kappa = \omega_1$. By Zorn's lemma, every antichain is contained in a maximal antichain. A maximal antichain A is predense: every condition is compatible with some member of A, for otherwise that condition could be adjoined to A.

These elementary facts turn forcing questions into cardinal bounds. If A is a maximal antichain deciding the value of a name $\dot{f}(\xi)$, then the set of values that $\dot{f}(\xi)$ can take has cardinal at most $|A|$. Under κ-cc, fewer than κ many possibilities therefore occur at each coordinate. The preservation theorems below repeatedly use exactly this "decide by an antichain, then count possible values" principle. □

Theorem 54.2 (ccc preservation of ω_1)**.** *If $\mathbb{P}$ is ccc and G is generic, then*

$$\omega_1^{V[G]} = \omega_1^V.$$

Equivalently, ccc forcing cannot add a countable sequence cofinal in the ground-model ω_1.

Proof.

Formal derivation spine

Formal spine: ccc cannot add a countable cofinal map into ω_1^V. If $p \Vdash \dot{f} : \omega \to \omega_1^V$, choose a countable maximal antichain A_n deciding $\dot{f}(n)$ and let S_n be its possible values. Then

$$S := \bigcup_{n<\omega} S_n$$

is countable in V, hence

$$\gamma := \sup S < \omega_1^V.$$

Thus $p \Vdash \operatorname{ran}(\dot{f}) \subseteq \gamma + 1$, contradicting cofinality.

Proof map. Assume a name is forced to be a cofinal map $\omega \to \omega_1^V$. For each input n, a maximal antichain deciding the value is countable by ccc. The union of the countably many sets of possible values is countable in the ground model, hence bounded below ω_1^V. No generic choice among those values can be cofinal.

Suppose $\mathbb{P}$ is ccc. If a name $\dot{f}$ is forced to map ω cofinally into the ground-model ω_1, for each n take a maximal antichain deciding $\dot{f}(n)$. Each antichain is countable, so the set of possible values of all $\dot{f}(n)$ is a countable union of countable sets and is bounded below ω_1^V, contradiction. Hence ω_1 is preserved. The general cardinal/cofinality preservation statements used later are proved analogously by bounding the set of possible values with antichains of size below the relevant cardinal. We state only the exact hypotheses needed for Cohen forcing and do not claim that arbitrary ccc forcing preserves every cofinality without qualification. □

Theorem 54.3 (Cardinal preservation under the chain-condition hypotheses used later)**.** *Let κ be regular and let $\mathbb{P}$ satisfy the κ-chain condition. If $G \subseteq \mathbb{P}$ is generic, then every ground-model cardinal $\lambda \geq \kappa$ remains a cardinal in $V[G]$, and every ground-model ordinal α with $\operatorname{cf}^V(\alpha) \geq \kappa$ satisfies*

$$\operatorname{cf}^{V[G]}(\alpha) = \operatorname{cf}^V(\alpha).$$

For $\kappa = \omega_1$, this gives the cardinal/cofinality preservation needed for ccc Cohen forcing.

Proof.

Formal derivation spine

Formal spine: a κ-cc name has fewer than κ possible values at each coordinate. For a name $\dot{f}$ and ξ in its domain, choose a maximal antichain A_ξ deciding $\dot{f}(\xi)$, and let

$$B_\xi := \{\beta : \exists p \in A_\xi \ (p \Vdash \dot{f}(\xi) = \check{\beta})\}.$$

Then $|B_\xi| < \kappa$.

First suppose a condition forces a surjection

$$\dot{f} : \mu \longrightarrow \lambda, \qquad \mu < \lambda, \qquad \lambda \geq \kappa,$$

where λ is a ground-model cardinal. For every $\xi < \mu$, choose a maximal antichain A_ξ deciding $\dot{f}(\xi)$. The κ-chain condition gives $|A_\xi| < \kappa$, hence $|B_\xi| < \kappa$.

Let $B = \bigcup_{\xi<\mu} B_\xi$. If $\mu < \kappa$, regularity of κ gives $|B| < \kappa \leq \lambda$. If $\kappa \leq \mu < \lambda$, then

$$|B| \leq \mu \cdot \sup_{\xi<\mu} |B_\xi| \leq \mu \cdot \kappa = \mu < \lambda.$$

Every possible value of $\dot{f}$ lies in B, so no condition can force $\dot{f}$ to be onto λ. Therefore no ground-model cardinal at least κ is collapsed.

For cofinality, let $\delta = \mathrm{cf}^V(\alpha) \geq \kappa$, and suppose a condition forces a cofinal map $\dot{g} : \mu \to \alpha$ with $\mu < \delta$. Construct sets $B_\xi \subseteq \alpha$ of possible values exactly as above. Their union has cardinality less than δ: use regularity of κ when $\mu < \kappa$, and $\max(\mu, \kappa) < \delta$ otherwise. Since $\delta = \mathrm{cf}^V(\alpha)$, the set $B = \bigcup_{\xi<\mu} B_\xi$ is bounded in α. Thus $\dot{g}$ cannot be cofinal. Hence

$$\mathrm{cf}^{V[G]}(\alpha) = \mathrm{cf}^V(\alpha).$$

Taking $\kappa = \omega_1$ yields the ccc preservation theorem used in the Cohen-forcing volume. □

Theorem 54.4 (Closure/distributivity preservation principles in the limited scope needed for comparison)**.** *Let κ be regular. If $\mathbb{P}$ is $< \kappa$-distributive, then for every $\lambda < \kappa$, every $V[G]$-function $f : \lambda \to \mathrm{Ord}^V$ already belongs to V. In particular, $\mathbb{P}$ adds no new sequences of ground-model ordinals of length $< \kappa$. If $\mathbb{P}$ is $< \kappa$-closed, the same conclusion follows by recursively deciding the coordinates of a name along a descending sequence of conditions and using closure at limit stages.*

Proof.

Formal derivation spine

Formal spine: distributivity/closure blocks short new ordinal sequences. For a name $\dot{f} : \lambda \to \mathrm{Ord}^V$ with $\lambda < \kappa$, let

$$D_\alpha := \{p : p \text{ decides } \dot{f}(\alpha)\}.$$

If $\mathbb{P}$ is $< \kappa$-distributive, $\bigcap_{\alpha<\lambda} D_\alpha$ is dense and one condition decides all coordinates. Under $< \kappa$-closure, recursively choose a descending deciding sequence and take lower bounds at limit stages.

Two preservation principles are needed for comparison. If $\mathbb{P}$ is $< \kappa$-closed and $\lambda < \kappa$, then a name for a λ-sequence of ground-model ordinals can be decided coordinate by coordinate along a descending sequence of length λ; closure supplies a common lower bound, which therefore forces the entire sequence to equal a ground-model sequence. If $\mathbb{P}$ is $< \kappa$-distributive, intersecting fewer than κ dense open sets is dense; applying this to the dense sets deciding successive coordinates yields the same "no new shorter sequences" conclusion. Thus either hypothesis prevents new functions from any $\lambda < \kappa$ into the ground-model ordinals. This is the exact comparison principle used here; the Cohen argument itself relies instead on the separately proved chain-condition lemmas. □

Worked example

Worked Example 54.5 (Why ccc preserves ω_1)**.** If a name $\dot{f} : \omega \to \omega_1^V$ were forced cofinal, choose for each n a maximal antichain deciding $\dot{f}(n)$. Under ccc each antichain is countable, so only countably many ordinals can occur as possible values. Their supremum is still below ω_1^V, contradicting cofinality.

A useful warning

The ccc does not mean that every cardinal or cofinality is automatically preserved in every context. Use only the preservation theorem whose hypotheses have actually been proved.

Looking ahead

The next chapter, *Nice names and preservation bookkeeping*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 55

Nice names and preservation bookkeeping

Definitions and setup

Formal definitions used in this chapter

Nice name for a subset of ω. A name assembled from an antichain A_n for each n, containing pairs $\langle \check{n}, p\rangle$.
Cardinal bookkeeping. First prove the relevant ground ordinals remain cardinals; only then identify them with extension cardinals in counting arguments.

A nice name for a subset of ω is determined, for each integer n, by an antichain of conditions forcing n into the set. Under ccc each such antichain is countable. Thus counting real names reduces to counting countable subsets/sequences from the forcing poset in the ground model. The construction normalizes arbitrary names without changing their forced value.

How this chapter fits together

Nice names compress arbitrary names for subsets of ω into countable antichain data. This normal form is the counting device behind the upper bound on the continuum in the Cohen model. The bookkeeping also keeps ground-model cardinals distinct from their extension interpretations until preservation has been established.

Theorem 55.1 (Nice-name theorem for subsets of omega under ccc forcing)**.** *If $\mathbb{P}$ is ccc and $\dot{A}$ is a name for a subset of ω, then there is a nice name $\dot{B}$ with*

$$\Vdash \dot{A} = \dot{B}, \qquad \dot{B} = \{\langle \check{n}, p\rangle : n < \omega,\ p \in A_n\},$$

where each A_n is an antichain.

Proof.

Formal derivation spine

Formal spine for Nice-name theorem for subsets of omega under ccc forcing.

$$\dot{B} = \{\langle \check{n}, p\rangle : p \in A_n,\ p \Vdash n \in \dot{A}\}$$

Proof map. For each n, replace all information about whether $n \in \dot{r}$ by one maximal antichain deciding that statement, retaining only its positive side. The resulting name is forced equal to $\dot{r}$. Under ccc each antichain is countable, so the name is coded by countable ground-model data.

For a name $\dot{A}$ forced to be a subset of ω, for each n choose a maximal antichain A_n deciding $n \in \dot{A}$, and retain the conditions in A_n forcing membership. The resulting

$$\dot{B} = \{\langle \check{n}, p\rangle : p \in A_n,\ p \Vdash n \in \dot{A}\}$$

is a nice name and is forced equal to $\dot{A}$. Under ccc each A_n is countable. Counting countable subsets of the forcing poset therefore bounds the number of nice real names by the appropriate ground-model cardinal exponent. This is the upper-bound mechanism for the continuum calculation in Volume IX. □

Proposition 55.2 (Cardinal counting of nice names)**.** *If $|\mathbb{P}| = \kappa$ and $\mathbb{P}$ is ccc, then the number of nice names for subsets of ω is at most*

$$(\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0}.$$

Proof.

Formal derivation spine

Formal spine for Cardinal counting of nice names.

$$(\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0}$$

Assume $|\mathbb{P}| = \kappa$ and $\mathbb{P}$ is ccc. A nice name for a real is determined by a sequence $\langle A_n : n < \omega\rangle$ in which each A_n is an antichain of conditions that force n into the named set. By ccc, every A_n is countable.

There are at most $\kappa^{\aleph_0}$ countable subsets of $\mathbb{P}$, hence at most

$$(\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0}$$

countable sequences of such subsets. Therefore there are at most $\kappa^{\aleph_0}$ nice names for reals. If the ground-model arithmetic gives $\kappa^{\aleph_0} = \kappa$, as in the later GCH application, this becomes the upper bound of κ many possible extension reals.

The count is intentionally performed before evaluating any generic filter. It gives a ground-model set of canonical names that surjects onto all extension reals after valuation. Later preservation results are then used to interpret the ground-model cardinal κ as the same cardinal in the extension. □

Theorem 55.3 (Cofinality preservation in the cases used by Cohen forcing)**.** *Let $\mathbb{P} = \mathrm{Add}(\omega, \lambda)$ for any cardinal λ. Since $\mathbb{P}$ is ccc, for every ground-model ordinal α with uncountable cofinality,*

$$\mathrm{cf}^{V[G]}(\alpha) = \mathrm{cf}^{V}(\alpha).$$

In particular,

$$\mathrm{cf}^{V[G]}(\omega_1^V) = \omega_1^V, \qquad \mathrm{cf}^{V[G]}(\omega_2^V) = \omega_2^V,$$

and the corresponding cardinals are preserved.

Proof.

Formal derivation spine

Formal spine for Cofinality preservation in the cases used by Cohen forcing.

$$\theta := \mathrm{cf}^{V}(\alpha) > \omega.$$

$$\dot{f} : \check{\mu} \longrightarrow \check{\alpha}$$

$$A_\xi = \{\beta < \alpha : \exists q \in D_\xi\ q \Vdash \dot{f}(\check{\xi}) = \check{\beta}\}.$$

Let $\mathbb{P} = \mathrm{Add}(\omega, \lambda)$. It is ccc. Fix a ground-model ordinal α with

$$\theta := \mathrm{cf}^V(\alpha) > \omega.$$

Suppose, toward a contradiction, that some $p \in \mathbb{P}$ forces

$$\dot{f} : \check{\mu} \longrightarrow \check{\alpha}$$

to be cofinal for an ordinal $\mu < \theta$. For each $\xi < \mu$, choose a maximal antichain D_ξ below p deciding $\dot{f}(\xi)$. Since $\mathbb{P}$ is ccc, D_ξ is countable. Let

$$A_\xi = \{\beta < \alpha : \exists q \in D_\xi \; q \Vdash \dot{f}(\check{\xi}) = \check{\beta}\}.$$

Then $|A_\xi| \leq \aleph_0$. Set $A = \bigcup_{\xi<\mu} A_\xi$. Because $\mu < \theta$ and θ is an uncountable regular cardinal,

$$|A| \leq \max(|\mu|, \aleph_0) < \theta.$$

By the definition of $\theta = \mathrm{cf}^V(\alpha)$, a subset of α of cardinality below θ is bounded, so there is $\gamma < \alpha$ with $A \subseteq \gamma$. Every value of $\dot{f}$ forced by a condition below p belongs to the corresponding $A_\xi \subseteq A$. Hence

$$p \Vdash \mathrm{ran}(\dot{f}) \subseteq \check{\gamma},$$

contradicting the assumption that p forces $\dot{f}$ cofinal in α. Therefore no shorter cofinal sequence is added, and

$$\mathrm{cf}^{V[G]}(\alpha) = \mathrm{cf}^V(\alpha).$$

Applying this to ω_1^V and ω_2^V, together with the earlier ccc cardinal-preservation theorem, gives the two displayed special cases and preservation of the corresponding cardinals. □

Proposition 55.4 (Ground-model versus extension cardinal-arithmetic bookkeeping)**.** *When a forcing theorem states a ground-model cardinal κ remains a cardinal, the same ordinal denotes κ in the extension. Cardinal-exponent calculations must therefore separate (i) preservation of the relevant ordinals/cardinals and (ii) counting new names in the extension.*

Proof.

Formal derivation spine

Formal spine: separate preservation from counting. If forcing preserves the ground-model cardinal κ, then the same ordinal remains a cardinal in the extension:

$$\kappa^{V[G]} = \kappa^V \quad \text{as an ordinal/cardinal identity.}$$

A statement such as $2^\lambda = \kappa$ additionally requires independent bounds

$$\kappa \leq |\mathcal{P}(\lambda)|^{V[G]} \leq \#\{\text{relevant names in } V\}.$$

Forcing calculations mix two kinds of cardinal statements, and the bookkeeping must keep them separate.

First, the *number of names* is counted in the ground model. Thus an estimate such as $\kappa^{\aleph_0} = \kappa$ is used as a statement of ground-model cardinal arithmetic when bounding nice names. Second, the resulting inequality is intended to describe cardinals in the extension. To make that translation, one must already know that the relevant ground-model cardinals have not been collapsed and that the relevant cofinalities are preserved.

For Cohen forcing the ccc theorems provide this bridge. Consequently a ground-model count of at most ω_2^V real names may be read in the extension as "at most $\aleph_2$ reals" only after $\omega_2^V = \omega_2^{V[G]}$ has been established. This distinction prevents a hidden circularity in continuum computations. □

Worked example

Worked Example 55.5 (Normalizing a real name)**.** For each n, choose a maximal antichain deciding whether $n \in \dot{r}$, and keep only conditions forcing membership. The resulting name is determined by the sequence of these antichains. Under ccc each is countable, so counting reals reduces to counting countable subsets of the forcing poset.

A useful warning

Write ω_1^V or ω_2^V until the preservation theorem identifies those ordinals with the extension cardinals. Suppressing the superscript too early hides a real logical step.

End-of-volume perspective

The forcing apparatus is now strong enough for precise consistency transfer and cardinal bookkeeping. Cohen forcing can therefore be treated as an application rather than as a mysterious new construction.

Part IX

Volume IX: Cohen Forcing and Independence of CH

Preface

Cohen forcing is now an application of the general machinery. Both the lower and upper bounds for the continuum are calculated explicitly.

Position in the series

Volume IX applies the forcing machinery to Cohen forcing and computes the continuum after adding many Cohen reals, yielding the relative consistency of *eg*CH.

Sources and further reading

The continuum calculation is Cohen's independence argument in modern forcing language; compare [9, 10, 11] and the later textbook accounts [13, 14].

Notation for this volume

$\mathrm{Add}(\omega, \kappa) = \mathrm{Fn}(\kappa \times \omega, 2, <\omega)$ with reverse inclusion. Coordinate generics are c_α.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 56

Adding Cohen reals

Definitions and setup

Formal definitions used in this chapter

Cohen forcing. $\mathrm{Add}(\omega,\kappa) = \mathrm{Fn}(\kappa\times\omega, 2, <\omega)$, ordered by reverse inclusion.
Coordinate real. $c_\alpha(n) = (\bigcup G)(\alpha, n)$.

$\mathrm{Add}(\omega,\kappa)$ consists of finite partial functions $\kappa\times\omega\to 2$, ordered by reverse inclusion. The generic union is total because each coordinate-decision set is dense. Its α-th row is a Cohen real c_α. Finite support makes coordinates easy to permute and easy to separate by dense sets.

How this chapter fits together

Cohen forcing $\mathrm{Add}(\omega,\kappa)$ consists of finite partial binary assignments to $\kappa\times\omega$. A generic filter's union is total, and each coordinate gives a new binary sequence. Dense sets show that the coordinate sequences are generic and pairwise distinct.

Proposition 56.1 (Definition and separativity of $\mathrm{Add}(\omega,\kappa)$)**.** *For an infinite cardinal κ,*

$$\mathrm{Add}(\omega,\kappa) = \mathrm{Fn}(\kappa\times\omega, 2, <\omega)$$

ordered by reverse inclusion is separative: if $p\not\leq q$, some extension of p is incompatible with q.

