Formal Definition of a Vocabulary:

A vocabulary Σ is formally made up of three elements:

- The set Φ of **function symbols**, each representing a function from D^k to D for some k. Φ includes the **constant symbols** of the vocabulary, which are thought of as function symbols with $k = 0$.
- The set Π of **predicate symbols**, each representing a **relation** on D, a function from D^k to $\{0, 1\}$ for some k. The **equality sign** is included in Π as a binary relation, written in its usual infix form, e.g. "$s = t$".
- The **arity function** r, which assigns the number of arguments k to each symbol in Φ and Π.

Inductive Definition of FOL Formulas:

Once we fix a vocabulary Σ we have a set $L(\Sigma)$ of **well-formed formulas**. Entities within formulas have two types, "object" and "boolean". We define valid formulas by induction:

Variables: We have an infinite set

$$V = \{x, y, z, x_1, y_1, z_1, \ldots\}$$

Terms: A **term** is a variable, or a function applied to the correct number of terms. A constant is a special case of the latter.

Formulas: A string is a **well-formed formula** if it is:

- An **atomic formula**, which is a predicate symbol applied to the correct number of terms, or the special atomic formula "$s = t$" where s and t are terms,
- A **boolean operator** applied to the correct number of formulas, or
- A **quantified formula** "$\exists x : P$" or "$\forall x : P$" where x is a variable and P is a formula.

A *structure* — also called a *model* — of a vocabulary $\Sigma = (\Phi, \Pi, r)$ is a pair $\mathcal{A} = (U, \mu)$ such that:

$$U = |\mathcal{A}| \neq \emptyset$$

$$\begin{aligned} \mu : V &\to U \\ x &\mapsto x^{\mathcal{A}} \end{aligned}$$

$$\begin{aligned} \mu : \Phi &\to \text{ total functions on } U^{O(1)} \\ \mu : f &\mapsto f^{\mathcal{A}} : U^{r(f)} \to U \end{aligned}$$

$$\begin{aligned} \mu : \Pi &\to \text{ relations on } U^{O(1)} \\ \mu : R &\mapsto R^{\mathcal{A}} \subseteq U^{r(R)} \end{aligned}$$

In propositional logic a **model** was an assignment of a truth value to every atomic formula. In FOL the model must tell us what the objects are, and what the relation and function symbols mean. If the **universe** or domain is finite, we can specify this information by finite lookup tables for each function and relation.

Example of a Binary String Structure:

Let w be the string "01101".

$\mathcal{A}_w = \langle\{0, 1, \ldots, 4\}, <, \{1, 2, 4\}\rangle \in$ STRUC$[\Sigma_s]$

$$\begin{aligned}\Sigma_s \;&=\; (\emptyset, \{=, <, S\}, \{\langle =, 2\rangle, \langle <, 2\rangle, \langle S, 1\rangle\}) \\ &=\; (; <^2, S^1)\end{aligned}$$

1. $(\exists x)(\forall y)(y \leq x \;\wedge\; S(x))$
2. $(\forall xy)((x < y \wedge \neg S(x) \wedge \neg S(y)) \rightarrow (\exists z)(x < z < y))$

Tarski's Inductive Definition of Truth:

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

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

Play Tarski's Truth Game!!!

world: $\mathcal{W}$; sentence: φ; players: A, B

A asserts that $\mathcal{W} \models \varphi$; B denies that $\mathcal{W} \models \varphi$.

The game rules depend inductively on the formula φ:

φ is atomic: A wins iff $\mathcal{W} \models \varphi$.

$\varphi \equiv \alpha \vee \beta$: A asserts $\mathcal{W} \models \alpha$ or A asserts $\mathcal{W} \models \beta$.

$\varphi \equiv \alpha \wedge \beta$: B denies $\mathcal{W} \models \alpha$ or B denies $\mathcal{W} \models \beta$.

$\varphi \equiv \neg\alpha$: A and B switch rôles, and B asserts $\mathcal{W} \models \alpha$.

$\varphi \equiv \exists x(\psi)$: A chooses an element from $|\mathcal{W}|$, assigning it a name n. A asserts that $\mathcal{W}' \models \psi[x \leftarrow n]$.

$\varphi \equiv \forall x(\psi)$: B chooses an element from $|\mathcal{W}|$, assigning it a name n. B denies that $\mathcal{W}' \models \psi[x \leftarrow n]$.

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.

Example of an FOL Proof:

DeMorgan's Law For Quantifiers:

Given $\neg\forall x : P(x)$, prove $\exists x : \neg P(x)$.

Though this is a well-known fact, no Fitch rule gives it to you directly. Here's a proof:

1. Assume $\neg\forall x : P(x)$.
2. Assume $\neg\exists x : \neg P(x)$.
3. Let c be arbitrary.
4. Assume $\neg P(c)$.
5. $\exists x : \neg P(x)$, by $\exists$-intro.
6. $\perp$, by $\perp$-intro from 2, 5.
7. $P(c)$, by $\perp$-elim.
8. $\forall x : P(x)$, by $\forall$-intro.
9. $\perp$, by $\perp$-intro from 1, 8.
10. $\neg\neg\exists x : \neg P(x)$, by $\perp$-elim.
11. $\exists x : \neg P(x)$, by $\neg$-elim.

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.

First, the $\rightarrow$-elim case, to demonstrate how the previous proofs for the propositional Fitch rules carry over. Suppose that the last step uses $\rightarrow$-elim to derive Q from $P \rightarrow Q$ and P. Let $\mathcal{M}$ be any structure such that $\mathcal{M} \models P \rightarrow Q$ and $\mathcal{M} \models P$. By the definition of truth for $\rightarrow$, it must be that $\mathcal{M} \models Q$, which is what we need to prove.

The $\exists$-Elim Case:

Suppose that with the premise $\exists x : P(x)$ in force we said "Let c be such that $P(c)$", and from this we derived the statement R (in which c does not occur). The last step was to conclude R outside of this derivation. Now let $\mathcal{M}$ be a structure such that $\mathcal{M} \models \exists x : P(x)$ and also any other premises used in our derivation.

By the definition of truth for $\exists$, there must be some object b in the universe of $\mathcal{M}$ such that $\mathcal{M} \models P(b)$. Let $\mathcal{M}'$ be the structure obtained from $\mathcal{M}$ by changing the binding of variable c so that its value is b.

We know that $\mathcal{M}' \models P(c)$ by the meaning of c. The proof steps we used to get R from $P(c)$ and the other premises are all sound, so we know that $\mathcal{M}' \models R$. To conclude that $\mathcal{M} \models R$ we need the **irrelevant variable rule**, the easily proved fact that changing the binding of a variable that does not occur in R cannot affect the truth of R.

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 $\perp$, 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 $\perp$ from $\mathcal{T} \cup \{\neg S\}$ and thus prove S from $\mathcal{T}$ using $\perp$-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 sel_0 be the original set of formulas over the original vocabulary. Let sel_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 sel_0. Let sel_{i+1} be the set of valid formulas over the vocabulary that has witnessing constants for all one-free-variable formulas in sel_i, for each number i. Our final set of formulas $\mathcal{L}$ is the union of sel_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 sel_i contains a witnessing constant for a formula containing another witnessing constant b, then the date of birth of b is less than i.

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 14.1 *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 $c_{P(x)}$. ♠