$$
\begin{array}{rcl}
(|\mathcal{A}|, \mu) \models t_1 = t_2 & \Leftrightarrow & \mu(t_1) = \mu(t_2) \\
(|\mathcal{A}|, \mu) \models R_j(t_1, \ldots, t_{r(R_j)}) & \Leftrightarrow & \langle \mu(t_1), \ldots, \mu(t_{r(R_j)}) \rangle \in R_j^{\mathcal{A}} \\
(|\mathcal{A}|, \mu) \models \neg\varphi & \Leftrightarrow & (|\mathcal{A}|, \mu) \not\models \varphi \\
(|\mathcal{A}|, \mu) \models \varphi \vee \psi & \Leftrightarrow & (|\mathcal{A}|, \mu) \models \varphi \text{ or } (|\mathcal{A}|, \mu) \models \psi \\
(|\mathcal{A}|, \mu) \models (\forall x)\varphi & \Leftrightarrow & (\text{for all } a \in |\mathcal{A}|) \\
 & & (|\mathcal{A}|, \mu, a/x) \models \varphi
\end{array}
$$

where
$$
(\mu, a/x)(y) = \begin{cases} \mu(y) & \text{if } y \neq x \\ a & \text{if } y = x \end{cases}
$$

Fitch Proofs for FOL

The Fitch proof system of [BE] can prove FOL formulas as well as propositional ones. We have to add six new proof rules to deal with the new concepts of **identity** and **quantifiers**:

- $=$**-Intro:** Derive $n = n$ (cf. *Atlas Shrugged*?)
- $=$**-Elim:** From $P(n)$ and $n = m$, derive $P(m)$
- $\forall$**-Intro:** (Ordinary form) If for a new variable c you derive $P(c)$, derive $\forall x : P(x)$
- $\forall$**-Intro:** (General conditional form) If from $P(c)$, for a new variable c, you derive $Q(c)$, conclude $\forall x : P(x) \rightarrow Q(x)$
- $\forall$**-Elim:** From $\forall x : S(x)$, derive $S(c)$
- $\exists$**-Intro:** From $S(c)$, derive $\exists x : S(x)$
- $\exists$**-elim:** If from $S(c)$, for a new variable c, you derive Q, then you may derive Q from $\exists x : S(x)$

Coming Attractions:

We will prove Fitch to be **sound** for FOL, following [BE] Section 18.3 with some details on HW#4. The basic idea is very similar to soundness for propositional Fitch. We show by induction on steps of any proof that each statement is true in any **structure** in which all of its premises are true (instead of for any truth assignment).

Then we will prove the **completeness** of Fitch for FOL, following [BE] Chapter 19 with some details on HW#5. The goal is to prove that any FOL-valid sentence can be proved in Fitch. We will do this as follows:

- Define an infinite set of sentences called the **Henkin theory**,
- Show that any propositional extension of the Henkin theory has a model,
- Use propositional completeness to get a propositional Fitch proof of any FOL-valid sentence from the Henkin theory, and finally
- Show that in Fitch we can eliminate every use of the Henkin theory in this proof, to get a Fitch proof of the FOL-valid sentence.

Soundness of Fitch for FOL:

Once again we prove, by induction on all steps in any proof, that every statement is a **FOL consequence** of the premises in force when it occurs. This means that if $\mathcal{M}$ is any structure such that $\mathcal{M} \models P_i$ for every premise in force when Q occurs, then $\mathcal{M} \models Q$.

Since the last step of the proof can use any of our eighteen proof rules, we need eighteen cases in our inductive step. We'll do two of these cases, with a few others to follow on HW#4.

The $\rightarrow$-Elim Case:

The $\exists$-Elim Case:

The cases of the other new Fitch rules are either similar to this case or are even easier.

Making an Existentially Complete Structure:

We now begin the proof of completeness for Fitch. Given a set of sentences $\mathcal{T}$ from which we cannot prove $\bot$, we want to show that $\mathcal{T}$ has a model, a structure in which all of its sentences are true. (This is an equivalent form of completeness: if $\mathcal{T} \cup \{\neg S\}$ has *no* model, it must be that we can derive $\bot$ from $\mathcal{T} \cup \{\neg S\}$ and thus prove S from $\mathcal{T}$ using $\bot$-elim.)

Our first step is to convert $\mathcal{T}$ into an **existentially complete** set of sentences over an expanded vocabulary. We do this by adding an infinite set of **witnessing constants** to the vocabulary. For every formula $P(x)$ in the vocabulary, with exactly one free variable. we add a new constant named $c_{P(x)}$. Eventually we will insist that if there is *any* element such that $P(x)$ is true, then $P(c_{P(x)})$ will be true.

