Introduction to Logic

Logic

A foundational branch of mathematics (the other foundation is set theory) that deals with the assignment of truth values to statements, also called deduction or proof.

- Statements, a.k.a., sentences, are either true, false, or neither.
- Propositions are either true or false.

Example:
Assuming the axioms of standard arithmetic,

- $x + 5 = 7$ is a statement if we don't know the value of x
- If we know the value of x, $x + 5 = 7$ is a proposition. In particular:
  - $x + 5 = 7$ is true if $x = 2$, and
  - $x + 5 = 7$ is false if $x \neq 2$

Proofs

- Axioms are propositions that are assumed to be true. Consider them the starting rules of a system.
- Proofs are chains of statements that lead from axioms to a final statement, called a conclusion, such that if each statement in the proof is true, the conclusion must be true.

Truth

Mathematical logic does not deal with the meaning of truth. It only deals with the systematic treatment of true and false statements. The meaning of truth is a matter for philosophy.

Theorems

- Theorems are statements that have been proved.
  - We usually use *theorem* to refer to major results in mathematics.
- Lemmas are theorems that are useful primarily in proving other theorems.
- Corrollaries are theorems that follow "quickly" (one or two deduction steps) from a theorem.

Example: Context-free Grammars

Consider the following system of axioms:

- $S \to aSb$
- $S \to \epsilon$, where ϵ means empty string

The set of axioms above is in instance of a *grammar*. A grammar is

- a set of *terminal symbols*, which can be an *alphabet* (a.k.a. *lexicon*) or set of words; here $\{a, b\}$,
- a set of *nonterminal symbols* which act as syntactic vairables in the grammar; here $\{S\}$,
- a distinguished nonterminal that acts as the start symbol; here S, and
- a set of axioms of the form $\alpha \to \beta$, called *production* or *grammar* rules.

A *context-free grammar* contains production rules with only nonterminals on the left.

Example: Language Generation

A production rule says that the symbol on the left can be replaced by the sequence of symbols on the right. A sequence of subsititions consitutes a *proof* that the final sequence of symbols, the conclusion, is a sentence in the language specified by the grammar. For example

S	the start symbol
aSb	by Rule 1, $S \to aSb$
$aaSbb$	by Rule 1, $S \to aSb$
$aabb$	by Rule 2, $S \to \epsilon$

is a proof that the sequence of symbols, or *string*, *aabb* is a sentence of the language, generated by successive applications of axioms, or production rules, in the langauge's grammar.

Parsing

The opposite problem, determining whether a given string is a valid sentence in a language, can be solved by finding a sequence of production rules that produce the string. This sequence of production rules can be represented as a tree, called a *parse tree*, and the program that does this is called a *parser*. A parser is an essential component of language analyzers such as interpreters and compilers.