Proof.

Formal derivation spine

Formal spine for Definition and separativity of $\mathrm{Add}(\omega,\kappa)$.

$$\mathrm{Add}(\omega,\kappa) = \mathrm{Fn}(\kappa\times\omega, 2, <\omega),$$

For an infinite cardinal κ, define

$$\mathrm{Add}(\omega,\kappa) = \mathrm{Fn}(\kappa\times\omega, 2, <\omega),$$

the set of finite partial functions from $\kappa\times\omega$ to 2, ordered by reverse inclusion: $q\leq p$ iff $q\supseteq p$. Thus a stronger condition specifies more bits.

The forcing is separative. If $p\not\leq q$, either p already disagrees with q on a common coordinate or q specifies a coordinate not fixed by p. In the latter case extend p at that coordinate with the opposite bit; the resulting condition is incompatible with q. Hence every failure of the order can be witnessed by a stronger condition

incompatible with the alleged upper condition, exactly as separativity requires. *Why reverse inclusion is the order.* A condition is information: it specifies finitely many bits. Extending a partial function adds information and therefore makes the condition stronger, so $q \leq p$ means $q \supseteq p$. In the separativity argument, if $p \not\leq q$, the goal is to strengthen p until it explicitly conflicts with q. If they already disagree, nothing must be added; if q mentions a fresh coordinate, assigning the opposite bit at that coordinate creates the required incompatibility. □

Theorem 56.2 (The generic object yields kappa Cohen reals)**.** *If $G \subseteq \mathrm{Add}(\omega, \kappa)$ is generic and $g = \bigcup G$, then $g : \kappa \times \omega \to 2$ is total. For each $\alpha < \kappa$,*

$$c_\alpha(n) = g(\alpha, n)$$

defines a Cohen real over the ground model.

Proof.

Formal derivation spine

Formal spine for The generic object yields kappa Cohen reals.

$$D_{\alpha,n} = \{p : (\alpha, n) \in \operatorname{dom} p\}$$

Let G be generic and put $g = \bigcup G$. Directedness of G guarantees that the union is a function: two conditions in G have a common extension and therefore cannot assign conflicting values to one coordinate. For every (α, n), the set

$$D_{\alpha,n} = \{p : (\alpha, n) \in \operatorname{dom} p\}$$

is dense, so genericity makes g total on $\kappa \times \omega$.

For each $\alpha < \kappa$, define $c_\alpha(n) = g(\alpha, n)$. If $D \subseteq 2^{<\omega}$ is a ground-model dense open set, the conditions whose α-section extends an element of D form a dense subset of $\mathrm{Add}(\omega, \kappa)$. Hence c_α meets every such D and is a Cohen real over the ground model. *Two separate uses of genericity.* The dense sets $D_{\alpha,n}$ make each coordinate sequence total: every bit is eventually decided. A different family of dense sets, obtained by lifting a dense open set $D \subseteq 2^{<\omega}$ to the α-coordinate of the product forcing, makes that total sequence Cohen generic. Totality and Cohen genericity are therefore distinct claims, and both are proved by meeting the appropriate dense sets. □

Proposition 56.3 (Distinct coordinates yield distinct reals)**.** *If $\alpha \neq \beta < \kappa$, then*

$$\Vdash c_\alpha \neq c_\beta.$$

Indeed, conditions forcing $c_\alpha(n) \neq c_\beta(n)$ for some fresh n are dense.

Proof.

Formal derivation spine

Formal spine for Distinct coordinates yield distinct reals.

$$D_{\alpha,\beta} = \{q : \exists n\ q(\alpha, n) \neq q(\beta, n)\}$$

Fix distinct coordinates $\alpha \neq \beta < \kappa$. Given any condition p, choose $n \in \omega$ outside the finite set of integers occurring in the α- or β-sections of $\operatorname{dom} p$. Extend p by assigning $p(\alpha, n) = 0$ and $p(\beta, n) = 1$. Thus the set

$$D_{\alpha,\beta} = \{q : \exists n\ q(\alpha, n) \neq q(\beta, n)\}$$

is dense. A generic filter meets it, so the union generic satisfies $c_\alpha(n) \neq c_\beta(n)$ for some n. Therefore $c_\alpha \neq c_\beta$. Since this holds for every ground-model pair of distinct coordinates, the map $\alpha \mapsto c_\alpha$ is injective in the extension.

Further explanation. For distinct $\alpha, \beta < \kappa$, the set of conditions that assign opposite values to (α, n) and (β, n) for some fresh n is dense. Given any finite condition, choose n outside both coordinate domains and extend the condition by setting, say, $p(\alpha, n) = 0$ and $p(\beta, n) = 1$. A generic filter meets this dense set, so $c_\alpha(n) \neq c_\beta(n)$ for some n. Hence the coordinate reals are pairwise distinct. □

Worked Theorem 56.4 (One Cohen real: explicit dense sets and genericity tests)**.** *For* $\mathrm{Add}(\omega, 1) = 2^{<\omega}$*, the sets deciding the n-th bit and the sets forcing disagreement with each ground-model real are dense. Hence a generic filter has a total union* $c \in 2^\omega$ *that is not equal to any ground-model real.*

Proof.

Formal derivation spine

Formal spine for One Cohen real: explicit dense sets and genericity tests.

$$E_x = \{p : \exists n \in \operatorname{dom} p \ (p(n) \neq x(n))\}$$

Specialize to $\mathrm{Add}(\omega, 1) = \mathrm{Fn}(\omega, 2, <\omega)$. The dense set $D_n = \{p : n \in \operatorname{dom} p\}$ forces the generic union to decide the n-th bit. For a fixed ground-model real x, the set

$$E_x = \{p : \exists n \in \operatorname{dom} p \ (p(n) \neq x(n))\}$$

is dense: extend any condition at a fresh coordinate with the opposite bit. Therefore the generic union is total and differs from every ground-model real.

This concrete calculation is the prototype for the many-coordinate forcing. Adding a new real, however, does not by itself determine the new value of the continuum; the later name-counting argument is still required. *Two families of dense requirements are visible here.* The sets D_n guarantee that the generic union is a total binary sequence. The sets E_x, one for each ground-model real x, guarantee novelty by forcing disagreement somewhere. A Cohen real is therefore not merely an infinite string of bits; it is a string chosen so as to meet every ground-model dense open requirement. □

Worked example

Worked Example 56.5 (Forcing two coordinates to differ)**.** Given distinct $\alpha, \beta < \kappa$ and a condition p, choose a fresh n not mentioned at either coordinate. Extend p by assigning $(\alpha, n) \mapsto 0$ and $(\beta, n) \mapsto 1$. Thus the set of conditions forcing $c_\alpha \neq c_\beta$ is dense.

A useful warning

The union of a generic filter is total only because the filter meets the dense set deciding each coordinate. Totality is not automatic from directedness alone.

Looking ahead

The next chapter, *The ccc*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 57

The ccc

Definitions and setup

Formal definitions used in this chapter

Δ-system. A family of sets with a fixed pairwise intersection called its root.
ccc. Every antichain is countable.

The Δ-system lemma says that an uncountable family of finite sets has an uncountable subfamily with common pairwise intersection (the root). Applied to finite domains of Cohen conditions, and followed by thinning so root assignments agree, it produces compatible pairs. Hence Cohen forcing is ccc. This combinatorial theorem is the owner of the later cardinal-preservation input.

How this chapter fits together

The Δ-system lemma converts an uncountable family of finite domains into a large subfamily with the same pairwise intersection. After thinning once more so that conditions agree on that root, any two Cohen conditions are compatible. Hence no uncountable antichain exists.

Theorem 57.1 (Delta-system lemma)**.** *Every uncountable family of finite sets contains an uncountable subfamily forming a Δ-system: there is a fixed finite root r such that*

$$A \cap B = r$$

for all distinct members A, B of the subfamily.

Proof.

Formal derivation spine

Formal spine: induct on the common finite size. After thinning to sets of one size n, either some point belongs to uncountably many members, reducing the remainders to size $n-1$, or every point occurs only countably often, allowing an uncountable pairwise-disjoint subfamily. In both cases obtain an uncountable family $\mathcal{B}$ and finite r with

$$A \cap B = r \qquad (A \neq B \in \mathcal{B}).$$

Thin the given uncountable family first so that every member has the same finite size n. We induct on n. For $n = 0$ the family consists only of $\varnothing$, so the assertion is trivial after deleting repetitions.

Assume the result for sets of size n, and let $\mathcal{A}$ be an uncountable family of sets of size $n+1$. There are two cases.

If some point x belongs to uncountably many members of $\mathcal{A}$, restrict to that uncountable subfamily and remove x from every member. The resulting family consists of n-element sets, so by induction it has an uncountable Δ-subsystem with root r. Restoring x gives a Δ-system with root $r \cup \{x\}$.

Otherwise every point belongs to at most countably many members of $\mathcal{A}$. Recursively choose $A_\xi \in \mathcal{A}$ for $\xi < \omega_1$ so that A_ξ is disjoint from all earlier choices. At stage $\xi < \omega_1$, the union $U_\xi = \bigcup_{\eta<\xi} A_\eta$ is countable. Each point of U_ξ occurs in only countably many members of $\mathcal{A}$, so only countably many members of $\mathcal{A}$ meet U_ξ. Since $\mathcal{A}$ is uncountable, choose A_ξ outside them. The resulting uncountable family is pairwise disjoint, hence a Δ-system with root $\varnothing$.

The two cases complete the induction. □

Theorem 57.2 (Add(ω, κ) is ccc)**.** *For every cardinal κ,* Add(ω, κ) *satisfies the countable chain condition: every antichain is countable.*

Commented symbolic trace: the Δ-system ccc argument

A Cohen condition has a finite domain. From an uncountable family of finite domains, thin to a Δ-system whose pairwise intersections are the same finite root. There are only finitely many assignments on that root, so thin again until all conditions agree there. Outside the root the domains are disjoint. Therefore any two surviving conditions have a union that is still a function, hence are compatible. This directly contradicts the possibility of an uncountable antichain.

Proof.

Formal derivation spine

Formal spine: apply the Δ-system lemma to Cohen supports. Thin an uncountable family to domains with common root r, then thin again so all conditions agree on r. For distinct survivors p, q,

$$p \restriction r = q \restriction r, \qquad (\operatorname{dom} p \setminus r) \cap (\operatorname{dom} q \setminus r) = \varnothing,$$

so $p \cup q$ is a condition extending both. Hence no uncountable antichain exists.

Proof map. Apply the Δ-system lemma to the finite domains of an uncountable family of Cohen conditions. There are only finitely many assignments on a fixed finite root, so thin to an uncountable subfamily agreeing there. Any two remaining conditions have disjoint non-root domains and identical root values; their union is therefore a common extension.

Let $\mathcal{A}$ be an uncountable family of conditions in Add(ω, κ). Their domains are finite subsets of $\kappa \times \omega$. By the Δ-system lemma, thin $\mathcal{A}$ to an uncountable subfamily whose domains have a common finite root r and are otherwise pairwise disjoint. There are only finitely many assignments $r \to 2$, so thin once more until all conditions agree on r. Any two conditions in the resulting family have union a finite partial function: they agree on the intersection and have disjoint off-root domains. Hence they are compatible. Therefore no uncountable antichain exists and Add(ω, κ) is ccc.

Detailed formal verification. The ccc proof has two finite combinatorial reductions. First apply the Δ-system lemma to the finite domains of an alleged uncountable antichain. Second thin so all functions agree on the fixed root. Then for distinct conditions

$$p \cup q$$

is a common extension, contradicting antichainhood. Every compatibility assertion is therefore reduced to equality of finitely many bits on the root. □

Remark 57.3 (Source note)**.** The ccc proof for Cohen forcing is the standard finite-domain Δ-system argument; compare [9, 10, 11, 13].

Proposition 57.4 (Finite-support compatibility analysis)**.** *If two Cohen conditions have finite domains whose intersection is a common root on which the conditions agree, then their union is a condition extending both. Thus Δ-system thinning reduces ccc to compatibility on finitely many root coordinates.*

Proof.

Formal derivation spine

Formal spine: Cohen compatibility is agreement on overlap. If

$$p \restriction (\operatorname{dom} p \cap \operatorname{dom} q) = q \restriction (\operatorname{dom} p \cap \operatorname{dom} q),$$

then

$$p \cup q \in \operatorname{Add}(\omega, \kappa), \qquad p \cup q \leq p, q.$$

For a Δ-system, the overlap is exactly the common root, so root agreement is the whole compatibility test.

A Cohen condition is a finite partial function. Two partial functions p, q have a common extension exactly when they agree on every point where both are defined. Under the stated hypothesis they agree on

$$r := \operatorname{dom} p \cap \operatorname{dom} q.$$

Hence $p \cup q$ is still a function. Its domain is the finite set $\operatorname{dom} p \cup \operatorname{dom} q$, so $p \cup q \in \operatorname{Add}(\omega, \kappa)$. Because the forcing order is reverse inclusion,

$$p \cup q \leq p \qquad \text{and} \qquad p \cup q \leq q.$$

Thus p and q are compatible.

For a Δ-system of domains, every pairwise intersection is the same finite root r. Therefore, after thinning so that the conditions have the same values on r, the displayed compatibility criterion applies to every pair. No global property of the off-root coordinates is needed beyond their pairwise disjointness. □

Corollary 57.5 (Preservation consequences for $\operatorname{Add}(\omega, \kappa)$)**.** *Because $\operatorname{Add}(\omega, \kappa)$ is ccc, it preserves ω_1 and, under the cardinal-preservation theorem of Volume VIII, the uncountable ground-model cardinals used in the target continuum calculation.*

Proof.

Formal derivation spine

Formal spine: import the preservation theorem only after ccc is proved. From

$$\operatorname{Add}(\omega, \kappa) \text{ is ccc}$$

and Volume VIII's ccc preservation theorem,

$$\omega_1^{V[G]} = \omega_1^V.$$

In the target case $\kappa = \omega_2^V$, the corresponding cardinal/cofinality preservation result gives

$$\omega_2^{V[G]} = \omega_2^V,$$

so later continuum inequalities may be read as inequalities involving $\aleph_1, \aleph_2$ in the extension.

The preceding theorem proves that $\operatorname{Add}(\omega, \kappa)$ is ccc. Apply the ccc preservation theorem from Volume VIII. It gives directly

$$\omega_1^{V[G]} = \omega_1^V.$$

In the continuum calculation used later we take a ground model of GCH and $\kappa = \omega_2^V$. The cardinal-preservation result established in Volume VIII applies to this Cohen poset and yields that the ground ordinal ω_2^V is not collapsed; hence

$$\omega_2^{V[G]} = \omega_2^V.$$

These are preservation statements about the identities of ground-model ordinals as cardinals. They do not yet count reals. The lower bound $2^{\aleph_0} \geq \aleph_2$ will come from the coordinate Cohen reals, and the upper bound will come separately from nice-name counting. Keeping these steps separate prevents a cardinal-arithmetic conclusion from being smuggled into the preservation theorem itself. □

Worked example

Worked Example 57.6 (A root calculation)**.** Suppose conditions p_i have domains forming a Δ-system with root r, and all $p_i \restriction r$ are equal. For $i \neq j$, the non-root parts of the domains are disjoint, so $p_i \cup p_j$ is still a function. It extends both conditions, proving compatibility.

A useful warning

It is not enough for finite domains merely to intersect in the same root; the conditions must also agree on the values assigned on that root.

Looking ahead

The next chapter, *Lower bound for the continuum*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 58

Lower bound for the continuum

Definitions and setup

Formal definitions used in this chapter

Continuum. $2^{\aleph_0} = |\mathcal{P}(\omega)|$ in the current universe.
Lower-bound strategy. Produce an injection from a preserved cardinal κ into the reals.

The lower bound comes from distinct generic coordinates, but cardinal language is used only after ccc preservation identifies the ground-model indexing cardinal with the corresponding extension cardinal. In the target case, ω_2^V remains $\aleph_2^{V[G]}$, and $\alpha \mapsto c_\alpha$ is an injection into the extension reals.

How this chapter fits together

The coordinate generics give an injection from κ into the reals of the extension. Turning that injection into the cardinal inequality $2^{\aleph_0} \geq \kappa$ requires knowing that the ground-model cardinal κ has not been collapsed. The preservation theorems therefore precede the continuum comparison.

Theorem 58.1 (Cardinal preservation for the specific Cohen forcing used)**.** *In the target case* $\mathbb{P} = \mathrm{Add}(\omega, \omega_2^V)$ *over a GCH ground model, ccc implies*

$$(\omega_1^V)^{V[G]} = \omega_1^{V[G]}, \qquad (\omega_2^V)^{V[G]} = \omega_2^{V[G]}.$$

Proof.

Formal derivation spine

Formal spine for Cardinal preservation for the specific Cohen forcing used.

$$\kappa \hookrightarrow 2^\omega,$$

The forcing $\mathrm{Add}(\omega, \kappa)$ is ccc by the Δ-system argument. Volume VIII proved that ccc forcing preserves every ground-model infinite cardinal: a hypothetical new surjection from a smaller cardinal onto a larger one would, coordinate by coordinate, have only countably many possible values, and the union of those possibilities would still be too small to cover the target.

Applying that theorem to the present forcing shows that the ground-model cardinals used in the continuum calculation are not collapsed. In particular, when $\kappa = \omega_2^V$, this ordinal remains a cardinal of the extension and may later be identified with $\aleph_2^{V[G]}$. *What is being preserved.* Forcing never changes the ordinals, but

an old ordinal can cease to be a cardinal if a new bijection or surjection collapses it. The ccc preservation theorem rules out the relevant new surjections. Hence the ordinal that the ground model called ω_2 is still the second uncountable cardinal in the extension. This identification is essential before the later statement "there are $\aleph_2$ Cohen reals" can be interpreted as a lower bound on the extension continuum. □

Theorem 58.2 (At least kappa distinct reals in the extension)**.** *The map $\alpha \mapsto c_\alpha$ is an injection*

$$\kappa \hookrightarrow \mathcal{P}(\omega)^{V[G]}.$$

Hence the extension contains at least κ distinct reals.

