Chapter 1

Predicate Logic: Syntax

In the last few chapters we developed rules for the syntax and semantics of the language *SL*. We learned how to use two powerful methods, truth-tables and truth-trees, for testing sentences and sets of sentences to analyze the logical properties that we defined for the language. Finally, we developed a system that enables us to derive new sentences of *SL* from given sentences of *SL*.

An important feature of *SL* is that it is*decidable*. Truth-tables and truth-trees are mechanical methods for verifying truth-functional properties like validity, consistency, etc., in the sense that each step of the method is determined by a rule and the previous steps. To say that *SL* is decidable is to say that, for questions about truth-functional concepts, our methods will always give us a definite yes or no answer in a finite number of steps. So, we can, simply by following a set of rules, determine the validity of *any* argument in sentential logic. There will never be an instance for which the validity of an argument in *SL* cannot be determined.

Sentential logic is a powerful tool, but there are times when it fails. Consider this argument:

1. All dogs are mammals.
2. All cats are mammals.
3. Either Lola is a dog or Lola is a cat.

∴ Lola is a mammal.

Symbolizing this argument in *SL* results in something like this:

1. D
2. C
3. $\underline{\mathrm{E} \vee \mathrm{F}}$

∴ G

The natural language argument above is obviously valid, but the corresponding argument in *SL* is not. The problem is that *SL* cannot capture the logical relations between "All dogs are mammals", "Lola is a dog", and "Lola is a mammal." Those relations are determined by the internal structures of the sentences, and the internal structures of sentences are invisible to *SL*, because the smallest logical unit in *SL* is an entire sentence.

In this chapter, we will begin to develop a new language, *PL* (for predicate logic), that will allow us to express some of the internal structures of sentences. The argument above has a valid symbolization in *PL*. Unfortunately, this power comes with a cost. *PL* is not a decidable system. There is no method that we can use that is guaranteed to always tell us if an argument in *PL* is valid or a set of sentences in *PL* is consistent.

1.1 Singular Terms and Predicates

Predicate logic identifies three important components of sentences in natural languages: individual constants, predicates, and quantity terms. Individual constants are a type of singular term. A singular term