Midterm Study Guide

Handout 17
CSCI 334: Spring 2020

Reading

1. "A Brief Overview of C"
2. "Manual Memory Management"
3. "Passing Pointers by Value"
4. "Introduction to the Lambda Calculus, Part 1"
5. "Introduction to the Lambda Calculus, Part 2"
6. "Grammars and Parse Trees"
7. "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I"

Problems

Q1. (*10 points*) ..... Terminology

Review the required readings for this assignment, specifically looking for definitions for new terminology. Write down the term and a brief definition.

Hint: Look for italicized words. Italics are sometimes used for emphasis, but it is also often used to draw your attention to new technical terms. For example, in "A Brief Overview of C", the first italicized phrase that also comes with a definition is "executable binary." So write something like:

Executable binary: An executable program written in binary machine code.

Find as many technical terms as you can and define them.

Q2. (*10 points*) ..... Memory Leak

Does the following program leak memory? Why or why not?

```
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>

int main() {
    int *i = malloc(sizeof(int));
    if (!i) {
        printf("Out of memory!\n");
        exit(1);
    }
    *i = 0;

    while(true) {
        printf("i: %d\n", (*i)++);
    }

    return 0;
}
```

Q3. (*10 points*) ..... Church Numerals

Church encoding is a means of representing data and operators in the lambda calculus. The data and operators form a mathematical structure which is embedded in the lambda calculus. The *Church numerals* are a representation of the natural numbers using lambda notation. The method is named for Alonzo Church, who first encoded data in the lambda calculus this way.

The natural numbers are written using Church numerals as follows.

Number	Lambda Expression
0	$\lambda f.\lambda x.x$
1	$\lambda f.\lambda x.fx$
2	$\lambda f.\lambda x.f(fx)$
3	$\lambda f.\lambda x.f(f(fx))$
...	...
n	$\lambda f.\lambda x.f^n x$

Addition by one can be achieved using the successor function, defined as

$$\text{succ} \equiv \lambda n.\lambda f.\lambda x.f(nfx)$$

Prove that $0 + 1 = 1$.

Q4. (*10 points*) ..... Backus-Naur Form

Our augmented grammar for the lambda calculus is:

```
<expression>   ::= <value>
                |  <abstraction>
                |  <application>
                |  <parens>
                |  <arithmetic>

<value>        ::= v ∈ ℕ
                | <variable>
<variable>     ::=  α ∈ { a ... z }
<abstraction>  ::=  λ <variable>.<expression>
<application>  ::= <expression><expression>
<parens>       ::= (<expression>)
<arithmetic>   ::= (<op> <expression> ... <expression>)
<op>           ::= o  ∈ { +, - }
```

Add grammar support for exponentiation. For example, we should be able to parse the following expression:

$$((\lambda x.(+\ x\ 2))1)^2$$

and correctly evaluate it to `9`.