

# Forcing and the independence of the continuum hypothesis

Jesse Han

September 13, 2018

## Abstract

In these notes, intended as the plaintext part of the Flypitch project, we give a complete account of the independence of the continuum hypothesis from ZFC, with special attention paid to comparing the different approaches: generic sets, Boolean-valued models, and double-negation sheaves.

## 1 Preliminaries

### 1.1 First-order logic

#### 1.1.1 General logical symbols

**Definition 1.1.** We reserve the following general logical symbols:

$\neg$	not
$\vee$	or
$\wedge$	and
$\forall$	for all
$\exists$	exists
$=$	equals
$(, )$	parentheses
$(x_i)_{i:\mathbb{N}}, (y_i)_{i:\mathbb{N}}, (z_i)_{i:\mathbb{N}}$	variables

#### 1.1.2 First-order languages

**Definition 1.2.** A (first-order, one-sorted) **language**  $\mathcal{L}$  comprises the following data:

- (i) A collection of **constant symbols**  $\text{Const}(\mathcal{L})$ ,
- (ii) a collection of **relation symbols**  $\text{Rel}(\mathcal{L})$ ,
- (iii) a collection of **function symbols**  $\text{Funct}(\mathcal{L})$ , and
- (iv) an assignment of each symbol  $S : \text{Const}(\mathcal{L}) \cup \text{Rel}(\mathcal{L}) \cup \text{Funct}(\mathcal{L})$  to a natural number  $\text{arity}(S) : \mathbb{N}$ .

Whenever we interpret a language on some carrier  $A$ , we mean for constants  $c$  to be interpreted as elements of  $A^{\text{arity}(c)}$ , relations  $R$  to be interpreted as subsets of  $A^{\text{arity}(R)}$ , and for function symbols to be interpreted as functions  $A^{\text{arity}(f)} \rightarrow A$ .

**Example 1.3.** • The language of groups comprises a 1-ary constant symbol for the identity and a 2-ary function for group multiplication.

- The language of rings comprises constant symbols 0 and 1 and 2-ary functions for addition and multiplication.
- The language of set theory comprises just one 2-ary relation  $\in$ .

### 1.1.3 Terms, formulas, and sentences

**Definition 1.4.** A **term** is a string of symbols defined by structural induction as follows:

1. Any variable  $v$  is a term.
2. Any constant  $c$  is a term.
3. If  $t_1, \dots, t_n$  are terms of arities  $a_1, \dots, a_n$ , then  $(t_1, \dots, t_n)$  is a term of arity  $a_1 + \dots + a_n$ .
4. If  $t$  is a term and  $f$  is a function symbol with matching arities, then  $ft$  is a term.

Whenever we interpret our language on a carrier  $A$ , we mean for terms to be interpreted as functions into  $A$  which we can construct by composing existing constants (constant functions), basic functions (i.e. the interpretations of the function symbols), and variables (identity).

**Definition 1.5.** A **formula** is defined by structural induction as follows:

1. If  $t_1$  and  $t_2$  are terms of the same arity,  $t_1 = t_2$  is a formula.
2. If  $t$  is a term and  $R$  is a relation symbol, and  $t$  and  $R$  have the same arity, then  $Rt$  is a formula.
3. If  $\varphi$  is a formula,  $\neg\varphi$  is a formula.
4. If  $\varphi$  and  $\psi$  are formulas, then  $\varphi \vee \psi$  is a formula.
5. If  $\varphi$  and  $\psi$  are formulas, then  $\varphi \wedge \psi$  is a formula.
6. If  $\varphi$  is a formula containing a variable  $v$ , then  $\exists v\varphi$  is a formula.
7. If  $\varphi$  is a formula containing a variable  $v$ , then  $\forall v\varphi$  is a formula.

**Definition 1.6.** Let  $\varphi$  be a formula containing the variables  $x_1, \dots, x_n$ . We say that the variable  $x_k$  is **free** if  $x_k$  is not contained in a subformula of the form  $\exists x_k\psi$  or  $\forall x_k\psi$ .

$x_k$  is **bound** if it is not free.

**Definition 1.7.** A formula is a **sentence** (or **statement**) if it contains no free variables.

We write  $\text{Formulas}(\mathcal{L})$  for all the first-order formulas of  $\mathcal{L}$ , and we write  $\text{Sentences}(\mathcal{L})$  for all the first-order sentences of  $\mathcal{L}$ .