Proof.

Formal derivation spine

Formal spine for At least kappa distinct reals in the extension.

$$V[G] \models 2^{\aleph_0} \geq \aleph_2.$$

$$\kappa \hookrightarrow 2^{\omega},$$

The coordinate construction supplies an injection $\kappa \to \mathcal{P}(\omega)$, $\alpha \mapsto c_\alpha$, in the extension. The ccc preservation theorems from Volume VIII ensure that in the target case the ground-model ω_1, ω_2 remain the corresponding extension cardinals. Thus, when $\kappa = \omega_2^V$,

$$V[G] \models 2^{\aleph_0} \geq \aleph_2.$$

No cardinal comparison is made until preservation has been established.

Point specific to this result. The map $\alpha \mapsto c_\alpha$ is injective because dense sets force any two coordinate reals to differ somewhere. After preservation identifies κ with the same extension cardinal, this is the lower bound $2^{\aleph_0} \geq \kappa$.

Further explanation. The map $\alpha \mapsto c_\alpha$ is an injection from the ground-model index set κ into the set of reals of the extension because distinct coordinates give distinct reals. Once the required preservation theorem identifies the ground-model κ with the same cardinal in the extension, this injection becomes the cardinal inequality $2^{\aleph_0} \geq \kappa$. Thus the lower bound is a direct consequence of the geometry of the generic object, not of name counting. □

Proposition 58.3 (Preservation of ω_1 and ω_2 in the CH/GCH ground-model case)**.** *If the ground model satisfies CH/GCH and $G \subseteq \mathrm{Add}(\omega, \omega_2)$ is generic, then ccc preserves the ordinals and cardinal identities of ω_1 and ω_2.*

Proof.

Formal derivation spine

Formal spine for Preservation of ω_1 and ω_2 in the CH/GCH ground-model case.

$$\omega_1^V = \omega_1^{V[G]} \quad \text{and} \quad \omega_2^V = \omega_2^{V[G]}.$$

$$\kappa \hookrightarrow 2^{\omega},$$

Because $\mathrm{Add}(\omega, \omega_2)$ is ccc, Volume VIII's preservation theorem gives

$$\omega_1^V = \omega_1^{V[G]} \quad \text{and} \quad \omega_2^V = \omega_2^{V[G]}.$$

For ω_1, this is also the direct cofinality argument: a name for a countable cofinal sequence would have only countably many possible values at each coordinate, whose union is still countable and therefore bounded below ω_1^V. Preservation of all infinite cardinals rules out collapsing ω_2^V onto ω_1 or any smaller cardinal.

Hence the symbols $\aleph_1$ and $\aleph_2$ in the remaining continuum calculation denote the same ordinals in the ground model and the extension.

This is the exact point where the cardinal identities needed by the next two inequalities are fixed. From here onward the notation $\aleph_2$ is unambiguous across the ground model and extension for this forcing construction. □

Corollary 58.4 ($2^{\aleph_0} \geq \kappa$)**.** *In $V[G]$ after forcing with* $\mathrm{Add}(\omega, \kappa)$*,*

$$2^{\aleph_0} \geq \kappa$$

whenever the ground-model κ is preserved as a cardinal.

Proof.

Formal derivation spine

Formal spine for $2^{\aleph_0} \geq \kappa$.

$$\kappa \hookrightarrow 2^{\omega},$$

For every $\alpha < \kappa$, the generic union produces a real $c_\alpha \in 2^\omega$. If $\alpha \neq \beta$, the set of conditions which choose a fresh n and assign different bits to (α, n) and (β, n) is dense, so $c_\alpha \neq c_\beta$. Hence $\alpha \mapsto c_\alpha$ is an injection $\kappa \hookrightarrow \mathcal{P}(\omega)$ in the extension. Therefore $2^{\aleph_0} \geq \kappa$.

Further explanation. This is deliberately a short corollary, but one hypothesis is worth making visible: the index cardinal must have survived the forcing. With that preservation already proved, the injection $\alpha \mapsto c_\alpha$ has domain of cardinality κ in the extension itself. Hence the extension contains at least κ reals, which is exactly the displayed lower bound.

No upper bound is contained in this argument. The opposite inequality requires the independent nice-name counting theorem, which is why the continuum calculation is deliberately split into two proofs. □

Worked example

Worked Example 58.5 (The ω_2 lower bound)**.** With $\kappa = \omega_2^V$, the family $\{c_\alpha : \alpha < \omega_2^V\}$ consists of pairwise distinct reals. Since ccc preserves ω_2 in the stated ground-model setting, the same ordinal is $\aleph_2$ in the extension. Thus $2^{\aleph_0} \geq \aleph_2$.

A useful warning

New reals alone do not determine the new continuum. A lower bound must be paired with a separate upper-bound argument.

Looking ahead

The next chapter, *Upper bound by nice names*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 59

Upper bound by nice names

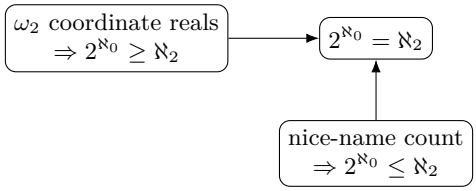

Figure 59.1: The Cohen continuum calculation is a squeeze: distinct coordinate generics give the lower bound and nice-name counting gives the upper bound.

Definitions and setup

Formal definitions used in this chapter

Nice name for a real. A $\mathbb{P}$-name $\dot{r}$ for a subset of ω is nice if

$$\dot{r} = \{\langle \check{n}, p\rangle : n < \omega,\ p \in A_n\}$$

for antichains $A_n \subseteq \mathbb{P}$. Under ccc, every A_n is countable.
Upper-bound strategy. Prove that every real name is forced equal to a nice name and then bound

$$\#\{\text{reals in } V[G]\} \leq \#\{\text{nice names in } V\}.$$

The latter is a ground-model cardinal-arithmetic calculation.

A nice name for a subset of ω is determined, for each integer n, by an antichain of conditions forcing n into the set. Under ccc each such antichain is countable. Thus counting real names reduces to counting countable subsets/sequences from the forcing poset in the ground model. The construction normalizes arbitrary names without changing their forced value.

How this chapter fits together

Every real has a nice name, and a nice name is coded by a countable sequence of antichains. Under GCH and $|\mathrm{Add}(\omega, \omega_2)| = \aleph_2$, there are only $\aleph_2$ such codes. This supplies the upper bound matching the coordinate lower bound.

Theorem 59.1 (Nice-name normal form for Cohen names for reals)**.** *Every* $\mathrm{Add}(\omega,\kappa)$*-name for a real is forced equal to a nice name*

$$\dot r = \{\langle \check n, p\rangle : n<\omega,\ p\in A_n\},$$

where each A_n *is an antichain of conditions forcing* $n\in\dot r$*.*

Proof.

Formal derivation spine

Formal spine: decide each integer on one maximal antichain. For every $n<\omega$, choose a maximal antichain A_n deciding $\check n\in\dot r$, and put

$$A_n^+ := \{p\in A_n : p\Vdash \check n\in\dot r\},\qquad \dot r^{\mathrm{nice}} := \{\langle \check n,p\rangle : n<\omega,\ p\in A_n^+\}.$$

Let $\dot r$ be a name with

$$1_{\mathbb P}\Vdash \dot r\subseteq \check\omega.$$

For each $n<\omega$, decision density gives a maximal antichain A_n every member of which forces either $\check n\in\dot r$ or $\check n\notin\dot r$. Retain the positive part A_n^+ and form the displayed name $\dot r^{\mathrm{nice}}$. This is a forcing name because it is a set of pairs of check names and conditions; its special form is what makes later counting possible.

We show that every generic filter gives the same real from the two names. Let G be generic. If $n\in(\dot r^{\mathrm{nice}})^G$, then some $p\in G\cap A_n^+$ forces $\check n\in\dot r$. The Truth Lemma yields

$$n\in\dot r^G.$$

Conversely, suppose $n\in\dot r^G$. Since A_n is maximal, G meets A_n in exactly one condition p. That condition decides the membership statement. It cannot force $\check n\notin\dot r$, because the Truth Lemma would then give $n\notin\dot r^G$. Hence $p\in A_n^+$, so

$$n\in(\dot r^{\mathrm{nice}})^G.$$

Thus

$$(\dot r^{\mathrm{nice}})^G=\dot r^G$$

for every generic G. The Fundamental Theorem of Forcing therefore gives

$$1_{\mathbb P}\Vdash \dot r^{\mathrm{nice}}=\dot r.$$

The ccc is not needed for normalization itself; it is used in the next result to make each antichain A_n countable and hence to count the possible nice names. □

Proposition 59.2 (Counting nice names under ground-model GCH)**.** *Assume ground-model GCH and let* $\kappa\ge\aleph_2$ *be regular with* $\kappa^{\aleph_0}=\kappa$*. Then the set of nice* $\mathrm{Add}(\omega,\kappa)$*-names for reals has cardinality at most* κ*.*

How to read the symbols: counting nice names is counting structured programs

A nice name for a real is not an arbitrary set-theoretic name. For each $n<\omega$, one countable antichain records the conditions that force n into the real. Thus the name is coded by a countable sequence of countable subsets of $\mathbb P$. If $|\mathbb P|=\kappa$, there are at most $\kappa^{\aleph_0}$ choices for one countable subset and at most $(\kappa^{\aleph_0})^{\aleph_0}=\kappa^{\aleph_0}$ sequences. Under the stated GCH arithmetic this becomes κ.

Proof.

Formal derivation spine

Formal spine for Counting nice names under ground-model GCH.

$$\{\text{nice names}\}\twoheadrightarrow (2^\omega)^{V[G]}.$$

$$\langle A_n^+ : n<\omega\rangle,$$

$$\{\text{nice real names}\}\hookrightarrow \big([\mathbb P]^{\le\aleph_0}\big)^\omega.$$

Proof map. A nice name is a countable sequence of countable antichain data. When $|\mathbb{P}| = \kappa$, each antichain is a countable subset of $\mathbb{P}$, so the number of possible antichains is at most $\kappa^{\aleph_0}$. A countable sequence of them contributes another exponent by $\aleph_0$, which collapses to the same cardinal under the stated GCH calculation.

Every real in the extension has a nice name. Such a name is determined by a sequence $\langle A_n : n < \omega \rangle$ of antichains, each countable by ccc. For $\mathbb{P} = \mathrm{Add}(\omega, \kappa)$ with $|\mathbb{P}| = \kappa$, the number of countable subsets of $\mathbb{P}$ is $\kappa^{\aleph_0}$. Under the chosen ground-model GCH hypothesis and regular $\kappa = \aleph_2$, this equals κ. Hence there are at most κ nice names for reals, so $2^{\aleph_0} \leq \kappa$. Combined with the coordinate lower bound, $2^{\aleph_0} = \kappa = \aleph_2$.

Detailed formal verification. The upper bound has the opposite logical direction: normalize every real name, count the normalized names in the ground model, and then use the surjection

$$\{\text{nice names}\} \twoheadrightarrow (2^{\omega})^{V[G]}.$$

Under $\kappa^{\aleph_0} = \kappa$, this gives at most κ reals. Nothing in this counting step produces new reals; it bounds how many values all possible names can have.

Second detailed calculation. Write the counting argument as a chain of injections. A nice real name is determined by a sequence

$$\langle A_n^+ : n < \omega \rangle,$$

where each A_n^+ is a countable antichain of $\mathbb{P} = \mathrm{Add}(\omega, \kappa)$. Therefore

$$\{\text{nice real names}\} \hookrightarrow \left([\mathbb{P}]^{\leq \aleph_0}\right)^{\omega}.$$

Since $|\mathbb{P}| = \kappa$,

$$|[\mathbb{P}]^{\leq \aleph_0}| \leq \kappa^{\aleph_0}.$$

Hence

$$\#\{\text{nice real names}\} \leq (\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0 \cdot \aleph_0} = \kappa^{\aleph_0}.$$

Under the stated hypothesis $\kappa^{\aleph_0} = \kappa$, this becomes the required κ-bound. The argument counts names in the ground model; the next theorem converts that name count into an upper bound on reals in the extension. □

Theorem 59.3 (Upper bound $2^{\aleph_0} \leq \kappa$ in the target case)**.** *Assume κ is regular, $\kappa^{\aleph_0} = \kappa$ in the ground model, and $G \subseteq \mathrm{Add}(\omega, \kappa)$ is generic. Then every real in $V[G]$ is represented by a nice name, there are at most κ such names, and therefore*

$$V[G] \models 2^{\aleph_0} \leq \kappa.$$

Proof.

Formal derivation spine

Formal spine for Upper bound $2^{\aleph_0} \leq \kappa$ in the target case.

$$\{\text{nice names}\} \twoheadrightarrow (2^{\omega})^{V[G]}.$$

By the nice-name theorem, every real in the extension is represented by a sequence $\langle A_n : n < \omega \rangle$ of antichains of $\mathbb{P}$, where the positive part of A_n decides membership of n. Under ccc each A_n is countable. Thus the number of possible nice names is at most the number of countable sequences of countable subsets of $\mathbb{P}$, bounded by $(|\mathbb{P}|^{\aleph_0})^{\aleph_0} = |\mathbb{P}|^{\aleph_0}$. In the target case $|\mathbb{P}| = \kappa = \aleph_2$ and ground-model GCH gives $\kappa^{\aleph_0} = \kappa$. Hence $2^{\aleph_0} \leq \kappa$.

Further explanation. The estimate is obtained by counting names, not by trying to enumerate the actual reals after forcing. Every real has a nice name, and a nice name is coded by countably many antichains. Under ccc those antichains are countable; under the ground-model cardinal arithmetic assumed in the theorem, the total number of such codes is at most κ. Since valuation maps names onto extension reals, this bounds the continuum from above by κ. □

Theorem 59.4 (Exact continuum calculation $2^{\aleph_0} = \kappa$)**.** *Assume κ is a preserved regular ground-model cardinal with $\kappa^{\aleph_0} = \kappa$, and let $G \subseteq \mathrm{Add}(\omega, \kappa)$ be generic. The κ distinct coordinate Cohen reals give*

$$2^{\aleph_0} \geq \kappa,$$

while nice-name counting gives

$$2^{\aleph_0} \leq \kappa.$$

Hence

$$V[G] \models 2^{\aleph_0} = \kappa.$$

Uses: the coordinate lower bound, ccc cardinal preservation, and the nice-name upper bound.

Proof.

Formal derivation spine

Formal spine for Exact continuum calculation $2^{\aleph_0} = \kappa$.

$$\alpha \mapsto c_\alpha$$

$$V[G] \models 2^{\aleph_0} \geq \kappa. \tag{C1}$$

$$\#\{\text{nice real names}\} \leq (\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0} = \kappa. \tag{C2}$$

Proof map. The equality is deliberately assembled from independent inequalities. Coordinate generics produce κ distinct reals, yielding $2^{\aleph_0} \geq \kappa$ after preservation. Nice-name counting yields $2^{\aleph_0} \leq \kappa$. Nothing about forcing alone makes the two bounds equal; the proof needs both mechanisms.

the lower-bound theorem from the coordinate Cohen reals gives $2^{\aleph_0} \geq \kappa$ from the coordinate Cohen reals. the upper-bound theorem from nice-name counting gives $2^{\aleph_0} \leq \kappa$ from nice-name counting under the stated ground-model cardinal arithmetic. The two inequalities are evaluated in the same extension, and the preservation theorem identifies the relevant ground cardinal κ with its extension cardinal. Therefore $2^{\aleph_0} = \kappa$.

Why the squeeze is legitimate. Both inequalities are statements inside the same extension $V[G]$. The lower bound counts actual pairwise distinct coordinate reals in $V[G]$. The upper bound counts all possible nice names in the ground model and uses the Truth Lemma to know that every extension real is represented by one of them. Cardinal preservation identifies the ground-model parameter κ used in the forcing definition with the extension cardinal appearing on both sides. Only after those identifications do the two inequalities combine to an equality.

Detailed formal verification. Two independent estimates determine the continuum. First, the coordinate generics are pairwise distinct, so

$$\alpha \mapsto c_\alpha$$

is an injection $\kappa \hookrightarrow 2^\omega$. Hence

$$V[G] \models 2^{\aleph_0} \geq \kappa. \tag{C1}$$

Second, every real has a nice name. A nice name is coded by a sequence $\langle A_n : n < \omega \rangle$ of antichains, and ccc makes each A_n countable. Thus

$$\#\{\text{nice real names}\} \leq (\kappa^{\aleph_0})^{\aleph_0} = \kappa^{\aleph_0} = \kappa. \tag{C2}$$

Every real in the extension is the value of one of these names, so

$$V[G] \models 2^{\aleph_0} \leq \kappa. \tag{C3}$$

The preservation theorem is used before (C1)–(C3) to ensure that the ground ordinal κ is still the same cardinal in $V[G]$. Combining (C1) and (C3) gives

$$V[G] \models 2^{\aleph_0} = \kappa.$$

□

Worked example

Worked Example 59.5 (Counting names rather than values)**.** There may be many different names for the same real, so counting all arbitrary names is wasteful. Nice-name normalization chooses a restricted form: for each n, a countable antichain records the conditions forcing n into the real. The number of these structured codes is at most $(\aleph_2^{\aleph_0})^{\aleph_0} = \aleph_2$ under the ground-model GCH calculation.

A useful warning

The equality $\kappa^{\aleph_0} = \kappa$ is a ground-model cardinal-arithmetic input here; it must not be inferred from ccc alone.

Looking ahead

The next chapter, *CH and its failure*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 60

CH and its failure

Definitions and setup

Formal definitions used in this chapter

CH. $2^{\aleph_0} = \aleph_1$.
Failure of CH in the target Cohen extension. Establish the exact equality $2^{\aleph_0} = \aleph_2$, not merely the existence of new reals.

Adding one Cohen real and adding $\aleph_2$ Cohen reals behave differently. A countable forcing over CH has only $\aleph_1$ nice real names, so CH can survive. By contrast $\mathrm{Add}(\omega, \omega_2)$ supplies $\aleph_2$ distinct reals and still admits an $\aleph_2$ nice-name upper bound under GCH. The exact continuum calculation, not merely novelty of the generic real, yields $\neg$CH.

