Chapter 2 A Simple Compiler

AC (addition calculator). We first define ac informally, **Types (Integer, Float), Keywords (f, i, p), Variables.** The ac language offers only 23 possible variable names, drawn from the lowercase Roman alphabet and excluding the three reserved keywords f, i, and p. Variables must be declared prior to using them.

Phases of a SC. The scanner reads a source a c program as a text file and produces a stream of tokens. The parser processes tokens produced by the scanner, determines the syntactic validity of the token stream, and creates an abstract syntax tree (AST). The AST created by the parsing task is next traversed to create a symbol table. This table associates type and other contextual information with variables used in an ac program. The AST is next traversed to perform semantic analysis. For ac, such analysis is fairly minimal. Semantic analysis often decorates or transforms portions of an AST as the actual meaning of such portions becomes more clear. For example, an AST node for the + operator may be replaced with the actual meaning of +, which may mean floating point or integer addition. Finally, the AST is traversed to generate a translation of the original program. Necessities such as register allocation and opportunities for program optimization may be implemented as phases that precede code generation.

Recursive descent is one of the simplest parsing techniques used in practi- cal compilers. The name is taken from the mutually recursive parsing routines that, in effect, descend through a derivation tree.

The scanner and parser together accomplish the syntax analysis phase of a compiler. They ensure that the compiler's input conforms to a language's token and CFG specifications. context sensitivity, overlation, syntax-directedtranslationcanperform almost all aspects of program translation during syntax analysis.

The AST serves as a common, intermediate representation for a program for all phases after syntax analysis. Such phases may make use of information in the AST, decorate the AST with more information, or transform the AST.

The next phase to consider is semantic analysis, which is really a catchall term for any post-parsing processing that enforces aspects of a language's definition that are not easily accommodated by syntax analysis.

Symbol-table construction is a semantic- processing activity that traverses the AST to record all identifiers and their types in a symbol table.

Most programming language specifications include a type hierarchy that compares the language's types in terms of their generality. Our ac language follows in the tradition of Java, C, and C++, in which a float type is considered wider (i.e., more general) than an integer. This is because every integer can be represented as a float. On the other hand, narrowing a float to an integer loses precision for some float values. Most languages allow automatic widening of type, so an integer can be converted to a float without the programmer having to specify this conver- sion explicitly. On the other hand, a float cannot become an integer in most languages unless the programmer explicitly calls for this conversion.

Code generation consists of generating source code that is suitable for the dc program, which is a simple calculator based on a stack machine model.

ch