By convention, we always include sentences called **true** and **false**.

### 1.1.4 Predicate calculus and provability

Throughout this section, we fix a language  $\mathcal{L}$ .

**Definition 1.8.** A **propositional function** is a function  $f : \text{Prop}^k \rightarrow \text{Prop}$ , for some  $1 < k : \mathbb{N}$  which we define inductively as follows:

1. Each projection  $(P_1, \dots, P_k) \mapsto P_j$  is a propositional function.
2. If  $f$  and  $g$  are propositional functions, so are

$$\neg f, f \wedge g, f \vee g, f \rightarrow g, \text{ and } f \leftrightarrow g,$$

where the operations above are carried out pointwise in  $\text{Prop}$ .

$f$  is a **tautology** if  $\vdash \forall \vec{p} : \text{Prop}^k, f\vec{p} \leftrightarrow \text{true}$ .

**Definition 1.9.** A **propositional combination** is a function  $f : \text{Sentences}(\mathcal{L})^k \rightarrow \text{Sentences}(\mathcal{L})^k$ , for some  $1 < k : \mathbb{N}$  which we define inductively as follows:

1. Each projection  $(B_1, \dots, B_k) \mapsto B_j$  is a propositional combination.
2. If  $f$  and  $g$  are propositional combinations, so are

$$\neg f, f \wedge g, f \vee g, f \rightarrow g, \text{ and } f \leftrightarrow g,$$

where the operations are carried out pointwise in  $\text{Sentences}(\mathcal{L})$ .

By sending projections to projections and symbols  $(\neg, \wedge, \vee, \rightarrow, \leftrightarrow)$  to the corresponding operations on  $\text{Prop}$ , every propositional combination  $f : \text{Sentences}(\mathcal{L})^k \rightarrow \text{Sentences}(\mathcal{L})^k$  can be realized as a propositional function  $\mathbf{r}(f) : \text{Prop}^k \rightarrow \text{Prop}$ .

**Definition 1.10.** The **predicate calculus** comprises the following rules for deducing sentences from other sentences. We call deducible sentences **valid**, and write  $\vdash_{\mathcal{L}} \varphi$  to mean that the  $\mathcal{L}$ -sentence  $\varphi$  is valid (and to disambiguate from  $\vdash$ , which when used unadorned means “provable in the metatheory”).

- (a) (Rule of the propositional calculus) if  $f$  is a propositional combination taking  $k$  arguments such that  $\mathbf{r}(f)$  is a tautology, then for any  $k$  sentences  $A_1, \dots, A_k$ , the value of the propositional combination  $f(\varphi_1, \dots, \varphi_k)$  is a valid sentence.
- (b) (Rule of modus ponens) If  $A$  and  $A \rightarrow B$  are valid, then  $B$  is valid.
- (c) (Rules of equality)
  - (i)  $\forall x, x = x$ ,  $\forall x \forall y, x = y \wedge y = x$ , and  $\forall x \forall y \forall z, x = y \wedge y = z \rightarrow x = z$  are all valid.
  - (ii) Let  $\varphi(x)$  be a formula whose only free variable is  $x$ . Then

$$\forall x \forall y, (x = y) \rightarrow (\varphi(x) \rightarrow \varphi(y))$$

is valid.

- (d) (Change of variable) If  $A$  is a sentence and  $A'$  represents  $A$  with all instances of a variable  $x$  switched to  $y$ , then  $A \leftrightarrow A'$  is valid.
- (e) (Rule of specialization “ $\forall$ -elimination”) Let  $c$  be any constant symbol, and let  $\varphi(x)$  be a formula whose only free variable is  $x$ . Then  $(\forall x \varphi(x)) \rightarrow \varphi(c)$  is valid.
- (f) (“ $\neg$ -introduction”) If  $\neg A \leftrightarrow (A \rightarrow \text{false})$  is valid.