How this chapter fits together

The contrast between one Cohen real and ω_2 Cohen reals is pedagogically important. Adding a real need not destroy CH: over a CH ground model, the count of nice names for one Cohen real remains $\aleph_1$. Adding ω_2 many coordinate reals over GCH forces the lower bound up to $\aleph_2$, while the same nice-name machinery holds the upper bound there.

Worked Theorem 60.1 (Adding one Cohen real over CH can preserve CH)**.** *If* $V \models \mathrm{CH}$ *and* $G \subseteq \mathrm{Add}(\omega, 1)$ *is generic, then*

$$V[G] \models \mathrm{CH}.$$

Thus adding a single Cohen real need not destroy CH.

Proof.

Formal derivation spine

Formal spine: one Cohen real over CH still leaves only $\aleph_1$ reals. Since $\mathrm{Add}(\omega, 1)$ is countable, every nice name for a real is coded by a countable sequence of subsets of a countable set. Under CH,

$$\#\{\text{nice real names}\} \leq (2^{\aleph_0})^{\aleph_0} = \aleph_1.$$

ccc preserves ω_1, while the old reals give the reverse inequality, hence

$$V[G] \models 2^{\aleph_0} = \aleph_1.$$

Assume the ground model satisfies CH and force with the countable poset $\mathrm{Add}(\omega, 1)$. It is ccc, so ω_1 is preserved. Every extension real has a nice name, and because the forcing is countable, a nice name is coded by a countable sequence of subsets of a countable set, hence by a ground-model real. Under CH there are only $\aleph_1$ such codes. Thus the extension has at most $\aleph_1$ reals, while it contains the old $\aleph_1$ reals. Consequently it still satisfies $2^{\aleph_0} = \aleph_1$. The example proves that adding a new real does not by itself force failure of CH.

Why this does not contradict the existence of a new real. CH says that the total number of reals is $\aleph_1$; it does not say that every real already belongs to the ground model. The forcing adds at least one genuinely new real, but a set of size $\aleph_1$ can acquire new elements while still having cardinality $\aleph_1$, provided the old and new reals together are still bounded by $\aleph_1$. Nice-name counting supplies exactly that global upper bound. □

Theorem 60.2 (Add(omega,ω_2) over GCH forces $2^{\aleph_0} = \aleph_2$)**.** *If $V \models$ GCH and $G \subseteq \mathrm{Add}(\omega, \omega_2^V)$ is generic, then*

$$V[G] \models 2^{\aleph_0} = \aleph_2.$$

Commented symbolic trace: the continuum is pinned between the same cardinal

The lower bound and upper bound come from different mechanisms. Coordinate generics give an injection $\omega_2 \to \mathcal{P}(\omega)$, so $2^{\aleph_0} \geq \aleph_2$. Nice-name counting gives at most $\aleph_2$ reals, so $2^{\aleph_0} \leq \aleph_2$. The ccc preservation theorem is needed before writing these inequalities with the extension's $\aleph_2$: it guarantees that the ground-model ω_2 has not collapsed.

Proof.

Formal derivation spine

Formal spine for Add(omega,ω_2) over GCH forces $2^{\aleph_0} = \aleph_2$.

$$2^{\aleph_0} = \aleph_2,$$

$$\kappa := \omega_2^V.$$

$$\kappa^{\aleph_0} = \kappa,$$

Let the ground model satisfy GCH and force with $\mathbb{P} = \mathrm{Add}(\omega, \omega_2)$. By the ccc theorem above, $\mathbb{P}$ is ccc; by the preservation results, the ground ω_1 and ω_2 remain $\aleph_1$ and $\aleph_2$ in the extension. The ω_2 coordinate generics are pairwise distinct reals, giving $2^{\aleph_0} \geq \aleph_2$. Conversely every real has a nice name, and the number of nice names is at most $(\aleph_2)^{\aleph_0} = \aleph_2$ by GCH, giving $2^{\aleph_0} \leq \aleph_2$. Hence

$$2^{\aleph_0} = \aleph_2,$$

so CH fails. The formal relative-consistency conclusion is taken only after applying the metatheoretic transfer theorem of Volume VIII.

Further explanation. The theorem is the point at which all four ingredients meet: ccc preserves the relevant cardinals, coordinate generics give $2^{\aleph_0} \geq \aleph_2$, nice-name counting gives $2^{\aleph_0} \leq \aleph_2$, and the ground-model GCH hypothesis supplies the cardinal arithmetic needed in that count. Therefore the extension computes the continuum exactly as $\aleph_2$. Since $\aleph_2 > \aleph_1$, CH fails there.

Detailed formal verification. The one-real and many-real cases must be calculated separately. For $\mathrm{Add}(\omega, 1)$, the forcing is countable, so under CH the set of nice names has size at most $\aleph_1$; ccc preserves $\aleph_1$, hence CH remains true. For $\mathrm{Add}(\omega, \omega_2)$, the coordinate lower bound is $\aleph_2$, while GCH gives the matching nice-name upper bound. Thus the amount of Cohen forcing, not merely the existence of a new real, determines the continuum.

Second detailed calculation. In the target application put

$$\kappa := \omega_2^V.$$

GCH gives

$$\kappa^{\aleph_0} = \kappa,$$

and ccc preserves ω_1^V and ω_2^V, so in the extension

$$\kappa = \aleph_2^{V[G]}.$$

The lower-bound theorem gives

$$V[G] \models 2^{\aleph_0} \geq \aleph_2,$$

while the nice-name theorem and the preceding cardinal calculation give

$$V[G] \models 2^{\aleph_0} \leq \aleph_2.$$

Therefore

$$V[G] \models 2^{\aleph_0} = \aleph_2.$$

Since CH is the equation $2^{\aleph_0} = \aleph_1$, and ccc also preserves $\aleph_1$, the same extension satisfies $\neg$CH. □

Remark 60.3 (Source note)**.** Cohenś independence proof combines cardinal preservation, many generic reals, and an upper bound on real names. Compare [9, 10, 11].

Corollary 60.4 (The Cohen extension satisfies not-CH)**.** *In the extension of the preceding theorem,*

$$V[G] \models \neg\mathrm{CH}.$$

Proof.

Formal derivation spine

Formal spine for The Cohen extension satisfies not-CH.

$$2^{\aleph_0} = \aleph_2.$$

$$2^{\aleph_0} \neq \aleph_1,$$

The preceding theorem gives, in the Cohen extension,

$$2^{\aleph_0} = \aleph_2.$$

The preservation theorem gives the usual strict inequality $\aleph_1 < \aleph_2$ in that same extension. Therefore

$$2^{\aleph_0} \neq \aleph_1,$$

which is exactly the negation of CH. No new forcing argument is needed here; this corollary records the logical consequence of the exact continuum computation.

The role of this result is therefore logical rather than combinatorial: the preceding forcing computation has already done all of the set-theoretic work. Here we merely translate the strict cardinal inequality into the standard sentence $\neg$CH that will be used by the consistency-transfer theorem. *Logical translation.* In ZFC, CH is the equation $2^{\aleph_0} = \aleph_1$. The previous theorem establishes the incompatible equation $2^{\aleph_0} = \aleph_2$, and cardinal preservation guarantees $\aleph_1 \neq \aleph_2$ in the same model. Thus the inference to $\neg$CH is simply substitution into the definition of CH; all forcing-specific work has already been discharged. □

Worked example

Worked Example 60.5 (One real versus ω_2 reals)**.** For $\mathrm{Add}(\omega, 1)$, the forcing is countable, so nice real names are coded by reals from the ground model; CH bounds them by $\aleph_1$. For $\mathrm{Add}(\omega, \omega_2)$, the coordinate family already has size $\aleph_2$. The two forcings use the same local finite conditions but produce different continuum sizes because the number of coordinates changes the lower bound.

A useful warning

The slogan “Cohen forcing makes CH false” is too vague. The specific forcing and ground-model cardinal arithmetic determine the resulting continuum.

Looking ahead

The next chapter, *Homogeneity and relative consistency*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 61

Homogeneity and relative consistency

Definitions and setup

Formal definitions used in this chapter

Homogeneous forcing. Automorphisms can move any condition to one compatible with any prescribed condition (in the weak homogeneity form used here).
Independence. A sentence φ is independent of T, relative to $\mathrm{Con}(T)$, when both $T + \varphi$ and $T + \neg\varphi$ are relatively consistent with T.

Cohen forcing admits many coordinate permutations. Homogeneity says conditions can be moved by automorphisms into compatible positions, so parameter-free/ground-parameter statements are not decided differently on unrelated local pieces. For relative consistency, however, homogeneity is secondary: the main load-bearing steps are ccc, name counting, and the Volume VIII consistency-transfer theorem.

How this chapter fits together

Automorphisms of Cohen forcing show that no condition can encode privileged ground-model-free information about the generic coordinates. Homogeneity is then combined with the formal consistency-transfer machinery of Volume VIII. Finally, the positive constructible model and the negative Cohen model give the two sides of CH independence.

Theorem 61.1 (Homogeneity of Cohen forcing)**.** *Let $\mathbb{P} = \mathrm{Add}(\omega, \kappa)$ with largest condition $1_{\mathbb{P}}$. For any $p, q \in \mathbb{P}$ there is a coordinate permutation π inducing an automorphism of $\mathbb{P}$ such that $\pi(p)$ is compatible with q. Consequently, if $\varphi(\check{a}_1, \dots, \check{a}_n)$ is a sentence whose parameters are canonical ground-model names and some condition forces it, then*

$$1_{\mathbb{P}} \Vdash \varphi(\check{a}_1, \dots, \check{a}_n).$$

Proof.

Formal derivation spine

Formal spine: permute the entire countable coordinate set used by conditions. For finite domains $d_p, d_q \subseteq \kappa \times \omega$, choose a permutation π of $\kappa \times \omega$ such that

$$\pi[d_p] \cap d_q$$

contains only protected coordinates on which the two conditions already agree. Then $\pi(p) \cup q$ is a condition.

An arbitrary permutation π of the coordinate set $\kappa \times \omega$ induces an automorphism of $\mathrm{Add}(\omega, \kappa)$ by

$$(\pi p)(\pi(\alpha, n)) := p(\alpha, n).$$

Let p, q be conditions. Their domains are finite. Fix any finite protected set of coordinates that must remain unchanged. Because $\kappa \times \omega$ is infinite even when $\kappa = 1$, choose a permutation fixing the protected set pointwise and moving every unprotected coordinate in $\mathrm{dom}(p)$ away from $\mathrm{dom}(q)$. On any protected overlap the values were required to agree. Hence

$$\pi(p) \cup q$$

is a finite partial function and therefore a common extension of $\pi(p)$ and q. This proves weak homogeneity. No preservation of the second coordinate is required; imposing that unnecessary restriction would fail to prove the $\kappa = 1$ case.

Now suppose

$$p \Vdash \varphi(\check{a}_1, \dots, \check{a}_n) \tag{1}$$

for canonical ground-model names. Every automorphism fixes each check name, so the symmetry lemma gives

$$\pi(p) \Vdash \varphi(\check{a}_1, \dots, \check{a}_n). \tag{2}$$

If the largest condition did not force φ, decision density would give $q \Vdash \neg\varphi$. Choose π so that $\pi(p)$ is compatible with q, and let $r \leq \pi(p), q$. Monotonicity applied to (2) gives $r \Vdash \varphi$, while $r \leq q$ gives $r \Vdash \neg\varphi$, a contradiction. Thus

$$1_{\mathbb{P}} \Vdash \varphi(\check{a}_1, \dots, \check{a}_n).$$

The restriction to canonical ground-model parameters is essential because arbitrary names need not be fixed by the automorphism group. □

Theorem 61.2 (Con(ZFC) implies Con(ZFC + not-CH) using Volume VIII transfer)**.** *Using the formal forcing consistency-transfer theorem of Volume VIII,*

$$\mathrm{Con}(\mathrm{ZFC}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

Proof.

Formal derivation spine

Formal spine for Con(ZFC) implies Con(ZFC + not-CH) using Volume VIII transfer.

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{GCH}).$$

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{GCH}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

$$\mathrm{Con}(\mathrm{ZFC}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

Assume Con(ZFC). The constructible-universe argument of Volume VI, together with the formal metatheoretic transfer now available in Volume VIII, gives

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{GCH}).$$

Over that strengthened but equiconsistent ground theory, the forcing calculation of this volume proves that $\mathrm{Add}(\omega, \omega_2)$ forces ZFC together with $2^{\aleph_0} = \aleph_2$, hence $\neg$CH.

Apply the formal forcing consistency-transfer theorem of Volume VIII to this calculation. It yields

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{GCH}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

The first consistency implication moved us to an equiconsistent ground theory where the cardinal arithmetic is convenient; the second moved from that ground theory to the Cohen extension. Ordinary transitivity of implication now composes these two metatheoretic statements:

$$\mathrm{Con}(\mathrm{ZFC}) \Longrightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

At no stage is a countable transitive model inferred merely from consistency.

The intermediate use of GCH is a convenience for the clean name-counting calculation, not an additional consistency assumption. Gödel's constructible-universe theorem supplies that ground theory at no extra consistency strength relative to ZFC. □

Theorem 61.3 (Relative independence of CH from ZFC after combining with L)**.** *If ZFC is consistent, then neither CH nor ¬CH is provable from ZFC: constructibility gives*

$$\mathrm{Con}(\mathrm{ZFC}) \Rightarrow \mathrm{Con}(\mathrm{ZFC} + \mathrm{CH}),$$

while Cohen forcing gives

$$\mathrm{Con}(\mathrm{ZFC}) \Rightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

Uses: the constructible model of CH, the Cohen extension with ¬CH, and formal consistency transfer.

Proof.

Formal derivation spine

Formal spine for Relative independence of CH from ZFC after combining with L.

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{CH}),$$

$$\mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

Assume ZFC is consistent. The constructible universe gives the positive branch

$$\mathrm{Con}(\mathrm{ZFC} + \mathrm{CH}),$$

indeed the stronger $\mathrm{Con}(\mathrm{ZFC} + \mathrm{GCH})$. The preceding Cohen-forcing theorem gives the negative branch

$$\mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH}).$$

If ZFC proved CH, then adding ¬CH to ZFC would make an inconsistent theory, contradicting the negative branch. Hence $\mathrm{ZFC} \nvdash \mathrm{CH}$. If ZFC proved ¬CH, then ZFC + CH would be inconsistent, contradicting the constructible positive branch. Hence $\mathrm{ZFC} \nvdash \neg\mathrm{CH}$. Therefore CH is independent of ZFC, conditional on the consistency of ZFC.

This final step uses consistency only to ensure that the two model constructions really witness compatible extensions of the same base theory. It does not claim that ZFC can internally prove its own consistency or the displayed relative-consistency statements. □

Worked example

Worked Example 61.4 (Moving finite support by a permutation)**.** Given two finite Cohen conditions, a permutation of the unused coordinate indices can move the non-root support of one away from the other while fixing a prescribed finite root. The image becomes compatible with the second condition. This finite-support mobility is the concrete source of Cohen homogeneity.

A useful warning

Relative consistency is the conclusion $\mathrm{Con}(\mathrm{ZFC}) \Rightarrow \mathrm{Con}(\mathrm{ZFC} + \neg\mathrm{CH})$, not an absolute proof that ZFC is consistent.

End-of-volume perspective

Together with L, the Cohen extension gives the two relative-consistency directions for CH. The final volume applies a symmetry restriction to a generic extension to separate the Axiom of Choice from ZF.

Part X

Volume X: Symmetric Extensions and Independence of Choice

Preface

Symmetric extensions are treated independently from ordinary forcing. The hard ZF axioms in the symmetric submodel receive their own proved results before Choice is allowed to fail.

Position in the series

Volume X passes from ordinary generic extensions to symmetric submodels. The support calculus preserves ZF while allowing Choice to fail, completing the second Cohen independence branch.

Sources and further reading

The permutation/symmetric-model method is developed as the pure-set counterpart of the Fraenkel–Mostowski idea. Standard treatments and the basic Cohen model may be compared with [12, 13].

Notation for this volume

$\pi\tau$ is the recursive action of a forcing automorphism on a name; $\mathrm{sym}(\tau)$ is its stabilizer; N is the class of values of hereditarily symmetric names.

How to read symbolic arguments in this volume

When a formula appears, read it in layers rather than as one visual block. First identify the *ambient language*: are we speaking inside a formal theory, about a structure, or in the surrounding metatheory? Next mark the objects being manipulated (formulas, codes, sets, names, conditions, ordinals) and the operation being applied to them. Finally check the side conditions–free variables, rank bounds, compatibility, genericity, transitivity, or consistency hypotheses. Whenever a displayed derivation changes level, the text says so explicitly. A displayed line is therefore meant to be read like a line of well-commented code: the symbols perform the operation, and the surrounding prose records its type, preconditions, and purpose.

Chapter 62

Fraenkel–Mostowski precursor

Definitions and setup

Formal definitions used in this chapter

Atom. An urelement with no members, distinct from all pure sets.
Support. A finite set of atoms whose pointwise stabilizer fixes the object.
Dedekind-finite. An infinite set with no injection from ω.

The Fraenkel–Mostowski method begins with atoms and a permutation group. A finite support controls an object's invariance. Hereditarily symmetric objects form an inner universe of set theory with atoms in which Choice may fail. This is pedagogically useful because the support contradiction is transparent, but atoms are not allowed in ordinary ZF; the later symmetric forcing construction removes them.

How this chapter fits together

The permutation model with atoms is a warm-up for symmetric forcing. Finite supports make the full set of atoms invariant while preventing an enumeration from being invariant. This produces an infinite Dedekind-finite set and reveals the mechanism by which Choice will later fail. Because atoms are urelements, however, this first model is a model of ZFA rather than pure ZF.

Proposition 62.1 (Permutation actions on sets with atoms)**.** *Let A be a set of atoms and $G \leq \mathrm{Sym}(A)$. The action of G extends recursively to the cumulative hierarchy over atoms by*

$$\pi x = \{\pi y : y \in x\}$$

for non-atoms, while πa is the given permutation on atoms.