(Note that there is nothing wrong with a vocabulary being infinite! In almost any imaginable computer science application we will want the vocabulary we use to be finite, but everything we have proved *about* FOL systems and Fitch has applied equally well to infinite vocabularies.)

An Interesting Technicality:

We want to have a constant $c_{P(x)}$ for every formula $P(x)$ over the vocabulary. But of course we mean every formula over the *new, improved* vocabulary with the witnessing constants already in it! This leads us to an apparent circularity in the definition.

But we can get around this problem. Let $\mathcal{L}_0$ be the original set of formulas over the original vocabulary. Let $\mathcal{L}_1$ be the set of valid formulas over the vocabulary that includes the original one and witnessing constants for all one-free-variable formulas in $\mathcal{L}_0$. Let $\mathcal{L}_{i+1}$ be the set of valid formulas over the vocabulary that has witnessing constants for all one-free-variable formulas in $\mathcal{L}_i$, for each number i. Our final set of formulas $\mathcal{L}$ is the union of $\mathcal{L}_i$ for all i.

Every witnessing contant has a **date of birth**, the number of the phase of this construction on which it is created. It's easy to see that if a formula of $\mathcal{L}_i$ contains a witnessing constant for a formula containing another witnessing constant b, then the date of birth of b is less than i.

Example of Witnessing Constants:

Let Σ_0 be Σ_T, the Tarski's World vocabulary.

Let $\mathcal{L}_0$ be all formulas over Σ_0.

Let Σ_1 be Σ_0 together with all witnessing constants for one-free-variable formulas in $\mathcal{L}_0$, such as $c_{\text{Cube}(x)}$.

If $c_{\text{Cube}(x)}$ is a true witnessing constant, then $\text{Cube}(c_{\text{Cube}(x)})$ will be true iff $\exists x : \text{Cube}(x)$.

Let $\mathcal{L}_1$ be all formulas over Σ_1, such as

$$\text{Smaller}(y, c_{\text{Cube}(x)}).$$

Then Σ_2 includes witnessing constants for formulas in $\mathcal{L}_1$, such as $c_{\text{Smaller}(y, c_{\text{Cube}(x)})}$.

For this last constant to be a true witnessing constant, we would have

$$\text{Smaller}(c_{\text{Smaller}(y, c_{\text{Cube}(x)})}, c_{\text{Cube}(x)}) \Leftrightarrow$$

$$\exists y : \text{Smaller}(y, c_{\text{Cube}(x)}).$$

The Henkin Axioms:

We want to apply our completeness result for *propositional* Fitch in order to get the completeness result we want for full Fitch. To do this we will create a set of **axioms** for the augmented vocabulary (with the witnessing constants). Every statement that is an FOL consequence of some premises will be a **first-order consequence** of those premises plus the Henkin axioms.

The five classes of Henkin axioms will correspond to the non-propositional proof rules of Fitch. Let $P(x)$ be any formula with exactly one free variable and let c and d be any constants. The Henkin axioms $\mathcal{H}$ consist of:

H1 Every statement of the form $\exists x : P(x) \to P(c_{P(x)})$,

H2 Every statement of the form $P(c) \to \exists x : P(x)$,

H3 Every statement of the form $(\neg\forall x : P(x)) \leftrightarrow (\exists x : \neg P(x))$,

H4 Every statement of the form $c = c$, and

H5 Every statement of the form $(P(c) \wedge (c = d)) \to P(d)$.

Proposition 15.1 *Given any model $\mathcal{M}$ of the vocabulary for $\mathcal{L}_0$, we can interpret the witnessing constants to get a model $\mathcal{M}'$ of the vocabulary for $\mathcal{L}$ such that $\mathcal{M}' \models \mathcal{H}$.*

Proof: The statements in H2, H3, H4, and H5 are true in every FOL structure because they are provable in Fitch and Fitch is sound. We proved half of the generic H3 statement earlier, and the other half is similar. Statements in H2, H4, and H5 have one-line proofs using $\exists$-intro, $=$-intro, and $=$-elim respectively.

So all we need to do is pick the witnessing constants to satisfy all the H1 statements. For every formula $P(x)$ with one free variable, we assign $c_{P(x)}$ to be an element b such that $\mathcal{M} \models P(b)$, if any such element exists. (If no such element exists, any element of the domain will do – why?) More precisely, we pick a b such that $\mathcal{M}' \models P(b)$, where $\mathcal{M}'$ refers to $\mathcal{M}$ with the partial assignment of values for witnessing constants with dates of birth less than that of $c_{P(x)}$. ♠

Continuing With Completeness:

From this proposition we know that if $\mathbf{T} \cup \mathcal{H} \models S$, then $\mathbf{T} \models S$. That is, if a statement holds in any structure $\mathcal{M}'$ in which all the sentences of both $\mathbf{T}$ and $\mathcal{H}$ are true, it holds in any model $\mathcal{M}$ in which $\mathbf{T}$ is true. This is because given $\mathcal{M}$, we can make $\mathcal{M}'$ which satisfies $\mathcal{H}$, still satisfies T, and has the same truth value for S.