- (g) (Generalization of constants “ $\forall$ -introduction”) Let  $B$  be a sentence which does not contain the constant  $c$  or the variable  $x$ . Let  $\varphi(x)$  be some formula such that  $\varphi(x) \rightarrow B$  is valid. Then  $\exists x\varphi(x) \rightarrow B$  is also valid.<sup>1</sup>
- (h) (de Morgan laws) Let  $\varphi(x)$  have  $x$  as its only free variable. Let  $B$  be a sentence which does not contain  $x$ . Then the following are valid statements:

$$\begin{aligned} (\neg(\forall x\varphi(x))) &\leftrightarrow (\exists x\neg\varphi(x)) \\ ((\forall x\varphi(x)) \wedge B) &\leftrightarrow ((\forall x(\varphi(x) \wedge B)) \\ ((\exists x\varphi(x)) \wedge B) &\leftrightarrow ((\exists x(\varphi(x) \wedge B)) \end{aligned}$$

**Definition 1.11.** Let  $S$  be a collection of sentences.

- (i) We say that  $A$  is provable from  $S$  if there exist finitely many  $B_1, \dots, B_n : S$  such that  $(B_1 \wedge \dots \wedge B_n) \rightarrow A$  is valid.
- (ii) We say that  $S$  is consistent if **false** is not valid.

**Remark 1.12.** One may wonder why we work with a type of formulas and not with a collection of Props directly. The problem with this is that everything needs to be typed, and so to reason about a predicate (say “ $\in$ ”) using Prop, we need some carrier type  $A$  such that  $\in : A \rightarrow A \rightarrow \text{Prop}$ , so that e.g.  $\in$  satisfies the axioms of set theory. But then what does it mean for some other type  $B$  to have an interpretation of  $\in$  and the axioms it satisfies? There then needs to be a separate predicate  $\in_B : B \rightarrow B \rightarrow \text{Prop}$  satisfying the same *kind* of Props as  $\in : A \rightarrow A \rightarrow \text{Prop}$ . We could proceed to define a typeclass of such  $(B, \in_B)$ , and we would then be working with models of set theory, but we would lack a way to reason syntactically about the axioms themselves.

## 1.2 Models and satisfiability

For the remainder of this section we fix a language  $\mathcal{L}$ .

**Definition 1.13.** An  $\mathcal{L}$ -theory is a collection of sentences from  $\text{Sentences}(\mathcal{L})$ .

**Definition 1.14.** An  $\mathcal{L}$ -structure comprises the following data:

- (i) A carrier type  $A$ ,
- (ii) an assignment of every  $c : \text{Const}(\mathcal{L})$  to a  $c^A : A^{\text{arity}(c)}$ ,
- (iii) an assignment of every  $R : \text{Rel}(\mathcal{L})$  to a subtype  $R^A : A^{\text{arity}(R)} \rightarrow \text{Prop}$ ,
- (iv) an assignment of every  $f : \text{Funct}(\mathcal{L})$  to a function  $f^A : A^{\text{arity}(f)} \rightarrow A$ .

**Definition 1.15.** Let  $A$  be an  $\mathcal{L}$ -structure. Using the data of  $A$  being an  $\mathcal{L}$ -structure, we can inductively assign to every term  $t$  (of arity  $k$  and containing  $n$  free variables) a **realization**  $\mathbf{r}(t) : A^n \rightarrow A^k$ , as follows:

1. If  $t = v$  for a variable  $v$ ,  $\mathbf{r}(t) = \text{id}_A = \lambda v, v$ .
2. If  $t = c$  for a constant symbol  $c$ ,  $\mathbf{r}(t) = A^0 \xrightarrow{c^A} A$ .

---

<sup>1</sup>In particular, using the next rule, if  $\neg\varphi(c) \rightarrow \text{false}$  is valid, so is  $\exists x\neg\varphi(x) \rightarrow \text{false}$ , so is  $\neg\exists x\neg\varphi(x)$ , and therefore so is  $\forall x\varphi(x)$ .

3. If  $t = (t_1, \dots, t_m)$ , then  $\mathbf{r}(t) = \mathbf{r}(t_1) \times \dots \times \mathbf{r}(t_m)$ .
4. If  $t = f(t_0)$  for some function symbol  $f$ , then  $\mathbf{r}(t) = f^A \circ \mathbf{r}(t_0)$ .