Proof.

Formal derivation spine

Formal spine for Permutation actions on sets with atoms.

$$\pi(x) = \{\pi(y) : y \in x\}.$$

$$\pi(x \cup y) = \pi(x) \cup \pi(y), \qquad x \in y \iff \pi(x) \in \pi(y),$$

$$\mathrm{Fix}(S) \leq \mathrm{Stab}(x).$$

Let A be a set of atoms and let G be a group of permutations of A. Extend each $\pi \in G$ recursively to the cumulative hierarchy over the atoms by

$$\pi(x) = \{\pi(y) : y \in x\}.$$

For atoms the value is given by the original permutation; for ordinary sets the displayed clause recursively moves every member. Foundation makes that recursion well founded. We now verify that this recursive action respects the set-theoretic operations and membership relation:

$$\pi(x \cup y) = \pi(x) \cup \pi(y), \qquad x \in y \iff \pi(x) \in \pi(y),$$

and $(\pi \circ \sigma)(x) = \pi(\sigma(x))$. Thus G acts by automorphisms of the set-with-atoms universe. The permutation model is obtained not by changing membership, but by restricting attention to objects whose behavior under this action is controlled by small supports. □

Theorem 62.2 (Finite-support symmetry calculus in the atom model)**.** *A finite set $S \subseteq A$ supports an object x if*

$$\mathrm{Fix}(S) \leq \mathrm{Stab}(x).$$

Finite supports are closed under the finite set-forming operations used in the Fraenkel–Mostowski universe.

Proof.

Formal derivation spine

Formal spine for Finite-support symmetry calculus in the atom model.

$$\mathrm{Fix}(S) = \{\pi \in G : \pi(a) = a \text{ for every } a \in S\}.$$

$$\mathrm{Fix}(S) \leq \mathrm{Stab}(x).$$

For a finite set of atoms S, let

$$\mathrm{Fix}(S) = \{\pi \in G : \pi(a) = a \text{ for every } a \in S\}.$$

Say that S supports x if every permutation in $\mathrm{Fix}(S)$ fixes x. If S supports x and T supports y, then $S \cup T$ supports $\{x, y\}$, ordered pairs built from x, y, and every finite construction from them. If S supports x, then $\pi[S]$ supports $\pi(x)$.

Define a set to be hereditarily symmetric when it has a finite support and every object in its transitive closure does as well. Rank induction shows that the hereditarily symmetric objects are closed under the elementary set operations. This is the finite-support calculus that the later forcing construction abstracts using a normal filter of subgroups. □

Theorem 62.3 (Existence of an infinite Dedekind-finite set in the basic atom model)**.** *In the basic Fraenkel–Mostowski model with a countably infinite set A of atoms and finite supports, A is infinite but Dedekind-finite: there is no injection $\omega \to A$ belonging to the symmetric universe.*

Proof.

Formal derivation spine

Formal spine for Existence of an infinite Dedekind-finite set in the basic atom model.

$$f(n) = \pi(f)(n) = \pi(f(n)) = b,$$

$$\mathrm{Fix}(S) \leq \mathrm{Stab}(x).$$

The whole atom set A has empty support, because every permutation sends A to itself. It is infinite, since for every n it contains n distinct atoms. Suppose there were an injection $f : \omega \to A$ in the finite-support

universe. Let S be a finite support for f. Choose n with $f(n) \notin S$ and an atom $b \notin S \cup \{f(n)\}$. Let π swap $f(n)$ and b while fixing S pointwise.

Because S supports f, $\pi(f) = f$, and π fixes the natural number n. Hence

$$f(n) = \pi(f)(n) = \pi(f(n)) = b,$$

contradiction. Thus A is an infinite Dedekind-finite set in the basic Fraenkel–Mostowski model. *Why the contradiction uses a permutation rather than cardinal arithmetic.* The support S records every atom on which the name of f is allowed to depend. Once an output value lies outside S, a permutation can move that value without changing the name of the function. Evaluating the unchanged function at the unchanged input must then give both the old and moved atoms, which is impossible. □

Proposition 62.4 (Why this does not yet give a model of pure ZF)**.** *The Fraenkel–Mostowski construction of the preceding theorem is a model of set theory with atoms (ZFA), not a model of pure ZF, because the atoms have no members and are not sets generated from the empty set by the ordinary membership hierarchy.*

Proof.

Formal derivation spine

Formal spine for Why this does not yet give a model of pure ZF.

$$\mathrm{Fix}(S) \leq \mathrm{Stab}(x).$$

The preceding construction is a model of set theory *with atoms* (usually ZFA), not yet a model of ordinary pure ZF. Atoms are urelements: they may belong to sets but they are not themselves sets assembled from lower-rank members. Thus the underlying ontology is different from the pure cumulative hierarchy required by ZF.

Historically one can transfer permutation-model information from atoms to pure sets by additional representation theorems, but that is not the route used in this series. Instead, the next chapters reproduce the same symmetry/support mechanism directly on forcing names. The resulting symmetric submodel consists entirely of ordinary sets and is verified axiom by axiom to satisfy ZF before Choice is shown to fail. □

Worked example

Worked Example 62.5 (Moving an unsupported atom)**.** If a function $f : \omega \to A$ has finite support S, choose an output atom $a \notin S$ and swap it with another atom $b \notin S$. The permutation fixes f and the input integer but moves the output, a contradiction. The support argument is exactly the one later transferred to Cohen names.

A useful warning

The atom model demonstrates failure of Choice, but it does not by itself establish $\mathrm{Con}(\mathrm{ZF} + \neg\mathrm{AC})$ because ZF has no atoms.

Looking ahead

The next chapter, *Automorphisms of forcing and names*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 63

Automorphisms of forcing and names

Definitions and setup

Formal definitions used in this chapter

Automorphism of forcing. An order automorphism $\pi : \mathbb{P} \to \mathbb{P}$.
Action on names. $\pi\tau = \{\langle \pi\sigma, \pi p\rangle : (\sigma, p) \in \tau\}$.

An automorphism of a forcing poset acts recursively on names by applying the automorphism to every attached condition and constituent name. The symmetry lemma states that forcing is equivariant under this action. The stabilizer of a name is the subgroup fixing it. These definitions let group-theoretic supports control semantic facts about the eventual interpretations.

How this chapter fits together

A forcing automorphism acts first on conditions and then recursively on names. Valuation is covariant under this action, and the symmetry lemma transports forcing truth along automorphisms. These three layers must match before support arguments can be made inside a forcing construction.

Theorem 63.1 (Recursive automorphism action on forcing names)**.** *If π is an automorphism of a forcing notion $\mathbb{P}$, define recursively*

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}.$$

This gives an action of the automorphism group on all forcing names.

How to read the symbols: automorphisms act on every layer of a name

If π moves forcing conditions, it must also move all conditions occurring inside a name, and recursively all lower-rank names appearing there. The definition

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}$$

performs that recursive renaming. The covariance identity $(\pi\tau)^{\pi G} = \tau^G$ says that renaming both the program (the name) and its input (the generic filter) leaves the resulting object unchanged.

Proof.

Formal derivation spine

Formal spine for Recursive automorphism action on forcing names.

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}.$$

$$(\pi\tau)^{\pi G} = \tau^G.$$

Let π be an automorphism of the forcing poset. Define its action on names by rank recursion:

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}.$$

Every constituent σ has smaller name rank, so the recursion is well founded. Induction gives $\mathrm{id}(\tau) = \tau$ and $(\pi\rho)(\tau) = \pi(\rho\tau)$, so this is genuinely a group action. The construction preserves name rank and sends ground check names to themselves because their conditions are the top condition and their constituents are recursively fixed.

Further explanation. The recursion is on name rank. Once $\pi\sigma$ has been defined for every name occurring below τ, the displayed set of pairs defines $\pi\tau$. Rank is preserved because automorphisms change conditions but not the membership depth of names. A second induction proves $\pi(\rho \text{ occurring in } \tau)$ occurs in $\pi\tau$, so the action respects the entire hereditary name structure needed later for symmetric names. □

Proposition 63.2 (Valuation commutes with the automorphism action)**.** *For every automorphism π, name τ, and generic filter G,*

$$(\pi\tau)^{\pi G} = \tau^G.$$

Equivalently, valuation is equivariant with respect to simultaneous transport of names and the generic filter.

Proof.

Formal derivation spine

Formal spine for Valuation commutes with the automorphism action.

$$(\pi\tau)^{\pi G} = \tau^G.$$

$$(\pi\tau)^{\pi G} = \{(\pi\sigma)^{\pi G} : \exists p \in G\, \langle \pi\sigma, \pi p\rangle \in \pi\tau\}.$$

We prove by induction on name rank that

$$(\pi\tau)^{\pi G} = \tau^G.$$

Indeed, $(\pi\tau)^{\pi G}$ consists of the values $(\pi\sigma)^{\pi G}$ for pairs $\langle \pi\sigma, \pi p\rangle \in \pi\tau$ with $\pi p \in \pi G$. This is equivalent to $\langle \sigma, p\rangle \in \tau$ with $p \in G$, and the induction hypothesis turns the constituent value into σ^G. Thus the two sets have exactly the same elements. This covariance is the semantic counterpart of the syntactic symmetry lemma for forcing.

Further explanation. The induction compares the same pair at every rank. We have

$$(\pi\tau)^{\pi G} = \{(\pi\sigma)^{\pi G} : \exists p \in G\, \langle \pi\sigma, \pi p\rangle \in \pi\tau\}.$$

By the induction hypothesis $(\pi\sigma)^{\pi G} = \sigma^G$, and the condition clause is equivalent to $\langle \sigma, p\rangle \in \tau$. The resulting set is exactly τ^G. This covariance identity is why automorphisms of names correctly model symmetry of the interpreted objects. □

Theorem 63.3 (Symmetry lemma for forcing)**.** *For every forcing formula φ, automorphism π, condition p, and names $\bar\tau$,*

$$p \Vdash \varphi(\bar\tau) \quad \Longleftrightarrow \quad \pi p \Vdash \varphi(\pi\bar\tau).$$

Commented symbolic trace: the symmetry lemma is equivariance of forcing

The equivalence

$$p \Vdash \varphi(\bar{\tau}) \quad \Longleftrightarrow \quad \pi p \Vdash \varphi(\pi\bar{\tau})$$

says that forcing commutes with a relabelling of coordinates. Atomic forcing is checked first using the recursive action on names; Boolean connectives preserve the equivalence immediately from the induction hypothesis; quantifiers use that π is a bijection on names. This equivariance is what later allows finite-support information to control which objects survive in the symmetric model.

Proof.

Formal derivation spine

Formal spine for Symmetry lemma for forcing.

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}.$$

$$p \Vdash \varphi(\bar{\tau}) \iff \pi p \Vdash \varphi(\pi\bar{\tau}).$$

$$(\pi\tau)^{\pi G} = \tau^{G}.$$

For an automorphism π of $\mathbb{P}$, define recursively

$$\pi\tau = \{\langle \pi\sigma, \pi p\rangle : \langle \sigma, p\rangle \in \tau\}.$$

Induction on name rank gives $(\pi\tau)^{\pi G} = \tau^{G}$. A simultaneous induction on the definition of forcing proves the symmetry lemma

$$p \Vdash \varphi(\bar{\tau}) \iff \pi p \Vdash \varphi(\pi\bar{\tau}).$$

The stabilizer $\mathrm{sym}(\tau) = \{\pi : \pi\tau = \tau\}$ is a subgroup. These facts make support arguments compatible with forcing truth rather than merely with raw names.

Point specific to this result. The proof is an induction on the forcing relation: automorphisms preserve order, compatibility, and the recursive action on names. Therefore applying an automorphism to both a condition and every parameter preserves exactly what that condition forces.

Further explanation. For atomic formulas, the claim follows from the recursive definitions of forcing together with preservation of compatibility under automorphisms. Boolean connectives follow immediately by induction. In the existential step, if p forces that some name is a witness, applying π to that witness and to every stronger condition shows that πp forces the transformed existential statement. Applying π^{-1} gives the reverse implication. Thus the forcing relation is equivariant under the entire automorphism group. □

Remark 63.4 (Source note)**.** The symmetry lemma is the formal bridge between automorphisms of forcing and invariance of symmetric names; compare the symmetric-model treatments in [11, 12, 13].

Proposition 63.5 (Stabilizers and support calculus for names)**.** *For a name τ,*

$$\mathrm{sym}(\tau) = \{\pi : \pi\tau = \tau\}$$

is a subgroup of the automorphism group. A support condition may be expressed by requiring a designated subgroup such as $\mathrm{Fix}(S)$ *to lie inside* $\mathrm{sym}(\tau)$*.*

Proof.

Formal derivation spine

Formal spine for Stabilizers and support calculus for names.

$$\mathrm{sym}(\tau) := \{\pi \in \mathcal{G} : \pi\tau = \tau\}.$$

$$(\pi\rho)\tau = \pi(\rho\tau) = \pi\tau = \tau;$$

$$H \le \operatorname{sym}(\tau).$$

For a name τ, define

$$\operatorname{sym}(\tau) := \{\pi \in \mathcal{G} : \pi\tau = \tau\}.$$

It is a subgroup: the identity fixes τ; if π, ρ fix τ, then

$$(\pi\rho)\tau = \pi(\rho\tau) = \pi\tau = \tau;$$

and if $\pi\tau = \tau$, then applying π^{-1} gives $\pi^{-1}\tau = \tau$. A subgroup $H \le \mathcal{G}$ supports τ when

$$H \le \operatorname{sym}(\tau).$$

If H supports τ, then the conjugate subgroup $\pi H \pi^{-1}$ supports $\pi\tau$, because

$$(\pi h \pi^{-1})(\pi\tau) = \pi(h\tau) = \pi\tau.$$

This support transport is the group-theoretic input used by normal filters. *Point specific to this result.*

A support is a small set of coordinates whose pointwise stabilizer fixes the name. Intersections/unions of supports correspond to intersections of stabilizer subgroups, providing the closure calculations used for pairing, separation, and the Cohen family.

Further explanation. If $H = \operatorname{sym}(\tau)$, then H records exactly the automorphisms under which the name is unchanged. A support S is a concrete device for producing a subgroup contained in H: every permutation fixing S pointwise fixes the name. Intersections of supports support finite tuples, and conjugating a support subgroup corresponds to transporting the support by the automorphism. These elementary facts are precisely what normality of the filter packages abstractly. □

Worked example

Worked Example 63.6 (Acting on a coordinate name)**.** If π permutes Cohen coordinates and $\dot{c}_n$ is the standard name for the n-th coordinate real, then recursively $\pi\dot{c}_n = \dot{c}_{\pi(n)}$. Consequently $(\pi\dot{c}_n)^{\pi G} = \dot{c}_n^G$. The formula makes clear why both the name and the generic filter move.

A useful warning

Fixing a name syntactically and fixing its value in one generic extension are different statements. The support theory is formulated at the level of names and justified semantically by the symmetry lemma.

Looking ahead

The next chapter, *Symmetric and hereditarily symmetric names*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 64

Symmetric and hereditarily symmetric names

Definitions and setup

Formal definitions used in this chapter

Normal filter of subgroups. A family $\mathcal{F}$ of subgroups of $\mathcal{G}$ is a normal filter iff

$$H \in \mathcal{F},\ H \le K \le \mathcal{G} \Longrightarrow K \in \mathcal{F},$$

$$H, K \in \mathcal{F} \Longrightarrow H \cap K \in \mathcal{F},$$

and

$$H \in \mathcal{F},\ \pi \in \mathcal{G} \Longrightarrow \pi H \pi^{-1} \in \mathcal{F}.$$

Symmetric name. A name τ is symmetric iff

$$\mathrm{sym}(\tau) := \{\pi \in \mathcal{G} : \pi\tau = \tau\} \in \mathcal{F}.$$

Hereditarily symmetric name. Recursively,

$$\tau \in \mathrm{HS} \Longleftrightarrow \mathrm{sym}(\tau) \in \mathcal{F} \ \wedge\ \forall \langle \sigma, p \rangle \in \tau\ (\sigma \in \mathrm{HS}).$$

Fix a normal filter $\mathcal{F}$ of subgroups of the automorphism group. A name is symmetric when its stabilizer belongs to $\mathcal{F}$; it is hereditarily symmetric when every name in its transitive closure is symmetric. The symmetric model N consists of values of hereditarily symmetric names. Normality is what makes symmetry stable under conjugation and hence under the forcing automorphism action.

How this chapter fits together

A normal filter of subgroups $\mathcal{F}$ determines exactly which stabilizers are admitted: a name τ is symmetric precisely when $\mathrm{sym}(\tau) \in \mathcal{F}$. A symmetric name has an allowed stabilizer; a hereditarily symmetric name requires the same property all the way down its transitive closure. Evaluating precisely these names gives the intermediate model N, which contains the ground model and all ordinals but not every set in the full generic extension.

Proposition 64.1 (Normal filters of subgroups and symmetric names)**.** *Let $\mathcal{G} \leq \mathrm{Aut}(\mathbb{P})$. A normal filter of subgroups is a nonempty family $\mathcal{F}$ such that*

$$H \in \mathcal{F},\ H \leq K \leq \mathcal{G} \Rightarrow K \in \mathcal{F}, \qquad H, K \in \mathcal{F} \Rightarrow H \cap K \in \mathcal{F},$$

and

$$H \in \mathcal{F},\ \pi \in \mathcal{G} \Rightarrow \pi H \pi^{-1} \in \mathcal{F}.$$

A name τ is symmetric when $\mathrm{sym}(\tau) := \{\pi \in \mathcal{G} : \pi\tau = \tau\} \in \mathcal{F}$*, and it is hereditarily symmetric when it is symmetric and every constituent name occurring recursively in it is hereditarily symmetric.*

Proof.