But our goal here is to show that if $\mathbf{T} \models S$, then $\mathbf{T} \vdash S$. Later in this lecture we'll show that if $\mathbf{T} \models S$, then S is a propositional consequence of $\mathbf{T} \cup \mathcal{H}$ and thus, by *propositional* completeness), $\mathbf{T} \cup \mathcal{H} \vdash S$. This, as we will now see, is good enough to reach our goal.

Theorem 15.2 (The Elimination Theorem) *Suppose that S is the conclusion of a Fitch proof whose premises are $P_1, \ldots, P_k$ plus a finite set of sentences from $\mathcal{H}$. Then there is a Fitch proof of S from $P_1, \ldots, P_k$ alone.*

Proof:

We can use Fitch directly to prove anything in H2, H3, H4, and H5. The interesting case is to eliminate an arbitrary H1 axiom. We will need a series of lemmas about Fitch proofs:

Lemma 15.3 *If* $\mathbf{T} \cup \{P\} \vdash Q$*, then* $\mathbf{T} \vdash (P \to Q)$.

Proof: With the necessary premises from **T** as assumptions, assume P, prove Q as in the given proof, and then conclude $P \to Q$ by $\to$-elim. ♠

More Lemmas:

Lemma 15.4 *If* $\mathbf{T} \vdash (P \to Q)$ *and* $\mathbf{T} \vdash (\neg P \to Q)$*, then* $\mathbf{T} \vdash Q$*.*

Proof: With the necessary premises in place, derive $P \vee \neg P$ from scratch. Then assume P, prove $P \to Q$, and derive Q by $\to$-elim. Then assume $\neg P$ and derive Q from it in the same way. Then derive Q from $P \vee \neg P$ by $\vee$-elim. ♠

Lemma 15.5 *If* $\mathbf{T} \vdash (P \to Q) \to R$*, then* $\mathbf{T} \vdash \neg P \to R$ *and* $\mathbf{T} \vdash Q \to R$*.*

Proof: Use $\bot$-elim, $\to$-intro. ♠

Still More Lemmas:

Lemma 15.6 *If* $\mathbf{T} \vdash P(c) \to Q$*, and* c *does not appear in* $\mathbf{T}$ *or* Q*, then* $\mathbf{T} \vdash (\exists x : P(x)) \to Q$*.*

Proof: Assume $\exists x : P(x)$, use $\exists$-elim to get $P(c)$, derive $P(c) \to Q$ from $\mathbf{T}$ by the give proof, then conclude Q by $\to$-elim. ♠

Lemma 15.7 *If* $\mathbf{T} \cup \{(\exists x : P(x)) \to P(c)\} \vdash Q$*, then* $\mathbf{T} \vdash Q$*.*

Proof: From the above lemmas and the given proof, we know $\mathbf{T} \vdash ((\exists x : P(x)) \to P(c)) \to Q$, $\mathbf{T} \vdash \neg\exists x : P(x) \to Q$, and $\mathbf{T} \vdash P(c) \to Q$. From this last we get $(\exists x : P(x)) \to Q$, and this is enough to derive Q.

♠

But now we are done with the Elimination Theorem, as repeated use of this last lemma can eliminate each H1 premise in any proof from $\mathbf{T} \cup \mathcal{H}$.

♠

The Henkin Construction:

We now need only one more step to prove the completeness of Fitch.

Theorem 15.8 (Henkin Construction) *Let h be a truth assignment to $\mathcal{L}$, considering all sentences beginning with a quantifier as atomic, that makes every sentence in $\mathcal{H}$ true. Then there is a model $\mathcal{M}_h$ such that for any S, $\mathcal{M}_h \models S$ iff h makes S true.*

How does this construction give us completeness? Suppose S is a first-order consequence of a set of sentences **T**. We claim that S must be a *propositional* consequence of $\mathbf{T} \cup \mathcal{H}$. If it were not, there would be a truth assignment making **T** and $\mathcal{H}$ all true but making S false. From the Henkin construction, there would then be a structure modeling $\mathbf{T} \cup \{\neg S\}$, which contradicts our hypothesis.

Proof: (of Henkin construction) As in Exercise 18.12, we begin by taking the constants themselves, including all the witnessing constants, as the elements of our domain. We set the truth of the atomic formulas from the predicate symbols according to h.

This won't quite do. We are assured that sentences of the form $b = b$ are true because they are in $\mathcal{H}$, but it is possible for h to assign statements of the form $a = b$ true, where a and b are two *different* constants. To be a valid model, our $\mathcal{M}_h$ will have to make a and b the same object.