**Definition 1.16.** Let  $A$  be an  $\mathcal{L}$ -structure. Using the data of  $A$  being an  $\mathcal{L}$ -structure, we can inductively assign to every formula  $\varphi(x_1, \dots, x_n)$  (where  $x_1, \dots, x_n$  exhaust the free variables of  $\varphi$ ) a **realization**  $\mathbf{r}(\varphi) : A^n \rightarrow \mathbf{Prop}$ , as follows:

1. If  $\varphi$  is of the form  $t_1 = t_2$ , then  $\mathbf{r}(\varphi)$  is  $\mathbf{r}(t_1) = \mathbf{r}(t_2)$  (where symbolic equality is realized as true equality).
2. If  $\varphi$  is of the form  $R(t)$ ,  $\mathbf{r}(R(t))$  is  $R^A(\mathbf{r}(t))$ .
3. If  $\varphi$  is of the form  $\neg\psi$ , then  $\mathbf{r}(\varphi)$  is  $\neg\mathbf{r}(\psi)$ .
4. If  $\varphi$  is of the form  $\psi \vee \psi'$ , then  $\mathbf{r}(\varphi)$  is  $\mathbf{r}(\psi) \vee \mathbf{r}(\psi')$ .
5. If  $\varphi$  is of the form  $\psi \wedge \psi'$ , then  $\mathbf{r}(\varphi)$  is  $\mathbf{r}(\psi) \wedge \mathbf{r}(\psi')$ .
6. If  $\varphi$  is of the form  $\exists v\psi$ , then  $\mathbf{r}(\varphi)$  is  $\exists v\mathbf{r}(\varphi)$ .
7. If  $\varphi$  is of the form  $\forall v\psi$ , then  $\mathbf{r}(\varphi)$  is  $\forall v\mathbf{r}(\varphi)$ .

In particular, each sentence  $\varphi$  is sent to a  $\mathbf{Prop}$   $\mathbf{r}(\varphi)$ .

**Definition 1.17.** Let  $A$  be an  $\mathcal{L}$ -structure, and let  $\varphi$  be a sentence. We say that  $A$  **satisfies**  $\varphi$ , written

$$A \models \varphi,$$

if  $\vdash \mathbf{r}(\varphi)$ .

**Definition 1.18.** Let  $T$  be an  $\mathcal{L}$ -theory, and let  $A$  be an  $\mathcal{L}$ -structure. We say that  $A$  is a **model** of  $T$  if for every sentence  $\varphi : T$ ,  $A \models \varphi$ .

**Example 1.19.** (Line graph) The **language of graphs**  $\mathcal{L}_{\text{Graph}}$  comprises a single 2-ary relation symbol  $E$ .

The **theory of graphs**  $\text{Graph}$  comprises the sentence  $\forall x \forall y (E(x, y) \leftrightarrow E(y, x))$ .

The natural numbers  $\mathbb{N}$  can be viewed as a model of  $\text{Graph}$  as follows. We realize  $E$  as the set

$$(y = \text{succ } x) \vee (x = \text{succ } y) : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbf{Prop}$$

which is clearly symmetric.

**Example 1.20.** Let  $\mathbf{Mod}(\text{Graph})$  be the collection of graphs.<sup>2</sup> A **graph property** is a map  $P : \mathbf{Mod}(\text{Graph}) \rightarrow \mathbf{Prop}$  such that whenever  $G \simeq G'$ ,  $P(G) \leftrightarrow P(G')$ . We say that  $G$  *satisfies*  $P$  if  $P(G) \leftrightarrow \text{true}$ . A graph property is additionally said to be *monotone* if whenever  $G \subseteq G'$  is a subgraph, then  $P(G') \rightarrow P(G)$ .

For example, the property of being a complete graph is not monotone, while the property of being cycle-free is.

The **graph evasiveness conjecture** says that for every monotone graph property  $P$  and every  $n : \mathbb{N}$ , one needs to ask  $\binom{n}{2}$  questions of the form “is there an edge between  $v$  and  $w$ ” to determine if an arbitrary graph on  $n$  vertices satisfies  $P$ .

---

<sup>2</sup>Warning: this is “large”, so lives in the next universe up: one can interpret a trivial edge relation on *every* type in the current universe.