Formal derivation spine

Formal spine for Normal filters of subgroups and symmetric names.

$$H \in \mathcal{F} \Longrightarrow \pi H \pi^{-1} \in \mathcal{F} \qquad (\pi \in \mathcal{G}).$$

$$\tau \text{ symmetric} \Longleftrightarrow \mathrm{sym}(\tau) \in \mathcal{F}.$$

$$\tau \in \mathrm{HS} \Longleftrightarrow \tau \text{ is symmetric and } \forall \langle \sigma, p\rangle \in \tau\ (\sigma \in \mathrm{HS}).$$

A normal filter $\mathcal{F}$ on subgroups of $\mathcal{G}$ satisfies: if $H \in \mathcal{F}$ and $H \leq K$, then $K \in \mathcal{F}$; if $H, K \in \mathcal{F}$, then $H \cap K \in \mathcal{F}$; and

$$H \in \mathcal{F} \Longrightarrow \pi H \pi^{-1} \in \mathcal{F} \qquad (\pi \in \mathcal{G}).$$

Define

$$\tau \text{ symmetric} \Longleftrightarrow \mathrm{sym}(\tau) \in \mathcal{F}.$$

Define hereditary symmetry recursively by

$$\tau \in \mathrm{HS} \Longleftrightarrow \tau \text{ is symmetric and } \forall \langle \sigma, p\rangle \in \tau\ (\sigma \in \mathrm{HS}).$$

Thus symmetry is a stabilizer condition, while hereditary symmetry additionally constrains every constituent name. *Point specific to this result.* Normality under conjugation is what makes symmetry invariant under the group action on names. Closure upward and under finite intersections is what permits several supported parameters to be used together in one construction. □

Theorem 64.2 (Hereditary symmetry is preserved by the required name operations)**.** *Let $(\mathbb{P}, \mathcal{G}, \mathcal{F})$ be a symmetric system, let* HS *be the class of hereditarily symmetric names, and let $\tau, \sigma, \rho_1, \ldots, \rho_n \in$ HS. Then canonical pair and union names built from these names lie in* HS*, and $\pi\tau \in$ HS for every $\pi \in \mathcal{G}$. Moreover, for any formula $\varphi(x, \bar{y})$, the bounded subname*

$$\tau \restriction \varphi(\bar{\rho}) := \{\langle \eta, p\rangle \in \tau : p \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho})\}$$

is hereditarily symmetric. Every name in the transitive closure of an HS name is HS by definition.

Proof.

Formal derivation spine

Formal spine: combine supports by intersection and transport them by conjugation. If H supports τ and K supports σ, then $H \cap K \in \mathcal{F}$ supports the pair/union constructions. If H supports τ, then $\pi H \pi^{-1}$ supports $\pi\tau$.

Choose support subgroups in $\mathcal{F}$ for the finitely many names under consideration. Closure of $\mathcal{F}$ under finite intersections gives a common support H. Every element of H fixes each input name, hence fixes the canonical pair name and the canonical union name obtained from them. Their constituent names are already HS, so the resulting names are hereditarily symmetric.

If H supports τ and $\pi \in \mathcal{G}$, then for $h \in \pi H \pi^{-1}$, write $h = \pi k \pi^{-1}$ with $k \in H$. Then

$$h(\pi\tau) = \pi(k\tau) = \pi\tau.$$

Normality gives $\pi H \pi^{-1} \in \mathcal{F}$, and rank induction transports hereditary symmetry to all constituents. Thus $\pi\tau \in \mathrm{HS}$.

For the bounded subname, let $H \in \mathcal{F}$ be a common support for τ and the parameter names $\bar{\rho}$. If $h \in H$, the symmetry lemma gives

$$p \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho}) \iff hp \Vdash_{\mathrm{sym}} \varphi(h\eta, h\bar{\rho}) = \varphi(h\eta, \bar{\rho}).$$

Since $h\tau = \tau$, the action of h permutes exactly the pairs in $\tau \restriction \varphi(\bar{\rho})$. Hence H supports this subname. Every constituent η already occurs in the transitive closure of the HS name τ, so it is HS. Therefore the bounded subname is hereditarily symmetric. □

Proposition 64.3 (Evaluation of hereditarily symmetric names gives a transitive class N)**.** *For an M-generic filter G, define*

$$N = \{\tau^G : \tau \in \mathrm{HS}^M\}.$$

Then N is a transitive class.

Proof.

Formal derivation spine

Formal spine for Evaluation of hereditarily symmetric names gives a transitive class N.

$$\pi H \pi^{-1} \in \mathcal{F}$$

Let $N = \{\tau^G : \tau \text{ is hereditarily symmetric}\}$. If $x \in y \in N$, choose a hereditarily symmetric name τ with $y = \tau^G$. By definition of valuation, $x = \sigma^G$ for some constituent name σ in the transitive closure of τ and some condition in G. Hereditary symmetry of τ implies that σ is hereditarily symmetric, so $x \in N$. Hence N is transitive. Check names are fixed by every automorphism and are hereditarily symmetric, so the ground model is contained in N; in particular every ordinal, represented by its check name, lies in N.

Further explanation. Take $x \in y \in N$. Choose a hereditarily symmetric name τ with $\tau^G = y$. By the definition of valuation, $x = \sigma^G$ for some constituent name σ appearing below τ with a condition in G. Hereditary symmetry of τ implies hereditary symmetry of σ, so $x \in N$. Hence N is transitive. This argument also explains why ordinary symmetry of the top name is insufficient: symmetry must persist through all constituents. □

Theorem 64.4 (N contains the ground model and all ordinals)**.** *The symmetric model N contains the ground model and all ordinals:*

$$M \subseteq N \subseteq M[G], \qquad \mathrm{Ord}^N = \mathrm{Ord}^M.$$

Proof.

Formal derivation spine

Formal spine for N contains the ground model and all ordinals.

$$\pi \check{x} = \check{x}.$$

$$x = \check{x}^G \in N.$$

$$\pi H \pi^{-1} \in \mathcal{F}$$

For every ground-model set x, the check name $\check{x}$ is fixed by every automorphism: induction on rank gives

$$\pi \check{x} = \check{x}.$$

Its stabilizer is therefore the whole automorphism group, which belongs to every normal filter. The same induction shows that all constituents of $\check{x}$ are hereditarily symmetric. Hence

$$x = \check{x}^G \in N.$$

Thus the entire ground model lies in the symmetric submodel.

In particular every ground-model ordinal belongs to N. Ordinary forcing adds no new ordinals, and $N \subseteq M[G]$; therefore N and the ground model have exactly the same ordinals.

Thus the symmetric model changes which sets are available without changing the ordinal scale. This fact is used repeatedly when comparing finite supports, cardinal statements, and the ambient generic extension. □

Worked example

Worked Example 64.5 (Why hereditary symmetry is needed)**.** A name τ could itself be fixed by many automorphisms while containing a constituent name σ with no acceptable support. If $\sigma^G \in \tau^G$, transitivity of the intended symmetric model would then fail. Requiring symmetry recursively for every constituent prevents this problem.

A useful warning

Symmetry of a top-level name alone is not enough. The hereditary clause is what makes the evaluated class transitive.

Looking ahead

The next chapter, *ZF in a symmetric extension, elementary axioms*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Symmetric forcing lemma. Let HS be the class of hereditarily symmetric names. Restrict the forcing recursion to formulas whose parameters lie in HS, and in the quantifier clauses allow only names in HS. Denote the resulting relation by $p \Vdash_s \varphi$. The same induction used for the ordinary Truth Lemma, with the symmetry lemma at the existential step, gives

$$N \models \varphi(\bar{\tau}^G) \quad \Longleftrightarrow \quad (\exists p \in G)\; p \Vdash_s \varphi(\bar{\tau}) \qquad (\bar{\tau} \in \mathrm{HS}).$$

Moreover $p \Vdash_s \varphi(\bar{\tau})$ is invariant under forcing automorphisms: $p \Vdash_s \varphi(\bar{\tau})$ iff $\pi p \Vdash_s \varphi(\pi\bar{\tau})$. This is the semantic bridge used below when Separation and Collection are verified inside the symmetric submodel rather than in the full generic extension.

Chapter 65

ZF in a symmetric extension, elementary axioms

Definitions and setup

Formal definitions used in this chapter

Symmetric model. $N = \{\tau^G : \tau \in \mathrm{HS}\}$.
Symmetric forcing relation. The ordinary forcing relation restricted to hereditarily symmetric names, with the symmetry lemma used to keep witnesses within the support system.

Because N is a transitive subclass of a generic extension, Extensionality and Foundation are inherited, but set-formation axioms need names that remain hereditarily symmetric. Pairing/Union/Infinity use canonical supported name operations. Separation uses the symmetry lemma to show the defining subname has the common support. Collection is already nontrivial and prepares Replacement.

How this chapter fits together

The symmetric model is transitive, so Extensionality and Foundation are inherited, but the constructive axioms still require symmetric names. Pairing and Union combine finitely many supports. Separation uses the symmetry lemma to define an invariant subname. Collection is subtler because witnesses chosen independently may have unrelated supports; the proof closes a set of witnesses under the relevant group action.

Theorem 65.1 (Extensionality/Foundation/Pairing/Union/Infinity in N)**.** *The transitive symmetric model N satisfies Extensionality, Foundation, Pairing, Union, and Infinity.*

Proof.

Formal derivation spine

Formal spine: the elementary axioms are preserved by HS name operations. If $\sigma, \tau \in \mathrm{HS}$, the canonical pair and union names are again HS, with stabilizers containing a finite intersection of the original stabilizers. Check names satisfy

$$\check{x} \in \mathrm{HS}, \qquad \check{\omega}^G = \omega.$$

Since $N \subseteq M[G]$ is transitive, Extensionality and Foundation are inherited; the HS constructions give Pairing, Union, and Infinity.

Extensionality and Foundation hold in N because N is transitive and has the ambient membership relation. If $x = \sigma^G$ and $y = \tau^G$ are in N, choose support subgroups for σ and τ; their intersection belongs to the normal filter and supports the canonical pair name, so $\{x, y\} \in N$. The same closure under finite intersections supports the canonical union name, yielding Union. The check name $\check{\omega}$ is fixed by the whole group, so Infinity holds.

These constructions prove exactly the elementary axioms listed. Separation, Collection/Replacement, and Power Set require the symmetry lemma and separate support-bounding arguments and are treated in the next two chapters. □

Theorem 65.2 (Separation in N)**.** *If $a, \bar{p} \in N$ and $\varphi(x, \bar{p})$ is a formula, then*

$$\{x \in a : N \models \varphi(x, \bar{p})\} \in N.$$

Thus $N \models$ Separation.

Proof.

Formal derivation spine

Formal spine: Separation must use the symmetric forcing relation. If $a = \tau^G$ and $\bar{p} = \bar{\rho}^G$, define

$$\sigma := \{\langle \eta, q\rangle : \exists r\, (\langle \eta, r\rangle \in \tau,\ q \le r,\ q \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho}))\}.$$

Then $\sigma^G = \{x \in a : N \models \varphi(x, \bar{p})\}$.

Let $\tau, \bar{\rho} \in \mathrm{HS}$, with $a = \tau^G$ and $\bar{p} = \bar{\rho}^G$. Definability of the symmetric forcing relation in the ground model and Separation there show that the displayed σ is a set-name.

We compute its value. If $x \in \sigma^G$, choose η, q, r with $q \in G$, $\langle \eta, r\rangle \in \tau$, $q \le r$, and

$$q \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho}).$$

Then $r \in G$, so $x = \eta^G \in \tau^G = a$; the symmetric Truth Lemma gives

$$N \models \varphi(x, \bar{p}).$$

Conversely, suppose $x \in a$ and $N \models \varphi(x, \bar{p})$. Choose $\langle \eta, r\rangle \in \tau$ with $r \in G$ and $x = \eta^G$. The symmetric Truth Lemma gives some $q_0 \in G$ forcing $\varphi(\eta, \bar{\rho})$. Directedness of G supplies $q \in G$ with $q \le q_0, r$. Monotonicity gives $q \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho})$, hence $\langle \eta, q\rangle \in \sigma$ and $x \in \sigma^G$.

It remains to prove hereditary symmetry. Let $H \in \mathcal{F}$ support τ and every parameter name. For $\pi \in H$, the symmetry lemma for $\Vdash_{\mathrm{sym}}$ gives

$$q \Vdash_{\mathrm{sym}} \varphi(\eta, \bar{\rho}) \iff \pi q \Vdash_{\mathrm{sym}} \varphi(\pi\eta, \bar{\rho}).$$

Since $\pi\tau = \tau$, this equivalence sends the defining pairs of σ bijectively to themselves; thus $H \le \mathrm{sym}(\sigma)$. Every constituent η of σ is a constituent of the HS name τ, so it is HS. Therefore $\sigma \in \mathrm{HS}$, and the separated subset belongs to N. □

Lemma 65.3 (Symmetric collection lemma)**.** *If $\tau, \bar{p} \in \mathrm{HS}$ and symmetric forcing yields*

$$\Vdash_{\mathrm{sym}} \forall x \in \tau\, \exists y\, \varphi(x, y, \bar{p}),$$

then there is an HS name σ with

$$\Vdash_{\mathrm{sym}} \forall x \in \tau\, \exists y \in \sigma\, \varphi(x, y, \bar{p}).$$

Proof.

Formal derivation spine

Formal spine: choose HS witnesses on antichains and close them under one support subgroup. Let $H \in \mathcal{F}$ support τ and the parameters. For each constituent $\langle \rho, r\rangle \in \tau$, choose a maximal antichain $A_{\rho,r}$ below r and HS names η_a such that

$$a \Vdash_{\mathrm{sym}} \varphi(\rho, \eta_a, \bar{p}) \quad (a \in A_{\rho,r}).$$

Close the selected names under H and put them into one HS collecting name.

Work over the ground model $M \models \mathrm{ZFC}$, as fixed for the symmetric construction. Choose $H \in \mathcal{F}$ supporting τ and all parameter names. Fix a constituent $\langle \rho, r\rangle \in \tau$. The assumed symmetric forcing statement implies that below every condition $q \le r$ there is a stronger condition $s \le q$ and an HS name η with

$$s \Vdash_{\mathrm{sym}} \varphi(\rho, \eta, \bar{p}).$$

Using the maximum-principle argument inside M, choose a maximal antichain $A_{\rho,r}$ below r and, for each $a \in A_{\rho,r}$, an HS witness name $\eta_{\rho,r,a}$ satisfying

$$a \Vdash_{\mathrm{sym}} \varphi(\rho, \eta_{\rho,r,a}, \bar{p}). \tag{1}$$

There are set-many constituents and set-many antichain elements, so Choice in the ground model selects all these names as a set S.

Close S under the action of H:

$$S^* := \{\pi\eta : \pi \in H,\ \eta \in S\}.$$

This is a set because H and S are sets. Define

$$\sigma := \{\langle \eta, 1_{\mathbb{P}}\rangle : \eta \in S^*\}. \tag{2}$$

The group H permutes S^*, hence fixes σ. Every member of S^* is HS because automorphisms preserve hereditary symmetry. Thus $\sigma \in \mathrm{HS}$.

Finally let $x \in \tau^G$. Choose $\langle \rho, r\rangle \in \tau$ with $r \in G$ and $x = \rho^G$. The maximal antichain $A_{\rho,r}$ meets G in some a. By (1) and the symmetric Truth Lemma,

$$N \models \varphi(x, \eta_{\rho,r,a}^G, \bar{p}^G).$$

By (2), $\eta_{\rho,r,a}^G \in \sigma^G$. Therefore

$$\Vdash_{\mathrm{sym}} \forall x \in \tau\, \exists y \in \sigma\, \varphi(x, y, \bar{p}).$$

This proves symmetric Collection. The orbit closure is what gives all locally chosen witnesses one common allowed support. □

Theorem 65.4 (Collection in N)**.** *The symmetric model N satisfies Collection: if*

$$N \models \forall x \in a\, \exists y\, \varphi(x, y, \bar{p}),$$

then there is $b \in N$ such that

$$N \models \forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p}).$$

Uses: the symmetric forcing theorem, the symmetry lemma, and the symmetric Collection construction.

Proof.

Formal derivation spine

Formal spine: evaluate the symmetric Collection name. Choose HS names $\tau, \bar{\pi}$ with $\tau^G = a$ and $\bar{\pi}^G = \bar{p}$. The symmetric Collection lemma gives an HS name ρ such that

$$\Vdash_{\mathrm{sym}} \forall x \in \tau\, \exists y \in \rho\, \varphi(x, y, \bar{\pi}).$$

Set $b := \rho^G \in N$. The symmetric Truth Lemma yields

$$N \models \forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p}).$$

Apply the symmetric Collection lemma proved immediately above to a set $a = \sigma^G \in N$ and hereditarily symmetric parameters. It produces a hereditarily symmetric name τ whose value $b = \tau^G$ contains a witness for every $x \in a$. Since τ is hereditarily symmetric, $b \in N$. The Truth/Symmetry lemmas identify the forcing assertion with truth in N, so $N \models \forall x \in a\, \exists y \in b\, \varphi(x, y)$. Thus Collection holds in N.

Further explanation. The bounding name produced by the symmetric Collection lemma is itself hereditarily symmetric and therefore belongs to N. If $N \models \forall x \in a\, \exists y\, \varphi(x, y)$, apply that lemma to obtain $b \in N$ such that every $x \in a$ has some witness $y \in b$. This is exactly the Collection axiom as evaluated in the transitive model N. Replacement will be derived next by adding functionality and using Separation on this bounding set. □

Worked example

Worked Example 65.5 (A symmetric separating subname)**.** Given a hereditarily symmetric name τ and formula φ with supported parameters, retain a constituent $\langle \sigma, p \rangle$ only along conditions that force $\varphi(\sigma)$. Any automorphism fixing the common supports preserves both τ and the forcing relation, so it preserves the new subname. The result is therefore hereditarily symmetric.

A useful warning

Transitivity of N does not automatically give Separation or Collection. The required subsets and witness sets must themselves have hereditarily symmetric names.