The way we do this is to make our domain not the set of all constants but a set of *equivalence classes* of constants, where we consider constants a and b to be equivalent iff h makes the formula $a = b$ true. This is the essence of the construction – the rest is checking details to make sure we can get away with this.

Validating the Henkin Construction:

First we must make sure that if a and b are the same object, then $R(a)$ and $R(b)$ have the same truth value according to h. (And if $a = a'$ and $b = b'$ are both true according to h, that $S(a, b)$ and $S(a', b')$ have the same truth value, and so forth. This follows because h makes the H5 axioms true (the ones from $=$-elim), and h is propositionally consistent.

This is the base case of a big induction on all formulas S, showing that this model makes S true iff h does. We have dealt with atomic formulas (identity and other predicates), and the propositional steps of the induction are clear.

By using the H1 and H2 axioms we can carry out the inductive step for $\exists$. To handle $\forall$, we use the appropriate H3 axiom to convert the $\forall$ into an $\exists$ and two $\neg$'s, then apply the inductive steps for $\exists$ and $\neg$.

A final wrinkle comes in if our language has function symbols. For every constant d and function f, we need to define a value for $f(d)$. But since the statement $\exists x : f(d) = x$ is true, it holds for its own witnessing constant $c_{\exists x:f(d)=x}$, and because h says that any other constant c satisfying $f(d) = c$ is equal to this one, all such constants are equivalent and $f(d)$ is a unique object of the domain.

We have finished the proof of completeness for Fitch. ♠

Just as in propositional logic, we can apply completeness to get another useful property:

Theorem 15.9 (Compactness Theorem) *Let* Γ *be a set of sentences. Suppose every finite subset of* Γ *has a model. Then* Γ *has a model.*

Proof: If Γ is inconsistent (meaning that $\perp$ can be proved from Γ in Fitch), then some finite subset of Γ is inconsistent because Fitch proofs are finite.

But no finite subset of Γ can be inconsistent because that set has a model and Fitch is sound.

So Γ is consistent.

By completeness, then, Γ has a model.

♠

The Compactness Theorem has surprising consequences for number theory:

$$\text{Theory}(\mathbf{N}) \;=\; \{\varphi \in \mathcal{L}(\Sigma_N) \mid \mathbf{N} \models \varphi\}$$

$$\Gamma \;=\; \text{Theory}(\mathbf{N}) \;\cup\; \{c > 0, c > 1, c > 2, c > 3, \ldots\}$$

Corollary 15.10 Γ *has a model.*

There is a countable model of Theory($\mathbf{N}$) *that is not isomorphic to* $\mathbf{N}$.

$\mathcal{L}(\Sigma_N)$ *cannot uniquely characterize* $\mathbf{N}$.

Proof: Every finite subset of Γ is satisfiable by $(\mathbf{N}, i)$ for i sufficiently large.

By Compactness, Γ is satisfiable. ♠

Corollary 15.11 *"Connectedness" is not expressible in the first-order language of graphs,* $\mathcal{L}(\Sigma_g)$

Proof:

Suppose that $\chi \equiv$ "I am connected."

$$\Gamma \quad = \quad \{\chi\} \ \cup \ \{\mathrm{DIST}(s,t) > 1, \mathrm{DIST}(s,t) > 2, \ldots\}$$

$$\mathrm{DIST}(x_0, x_n) > n \quad \equiv$$

$$(\forall x_1 \cdots x_{n-1}) \bigvee_{i=0}^{n-1} (x_i \neq x_{i+1} \wedge \neg E(x_i, x_{i+1}))$$

Every finite subset of Γ is satisfiable.

By Compactness, Γ is satisfiable.

This is a contradiction.

Thus "Connectedness" is not expressible in the first-order language of graphs. ♠

The Lowenheim-Skolem Theorem:

Theorem 15.12 *If a set of first-order sentences has any model at all, it has a* **countable** *model.*

Proof: Take the truth assignment to $\mathcal{L}$ arising from the model and use the Henkin Construction to make a model from it. Since there were only countably many constants to put into equivalence classes, there can be only countably many classes and thus the new model has a countable domain. ♠

The set of real numbers and the set of countable binary sequences are **uncountable**. But if we define a first-order vocabulary to talk about either of these sets, we get a first-order theory, the set of sentences that are true. This theory has a countable model!

Thus there is a countable set, with "addition" and "multiplication" operations and a "zero" element, that satisfies exactly the same first-order sentences as does the reals. Of course the new operations are not the usual ones.

In first-order set theory even stranger things happen. You can prove that there are sets that are uncountable, bigger than the reals, and even bigger than that. So there is a countable model that has sets that *the model thinks* are uncountable. The reason this is possible is that the notion of one set "being an element" of another does *not* have its usual meaning in this model.