**Example 1.21.** (Peano arithmetic)

The **language of Peano arithmetic**  $\mathcal{L}_{\text{PA}}$  comprises:

1. A 1-ary constant 0.
2. Three function symbols succ, +,  $\times$ .

The **theory of Peano arithmetic** PA comprises:

1.  $\forall x, s(x) \neq 0$
2.  $\forall x \forall y, (s(x) = s(y)) \rightarrow x = y$
3.  $\forall x, x + 0 = x$
4.  $\forall x \forall y, x + s(y) = s(x + y)$
5.  $\forall x, x \times 0 = 0$
6.  $\forall x \forall y, x \times s(y) = (x \times y) + x$

Schema: For every  $\mathcal{L}_{\text{PA}}$ -formula  $\varphi(x)$  with one free variable  $x$ ,

$$(\varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(\text{succ } x))) \rightarrow \forall x \varphi(x).$$

The **standard model** of PA is  $\mathbb{N}$  with 0 realized as  $0 : \mathbb{N}$ , succ realized as  $\text{succ} : \mathbb{N} \rightarrow \mathbb{N}$ , + realized as  $+$  :  $\mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$ , and  $\times$  realized as  $\times$  :  $\mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$ .

By recursing on the inductive type of valid sentences and replacing every rule of the propositional calculus with the corresponding deduction rule for **Prop**, we can construct for every valid  $\mathcal{L}$ -sentence  $\varphi$  a proof that  $\mathbf{r}(\varphi) \leftrightarrow \text{true}$ .

That is the soundness theorem. (In what follows, taking  $\psi$  to be **true** yields the assertion in the previous paragraph.)

**Theorem 1.22.** (*Soundness theorem*) For every  $\mathcal{L}$ -structure and any sentences  $\varphi, \psi : \text{Sentences}(\mathcal{L})$ ,

$$\vdash_{\mathcal{L}} \varphi \rightarrow \psi \quad \Longrightarrow \quad \vdash \mathbf{r}(\varphi) \rightarrow \mathbf{r}(\psi).$$

This happens regardless of which  $\mathcal{L}$ -structure is doing the realizing. When the  $\mathcal{L}$ -structure itself is a model of a theory  $T$ , then whenever  $T \vdash_{\mathcal{L}} \psi$ , then since there is some sentence  $\varphi : T$  such that  $\vdash_{\mathcal{L}} \varphi \rightarrow \psi$ , **Prop**'s modus ponens tells us that the model satisfies  $\psi$  also.

**Example 1.23.** For example, suppose we're working in the language of graphs expanded with two 1-ary constants  $a$  and  $b$ , and we know that there is some model  $M$  such that  $M$  satisfies the sole axiom that  $E$  is symmetric. We can show

$$\vdash (\forall x \forall y, \mathbf{r}(E)(x, y) \leftrightarrow \mathbf{r}(E)(y, x)) \rightarrow \mathbf{r}(E)(\mathbf{r}(a), \mathbf{r}(b)) \leftrightarrow \mathbf{r}(E)(\mathbf{r}(b), \mathbf{r}(a))$$

because we already know the antecedent and can apply **Prop**'s  $\forall$ -elimination.

The converse of Theorem 1.22 is false. There may be some things which are incidentally true about the model which are not universally valid.

**Example 1.24.** Working again in the language of graphs, consider a complete graph on  $n$  vertices. Call this model  $M$ .  $M$  happens to satisfy the  $\mathcal{L}$ -sentence

$$(\forall x \forall y, E(x, y) \leftrightarrow E(y, x)) \rightarrow (\forall x \forall y \forall z, E(x, y) \wedge E(y, z) \rightarrow E(x, z)),$$

but this is not a valid  $\mathcal{L}$ -sentence. (Indeed, if it were, then the soundness theorem would imply that *every* graph has a transitive edge relation, which is not true.)

It will turn out that we can do the next best thing. If we rule out this kind of exception by requiring that  $M \models \varphi$  for every  $\mathcal{L}$ -structure  $M$  (resp. every model  $M$  of  $T$ ), then it follows that  $\vdash_{\mathcal{L}} \varphi$  (resp.  $T \vdash_{\mathcal{L}} \varphi$ ). This is the completeness theorem.

### 1.3 The completeness theorem

In this section, our goal will be to prove the **completeness theorem**:

**Theorem 1.25.** *Let  $T$  be an  $\mathcal{L}$ -theory.  $T$  is consistent if and only if there exists a model of  $T$ .*

First we will prove that if there exists a model  $M$  of  $T$ , then  $T$  is consistent.

