We read ∀ X as "For every object X. . . ." This example shows how we can use the logical connectives of propositional logic to put together our new "analyzed" sentences. This formalization says that for every object X, X is among the set of objects that are mortal if X is among the objects that are students.

To give an idea of what we expect of the deduction component of this logic, consider the first three sentences concerning Hector, mortality, and students. We have seen the formal expressions for the first two. The formal expression for the third, "Hector is mortal," is simply:

```
mortal(hector).
```

Clearly, if the first two sentences are true in our world, then the third statement, sadly, is also true. The deductive component should be able to capture this semantic relationship among the three sentences.

As another example of capturing an English sentence in predicate logic, consider the statement "All student papers not graded by their authors are graded by an instructor." When rendering this sentence in predicate logic, we first must express the condition that an object is a "student paper." Using predicates **paper**, **student**, and **wrote**, we can describe a student paper as an object X that is a paper such that there is some student Y who wrote X:

```
paper(X), ∃ Y (student(Y), wrote(Y, X))
```

This formula is true of any X that is a student paper. With predicates **instructor** and **graded**, we can express the situation that an instructor graded some object X:

```
∃ Z (instructor(Z), graded(Z, X))
```

To express that every object that is a student paper was graded by an instructor, we use *not* and *or* to express the implication, and quantify over all objects. We arrive at:

```
∀ X ¬(paper(X), ∃ Y (student(Y), wrote(Y, X), ¬graded(Y, X)));
   ∃ Z (instructor(Z), graded(Z, X))
```

To paraphrase this expression: "For every object X, it is not the case that X is a student paper and X was not graded by its author, or some instructor graded X."

These paragraphs have been a brief introduction to predicate logic. The following sections give its syntax, semantics, and deductive theory in more detail than we really care to consider.

5.2.1. *Syntax of Predicate Logic*

The grammar in Figure 5.3 defines the language of predicate logic. The grammar is meant to elucidate the structure of predicate logic expressions rather than to be easy to parse. *Predicate formulas* are similar in structure to propositional formulas,

since a predicate formula can be a disjunction of one or more *predicate terms,* each of which is a conjunction of one or more *predicate factors.* A predicate factor is a negated or unnegated *predicate unit,* which is either an *atom* (sometimes called an *atomic formula*), or a parenthesized predicate expression. We now have *quantifiers,* so a predicate formula can have a quantifier, with an associated variable, in front of it. The quantifier is either the *universal quantifier* (∀) or the *existential quantifier* (∃). The notation ∀ X is read "for all X," and the notation ∃ X is read "there exists X such that." Also, atoms are structured, being either a solitary predicate symbol, or a predicate symbol followed by a list of one or more *terms* enclosed in parentheses. Each term is either a variable or a constant.

Do not confuse "predicate term" with just "term." The first "term" is an algebraic notion indicating an expression that is the conjunction (product) of factors, and that is disjoined (added) with similar expressions. The second "term" means an entity that can be an argument in an atom. When we use "term" with no further qualification, we will always mean the second kind of "term." As before, we will use "expression" and "formula" more or less interchangeably.

```
PREDEXP → PREDFORM
PREDEXP → QUANT PREDEXP
QUANT → ∀ varsym
QUANT → ∃ varsym
PREDFORM → PREDTERM
PREDFORM → PREDTERM ; PREDFORM
PREDTERM → PREDFACT
PREDTERM → PREDFACT , PREDTERM
PREDFACT → PREDUNIT
PREDFACT → ¬ PREDUNIT
PREDUNIT → PREDATOM
PREDUNIT → ( PREDEXP )
PREDATOM → predsym
PREDATOM → predsym ( ARGLIST )
ARGLIST → TERM
ARGLIST → TERM , ARGLIST
TERM → csym
TERM → varsym

where predsym = lc (lc + uc + digit)*
         csym = lc (lc + uc + digit)*
       varsym = uc (lc + us + digit)*
           lc = any lowercase letter
           uc = any uppercase letter
        digit = any digit
```

Figure 5.3

The "history" of the machine H_M is given by $(\Xi, \Box, P)$,[1] the "initial state" is described by $(\Xi_0, \Box_0, P_0)$, and the set of possible $\mathscr{M}$ tape subsequences is designated by I^*. We say that M is halted at time $t \Leftrightarrow \forall t' > t$, $H_t = H_{t'}$ $(t, t' \in \mathscr{N})$; that M halts $\Leftrightarrow \exists t \in \mathscr{N}$, M is halted at time t; that p runs at time $t \Leftrightarrow$ the "initial state" occurs when P_0 is such that p appears at $\Box_0, P_0$; and that p runs $\Leftrightarrow \exists t \in \mathscr{N}$, p runs at time t. The formal definition of the viral set $(\mathscr{V})$ is then given by

