CHAPTER 17

Theory of Computation

(Solutions to Practice Set)

Review Questions

1. The three statements in our Simple Language are the increment statement, decrement statement, and loop statement. The increment statement adds 1 to the variable; the decrement statement subtracts 1 from the variable; the loop statement repeats an action (or a series of actions) while the value of the variable is not zero.
2. Algorithm S17.1 shows how we implement Y ← X.

   Algorithm S17.1 *Question 2*

   ```
   Y ← 0
   while (X)
   {
           decr (X)
           incr (Y)
   }
   ```

3. A problem that can be solved by our Simple Language can also be solved by the Turing machine.
4. A Turing machine is made of three components: a tape, a controller, and a read/write head. The tape, at any one time, holds a sequence of characters from the set of characters acceptable by the machine; the read/write head at any moment points to one symbol on the tape and is used to read and write characters; the controller controls the read/write head and is the theoretical counterpart of the central processing unit (CPU) in modern computers.
5. One way to delimit the data on a Turing machine tape is the use of two blanks, one at the beginning of the data and one at the end of the data.
6. The read/write head can move to left, right, or stay at the same place. At the same time, it may go to different state or remain in the same state.
7. A transition state diagram is a pictorial representation of a program written for the Turing machine.

8. Both tools show the same thing. The first uses a diagram; the second uses a table.
9. A Gödel number is an unsigned integer that is assigned to every program that can be written in a specific language. In the halting program, we represent a program as its Godel number when that program is the input to another program.
10. A polynomial solvable problem can be solved by a computer in a feasible time period. A non-polynomial solvable still can be solved by a computer, but not in an acceptable period of time.

Multiple-Choice Questions

11. a	12. c	13. b	14. d	15. a	16. d
17. c	18. b	19. d	20. c	21. c	22. a
23. c	24. d	25. c	26. b	27. d	

Exercises

28. See Algorithm S17.2.

Algorithm S17.2 *Exercise 28*

```
Temp ← 0
Y ← 0
while (X)
{
        decr (X)
        incr (Y)
        incr (Temp)
}
while (Temp)
{
        decr (Temp)
        incr (X)
}
```

29. See Algorithm S17.3. After assigning Y to Z, we increment Z (X times).

Algorithm S17.3 *Exercise 29*

```
Temp ← X                                  // See solution to Exercise 28
Z ← Y                                     // See solution to Exercise 28
while (Temp)
{
        decr (Temp)
        incr (Z)
}
```

30. See Algorithm S17.4.

Algorithm S17.4 *Exercise 30*

```
Temp ← X                                // See solution to Exercise 28
Z ← 0
while (Temp)
{
        decr (Temp)
        Z ← Z + Y                        // See algorithm 17.7 in the text
}
```

31. See Algorithm S17.5.

Algorithm S17.5 *Exercise 31*

```
Temp ← X                                // See solution to Exercise 28
Z ← 1
while (Temp)
{
        decr (Temp)
        Z ← Z × Y                        // See algorithm 17.8 in the text
}
```

32. See Algorithm S17.6. We assume that Y > X.

Algorithm S17.6 *Exercise 32*

```
while (X)
{
        decr (X)
        decr (Y)
}
```

33. See Algorithm S17.7.

Algorithm S17.7 *Exercise 33*

```
Temp ← X + 1
while (X)
{
        decr (X)
        A1
        Temp ← 0
}
while (Temp)
{
        decr (Temp)
        A2
}
```

34. The machine with the single instruction (A, 1, b, R, B) cannot perform any action when it is in the state shown in the text. It crashes.

35. The tape moves to the right and goes to state B as shown below:

	↓				
b	1	1	1	b	State: B

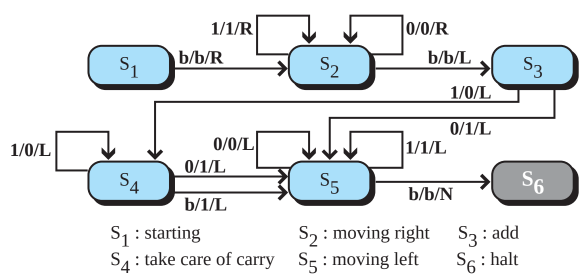
36. The machine goes through the following states:

	↓					
Start	b	1	1	1	b	State: A
		↓				
1	b	1	1	1	b	State: B
			↓			
2	b	#	1	1	b	State: B
				↓		
3	b	#	#	1	b	State: B
					↓	
4	b	#	#	#	b	State: B
				↓		
5	b	#	#	#	b	State: C
			↓			
6	b	#	#	1	b	State: C
		↓				
7	b	#	1	1	b	State: C
	↓					
8	b	1	1	1	b	State: C
		↓				
9	b	1	1	1	b	State: B

The last statement is the same as the first statement. The machine goes through an endless loop from statement 1 through statement 9.

