Homework 0

Due 7 February

Handout 2
CSCI 334: Spring, 2012

Reading

1. **(Required)** Mitchell, Chapters 1–2.
2. **(Recommended)** "Why Undergraduates Should Learn the Principles of Programming Languages", Stephen N. Freund (Williams College), Kim Bruce (Pomona College), Curtis Clifton (Rose-Hulman Institute of Technology), Kathi Fisler (WPI), Dan Grossman (University of Washington), Matthew Hertz (Canisius College), Johan Jeuring (Utrecht University), Gary T. Leavens (University of Central Florida), Andrew Myers (Cornell University), Larry Snyder (University of Washington), Simon Thompson (University of Kent), 2011.

   An overvew of why PL is worth studying and what the course objectives are. Draft Available on the course web page.

Problems

1. (*10 points*) ..... Partial and Total Functions

For each of the following function definitions, give the graph of the function. Say whether this is a partial function or a total function on the integers. If the function is partial, say where the function is defined and undefined.

For example, the graph of $\texttt{f(x)} = \texttt{if x} > 0 \texttt{ then x} + 2 \texttt{ else x/0}$ is the set of ordered pairs $\{\langle x, x+2\rangle \mid x > 0\}$. This is a partial function. It is defined on all integers greater than 0 and undefined on integers less than or equal to 0.

Functions:

(a) $\texttt{f(x)} = \texttt{if x} + 2 > 3 \texttt{ then x} * 5 \texttt{ else x/0}$

(b) $\texttt{f(x)} = \texttt{if x} < 0 \texttt{ then 1 else f(x} - 2\texttt{)}$

(c) $\texttt{f(x)} = \texttt{if x} = 0 \texttt{ then 1 else f(x} - 2\texttt{)}$

2. (*10 points*) ..... Deciding Simple Properties of Programs

Suppose you are given the code for a function $\texttt{Halt}_\emptyset$ that can determine whether a program P requiring no input halts. Can you solve the halting problem using $\texttt{Halt}_\emptyset$?

To be more precise, suppose I give you a C function $\texttt{Halt}_\emptyset$ with the following behavior:

$\texttt{Halt}_\emptyset(P)$ returns true if program P will halt without reading any input when executed.
$\texttt{Halt}_\emptyset(P)$ returns false if program P will not halt when executed.

You should not make any assumptions about the behavior of $\texttt{Halt}_\emptyset$ on arguments that do not consist of a syntactically correct program.

Can you write a Java program `Halt` that reads a program text P as input, reads an integer n as input, and then decides whether P halts when it reads n as input? Such a `Halt` program would have the following form, and it would print "yes" if P halts when it runs and reads input n and "no" if P does not halt when it runs and reads input n:

```
P = readString();
n = readInteger();
...
```

You may assume that any program P read by your program begins with a statement that reads a single integer from standard input, and then performs operations Q. That is, all P have the form:

```
x = readInteger(); Q
```

where Q is the rest of the program text, and Q does not perform any input.

If you believe that the halting problem can be solved if you are given $\texttt{Halt}_\emptyset$, then explain your answer by describing how a program solving the halting problem would work. To do this, just describe what replaces `...` in the `Halt` program definition above. If you believe that the halting problem cannot be solved using $\texttt{Halt}_\emptyset$, then explain briefly why you think not.

3. (*10 points*) ..... Background

This question is just to gather some background about you, and to set the stage for what's to come shortly. You need not write more than a sentence or short list for each part.

(a) Which computer science classes have you taken?

(b) Which programming languages have you used? What's the most complex program you've written in them?

(c) Pick your favorite language. Answer the following questions about it. A sentence or two is sufficient for each part.

i. Describe two programming errors that the compiler identifies and reports while compiling a program in that language.

ii. Describe two programming errors that can cause your program to halt with an error message or crash after you compile and start to run it.

iii. What *feature* of that language do you find most confusing or hard to use?

iv. Are there features that you learned but never use? Why not?