TRUTH AND MODELS

A structure is the equivalent of a truth assignment in sentential logic. There we thought of different possible worlds, resulting in differing assignments of truth and falsity to given sentences. Here, because of the greater complexity of the first-order predicate calculus, we need to consider more than mere truth and falsity: we need to consider the "structure" of the universe which gives rise them.

EXAMPLE: Consider the language of set theory. The parameters (if we adopt Enderton's definition of these) consist of the universal quantifier symbol $\forall$, and the two-place predicate symbol $\in$. Operating purely formally, we can use these to construct formulas by the three step process of terms, atomic formulas, and formulas. Consider the following possible "interpretations":

1. The standard interpretation

\[ \forall \] ranges over all subsets of a set \[ \forall \]
\[ \in \] corresponds to the relation \( \{<x, y>: y \text{ contains } x \text{ and } x \text{ is a singleton} \)\]

2. A number theory interpretation

\[ \forall \] ranges over the natural numbers 1, 2, 3, ...
\[ \in \] corresponds to the relation \( \{<m, n>: m < n \} \)

Now consider the sentence \( \forall x \forall y \neg y \in x \). In the standard interpretation this sentence asserts that there exists a set \( x \) having the property that for any point \( y \), \( y \) is not contained in \( x \). That is, it (correctly) asserts the existence of the empty set. In the number theory interpretation, however, this very same sentence asserts the existence of a least number \( x \) such that no other number \( y \) is strictly less than it. That is, it (correctly) asserts the existence of the number 1. Thus in both cases the sentence asserts something that is true, but for reasons that are very different.
In general, a structure \( U \) specifies

1. A nonempty set, denoted \( |U| \), that the universal quantifier ranges over.
2. To each \( n \)-place predicate symbol \( P \); an \( n \)-ary relation \( P^U \) defined on \( |U|^n \).
3. To each constant symbol \( c \); an element \( c^U \) of \( |U| \).
4. To each \( n \)-place function symbol \( f \); an operation \( f^U : |U|^n \rightarrow |U| \).

If a sentence \( \phi \) is true when interpreted by a structure \( U \), we say \( \phi \) is true in \( U \) and that \( U \) is a model of the sentence \( \phi \).

Recall that a term is something built up from the constants and variables by prefixing function symbols. That is, those expressions that result from successively applying term-building operations arising from the function symbols to the constants and variables. An atomic formula is the result of applying the predicate symbols to the terms. The formulas are the expressions generated from the atomic formulas by the sentential connectives and the universal quantifier.

Let \( f \) be a wff; let \( V \) denote the set of variables; let \( U \) be a structure; and let \( s : V \rightarrow |U| \). Our goal is to make the following

**DEFINITION**: The structure \( U \) satisfies \( \phi \) with \( s \), denoted \( \models_U \phi [s] \).

Heuristically, this means that if we translate any free variable \( s \) by \( s(x) \), and translate \( \phi \) by \( U \), then the result is true. Formally:

**I. Terms.** Define the extension \( s : T \rightarrow |U| \) by recursion.

1. For each variable \( x \), \( s(x) = s(x) \)
2. For each constant symbol \( c \), \( s(c) = c^U \).
3. Given terms \( t_1, t_2, \ldots, t_n \) and \( n \)-place function symbol \( f \),

\[
\overline{s}(ft_1t_2\ldots t_n) = f^U(\overline{s}(t_1), \ldots, \overline{s}(t_n)).
\]
II. Atomic formulas.

The definition is direct:

\[ I = \U \mathbf{t_1 t_2} [s] \iff \mathbf{s}(t_1) = \mathbf{s}(t_2). \]

\[ I = \U \mathbf{P t_1 t_2 \ldots t_n} [s] \iff \mathbf{<s}(t_1), \ldots, \mathbf{s}(t_n) > \U \mathbf{P}. \]

III. Other formulas.

Since the general formula is defined inductively, the definition of \( \mathbf{s} \) must be recursive.

1. For atomic formulas, the definition has already been given.
2. \( I = \U \mathbf{\neg f} [s] \iff \text{not } I = \U \mathbf{ f} [s]. \)
3. \( I = \U \mathbf{ f \land y} [s] \iff \text{either not } I = \U \mathbf{ f} [s] \text{ or } I = \U \mathbf{ y} [s] \text{ or both.} \)
4. \( I = \U \mathbf{x f} [s] \iff \text{for every } d \U \mathbf{U}, I = \U \mathbf{ f} [s(x \mid d)], \)

where \( s(x \mid d) \) coincides with \( s \) except that at \( x \) it assumes the value \( d \). Thus

\[ s(x\mid d)(y) = \begin{cases} s(y), & \text{if } y \neq x; \\ d, & \text{if } y = x. \end{cases} \]

Now we are able to give a formal definition of logical implication within the first-order predicate calculus.

DEFINITION: If \( \{ \} \) is a set of formulas and \( \mathbf{f} \) another formula, then the set \( \{ \} \) logically implies \( \mathbf{f} \), written \( I = \mathbf{ f} \), iff for every structure \( \U \) and every function \( s : V \U \) \( I \U \mathbf{s} \) such that \( \U \) satisfies every member of \( \{ \} \) with \( s \), \( \U \) also satisfies \( \mathbf{f} \) with \( s \).