Completeness for FOL

Completeness Theorem for *F*

- **Theorem.** Let *T* be a set of sentences of a first-order language *L* and let **S** be a sentence of the same language. If **S** is a first-order consequence of *T* , then T |- S.

Overview

- Adding Witnessing Constants
- The Henkin Theory
- The Elimination Theorem
- The Henkin Construction

Overview: Adding Witnessing Constants

- Let *L* be a fixed first-order language.
- We want to prove that if a sentence **S** of *L* is a first-order consequence of a set *T* of *L* sentences, then *T* |- **S**.
- The first step is to enrich *L* to a language L_H with infinitely many new constant symbols, known as *witnessing constants*, in a particular manner.

Overview: The Henkin Theory

- We next isolate a particular theory *H* in the enriched language L_H.
- This theory consists of various sentences which are not tautologies but are theorems of first-order logic, plus some additional sentences known as *Henkin witnessing axioms*.
- The latter take the form $\exists x P(x) \rightarrow P(c)$ where c is a witnessing constant.
- The particular constant is chosen carefully so as to make the Henkin Construction Lemma and Elimination Theorem true.

Overview: The Elimination Theorem

- The Henkin theory is weak enough, and the formal system *F* strong enough, to allow us to prove the following theorem.
- Let *p* be any formal first-order proof whose premises are all either sentences of *L* or sentences from *H*, with a conclusion that is also a sentence of *L*.
- We can eliminate the premises from *H* from this proof in favor of uses of the quantifier rules.
- More precisely, there exists a formal proof *p*' whose premises are those premises of *p* that are sentences of *L* and with the same conclusion as *p*.

Overview: The Henkin Construction

- On the other hand, the Henkin theory is strong enough, and the notion of first-order structure wide enough, to allow us to prove the following result.
- For every truth assignment *h* that assigns TRUE to all wffs in *H* there is a first-order structure $\mathfrak{M}_h$ such that $\mathfrak{M}_h \models S$ for all first-order sentences S assigned TRUE by *h*.
- This construction of the structure $\mathfrak{M}_h$ from the truth assignment *h* is sometimes called the *Henkin construction*.

Witnessing Constants

- Given any first-order language *K*, we construct a new first-order language *K'*.
- The language *K'* will have the same symbols as *K* except that it will have additional constant symbols.
- For example, if *K* is our blocks language, then in *K',* we will be able to say things like the following:
  1. $\exists x\,(\mathrm{Small}(x) \land \mathrm{Cube}(x)) \to \mathrm{Small}(c1) \land \mathrm{Cube}(c1)$
  2. $\exists z\,(z = a \land z = b) \to (c2 = a \land c2 = b)$
  3. $\exists y\,\mathrm{Between}(y, a, b) \to \mathrm{Between}(c3, a, b)$
  4. $\exists x\,\exists y\,\mathrm{Between}(a, x, y) \to \exists y\,\mathrm{Between}(a, c4, y)$

Witnessing Constants

- More generally, for each wff **P** of *L* with exactly one free variable, form a new constant symbol c_P, making sure to form different names for different wffs.
- This constant is called the *witnessing constant* for P.

Ensuring Uniqueness of Witnessing Constants

- How do we make sure that distinct wffs get distinct witnessing constants?
- Use a single symbol **c** not in the language K and have the new symbol be the expression **c** with the wff as a subscript.
- For example, in our above list, instead of the constant symbol c_1 we would use the symbol

  $$c_{(Small(x) \wedge Cube(x))}$$
- This may be an awkward symbol to write down, but it shows us how we can generate unique constants.

Extending the Language K

- The language *K’* consists of all the symbols of *K* plus all these new witnessing constants.
- They can now be used in wffs.
- For example, the language *K’* allows us to form sentences like:

  $\text{Smaller}(a, c_{\text{Between}(x,a,b)})$

Extending the Language K

- We also have sentences like:

  $\exists x\ \text{Smaller}(x, c_{\text{Between}(x,a,b)})$

- Using the earlier rule, we would need to be able to write:

  $\exists x\ \text{Smaller}(x, c_{\text{Between}(x,a,b)}) \rightarrow$

  $\text{Smaller}(c_{\text{Smaller}(x,\ c_{\text{Between}(x,a,b)})}, c_{\text{Between}(x,a,b)})$

Extending the Language K

- However, we have not added that particular witnessing constant in forming K'.
- We have to iterate the process of extending a language.
- Starting with a language L, we define an infinite sequence of larger and larger languages

  $$L_0 \subseteq L_1 \subseteq L_2 \subseteq \ldots$$

- Where $L_0 = L$
- Language L_{n+1} results by extending L_n, i.e. $L_{n+1} = L'_n$

Henkin Language

- The *Henkin language* L_H for L consists of all the symbols of L_n for any $n = 0, 1, 2, 3, \ldots$

Date of Birth Lemma

- Each witnessing constant c_P is introduced at a certain stage $n \geq 1$ of this construction.
- Let us call that stage the *date of birth* of c_P .
- **Lemma 1**. Let n+1 be the date of birth of c_P. If **Q** is any wff of the language L_n, then c_P does not appear in **Q**.
- **Proof**. Obvious from the construction of L_H.