*Proof.* Suppose  $T$  is inconsistent. If  $T$  has a model  $M$ , then by the soundness theorem  $M \models \text{false}$ . By definition, this means that in **Prop**, we have proved that  $\text{true} \leftrightarrow \text{false}$ , a contradiction.  $\square$

It then remains to show that if  $T$  is consistent,  $T$  has a model. We will first prove Theorem 1.25 in the case where the language  $\mathcal{L}$  of  $T$  is *relational*, i.e. has no function symbols. Later, we will show that to every theory  $\mathcal{L}$ -theory  $T$  we can associate a relational language  $\mathcal{L}_{\text{rel}}$  and an  $\mathcal{L}_{\text{rel}}$ -theory  $T_{\text{rel}}$  by replacing function symbols with their graph relations. Then we will show that if  $T$  is consistent, so is  $T_{\text{rel}}$ , and that every model of  $T_{\text{rel}}$  gives rise to a model of  $T$ , which will give the full completeness theorem.

Before proceeding, we prove a lemma, valid for any consistent theory in any language.

**Lemma 1.26.** *Suppose  $T$  is consistent. Let  $\varphi$  be an  $\mathcal{L}$ -sentence. Then  $T \cup \{\varphi\}$  is consistent or  $T \cup \{\neg\varphi\}$  is consistent.*

*Proof.* Suppose that both  $T \cup \{\varphi\}$  and  $T \cup \{\neg\varphi\}$  are inconsistent. Then there exist sentences  $\sigma$  and  $\rho$  from  $T$  such that

$$\vdash_{\mathcal{L}} (\sigma \wedge \varphi) \rightarrow \text{false} \quad \text{and} \quad \vdash_{\mathcal{L}} (\rho \wedge \neg\varphi) \rightarrow \text{false}.$$

By  $\neg$ -introduction, we get

$$\vdash_{\mathcal{L}} \neg(\sigma \wedge \varphi) \quad \text{and} \quad \vdash_{\mathcal{L}} \neg(\rho \wedge \neg\varphi)$$

and by  $\wedge$ -introduction, we get

$$\vdash_{\mathcal{L}} (\neg(\sigma \wedge \varphi)) \wedge (\neg(\rho \wedge \neg\varphi)).$$

Since the finitary de Morgan laws are tautologies in the sense of 1.8, it follows that

$$\vdash_{\mathcal{L}} \neg(\sigma \vee \varphi \vee \rho \vee \neg\varphi).$$

Since the metatheory satisfies the law of the excluded middle, we have that the law of the excluded middle for  $\mathcal{L}$ -formulas is a tautology in the sense of 1.8. Therefore,

$$\vdash_{\mathcal{L}} \neg(\sigma \vee \rho) \Leftrightarrow \vdash_{\mathcal{L}} \neg\sigma \wedge \neg\rho,$$

so by  $\wedge$ -elimination,  $\vdash_{\mathcal{L}} \neg\sigma$  and  $\vdash_{\mathcal{L}} \neg\rho$ , so  $T$  is inconsistent.  $\square$

**Theorem 1.27.** *Suppose that  $S$  is a relational theory, containing no quantifiers, and which is consistent. Then  $S$  has a model.*

*Proof.* We start by choosing a well-ordering of  $S$ , which induces a well-ordering of the constant and relation symbols which appear in  $S$ . In turn, this induces a lexicographic ordering on all sentences of the form  $c_i = c_j$  and  $R_\beta(c_1, \dots, c_n)$  where  $c_i, c_j$  and  $R_\beta$  are constant and relation symbols occurring in  $S$ . Collect these sentences into a single well-ordered set  $(F_\alpha)$ .

Now, we inductively decide whether the  $F_\alpha$  should be true or false consistent with  $S$ . We put  $G_0 \stackrel{\text{df}}{=} F_0$  if  $S \cup \{F_0\}$  is consistent; otherwise we put  $G_0 \stackrel{\text{df}}{=} \neg F_0$ . Similarly, for  $\beta > 0$  we put  $G_\beta \stackrel{\text{df}}{=} F_\beta$  if  $S \cup \{G_\alpha \mid \alpha < \beta\} \cup \{F_\beta\}$  is consistent, and we put  $G_\beta \stackrel{\text{df}}{=} \neg F_\beta$  otherwise.

