PhD Qualifying Exam On Programming Languages

Adriana Compagnoni

25 April 2005

General Remarks

- *This is an open-books, open-notes, but closed laptops exam.*
- The exercises refer to the text book *Essentials of Programming Languages* by Friedman, Wand and Haynes (Second Edition).

Exercises

Consider the Post-Turing language PT where all the information being processed is placed in one linear tape. The tape is divided into contiguous squares, and a reading head is positioned on top of one of the squares at any given time: we refer to that square as the square being scanned.

Each square of the tape contains a symbol from a finite alphabet $A = \{B, S_1, \cdots, S_n\}$, where B is the blank symbol. At any given point during computation the tape contains only finitely many non-blank symbols.

A PT program is a finite sequence of the following instructions.

(L)	PRINT S	replaces the symbol on the square being scanned with S.
(L)	IF S GOTO L'	if the symbol being scanned is S then GOTO the first instruction labeled L', otherwise, continue to the next instruction. If there is no instruction labeled L', halt.
(L)	RIGHT	scan the square immediately to the right of the square presently being scanned.
(L)	LEFT	scan the square immediately to the left of the square presently being scanned.

We often omit the label if it is unused in the program.

A *tape configuration* consists of a tape with finitely many non-blank squares and the reading head positioned on a determined square.

For example:

```
...B B B 1 1 1 B B B ...
       ^
```

is a tape configuration, where ^ indicates the square being scanned, and everywhere else in the tape the squares contain the symbol B. The input to a program is a tape configuration, and the output of a program is also a tape configuration.

Consider the following program and the alphabet $A = \{B, 1, M\}$. This program copies a string of 1's.

(A) RIGHT
    IF B GOTO E
    PRINT M
(B) RIGHT
    IF 1 GOTO B
(C) RIGHT
    IF 1 GOTO C
    PRINT 1
(D) LEFT
    IF 1 GOTO D
    IF B GOTO D
    PRINT 1
    IF 1 GOTO A

Given the following input tape configuration.

```
... B B B 1 1 1 B B B ...
        ^
```

The program halts with tape configuration:

```
... B B B 1 1 1 B 1 1 1 B B B...
                  ^
```

Implement the language PT. Notice that although the tape is infinite in both directions, only a finite number of squares contain non-blank symbols, and therefore it can be implemented with a finite (and extensible on demand) data structure.

1. Define a Scheme data-type to represent the tape.
2. Define Scheme data-types to represent programs and instructions.
3. Write an interpreter in Scheme for PT that, given a program `p` and an initial tape `t`, returns the corresponding output tape, if the program `p` applied to input tape `t` terminates.
4. Write a parser in Scheme for PT.

Good luck!