Project Proposal: Register Machine Interpreter

Wei-Lin Wu and Johnnie Chang

April 20, 2018

1 Background

In theoretical computer science, there have been many computational models proposed to study computability and complexity that are equivalent to one another in terms of computational power. Two best-known models are Turing machine and λ-calculus. *Register machine*, a lesser known model is put forth to discuss the Halting Problem and other decidability problems in the book Mathematical Logic by H.-D. Ebbinghaus, J. Flum and W. Thomas [1].

A register machine is associated with a fixed alphabet $\mathcal{A} = \{a_0, \ldots, a_r\}$ and has countably many registers, each of which can store a string over $\mathcal{A}$ of an arbitrary (finite) length.

A program for a register machine with alphabet $\mathcal{A}$ is a finite nonempty sequence of instructions that each takes one of the forms below:

- L LET $\mathrm{R}_i = \mathrm{R}_i + a_j$, where $L, i, j \in \mathbb{N}$ and $j \leq r$ (Add-instruction: Add the letter a_j at the end of the string in register $mathrmR_i$);
- L LET $\mathrm{R}_i = \mathrm{R}_i - a_j$, where $L, i, j \in \mathbb{N}$ and $j \leq r$ (Subtract-instruction: If the string in register R_i ends with the letter a_j, delete this a_j; otherwise leave the word unchanged);
- L IF $\mathrm{R}_i = \epsilon$ THEN L' ELSE L_0 OR $\ldots$ OR L_r, for $L, i, L', L_0, \ldots, L_r \in \mathbb{N}$, where ϵ denotes the empty string (Jump-instruction: If register R_i contains the empty string go to instruction labelled L'; if the string in register R_i ends with a_0 (or $a_1, \ldots, a_r$, respectively) go to instruction labelled L_0 (or $L_1, \ldots, L_r$, respectively));
- L PRINT, for $L \in \mathbb{N}$ (Print-instruction: Print as output the string stored in register R_0);
- L HALT, for $L \in \mathbb{N}$ (Halt-instruction: Halt).

In a program of length $n+1$, the prefixing labels of instructions are $0, 1, 2, \ldots, n$, every jump instruction refers to labels no greater than n, and only the last instruction is a halt instruction.

2 Goal and Motivation

We have found this book an excellent reference for mathematical logic and computability, and would like to write a register-machine interpreter in Haskell to run the programs given in this book for a deeper level of understanding and to experiment our ideas concerning the computational model of register machine.

3 Work Schedule and Risk Assessment

Week 4. Background reading: [1] Ch. 10

Week 5. Background reading: [1] Ch. 10

Week 6. Implementaion of the interpreter

Week 7. Implementaion of additional features and testing with examples in [1]

Week 8. Short paper; presentation preparation

References

[1] H.-D. Ebbinghaus, J. Flum, and W. Thomas, *Mathematical logic.* Springer Science & Business Media, 2013.