CS 5363: Programming Languages and Compilers

The University of Texas at San Antonio
Spring Semester, 2013

CS 5363: Compiler Project
TL13 Language Introduction

The TL13 programming language, is based on the P2K language used by Michael Franz in his Advanced Compiler Construction class at the University of California, Irvine. The P2K language was a simplified subset of Pascal, and TL13 has been simplified even further, so that you can experience writing an optimizing compiler without getting burdened by all of the complexities and details of a complete, standard programming language.

Detailed Information is available on TL13's

- Formal Syntax
- Informal Type Rules, and
- Informal Semantics

Example TL13 Program(s)

- Simple Programs
  - simple1.tl13
  - simple2.tl13
- "Real" Programs
  - sqrt.tl13 (equivalent Java program, program test inputs/outputs)
  - factorize.tl13 (equivalent Java program, program test inputs/outputs)
  - fibonacci.tl13 (equivalent Java program, program test inputs/outputs)
  - euclid.tl13 (equivalent Java program, program test inputs/outputs)

Lexical Features

The TL13 language, is lexically simple. All lexical items are to be separated by one or more whitespace characters (i.e., spaces, tabs, and returns). All identifiers start with an capital letter, and may contain only numbers and capital letters. All key words are start with lower-case letters or are a symbol. The symbols "(", ")", ":=", ";", "*", "div", "mod", "+", "-", "=", "!=", "<", "<=", ">", ">=" are used.

Data Types

Core TL13 supports 32-bit integers ("int"), booleans ("bool"). Variables are always declared to be of a

particular type.

Operators

TL13 has several infix binary operators that work on either integer operands. The multiplication "*", division "div", modulus "mod", addition "+", and subtraction "-" produce integer results. The comparison operators (i.e., equals "=", not equal "!=", less than "<", less-than or equal-to "<=", greater than ">", and greater-than or equal-to ">=") all produce boolean results.

Control Structures

TL13 is a structured programming language. The only control structures supported are "if" and "while" statements. Both take a boolean expression that guards the body of the control structure. In the case of an "if" statement, the statements after the "then" are executed if the expression is true, and the statements after the "else" (if there is one) are executed if the expression is false. In the case of the "while" statement, the loop is exited if the expression false; otherwise if the expression is true, the body will be executed, and then the expression will be re-evaluated.

Assignment

Assignments are a kind of statement rather than a kind of operator. The ":=" keyword is used to separate the left hand side (which is the variable being assigned to) from the right hand side, which is an expression that must be of the same type as the left hand side.

Built-in Procedures

Core TL13 does not support user-defined functions or procedures, but it does support one built-in procedures "writeInt" that outputs an integer and a new-line to the console (respectively), and one user-defined function, "readInt" that reads an integer from the console. The syntax for these is hard-coded into TL13's BNF grammar.

You are encouraged to post additional TL13 programs (or not-quite TL13 programs) to the Piazza forum.

Errata/Clarifications

- 4/10/2013: Added equivalent Java programs and input/output examples for "Real Programs".

CS 5363
Jeffery von Ronne
Department of Computer Science
The University of Texas at San Antonio

Last Modified: 4/10/2013