Chapter 12

Computability and Undecidability

1. What is the difference between computability and decidability?

   Solution: See Sec. 12.6.

2. What is the relation between computing problems and formal languages?

   Solution: See Sec. 12.6.

3. Let the alphabet set be {0, 1}. Let *A* and *B* lists be as defined below. Show that the Post Correspondence Problem does not have a solution for (*A, B*).

i	List A: w_i	List B: v_i
1	011	101
2	11	011
3	1101	110

   Solution: Only possible value for i_1 is 3 (first symbols should agree: 1101 and 110). i_2 must be such that the *B* string begins with a 1. i_2 = 3 is not possible since 11011101 does not match 110110 in the sixth place. i_2 = 1 leads to 1101011 and 110101. i_3 must be such that *B* string begins with a 1. Again i_3 = 3 leads to 11010111101 and 110101110. i_4 must also have a *B* string beginning with 1, and the mismatch continues ad infinitum. Therefore there is no solution to this instance of PCP.

4. Consider life on Planet *M* in the Andromeda galaxy. There, the Post Correspondence Problem is decidable! What are the consequences of this? Explain clearly.

   Solution: If PCP is decidable, any derivation in an unrestricted grammar becomes decidable. Therefore all recursively enumerable languages become recursive. Turing machines will no longer have a halting problem. There are no un-computable problems. Real numbers also become enumerable. Gödel's incompleteness theorem will be wrong. Hilbert's Entscheidungsproblem will be solvable. There will be a universal oracle which will answer all questions correctly. Every theorem in mathematics can be proved. Etc.

5. Show that greater(x, y) = 1 if $x > y$ else = 0 is computable.
   Solution: We first design a sign function:

```
sign(0) = 0;
sign(x+1) = 1;
```

   Then, we use the proper subtraction function (see Exercise 7) to define greater:

```
greater(x, y) = sign(subtract(y, x));
```

6. Show that $g(x, y) = x^y$ is primitive recursive.

Solution: We have shown in Sec. 12.8 that multiply(x, y) is a primitive recursive function. The exponentiation function is primitive recursive:

```
      g(x, 0) = successor(0);
    g(x, y+1) = multiply(x, g(x, y));
```

7. Show that subtraction is computable.

Solution: We consider proper subtraction of natural numbers only in which the first number must be greater than or equal to the second number; otherwise, the result is 0. First we define a predecessor function:

```
  predecessor(0) = 0;
predecessor(x+1) = x;
```

We can now define subtracting n from x as follows:

```
        subtract(0, x) = x;
subtract(n+1, x) = predecessor(subtract(n, x)).
```

Since subtraction is primitive recursive, it is computable.

8. Compute the values of Ackermann function $A(m, n)$ for each of m = 0, 1, 2, 3 and n = 0, 1, 2, 3, 4, 5. *Hint*: Apply the recursive definition of the function starting from the smallest values of m and n and moving to higher values incrementally.

Solution: The values are shown in the table below:

m*, *n	**0**	**1**	**2**	**3**	**4**	**5**
0	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	5	7	9	11	13
3	5	13	29	61	125	253