(1) $\forall M\ \forall V (M, V) \in \mathscr{V} \Leftrightarrow$
(2) $[V \subset I^*]$ and $[M \in \mathscr{M}]$ and $\forall v \in V\ \forall H\ \forall t, j \in \mathscr{N}$
(3) $[[P_t = j]$ and $[\Xi_t = \Xi_0]$ and $(\Box_{t,j}, \ldots, \Box_{t,j+|v|-1}) = v] \Rightarrow$
(4) $\exists v' \in V, \exists t', t'', j' \in \mathscr{N}$ and $t' > t$
(5) (1) $[[(j' + |v'|) \leq j]$ or $[(j + |v|) \leq j']]$ and
(6) (2) $[(\Box_{t',j'}, \ldots, \Box_{t',j'+|v'|-1}) = v']$ and
(7) (3) $[\exists t''[t < t'' < t']$ and $[P_{t''} \in \{j', \ldots, j' + |v'| - 1\}]]$

We will now review this definition line by line

(1) for all M and V, the pair (M, V) is a "viral set" if and only if
(2) V is a nonempty set of $\mathscr{M}$ sequences and M is a $\mathscr{M}$ and
For each virus v in V, For all histories of machine M,
For all times t and cells j
(3) If the tape head is in front of cell j at time t and $\mathscr{M}$ is in its initial state at time t and the tape cells starting at j hold the virus v
(4) then there is a virus v' in V, a time $t' > t$, and place j' such that
(5) (1) at place j' far enough away from v
(6) (2) the tape cells starting at j' hold virus v'
(7) (3) and at some time t'' between time t and time t' v' is written by M

To save space, we will use the expression

$$a \overset{B}{\Rightarrow} C$$

to abbreviate part of the previous definition starting at line (2) where a, B, and C are specific instances of v, M, and V, respectively, as follows:

$$\Big[\forall B \Big[\forall C [(B, C) \in \mathscr{V}] \Leftrightarrow$$

$$\Big[[C \subset CI^*] \text{ and } [M \in \mathscr{M}] \text{ and } \Big[\forall a \in C \big[a \overset{B}{\Rightarrow} C\big]\Big]\Big]\Big]\Big]$$

Before continuing, we should note some of the features of this definition and their motivation. We define the predicate V over all Turing Machines. We have also stated our definition so that a given element of a viral set may generate any number of other elements of that set depending on the rest of the tape. This affords additional generality without undue complexity or restriction. Finally, we have no so-called "conditional viruses" in that EVERY element of a viral set must ALWAYS generate another element of that set. If a conditional virus is desired, we may add conditionals that either cause or prevent a virus from being executed as a function of the rest of the tape, without modifying this definition.

We may also say that V is a "viral set" w.r.t. $M \Leftrightarrow [(M, V) \in \mathscr{V}]$ and define the term "virus" w.r.t. M as $\{[v \in V] : [(M, V) \in \mathscr{V}]\}$.

We say that "v evolves into v' for M" $\Leftrightarrow [(M, V) \in \mathscr{V}$ and $[v \in V]$ and $[v' \in V]$ and $[v \overset{M}{\Rightarrow} \{v'\}]]$, that "$v'$ is evolved from v for M" $\Leftrightarrow$ v evolves into v' for M, and that "v' is an evolution of v for M" $\Leftrightarrow$

$$(M, V) \in \mathscr{V}\ \exists i \in \mathscr{N}\ \exists V' \in V^i$$

$$[v \in V] \quad \text{and} \quad [v' \in V] \quad \text{and}$$

$$[\forall v_k \in V' [v_k \overset{M}{\Rightarrow} v_{k+1}]] \quad \text{and}$$

$$\exists l \in \mathscr{N}\ \exists m \in \mathscr{N}$$

$$[[l < m] \quad \text{and} \quad [v_l = v] \text{ and } [v_m = v']]$$

In other words, the transitive closure of $\overset{M}{\Rightarrow}$ starting from v, contains v'.

[1] For convenience, we drop the M subscript when we are dealing with a single machine except at the first definition of each term.

B.1 BASIC THEOREMS

At this point, we are ready to begin proving various properties of viral sets. Our most basic theorem states that any union of viral sets is also a viral set.

Theorem 1.

$$\forall M\ \forall U^* [\forall V \in U^* (M, V) \in \mathscr{V}] \Rightarrow [(M, \cup U^*) \in \mathscr{V}]$$