
Reading

1. Read Mitchell, Chapters 1–2.

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 $f(x) = \text{if } x > 0 \text{ then } x + 2 \text{ else } x/0$ 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) $f(x) = \text{if } x + 2 > 3 \text{ then } x * 5 \text{ else } x/0$
- (b) $f(x) = \text{if } x < 0 \text{ then } 1 \text{ else } f(x - 2)$
- (c) $f(x) = \text{if } x = 0 \text{ then } 1 \text{ else } f(x - 2)$

2. (10 points) Deciding Simple Properties of Programs

Suppose you are given the code for a function Halt_\emptyset that can determine whether a program P that requires no input halts. To be more precise, assume that you are writing a C or Java program that reads in another program P as a string. Your program is allowed to call Halt_\emptyset with the string P as an argument. A call to $\text{Halt}_\emptyset(P)$ 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 Halt_\emptyset on arguments that do not consist of a syntactically correct program.

Can you solve the halting problem using Halt_\emptyset ? More specifically, can you write a 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 you are given begins with a read statement that reads a single integer from standard input. Thus P has 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 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—there is no need to write the program out fully. If you believe that the halting problem cannot be solved using Halt_\emptyset , then explain briefly why you think not.

3. (10 points) Background

- (a) Which computer science classes have you taken?
- (b) Which programming languages have you used? What kind of programs have you written in each one?
- (c) Consider the programming language that you have used the most. 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 never use?