Looking ahead

The next chapter, *ZF in a symmetric extension, hard axioms*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 66

ZF in a symmetric extension, hard axioms

Definitions and setup

Formal definitions used in this chapter

Symmetric orbit closure. If $H \in \mathcal{F}$ and A is a set of names, write

$$H \cdot A = \{\pi\sigma : \pi \in H,\ \sigma \in A\}.$$

In symmetric Collection/Replacement, witnesses are chosen so that a common $H \in \mathcal{F}$ fixes the input data, and the bounding family is replaced by its set-sized H-orbit closure; this makes the resulting bounding name hereditarily symmetric.
Internal power set. $\mathcal{P}^N(a) = \mathcal{P}(a) \cap N$, represented by a bounded-rank HS name.

Replacement follows from symmetric Collection for functional relations. Power Set is subtler: for $a \in N$, one must collect exactly the subsets of a that also belong to N. A bounded-rank normalization of hereditarily symmetric subnames produces a ground-model set of candidate names, and symmetry of that candidate family gives an internal power set. Only after these two gates pass may we conclude $N \models \mathrm{ZF}$.

How this chapter fits together

Replacement and Power Set are the main closure tests for a symmetric model. Replacement follows from symmetric Collection plus Separation for a functional relation. Power Set requires a rank bound: every subset of a given symmetric set that lies in N must have an equivalent hereditarily symmetric subname within one set-sized family. Only after this is proved can one assert $N \models \mathrm{ZF}$.

Theorem 66.1 (Replacement in N)**.** *The symmetric model N satisfies Replacement. If $N \models \forall x \in a\, \exists! y\, \varphi(x, y, \bar{p})$, then the image*

$$\{y : \exists x \in a\, \varphi(x, y, \bar{p})\}$$

belongs to N.

Proof.

Formal derivation spine

Formal spine: functional Collection followed by Separation. From

$$N \models \forall x \in a\, \exists! y\, \varphi(x, y, \bar{p}),$$

Collection gives $b \in N$ with

$$N \models \forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p}).$$

Then Separation forms

$$r := \{y \in b : \exists x \in a\, \varphi(x, y, \bar{p})\}.$$

Assume

$$N \models \forall x \in a\, \exists! y\, \varphi(x, y, \bar{p}). \tag{1}$$

The symmetric Collection lemma, interpreted through its Truth Lemma, supplies a set $b \in N$ such that

$$N \models \forall x \in a\, \exists y \in b\, \varphi(x, y, \bar{p}). \tag{2}$$

Apply the already proved Separation schema in N to the set b:

$$r := \{y \in b : \exists x \in a\, \varphi(x, y, \bar{p})\} \in N. \tag{3}$$

By (2), every value associated with a member of a lies in r. Conversely, the definition (3) says that every member of r is associated with some $x \in a$. Functionality in (1) ensures that this is exactly the image of a under the relation φ, with no extraneous alternative values for any input. Thus r is the Replacement set. Power Set and Choice play no role in this deduction; the substantive symmetry work was already carried by Collection and Separation. □

Lemma 66.2 (Symmetric name bounding for subsets)**.** *Let $\sigma \in \mathrm{HS}^M$. The class definition of hereditary symmetry cuts out, by Separation in M, the set*

$$B_\sigma := \{\tau \subseteq \mathrm{dom}(\sigma) \times \mathbb{P} : \tau \in \mathrm{HS}^M\}.$$

For every $b \in N$ with $b \subseteq \sigma^G$, there is $\tau \in B_\sigma$ such that $\tau^G = b$. If $\pi\sigma = \sigma$, then $\pi[B_\sigma] = B_\sigma$.

How to read the symbols: the symmetric Power Set problem

The ambient extension $M[G]$ may have many subsets of $x \in N$ that are not in the symmetric model N. The target is therefore $\mathcal{P}^N(x) = \{y \in N : y \subseteq x\}$. The bounding lemma shows that every such y has an equivalent hereditarily symmetric name from one bounded-rank set. Only after producing that set of candidate names can we form a symmetric name whose valuation collects all internal subsets.

Proof.

Formal derivation spine

Formal spine: normalize an internal subset to a subname of σ.

$$\eta^* = \{\langle \rho, p\rangle \in \mathrm{dom}(\sigma) \times \mathbb{P} : p \Vdash_{\mathrm{sym}} \rho \in \eta\},$$

$$(\eta^*)^G = \eta^G, \qquad \eta^* \in B_\sigma.$$

Fix $a = \sigma^G$ with $\sigma \in \mathrm{HS}$, and let $b = \eta^G \in N$ satisfy $b \subseteq a$, where $\eta \in \mathrm{HS}$. Normalize η relative to σ by

$$\eta^* := \{\langle \rho, p\rangle \in \mathrm{dom}(\sigma) \times \mathbb{P} : p \Vdash_{\mathrm{sym}} \rho \in \eta\}. \tag{1}$$

The symmetric forcing relation is definable in the ground model, so (1) is a set. Let $H_\sigma, H_\eta \in \mathcal{F}$ support σ and η. If $\pi \in H_\sigma \cap H_\eta$, then the symmetry lemma preserves the defining condition in (1), and π also

preserves $\mathrm{dom}(\sigma)$. Thus $H_\sigma \cap H_\eta$ supports η^*. Every constituent ρ of η^* is a constituent of the hereditarily symmetric name σ, so $\eta^* \in \mathrm{HS}$.

We show

$$(\eta^*)^G = b. \tag{2}$$

If $x \in (\eta^*)^G$, then $x = \rho^G$ for some $p \in G$ with $p \Vdash_{\mathrm{sym}} \rho \in \eta$; the symmetric Truth Lemma gives $x \in \eta^G = b$. Conversely, if $x \in b \subseteq a = \sigma^G$, choose a constituent $\rho \in \mathrm{dom}(\sigma)$ with $x = \rho^G$. Since $x \in \eta^G$, the symmetric Truth Lemma gives $p \in G$ with $p \Vdash_{\mathrm{sym}} \rho \in \eta$. Hence $\langle \rho, p\rangle \in \eta^*$ and $x \in (\eta^*)^G$. This proves (2).

By construction $\eta^* \in B_\sigma$, so every internal subset has a representative in the single ground-model set B_σ. Finally, if $\pi\sigma = \sigma$, then $\tau \mapsto \pi\tau$ is a bijection of B_σ with itself: hereditary symmetry is preserved by automorphisms, and π preserves $\mathrm{dom}(\sigma) \times \mathbb{P}$. Thus the stabilizer of σ acts on B_σ by permutations. The Power Set name below will attach to each $\tau \in B_\sigma$ exactly those conditions forcing $\tau \subseteq \sigma$; the symmetry lemma then makes that entire set of pairs invariant. □

Theorem 66.3 (Power Set in N)**.** *For every $a \in N$, the internal power set*

$$\mathcal{P}^N(a) = \{x \in N : x \subseteq a\}$$

belongs to N. Thus $N \models$ Power Set.

Uses: the symmetric Truth Lemma and the preceding bounded-name normalization theorem.

Proof.

Formal derivation spine

Formal spine: collect all bounded HS candidates together with conditions forcing subsethood. For $a = \sigma^G$, set

$$\dot{\mathcal{P}}_N(\sigma) := \{\langle \tau, p\rangle \in B_\sigma \times \mathbb{P} : p \Vdash_{\mathrm{sym}} \tau \subseteq \sigma\}.$$

Then

$$\dot{\mathcal{P}}_N(\sigma)^G = \{b \in N : b \subseteq a\}.$$

Because $B_\sigma \times \mathbb{P}$ is a ground-model set and the symmetric forcing relation is definable, Separation in M forms the displayed name. Let $H \in \mathcal{F}$ support σ. If $h \in H$, then $h\sigma = \sigma$, $h[B_\sigma] = B_\sigma$, and the symmetry lemma gives

$$p \Vdash_{\mathrm{sym}} \tau \subseteq \sigma \iff hp \Vdash_{\mathrm{sym}} h\tau \subseteq \sigma.$$

Thus H fixes $\dot{\mathcal{P}}_N(\sigma)$. Its constituent names lie in $B_\sigma \subseteq \mathrm{HS}$, so the new name is hereditarily symmetric.

Let $b \in N$ and suppose $b \subseteq a = \sigma^G$. By the bounding lemma choose $\tau \in B_\sigma$ with $\tau^G = b$. The symmetric Truth Lemma applied to the true statement $\tau^G \subseteq \sigma^G$ yields a condition $p \in G$ such that

$$p \Vdash_{\mathrm{sym}} \tau \subseteq \sigma.$$

Hence $\langle \tau, p\rangle \in \dot{\mathcal{P}}_N(\sigma)$, so $b = \tau^G$ belongs to its valuation.

Conversely, if $b \in \dot{\mathcal{P}}_N(\sigma)^G$, then $b = \tau^G$ for some $\tau \in B_\sigma$ and some $p \in G$ with $p \Vdash_{\mathrm{sym}} \tau \subseteq \sigma$. The symmetric Truth Lemma gives $b \subseteq a$, and $\tau \in \mathrm{HS}$ gives $b \in N$. Therefore

$$\dot{\mathcal{P}}_N(\sigma)^G = \mathcal{P}^N(a),$$

so $\mathcal{P}^N(a) \in N$. □

Theorem 66.4 (Symmetric-model theorem: N satisfies ZF)**.** *The class N obtained from hereditarily symmetric names is a transitive model of every axiom of ZF.*

Proof.

Formal derivation spine

Formal spine: axiom-by-axiom ledger for $(N, \in)$.

ZF axiom/schema	proved by
Extensionality, Foundation	N transitive inside $M[G]$
Pairing, Union, Infinity	canonical HS name constructions
Separation	$\Vdash_{\mathrm{sym}}$ + symmetry lemma
Collection	symmetric collection lemma
Replacement	Collection + Separation
Power Set	B_σ and the condition-tagged Power Set name

The class N is transitive and contains the ground model and all ordinals. Because its membership relation is the ambient one, Extensionality is inherited immediately. Any nonempty $a \in N$ has an ambient $\in$-minimal element by Foundation in $M[G]$; transitivity puts that minimal element in N, so N satisfies Foundation.

For Pairing, Union, and Infinity, the preceding canonical-name constructions produce HS names from HS inputs and evaluate to the required sets. Separation was proved by forming a subname with the symmetric forcing relation and showing that a common support fixes it. The symmetric Collection lemma supplies one HS set containing witnesses for every element of a given domain. Functional Collection followed by symmetric Separation gives Replacement.

For Power Set, fix $a = \sigma^G \in N$. The bounding lemma gives a ground-model set B_σ containing an HS normalized name for every $b \in N$ with $b \subseteq a$. The condition-tagged name

$$\dot{\mathcal{P}}_N(\sigma) = \{\langle \tau, p\rangle \in B_\sigma \times \mathbb{P} : p \Vdash_{\mathrm{sym}} \tau \subseteq \sigma\}$$

is HS by the symmetry lemma and evaluates exactly to

$$\{b \in N : b \subseteq a\}.$$

Thus N satisfies the internal Power Set axiom.

These arguments establish every axiom and schema of ZF. No construction supplies a global well-order or a choice function, so the theorem deliberately concludes $N \models \mathrm{ZF}$, not $N \models \mathrm{ZFC}$. □

Remark 66.5 (Source note)**.** The basic symmetric-extension construction is a pure-set analogue of permutation-model arguments with atoms. The closure proof for ZF and the failure of Choice should be compared with [11, 12, 13].

Worked example

Worked Example 66.6 (Internal power set of a symmetric set)**.** Let $x = \tau^G \in N$. The ambient extension may contain many subsets of x whose names have no acceptable support. The Power Set axiom in N asks for the set $\{y \in N : y \subseteq x\}$. The bounding lemma supplies a set of hereditarily symmetric subnames of τ representing exactly those internal subsets, and a symmetric name for that set.

A useful warning

Do not import the ambient power set $\mathcal{P}^{M[G]}(x)$ into N. Many of those subsets are deliberately excluded by symmetry.

Looking ahead

The next chapter, *The basic Cohen symmetric model*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 67

The basic Cohen symmetric model

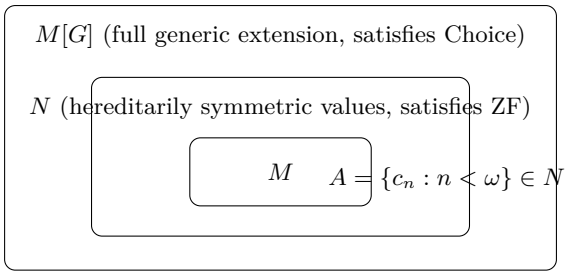

Figure 67.1: The symmetric model sits between the ground model and the full generic extension. Choice can hold in $M[G]$ while failing in the intermediate model N.

Definitions and setup

Formal definitions used in this chapter

Basic system. Force with $\mathrm{Add}(\omega, \omega)$; let finite permutations of ω act on coordinates; let the normal filter be generated by pointwise stabilizers $\mathrm{Fix}(S)$ for finite $S \subseteq \omega$.
Unordered Cohen family. $A = \{c_n : n < \omega\}$, represented by a globally symmetric name.

Use countably many Cohen coordinates together with the group of finite coordinate permutations and the normal filter generated by pointwise stabilizers of finite coordinate sets. Each individual Cohen real has a one-coordinate support. The unordered family of all coordinate reals is invariant under every permutation and therefore has empty support. Every hereditarily symmetric object nevertheless has some finite support.

How this chapter fits together

The basic model uses countably many Cohen coordinates, finite permutations of the coordinates, and the normal filter generated by pointwise stabilizers of finite sets. Individual coordinate reals have small supports; the unordered family of all of them has empty support. This asymmetry between the family and any attempted ordering of it is what destroys Choice.

Theorem 67.1 (The unordered Cohen family A belongs to N)**.** *In the basic Cohen symmetric system, if $\dot{c}_n$ is the n-th Cohen real name, then*

$$\dot{A} = \{\langle \dot{c}_n, 1\rangle : n < \omega\}$$

is hereditarily symmetric, so

$$A = \dot{A}^G = \{c_n : n < \omega\} \in N.$$

Proof.

Formal derivation spine

Formal spine for The unordered Cohen family A belongs to N.

$$\dot{A} = \{\langle \dot{c}_n, 1\rangle : n < \omega\}.$$

$$\pi\dot{A} = \dot{A}.$$

$$A = \dot{A}^G = \{c_n : n < \omega\}$$

In the basic Cohen symmetric system, let $\dot{c}_n$ be the name of the n-th coordinate Cohen real and define

$$\dot{A} = \{\langle \dot{c}_n, 1\rangle : n < \omega\}.$$

This is an *unordered* name: the index n is used only to list the constituents, not as structure that must survive in the symmetric model. A permutation π of the coordinate set sends $\dot{c}_n$ to $\dot{c}_{\pi(n)}$, and therefore merely permutes the constituents of $\dot{A}$. Hence

$$\pi\dot{A} = \dot{A}.$$

The entire family is therefore fixed even though its individual coordinate names are moved around. Consequently the family has empty support; $\dot{A}$ has empty support.

Each coordinate name $\dot{c}_n$ has the finite support $\{n\}$, and its lower-rank constituents have finite supports as well. Thus $\dot{A}$ is hereditarily symmetric. Consequently

$$A = \dot{A}^G = \{c_n : n < \omega\}$$

belongs to the symmetric submodel N, even though the sequence $\langle c_n : n < \omega\rangle$ need not belong to N. □

Theorem 67.2 (A is infinite in N)**.** *The unordered Cohen family $A = \{c_n : n < \omega\}$ is infinite in N: for every $k < \omega$, N contains at least k distinct members of A.*

Proof.

Formal derivation spine

Formal spine for A is infinite in N.

$$c_0, c_1, \ldots, c_{m-1}$$

$$A = \{c_n : n < \omega\}$$

For each natural number m, the coordinate reals

$$c_0, c_1, \ldots, c_{m-1}$$

are elements of A and are pairwise distinct by the dense-set argument from Cohen forcing. Their finite tuple and its range are built from finitely many individually supported objects, so the corresponding finite set belongs to N. Therefore N proves that A has at least m elements for every $m < \omega$.

If A were finite in N, it would be in bijection in N with some natural number m, contradicting the existence in N of $m + 1$ distinct coordinate reals. Hence $N \models$ "A is infinite." This does not produce an enumeration of A; that distinction is the point of the later Dedekind-finiteness theorem. □

Lemma 67.3 (Finite-support theorem for symmetric objects)**.** *In the basic Cohen symmetric system, every hereditarily symmetric name τ has a finite support $S \subseteq \omega$ such that*

$$\mathrm{Fix}(S) \leq \mathrm{sym}(\tau).$$

Proof.

Formal derivation spine

Formal spine for Finite-support theorem for symmetric objects.

$$\mathrm{Fix}(S)$$

$$\mathrm{Fix}(S) \leq \mathrm{sym}(\tau).$$

$$A = \{c_n : n < \omega\}$$

In the basic symmetric system, the normal filter is generated by the pointwise stabilizers

$$\mathrm{Fix}(S)$$

of finite coordinate sets $S \subseteq \omega$. If a name τ is symmetric, its stabilizer $\mathrm{sym}(\tau)$ lies in this generated filter. By definition of a generated filter, some finite S satisfies

$$\mathrm{Fix}(S) \leq \mathrm{sym}(\tau).$$

Thus every permutation fixing S pointwise fixes τ; S is a finite support for the name.

A hereditarily symmetric name has the same property at every constituent. Hence every object of the symmetric model has some representing name controlled by finitely many coordinates. This finite-support theorem is the exact input used in the permutation contradiction showing that no enumeration of the unordered Cohen family can belong to N. □

Worked example

Worked Example 67.4 (The unordered family has empty support)**.** For $\dot{A} = \{\langle \dot{c}_n, 1\rangle : n < \omega\}$, a coordinate permutation merely rearranges the constituents, so $\pi\dot{A} = \dot{A}$ for every π. Thus $\dot{A}$ has empty support even though each individual $\dot{c}_n$ is moved by permutations that move n.

A useful warning

A set can be highly symmetric even when none of its individual elements is fixed. That distinction is the key feature of the unordered Cohen family.

Looking ahead

The next chapter, *Failure of Choice*, uses the constructions just established. Keep the hypotheses of the present chapter explicit when carrying them forward.