37. Figure S17.37 shows the state diagram.

Figure S17.37 *Exercise 37*



38.

			$\downarrow$				
Start	…	b	b	b	b	…	State: S_1
	X = 0						
				$\downarrow$			
(S_1, b, b, R, S_2)	…	b	b	b	b	…	State: S_2
	X = 0						
			$\downarrow$				
$(S_2, b, 1, L, S_3)$	…	b	b	1	b	…	State: S_3 (halt)
	X = 1						

39.

			$\downarrow$				
Start	…	b	b	b	b	…	State: S_1
	X = 0						
				$\downarrow$			
(S_1, b, b, R, S_2)	…	b	b	b	b	…	State: S_2
	X = 0						
				$\downarrow$			
(S_2, b, b, N, S_3)	…	b	b	b	b	…	State: S_3 (halt)
	X = 0						

40.

a. (S_1, b, b, R, S_2) — S_1 is the ***starting*** state.
b. (S_2, b, b, N, S_3) — If X = 0, then go to state S_3 (halt).
c. $(S_2, 1, \#, R, M_R)$ — X is decremented and blank is replaced by #.
d. $(M_R, 1, 1, R, M_R)$ — M_R is the ***move right*** state.
e. (M_R, b, b, N, B_S) — B_S is the ***start of body loop*** state.
f. (B_H, b, b, L, M_L) — B_H is the ***halt of body loop*** state.
g. $(M_L, 1, 1, L, M_L)$ — M_L is the ***move left*** state.
h. $(M_L, \#, \#, L, M_L)$
i. (M_L, b, b, N, S_1)

At the end of the program the number of #s shows the value of X.

41.

a. (S_1, b, b, R, S_2) — S_1 is the ***starting*** state.
b. $(S_2, 1, 1, R, S_2)$ — S_2 is the ***move right*** state.
c. (S_2, b, b, L, S_3)
d. $(S_3, 1, b, L, S_3)$ — S_3 is the ***move left*** state. 1 is changed to b.
e. (S_3, b, b, N, S_4) — S_4 is the ***halt*** state.

42.

a. (S_1, b, b, R, S_2) — S_1 is the ***starting*** state.
b. (S_2, b, b, N, S_3) — S_3 is the ***halt*** state.
c. $(S_2, 1, b, R, M_R)$ — S_2 is the ***decrement X*** state.

d. (M_R, 1, 1, R, M_R) — M_R is the ***move right*** state.

e. (M_R, b, b, N, S_4) — S_4 is the ***start loop body*** state.

f. (S_4, b, b, R, S_5) — S_5 is the ***increment Y*** state.

g. (S_5, 1, 1, R, S_5)

h. (S_5, b, 1, L, S_6) — S_6 is the ***end loop body*** state.

i. (S_6, 1, 1, L, S_6) — S_6 is the ***end loop body*** state.

j. (S_6, b, b, L, M_L) — M_L is the ***move left*** state.

k. (M_L, 1, 1, L, M_L)

l. (M_L, b, b, N, S_1)

43. We use a single 1 to represent 0, two 1's to represent 1, three 1's to represent 2, …, and $n + 1$ 1's to represent n.

44. The following table shows the statements for the macro **X ← 0** and the Gödel number for each statement

Algorithm S17.8

while X$_1$	**// Gödel Number: CF1**
{	**// Gödel Number: D**
decr X$_1$	**// Gödel Number: BF1**
}	**// Gödel Number: E**

The Gödel number for the macro is then $(\text{CF1DBF1E})_{16}$. Notice that this micro does not preserve the value of X_1. The Gödel number for the macro will be longer if we want to preserve X_1.

45. The following table shows the statements for the macro and the Gödel number for each statement.

Algorithm S17.9

X$_2$ ← 0	**// Gödel Number: CF2DBF2E**
incr X$_2$	**// Gödel Number: AF2**
incr X$_2$	**// Gödel Number: AF2**

The Gödel number for the macro is then $(\text{CF2DBF2EAF2AF2})_{16}$. Notice that this micro does not preserve the value of X_2. The Gödel number for the macro will be longer if we want to preserve X_2.

46. The following table shows the statements. The Gödel number for the macro is $(\text{CF3DBF3ECF1DBF1AF3ECF2DBF2AF3E})_{16}$. Notice that this micro does not preserve the value of X_1 or X_2 The Gödel number for the macro will be longer if we want to preserve these two values.

Algorithm S17.10

```
X3 ← 0                  // Gödel Number: CF3DBF3E
while X1                // Gödel Number: CF1
{                       // Gödel Number: D
        decr X1         // Gödel Number: BF1
        incr X3         // Gödel Number: AF3
}                       // Gödel Number: E
while X2                // Gödel Number: CF2
{                       // Gödel Number: D
        decr X2         // Gödel Number: BF2
        incr X3         // Gödel Number: AF3
}                       // Gödel Number: E
```