Summary of Construction so Far

- We have added witnessing constants for each wff **P** with exactly one free variable.
- The free variable of **P** is going to be important.
- Hence, we often write the wff in a way that reminds us of the free variable, namely, as **P(x)**.
- Consequently, its witnessing constant is now denoted by $\mathbf{c_{P(x)}}$.
- By iterating our construction infinitely often, we have a situation where for each wff **P(x)** of L_H with exactly one free variable, the witnessing constant $\mathbf{c_{P(x)}}$ is also in L_H.
- This allows us to form the sentence $\exists x P(x) \rightarrow c_{P(x)}$ in L_H.

Henkin Witnessing Axiom

- The sentence $\exists x P(x) \rightarrow P(c_{P(x)})$ is known as the Henkin witnessing axiom for $P(x)$.
- The intuitive idea is that $\exists x P(x) \rightarrow P(c_{P(x)})$ asserts that if there is something that satisfies $P(x)$, then the object named by $c_{P(x)}$ provides an example (or “witness”) of one such.

Independence Lemma

- **Lemma 2. (Independence lemma)** If c_P and c_Q are two witnessing constants and the date of birth of c_P is less than or equal to that of c_Q, then c_Q does not appear in the witnessing axiom of c_P.
- **Proof:** If the date of birth of c_P is less than that of c_Q, the result follows from the date of birth lemma.
- If they have the same date of birth, it follows from the fact that different wffs of a language *K* have distinct new witnessing constants in *K'*.

Henkin Theory

- Definition. The *Henkin theory H* consists of all sentences of the following five forms, where **c** and **d** are any constants and P(x) is any formula (with exactly one free variable) of the language L_H:
- H1: All Henkin witnessing axioms

  $\exists x P(x) \rightarrow P(c_{P(x)})$
- H2: All sentences of the form

  $P(c) \rightarrow \exists x P(x)$

Henkin Theory

- H3: All sentences of the form

  $\neg\forall x P(x) \leftrightarrow \exists x \neg P(x)$
- H4: All sentences of the form

  $c = c$
- H5: All sentences of the form

  $(P(c) \wedge c = d) \rightarrow P(d)$

Parallels between H and Quantifier Rules

- There is a parallel between these sentences of *H* and the quantifier rules and identity rules of *F*:
- H1 corresponds roughly to ∃ **Elim**, in that both are justified by the same intuition.
- H2 corresponds to ∃ **Intro**,
- H3 reduces ∀ to ∃,
- H4 corresponds to = **Intro**, and
- H5 corresponds to = **Elim**.

Proposition 3

- **Proposition 3.** Let $\mathfrak{M}$ be any first-order structure for *L*. There is a way to interpret all the witnessing constants in the universe of $\mathfrak{M}$ so that, under this interpretation, all the sentences of *H* are true.
- **Proof:** (Sketch) The basic idea of the proof is that if $\mathfrak{M}$ |= ∃xP(x), then pick any element *b* of the domain that satisfies P(x) and let the witnessing constant $c_{P(x)}$ name *b*.
- If M |= ¬∃xP(x), then let $c_{P(x)}$ name any element of the domain.
- This takes care of the axioms of type H1.
- As for the other axioms, they are all logical truths, and so will turn out to be true no matter how we interpret the new constants.

Elimination Theorem

- **Proposition 4. (The Elimination Theorem)** Let *p* be any formal first-order proof with a conclusion **S** that is a sentence of *L* and whose premises are sentences $P_1, \ldots, P_n$ of *L* plus sentences from *H*. There exists a formal proof *p′* of **S** with premises $P_1, \ldots, P_n$ alone.
- **Proof.** The proof will take up the next several slides.

Deduction Theorem

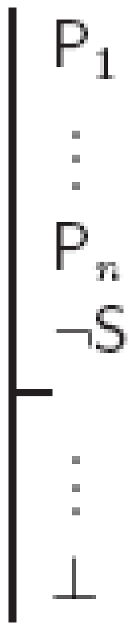
- **Proposition 5. (Deduction Theorem).** If $T \cup \{P\}$ |- Q then T |- P → Q
- **Proof.** The proof of this is very similar to the proof of Lemma 17.2. The details are left as an exercise. The theorem is also illustrated by the "You try it" exercise on p 535.

Reminder: Lemma 17.2

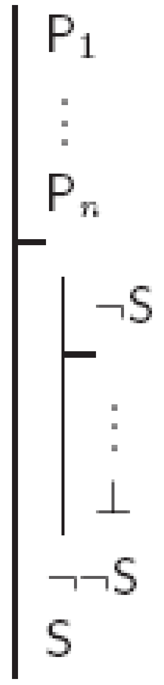
- **Lemma 2.** $T \cup \{\neg S\} \vdash_T \perp$ if and only if $T \vdash_T S$
- **Proof.** Assume $T \cup \{\neg S\} \vdash_T \perp$
- In other words, there is a proof of $\perp$ from premises ¬S and certain sentences $P_1,\ldots,P_n$ of *T*.

Reminder: Proof cont'd

Arrange the premises so that:



Rearrange to:



Reminder: Other direction

- **Proof.** Assume $T \vdash_T S$
- In other words, there is a proof of S from certain sentences $P_1,\ldots,P_n$ of *T*.
- You finish!

Proposition 6

- **Proposition 6.** If $T \cup \{P_1, . . . , P_n\}$ |- Q and, for each $i = 1, . . . , n$, T |- P_i then T |- Q.
- **Proof**. Use the Deduction Theorem and modus ponens repeatedly to get rid of each of the P_i. The details are left as an exercise.