Report for the IBM 650 Emulator in Scheme
Topic Paper #14

Alex Laird
CS-3210-01
3/21/09
Survey of Programming Languages
Spring 2009
Computer Science, (319) 360-8771
alexdlaird@cedarville.edu

Grading Rubric		Max	Earn
	On Time/Format	1	
	Correct	5	
	Clear	2	
	Concise	2	
	Total	10 pts	

Peer Reviewer

ABSTRACT
This paper discusses the completed IBM 650 emulator developed using the PLT Scheme programming language.

Keywords
IBM 650, PLT, Scheme, Emulator

1. OVERVIEW
The emulator has the capabilities of reading and writing from and to a text file (if two arguments are specified on instantiation), or reading and written from and to the command line (if no arguments are given).

The first batch of messages, up until a "+9999999999" is received and stored in a vector that emulates the memory. The second batch of messages, up until the stop command "+9000000000" followed by the program terminating "+9999999999" is stored in a vector that handles the messages. The final batch of messages is stored in a third vector until the end of the file is reached or the users terminates by typing "done."

2. PARSING
Each message is parsed according to the IBM 650 specifications given for this project. The lead ~# (where ~ is + or – and # is some number) is the command operator; the first set of ### is the memory location of the first value, the second set of ### is the memory location of the second value, and the final set of ### is the destination memory location.

3. TESTING
As was the case with the Python emulator, the test programs weren't all accurate, so Morehart's test program is probably the most conclusive when deciding how well the emulator was constructed.

4. ERROR HANDLING
Due to the functional and documentation limitations of the Scheme programming language, error handling in this emulator was kept to a minimum. Most situations where an error may occur were simply surrounded with the (when) statement, however, in general, it was heavily assumed that the user was providing valid input at all times.

5. MOST CONCLUSIVE TEST
Again, the overall best test that was run was created by Ryan Morehart of our class. The program conclusively tested every single operation the emulator was possible of. The output for this test file should is probably the most reputable and is also the most accurate. The Scheme emulator, as of 3/21/09, completed all of the tasks of this program correctly except the final one; though rigorous debugging was done on my part, I could not figure out where it was going wrong. I believe it is a loss of precision (or rather too much precision) at some point due to Scheme having a difficult dropping fractional bits down to the nearest integer.

Alex Laird, Cedarville University, Cedarville, Ohio, 45314