Lab: Simple modules
CSC 312P, “Programming language implementation: the good parts”
Department of Computer Science · Grinnell College
November 20, 2015

Exercises

A. Create a subdirectory called MODULES-LAB in the directory you’re using for labs in this course and copy my implementation of the SIMPLE-MODULES programming language into it. The /home/stone/courses/languages/code/SIMPLE-MODULES directory contains the seven language-specific modules that you need:

- tokens.scm
- syntax-trees.scm
- scanner.scm
- parser.scm
- expvals-and-environments.scm
- type-checker.scm
- interpreter.scm

If you don’t already have the four utility modules test.scm, natural-numbers.scm, character-sources.scm, list-of.scm you’ll also need to copy them into the parent directory of your MODULES-LAB directory (from /home/stone/courses/languages/code).

B. Our example languages lack many basic arithmetic operations. Write and test a SIMPLE-MODULES that starts with a module in which you implement the operations of integer addition, integer multiplication, and exponentiation with integer bases and natural-number exponents. In the main program body, use these operations to compute the value of $-17 \cdot (-12)^5 + 9$.

C. Exercise 8.2 in the text (page 289) points out a defect in the implementation of SIMPLE-MODULES: the add-module-defns-to-env adds all the variables that receive values in the body of the module to its environment, not just the ones that are specified in the interface.

1. Write a SIMPLE-MODULES program containing a module that defines at least two variables but does not export all of them (i.e., does not list all of the variables in the interface section), and then, in the main program body, tries to use a variable that the module defines but does not export. In the interpreter, comment out the call to the type-of-program procedure and execute the program to show that the error does actually occur.

2. Restore the call to type-of-program and demonstrate that the type checker detects and reports this error. Explain why the type checker finds this error even though the interpreter proper does not.

3. Revise the interpreter to eliminate the error.

Copyright © 2015 by John David Stone
This work is licensed under the Creative Commons Attribution–ShareAlike 4.0 International License. To view a copy of this license, visit https://creativecommons.org/licenses/by-sa/4.0/deed.enUS or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.