Reading

1. (Required) Mitchell, Chapters 1–2.


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

Problems

Q1. (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 \( f(x) = \begin{cases} x + 2 & \text{if } x > 0 \\ x/0 & \text{else} \end{cases} \) is the set of ordered pairs \( \{(x, x+2) | x > 0\} \). 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) \( f(x) = \begin{cases} x + 2 & \text{if } x + 2 > 3 \\ x/0 & \text{else} \end{cases} \)

   (b) \( f(x) = \begin{cases} 1 & \text{if } x < 0 \\ f(x-2) & \text{else} \end{cases} \)

   (c) \( f(x) = \begin{cases} 1 & \text{if } x = 0 \\ f(x-2) & \text{else} \end{cases} \)

Q2. (10 points) Deciding Simple Properties of Programs

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

   To be more precise, suppose I give you a Java function

   ```java
   boolean Halt_\emptyset(string program)
   ```

   where calling the function with the source code for program \( P \) has the following behavior:

   \( \text{Halt}_\emptyset(P) \) returns true if program \( P \) will halt without reading any input when executed.

   \( \text{Halt}_\emptyset(P) \) returns false if program \( P \) will not halt when executed.

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

   Can you write a Java program \( \text{Halt} \) 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 \( \text{Halt} \) 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 \( n \):

   ```java
   void Halt() {
       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 $P$ read at the start of your $\text{Halt}$ function will have the form
\[
x = \text{readInteger}(); Q
\]
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 $\text{Halt}_\emptyset$, then explain your answer by describing how a program solving the halting problem would work. To do this, just describe what replaces $\ldots$ in the $\text{Halt}$ program definition above. If you believe that the halting problem cannot be solved using $\text{Halt}_\emptyset$, then explain briefly why you think not.