Road map to the Completeness proof for FOL

Completeness 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.

Definition. 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. Example: $c_{(Small(x) \wedge Cube(x))}$

Definition. The language *K'* consists of all the symbols of *K* plus all these new witnessing constants.

Definition. 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 $L_0 = L$ and Language L_{n+1} is L'_n

Definition. The *Henkin language* L_H for *L* consists of all the symbols of L_n for any *n* = 0, 1, 2, 3, . . .

Definition. Each witnessing constant c_P is introduced at a certain stage n ≥ 1 of this construction. Let us call that stage the *date of birth* of c_P .

Lemma 1 (Date of Birth lemma). 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**.

Definition. The sentence $\exists x P(x) \rightarrow P(c_{P(x)})$ is known as the *Henkin witnessing axiom* for P(x).

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.

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:	$P(c) \rightarrow \exists x P(x)$
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:	$(P(c) \wedge c = d) \rightarrow P(d)$

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.

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.

Proposition 5. (Deduction Theorem) If $T \cup \{P\}$ |- Q then *T* |- P → Q

Proposition 6. If $T \cup \{P_1, \ldots, P_n\}$ |- Q and, for each $i = 1, \ldots, n$, *T* |- P_i then T |- Q.

Lemma 7. Let *T* be a set of first-order sentences of some first-order language *L*, and let **P, Q**, and **R** be sentences of *L*.

1. If T|- P → Q and T|- ¬P → Q then T|- Q.
2. If T|- (P → Q) → R then T|- ¬P → R and T|- Q → R.

Lemma 8. Let *T* be a set of first-order sentences of some first-order language *L* and let **Q** be a sentence. Let P(x) be a wff of *L* with one free variable and which does not contain c. If *T* |- P(c) → Q and c does not appear in *T* or Q, then *T* |- ∃xP(x) → Q.

Lemma 9. Let *T* be a set of first-order sentences of some first-order language *L* and let Q be a sentence of *L*. Let P(x) be a wff of *L* with one free variable which does not contain **c**. If *T* ∪ {∃xP(x) → P(c)} |- Q and **c** does not appear in *T* or Q, then *T* |- Q.

Lemma 10. Let *T* be a set of first-order sentences, let P(x) be a wff with one free variable, and let **c** and **d** be constant symbols. The following are all provable in *F*:

P(c) → ∃xP(x)
¬∀x P(x) ↔ ∃x ¬P(x)
(P(c) ∧ c = d) → P(d)
c = c

Theorem (Henkin Construction Lemma) Let *h* be any truth assignment for L_H that assigns TRUE to all the sentences of the Henkin theory *H*. There is a first-order structure $\mathfrak{M}_h$ such that $\mathfrak{M}_h$ |= S for all sentences S assigned TRUE by the assignment *h*.

Definition. Define a binary relation ≡ on the domain of M, i.e., the constants of L_H as follows: c ≡ d if and only if *h*(c = d) = TRUE.

Lemma 11. The relation ≡ is an equivalence relation.

Definition. This allows us to define our desired first-order structure $\mathfrak{M}_h$*:* The domain *D* of our desired first-order structure $\mathfrak{M}_h$ is the set of all such equivalence classes. We let each constant **c** of L_H name its own equivalence class [c]. To make the notation simpler, we assume that R is binary. Relation symbol **R** is the set {<[c], [d]> | *h*(R(c, d)) = TRUE}

Lemma 12. If c ≡ c', d ≡ d', and *h*(R(c, d)) = TRUE, then *h*(R(c', d')) = TRUE.

Lemma 13. For any sentence **S** of L_H, $\mathfrak{M}_h$ |= S if and only if *h*(**S**) = TRUE.