Chapter 68

Failure of Choice

Definitions and setup

Formal definitions used in this chapter

Finite-support contradiction. A purported enumeration/injection in N has finite support S; a permutation fixing S but moving an output coordinate must simultaneously fix and move the value.
Failure of AC. An infinite Dedekind-finite set cannot exist under Choice.

The finite-support contradiction targets a putative injection/enumeration of the unordered Cohen family. A finite support cannot protect every output coordinate; swapping a fresh output coordinate while fixing the support must both fix the function and move that output. This proves the family is infinite Dedekind-finite. Since AC implies every infinite set has a countably infinite subset, Choice fails in the symmetric model.

How this chapter fits together

The final contradiction assumes an injection from ω into the unordered Cohen family. A name for that function has finite support, so one can move an output coordinate outside the support while fixing the function name and input. Compatibility of the original and permuted conditions then forces two distinct coordinate reals to be equal. The resulting infinite Dedekind-finite set witnesses failure of Choice.

Theorem 68.1 (A is Dedekind-finite in N)**.** *In the basic Cohen symmetric model, the infinite set $A = \{c_n : n < \omega\}$ is Dedekind-finite:*

$$N \models \text{“there is no injection } \omega \to A\text{”}.$$

Uses: the finite-support theorem and a coordinate permutation fixing the support but moving one Cohen real.

Commented symbolic trace: finite support versus an alleged injection

Assume $f : \omega \to A$ belongs to the symmetric model. A name for f has finite support S. Choose an output coordinate $k \notin S$ and a fresh $k' \notin S$, then swap k and k' while fixing S. Support says the permutation fixes the function name and the input n; symmetry says the corresponding forcing statement is preserved; compatibility lets one condition see both conclusions. That condition would force $c_k = c_{k'}$, contradicting the already proved distinctness of coordinate Cohen reals.

Proof.

Formal derivation spine

Formal spine for A is Dedekind-finite in N.

$$\mathrm{Fix}(S) \le \mathrm{sym}(\dot f). \tag{D1}$$

$$f(n) = c_k. \tag{D2}$$

$$\pi \dot f = \dot f,$$

Proof map. Use forcing conditions rather than only values in one extension. Decide one value $\dot f(\check n) = \dot c_k$ with k outside the finite support. Choose a fresh coordinate k' so the transposed condition remains compatible with the original. The symmetry lemma makes the permuted condition decide the same function value as $\dot c_{k'}$; a common strengthening then contradicts distinctness of the coordinate reals.

Suppose, toward contradiction, that $f : \omega \to A$ is injective in the symmetric model, and choose a hereditarily symmetric name $\dot f$ supported by a finite coordinate set S. Some condition p decides $\dot f(\check n) = \dot c_k$ for some $k \notin S$; choose a fresh $k' \notin S \cup \{k\}$ outside the finite coordinate support of p except where necessary. Let π swap k and k' while fixing S. Then $\pi \dot f = \dot f$ and $\pi \check n = \check n$, so by the symmetry lemma πp forces $\dot f(\check n) = \dot c_{k'}$. The choice of fresh k' makes p and πp compatible; let q extend both. Then q forces $\dot c_k = \dot c_{k'}$, contradicting the dense-set proof that different coordinate Cohen reals are distinct. Thus no injection $\omega \to A$ belongs to N, while A is infinite. Hence A is Dedekind-finite.

Detailed formal verification. Suppose, toward contradiction, that $f : \omega \to A$ is injective in N. Choose an HS name $\dot f$ and a finite support $S \subseteq \omega$ such that

$$\mathrm{Fix}(S) \le \mathrm{sym}(\dot f). \tag{D1}$$

Since f is injective and $A = \{c_k : k < \omega\}$ is infinite, choose n and $k \notin S$ with

$$f(n) = c_k. \tag{D2}$$

Choose $k' \notin S \cup \{k\}$ and let π transpose k, k' while fixing every element of S. From (D1),

$$\pi \dot f = \dot f,$$

while $\pi \check n = \check n$ and

$$\pi \dot c_k = \dot c_{k'}.$$

Applying the automorphism/symmetry lemma to the forced equality represented by (D2) yields

$$f(n) = c_{k'}.$$

But the Cohen coordinate reals are pairwise distinct, so $c_k \neq c_{k'}$, contradicting injectivity and (D2). Therefore

$$N \models \neg\exists f\, (f : \omega \hookrightarrow A),$$

so A is Dedekind-finite. □

Remark 68.2 (Source note)**.** The finite-support permutation contradiction is the characteristic mechanism of the basic Cohen symmetric model; compare [11, 12].

Corollary 68.3 (N does not satisfy AC)**.** *The symmetric model N does not satisfy the Axiom of Choice, because AC implies that every infinite set has a countably infinite subset, while $A \in N$ is infinite and Dedekind-finite.*

Proof.

Formal derivation spine

Formal spine for N does not satisfy AC.

$$\omega \longrightarrow A$$

The previous theorem gives an infinite set $A \in N$ for which there is no injection $\omega \to A$ in N. If N satisfied the Axiom of Choice, then A could be well ordered. Because A is infinite, its well-order type would be an infinite ordinal and hence would contain an initial segment of order type ω. Composing that inclusion with the well-ordering bijection would give an injection

$$\omega \longrightarrow A$$

belonging to N, contradicting Dedekind-finiteness.

Therefore $N \not\models \mathrm{AC}$. The contradiction uses only the well-ordering consequence of AC proved in Volume V and the already established ZF model N. □

Theorem 68.4 (Con(ZF) implies Con(ZF + not-AC) using the formal transfer machinery)**.** *Using the formal consistency-transfer machinery for symmetric extensions,*

$$\mathrm{Con}(\mathrm{ZF}) \Longrightarrow \mathrm{Con}(\mathrm{ZF} + \neg\mathrm{AC}).$$

Uses: the symmetric-model ZF theorem, failure of Choice in the basic Cohen model, and formal consistency transfer.

Proof.

Formal derivation spine

Formal spine for Con(ZF) implies Con(ZF + not-AC) using the formal transfer machinery.

$$\mathrm{Con}(\mathrm{ZF}) \Longrightarrow \mathrm{Con}(\mathrm{ZF} + \neg\mathrm{AC}).$$

The symmetric construction is formalizable over finite fragments of ZF in the same metatheoretic manner as the ordinary forcing construction. The preceding chapters verify, uniformly from the ground axioms required by the construction, that the hereditarily symmetric names satisfy every axiom of any prescribed finite fragment of ZF and that the basic Cohen symmetric model contains an infinite Dedekind-finite set, hence satisfies ¬AC.

Assume Con(ZF). If ZF + ¬AC were inconsistent, one finite proof of contradiction would use only a finite fragment S of its axioms. Choose a finite ground fragment $S_0 \subseteq \mathrm{ZF}$ sufficient for the symmetric construction and the verification of S. Consistency of ZF gives consistency of S_0, completeness gives a model of S_0, and the formal symmetric construction supplies a model of S, contradiction. Hence

$$\mathrm{Con}(\mathrm{ZF}) \Longrightarrow \mathrm{Con}(\mathrm{ZF} + \neg\mathrm{AC}).$$

□

Worked example

Worked Example 68.5 (Finite support kills an enumeration)**.** Suppose $p \Vdash \dot{f}(\check{n}) = \dot{c}_k$ with k outside the finite support S of $\dot{f}$. Swap k with a fresh k' while fixing S, choosing k' so that p and πp are compatible. Symmetry gives $\pi p \Vdash \dot{f}(\check{n}) = \dot{c}_{k'}$. A common strengthening therefore forces $c_k = c_{k'}$, contradicting their previously proved distinctness.

A useful warning

The failure of Choice occurs in the symmetric intermediate model N, not in the full Cohen extension $M[G]$, which still satisfies Choice when the ground model does.

End-of-volume perspective

The basic symmetric Cohen model completes the second independence arc: ordinary forcing supplies a rich ambient ZFC extension, while hereditary symmetry selects an inner model of ZF in which an infinite Dedekind-finite set witnesses failure of Choice.

Glossary of recurring terms and notation

Axiom of Choice (AC)
: The assertion that every family of nonempty sets has a choice function; in ZF it is equivalent to the well-ordering theorem and to Zorn's lemma.

Boolean-valued model
: A semantics in which formulas receive values in a complete Boolean algebra rather than only 0 or 1. It provides a formal bridge from forcing calculations to relative consistency.

ccc
: The countable chain condition: every antichain is countable. For Cohen forcing it is proved by a Δ-system argument.

CH / GCH
: The Continuum Hypothesis is $2^{\aleph_0} = \aleph_1$. GCH asserts $2^{\kappa} = \kappa^{+}$ for every infinite cardinal κ.

Cohen real
: A binary sequence generic for the finite-partial-function forcing $\mathrm{Add}(\omega, 1)$ over the chosen ground model.

Completeness
: In first-order logic, semantic consequence implies formal derivability. This is distinct from a theory being complete, which means it decides every sentence of its language.

Condensation
: The theorem identifying the transitive collapse of suitable elementary substructures of L_θ with earlier levels L_β.

Consistency
: A theory is syntactically consistent if it does not derive a contradiction. Stronger notions used in Volume III include 1-consistency, ω-consistency, and Σ_1-soundness.

Constructible hierarchy
: $L_0 = \varnothing$, $L_{\alpha+1} = \mathrm{Def}(L_\alpha)$, and $L_\lambda = \bigcup_{\beta<\lambda} L_\beta$ at limits; L is their union over all ordinals.

Dedekind-finite
: A set with no injection from ω into it. In ZF such a set can be infinite; the basic symmetric Cohen model supplies the key example in Volume X.

Dense set
: A subset D of a forcing poset such that every condition has a stronger extension in D. Generic filters meet every dense set belonging to the ground model.

Diagonal lemma
: For a formula $\varphi(x)$, a suitable arithmetic theory proves $\theta \leftrightarrow \varphi(\ulcorner \theta \urcorner)$ for some sentence θ.

Elementary substructure
: $A \preccurlyeq M$ means that every first-order formula with parameters from A has the same truth value in A and M.

Forcing name A recursively built set of pairs $\langle \sigma, p \rangle$ whose value is interpreted relative to a generic filter.

Forcing relation $p \Vdash \varphi(\bar{\tau})$ is the ground-model relation expressing that every relevant generic extension containing p satisfies $\varphi(\bar{\tau}^G)$.

Generic filter A directed, upward-closed filter meeting every dense subset of the forcing poset that belongs to the ground model under discussion.

Gödel numbering An effective coding of finite syntactic objects by natural numbers, chosen so that formation, substitution, and proof checking are arithmetically representable.

Inner model A transitive class containing every ordinal and satisfying the specified fragment of set theory with the inherited membership relation.

Mostowski collapse A well-founded extensional relation is uniquely isomorphic to membership on a transitive set.

Nice name Under ccc forcing, a normalized name for a subset of a ground-model set, organized by antichains deciding membership. Nice names make counting arguments possible.

Primitive recursive A class of number-theoretic functions generated from basic functions by composition and primitive recursion. Much of syntactic coding is kept inside this class.

Provability predicate $\mathrm{Prov}_T(y) \equiv \exists p \, \mathrm{Prf}_T(p, y)$, where Prf_T represents the effective proof relation for T.

Reflection A finite collection of formulas, with chosen parameters, agrees between the universe and a sufficiently large rank-initial segment V_α.

Relative consistency A metamathematical implication such as $\mathrm{Con}(T) \Rightarrow \mathrm{Con}(T + \varphi)$. It is not an assertion that T proves its own consistency.

Satisfaction The recursively defined relation $M \models \varphi[\bar{a}]$ between a set-sized structure, a formula, and an assignment.

Skolem hull The closure of a parameter set under chosen Skolem functions; Tarski–Vaught then gives an elementary substructure.

Support A finite or otherwise controlled set of coordinates whose pointwise stabilizer fixes a symmetric name. Support is the mechanism that makes the failure of Choice visible in Volume X.

Symmetric name A forcing name whose stabilizer lies in a fixed normal filter of subgroups; hereditarily symmetric names are symmetric throughout their transitive name structure.

Transitive model A set or class M is transitive if $x \in y \in M$ implies $x \in M$. Transitivity gives important absoluteness properties but does not by itself imply closure under Power Set or Replacement.

Truth Lemma In a generic extension, $M[G] \models \varphi(\bar{\tau}^G)$ iff some $p \in G$ forces $\varphi(\bar{\tau})$.

ω-consistency A theory does not prove an existential statement while proving the negation of every standard numeral instance of its matrix.

Sources and references

[1] Leon Henkin. "The Completeness of the First-Order Functional Calculus." *Journal of Symbolic Logic* 14 (1949), 159–166.

[2] Herbert B. Enderton. *A Mathematical Introduction to Logic*, 2nd ed. Academic Press, 2001.

[3] Kurt Gödel. "Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik* 37 (1930), 349–360.

[4] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I." *Monatshefte für Mathematik und Physik* 38 (1931), 173–198.

[5] J. Barkley Rosser. "Extensions of Some Theorems of Gödel and Church." *Journal of Symbolic Logic* 1 (1936), 87–91.

[6] M. H. Löb. "Solution of a Problem of Leon Henkin." *Journal of Symbolic Logic* 20 (1955), 115–118.

[7] Alfred Tarski. "The Concept of Truth in Formalized Languages. In *Logic, Semantics, Metamathematics*, translated by J. H. Woodger, Oxford University Press, 1956, 152–278.

[8] Kurt Gödel. *The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of Set Theory*. Princeton University Press, 1940.

[9] Paul J. Cohen. "The Independence of the Continuum Hypothesis." *Proceedings of the National Academy of Sciences* 50 (1963), 1143–1148.

[10] Paul J. Cohen. "The Independence of the Continuum Hypothesis, II." *Proceedings of the National Academy of Sciences* 51 (1964), 105–110.

[11] Paul J. Cohen. *Set Theory and the Continuum Hypothesis*. W. A. Benjamin, 1966.

[12] Thomas J. Jech. *The Axiom of Choice*. North-Holland, 1973.

[13] Thomas Jech. *Set Theory: The Third Millennium Edition, Revised and Expanded*. Springer, 2003.

[14] Kenneth Kunen. *Set Theory*. College Publications, 2011.

Index

Absoluteness, 124
Adding Cohen reals, 233
Atomic forcing, 184
Automorphisms of forcing and names, 261
Axiom of Choice, 286
    failure in symmetric model, 282
Axiom of Choice and equivalents, 114

Boolean-valued model, 218, 286
Boolean-valued semantics, 208
Boolean-valued ZFC, 213

Canonical well-order and Choice, 162
Cantor theorem, 89
Cardinal arithmetic with choice, 117
Cardinals without assuming choice, 104
ccc, 286
CH, 286
CH and its failure, 248
Chain conditions and closure, 222
Coding finite objects, 32
Cohen forcing
    ccc, 237
    failure of CH, 249
Cohen real, 286
compactness
    first-order, 23
Compactness and Löwenheim–Skolem, 23
completeness, 286
    first-order, 21
    propositional, 8
Condensation, 165
Condensation lemma, 166, 286
consistency, 286
Consistency and soundness notions, 68
Consistency transfer, 217
constructible universe, 286
    Choice in L, 163
    GCH in L, 171
    ZF in L, 160
continuum
    Cohen forcing calculation, 246

Dedekind-finite set, 281, 286
Definability and Def(M), 142
dense set, 286
diagonal lemma, 56, 286

Elementary hulls, 131
elementary substructure, 286

Failure of Choice, 281
First-order syntax, substitution, and equality, 10
fixed-point lemma, 56
forcing
    name, 287
    Power Set, 200
    relation, 287
    Replacement, 198
Formal deduction, 16
Formalized provability, 72
Foundation and rank, 94
Fraenkel–Mostowski precursor, 258
Fundamental Theorem of Forcing, 193

GCH, 171, 286
GCH in L, 169
General forcing relation, 188
generic filter, 287
Gödel completeness theorem, 21
Gödel incompleteness theorem
    first, 62
    second, 78
Gödel numbering, 287
Gödel numbering of syntax, 46
Gödel's first theorem, 60

Hartogs lemma, 104
Henkin completeness, 19
Homogeneity and relative consistency, 252

independence
    Axiom of Choice, 283
    CH, 254
inner model, 287
Inner models and the Skolem paradox, 135

Lower bound for the continuum, 240
Löb and second incompleteness, 76
Löb theorem, 76
Löwenheim–Skolem theorem
    downward, 24

upward, 25

Mostowski collapse theorem, 108, 287

Names and valuation, 180
Natural numbers and recursion, 91
nice name, 227, 287
Nice names and preservation bookkeeping, 226

omega-consistency, 287
Ordinal arithmetic, 100
Ordinals, 97

Posets, dense sets, and generics, 176
Primitive recursion, 29
primitive recursive, 287
Proof and provability predicates, 49
Propositional proof systems, 7
provability predicate, 50, 287

Recursive and computably enumerable relations, 35
Reflection, 128
Reflection theorem, 129, 287
relative consistency, 287
    forcing transfer, 219
    not CH, 253
relativization, 121
Representability, 42
Rosser and essential undecidability, 64
Rosser incompleteness theorem, 65

satisfaction, 287
Satisfaction for set-sized structures, 120
Self-reference, 55
Separative quotients and Boolean completion, 205
Skolem hull, 132, 287
Skolem paradox, 136
Structures and satisfaction, 13
support, 287
Symmetric and hereditarily symmetric names, 265
symmetric model
    Power Set, 274
    Replacement, 273
symmetric name, 287
Syntax and propositional semantics, 3

Tarski undefinability theorem, 81
Tarski–Vaught criterion, 25
The basic Cohen symmetric model, 278
The ccc, 236
The L-hierarchy, 146
transitive model, 287
Truth and generic extension theorem, 192
Truth Lemma
    forcing, 192, 287
Truth, reflection, and iterated consistency, 81

Uniform definability and absoluteness of L, 150
Upper bound by nice names, 243

Weak arithmetic, 38
Well-founded relations and collapse, 108

ZF axioms and elementary constructions, 88
ZF axioms in L, part I, 154
ZF axioms in L, part II, 158
ZF in a symmetric extension, elementary axioms, 269
ZF in a symmetric extension, hard axioms, 273
ZFC in generic extensions, 197