From the previous lemma, at each stage  $\beta$  of this construction,  $S_\beta \stackrel{\text{df}}{=} S \cup \{G_\alpha \mid \alpha < \beta\}$  is consistent. Since any inconsistency is derivable from finitely many other sentences, the union

$$H \stackrel{\text{df}}{=} \bigcup_{\beta} S_\beta$$

is consistent.

Now, there is a natural equivalence relation on the collection  $\mathcal{C}$  of all constant symbols which occur in  $H$ , given by

$$c \sim_{\mathcal{C}} c' \iff c = c' : H.$$

Since  $\mathcal{C}$  is well-ordered, we may pick the least element of each  $\sim_{\mathcal{C}}$ -class, and collect them as  $\mathcal{C}'$ . We will make  $\mathcal{C}'$  into a model of  $H$ . First, we realize every constant symbol  $c$  as the chosen least representative of its  $\sim_{\mathcal{C}}$ -class.

For every ( $n$ -ary) relation symbol  $R_\beta$ , we realize  $R_\beta$  by putting

$$R_\beta^{\mathcal{C}}(c_1, \dots, c_n) \leftrightarrow (R(c_1, \dots, c_n) : H).$$

It remains to show that  $M$  is a model of  $S$ . Since  $S$  was quantifier-free, then by the inductive definition of formulas, every sentence in  $S$  is a Boolean combination of atomic sentences (precisely the  $F_\alpha$ ) or their negations. Let  $\varphi : S$ . We can additionally rearrange  $\varphi$  into a disjunctive normal form, so that

$$\varphi \equiv \bigvee_{i \leq n} \left( \bigwedge_{j \leq m_i} L_j^i \right),$$

where each  $L_j^i$  is an atomic or negated-atomic sentence. For each disjunctand  $\bigwedge_{j \leq m_i} L_j^i$ , we have each of the  $L_j^i$  belong to  $\{F_\alpha\}$ , so either  $L_j^i$  or  $\neg L_j^i$  belongs to the  $\{G_\alpha\}$ . It follows that if for every  $\bigwedge_{j \leq m_i} L_j^i$ , there exists some  $L_j^i$  such that  $\neg L_j^i$  is in  $\{G_\alpha\}$ , then  $H \vdash \neg\varphi$  is inconsistent. Therefore, there must be some disjunctand  $\bigwedge_{j \leq m_i} L_j^i$  such that every  $L_j^i$  is in  $\{G_\alpha\}$ .

Since  $M$  was designed to satisfy the  $G_\alpha$ , the propositional calculus implies that  $M \models \varphi$ . Since  $\varphi : S$  was arbitrary,  $M \models S$ .  $\square$



**Definition 1.28.** Let us say that two  $\mathcal{L}$ -theories  $T$  and  $T'$  are *equivalent* if every sentence of  $T$  can be proved from  $T'$  and every sentence of  $T'$  can be proved from  $T$ . It is easy to see that if  $T$  and  $T'$  are equivalent,  $T$  is consistent if and only if  $T'$  is consistent.

**Definition 1.29.** We say that a sentence  $\varphi$  is in **prenex normal form** if any quantifiers occurring in  $\varphi$  occur together at the beginning of  $\varphi$ . We say that a theory is in prenex normal form if every sentence in  $T$  is in prenex normal form.

**Lemma 1.30.** *Every theory  $T$  is equivalent to a theory  $T'$  in prenex normal form.*

*Proof.* Apply the change-of-variables rule and the de Morgan rules for quantifiers to change any sentence not in the desired form into one in  $T'$ .  $\square$

**Theorem 1.31.** *Now suppose that  $S$  is a relational theory, which possibly contains quantifiers, and is consistent. Then  $S$  has a model.*

*Proof.* Let  $T$  be a theory whose sentences are either quantifier-free or begin with a quantifier. We expand  $T$  (and the language) as follows: for every sentence in  $T$  of the form  $\exists x\varphi(x)$ , we expand the language by a new constant symbol  $c$  and adjoin to  $T$  the sentence  $\varphi(c)$ , and for every sentence in  $T$  of the form  $\forall x\varphi(x)$  and every constant  $c$  already occurring in  $T$ , we adjoin the sentence  $\varphi(c)$ . We call the result of this process  $T^*$ .

