Compiler Assignment 1

Written by Akilesh B CS13B1042

August 26, 2015

Objective:

To implement a Stack Machine Interpreter in Cool programming language with the following rules.

Command	Meaning
int	push the integer *int* on the stack
+	push a '+' on the stack
s	push an 's' on the stack
e	evaluate the top of the stack
d	display contents of the stack
x	stop

The 'd' command prints out the content of the stack, one element per line, beginning with the top of the stack. The behaviour of the 'e' command depends on the contents of the stack when 'e' is issued:

- If '+' is on the top of the stack, then the '+' is popped off the stack, the following two integers are popped and added, and the result is pushed back to the stack.
- If 's' is on top of the stack, then the 's' is popped and the following two items are swapped on the stack.
- If an integer is on the top of the stack or if the stack is empty, the stack is left unchanged.

Implementation

- To implement the stack, I used the list.cl in *~cool/examples* as a data structure for storing these elements.
- *List* class defines the operations on an empty list. *Cons* class inherits List class.
- *head_String()* returns the top most element of the Stack and *tail()* returns the list after popping the top most element.
- *init_String* method in *Cons* class adds a particular string s to the start of the list and returns the new list.
- *Main* class contains appropriate functions for performing the above mentioned operations.
- Every input element which I get from the user is a string, so to perform addition operation, I use the atoi.cl in *~cool/examples* to convert string to integer (*a2i*) , perform the addition and then convert back to string (*i2a*) and push it on the stack.

- *print_list* method is used for printing the list.
- Stack Machine Interpreter is successfully implemented through this Assignment in Cool language.