CECS 329, Homework Assignment 5, Fall 2025, Dr. Ebert

Directions: Please review the `Homework` section on page 6 of the syllabus including a list of all rules and guidelines for writing and submitting solutions.

Due Date: Monday, November 17th as a PDF-file upload to the HW4 Canvas dropbox.

URM Programs

An **Unlimited Register Machine (URM)** is a computing device that consists of an unlimited list of registers $R_1, R_2, \ldots$, where each register is capable of holding any natural number $n \geq 0$, regardless of how large. The following is a recursive definition for what it means to be a **URM program**.

Atomic Program Each of the following instructions is considered a (single-instruction) URM program. Note: indices i and j are both positive integers.

1. $z(i)$. Assigns register i the value of 0: $R_i \leftarrow 0$, and is then terminated.
2. $s(i)$. Adds 1 to register i: $R_i \leftarrow R_i + 1$, and is then terminated.
3. $c(i, j)$. Copies the contents of register i to register j: $R_j \leftarrow R_i$, and is then terminated.
4. **end**. Has the global effect of terminating all programs that are currently being executed.

Compound Program via Concatenation If P and Q are programs, then so is PQ, i.e. the concatenation of P followed by Q.

Compound Program via if If P is a program, then if$(i, j)[P]$ is also a program. To execute this program, first compare the values stored in R_i and R_j. If $R_i = R_j$, then execute P. Otherwise terminate program if$(i, j)[P]$.

Compound Program via while If P is a program, then while$(i, j)[P]$ is also a program. To execute this program, repeatedly do the following until while$(i, j)[P]$ has been terminated. Compare the values stored in R_i and R_j. If $R_i \neq R_j$, then execute P. Otherwise, in case $R_i = R_j$, terminate program while$(i, j)[P]$.

To perform a computation with a URM program P, the program inputs are successively placed in R_1 through R_m, where $m \geq 0$ is the number of inputs. All other registers are automatically initialized to 0. The output of the computation is equal the value stored in R_1 after P has been terminated.

The following is a URM program for computing $\lfloor n/2 \rfloor$ where n is the sole input. The program is written in **vertical indentation form** for the sake of readability.

while$(1, 2)$

$s(2)$

if$(1, 2)$ //n must be odd.

$c(3, 1)$ //copy output $\lfloor n/2 \rfloor$ to output register R_1

end

$s(3)$ //add 1 to the output/quotient

$s(2)$

if$(1, 2)$ //n must be even.

$c(3, 1)$ //copy output $n/2$ to R_1

end

Finally, we may also view a URM program as a word over the alphabet

$$\Sigma = \{z, s, c, 0, 1, \ldots, 9, \text{end}, \text{while}, \text{if}, (,), [,], \text{“,”}\}.$$

For example, the **word form** of the above program is the word

$$\text{while}(1, 2)[s(2), \text{if}(1, 2)[c(3, 1), \text{end}], s(3), s(2), \text{if}(1, 2)[c(3, 1), \text{end}]].$$

Problems

1. Use vertical indentation form to describe a URM program that on inputs x and y, outputs xy. (10 pts)
2. Provide the word form of your program from part a. (5 pts)
3. Provide the rules for a context-free grammar that describes the set of all URM programs, where each program is viewed as a word (in word form) over Σ. **Use capital letters for all variables and indicate the start variable**. You may assume that any nonempty sequence of digits represents a valid register number, so long as it has at least one nonzero digit. (20 pts)
4. Use your grammar to provide a left-most derivation of the program

   ```
   while(2,3)[s(1),s(3)]
   ```

   (10 pts)