We observe that whenever  $T$  is consistent, so is  $T^*$ : if  $T^* \vdash_{\mathcal{L}} \text{false}$ , then there are finitely many sentences  $\varphi_1, \dots, \varphi_n$  from  $T^*$  such that  $\vdash_{\mathcal{L}} (\bigwedge_i \varphi_i) \rightarrow \text{false}$ . We regroup this conjunction according to whether or not  $\varphi_i$  contains a new constant symbol or not, viz.

$$\vdash_{\mathcal{L}} \left( \bigwedge_i \varphi_i \right) \wedge \left( \bigwedge_j \psi_j(c_j) \right) \rightarrow \text{false},$$

where  $c_j$  are the new constant symbols. Applying the generalization of constants deduction rule and the de Morgan rules, we conclude that

$$\vdash_{\mathcal{L}} \left( \bigwedge_i \varphi_i \right) \wedge \left( \exists x_j \bigwedge_j \psi_j(x_j) \right) \rightarrow \text{false}$$

and therefore

$$\vdash_{\mathcal{L}} \left( \bigwedge_i \varphi_i \right) \wedge \left( \bigvee_j \neg \exists x_j \psi_j(x_j) \right).$$

So, for some  $j$ ,  $\vdash_{\mathcal{L}} \exists x_j \psi_j(x_j)$ , but by construction  $\exists x_j \psi_j(x_j) : T$  for  $\psi_j(c_j)$  to be in  $T^*$ . Therefore,  $T$  is not consistent.

Now let  $S$  be any consistent theory. We put  $S_0 \stackrel{\text{df}}{=} S$  and if  $S_n$  has already been defined, we put  $S_{n+1} \stackrel{\text{df}}{=} (S_n)^*$ . Then we obtain a consistent limit theory  $\bar{S} \stackrel{\text{df}}{=} \bigcup_{n \in \mathbb{N}} S_n$ , and we define the model  $M$  as we did in the quantifier-free case for the quantifier-free part of  $\bar{S}$ .  $\square$

**Proposition 1.32.** *Let  $T$  be an  $\mathcal{L}$ -theory, and let  $T_{\text{rel}}$  be the associated  $\mathcal{L}_{\text{rel}}$ -theory obtained by replacing function symbols with their graphs. Then any model  $M_{\text{rel}} \models T_{\text{rel}}$  can be viewed as a model  $M \models T$ .*

*Sketch.* For every function symbol  $f$  of  $\mathcal{L}$ , we interpret  $f$  as the function specified by the graph relation  $\Gamma_f$  in  $\mathcal{L}_{\text{rel}}$ , which was axiomatized in  $T_{\text{rel}}$  to be the graph of a function. This gives an  $\mathcal{L}$ -structure  $M$ . Since  $M \models T_{\text{rel}}$  and every sentence of  $T_{\text{rel}}$  is either a modified version of a sentence in  $T$  or asserts that a new relation is a graph of a function,  $M \models T$ .  $\square$

**Proposition 1.33.** *Let  $T$  be an  $\mathcal{L}$ -theory. If  $T$  is consistent, then  $T_{\text{rel}}$  is consistent.*

*Sketch.* Suppose towards the contrapositive that  $T_{\text{rel}}$  is inconsistent. Then there is a proof from  $T_{\text{rel}}$  of false. It suffices to show that replacing the graphs  $\Gamma_f$  by the functions  $f$  induces a deduction-preserving map from the valid  $\mathcal{L}_{\text{rel}}$ -sentences to the valid  $\mathcal{L}$ -sentences, for then we will have a proof from  $T$  of false. This can be done by induction and a case-by-case analysis of the rules of deduction.  $\square$

## 1.4 The Henkin construction

In this section, we present an alternate proof of the completeness theorem, due to Henkin, by building “term models”.

TODO(jesse)

## 1.5 The Löwenheim-Skolem theorem

TODO(kody)

## 2 ZFC

## 3 Generic sets: Cohen’s original proof

## 4 Boolean-valued models

## 5 Sheaves and filterquotients