Example 9.7 Subtraction of two integers

$$f(x, y) = x - y \text{ if } x > y$$
$$= 0 \text{ if } x \leq y.$$

Solution: The input tape is 0^x10^y. The calculation process is as follows.

i) Replace the first 0 by B, change the state, and traverse right. The transitional function is

$$\delta(q_0, 0) \rightarrow (q_1, B, R)$$

Traverse right for the remaining $'0'$ s of the first string. Getting the separator $'1'$, change the state from q_1 to q_2. The transitional functions are

$$\delta(q_1, 0) \rightarrow (q_1, 0, R)$$
$$\delta(q_1, 1) \rightarrow (q_2, 1, R)$$

ii) Replace the leftmost $'0'$ for the second number by $'1'$, change the state, and traverse left to find the replaced B, using transitional functions

$$\delta(q_2, 0) \rightarrow (q_3, 1, L)$$
$$\delta(q_3, 1) \rightarrow (q_3, 1, L)$$
$$\delta(q_3, 0) \rightarrow (q_3, 0, L)$$

Upon getting B, the state is changed to q_0 using the function

$$\delta(q_3, B) \rightarrow (q_0, B, R)$$

From the second iteration onwards, q_2 has to traverse $'1'$ using the transition function

$$\delta(q_2, 1) \rightarrow (q_2, 1, r)$$

If $x > y$, then the state q_2 will get B at the last of the second number representation. $x - y$ number of $'0'$ and $y + 1$ number of $'1'$ will remain. It changes its state and traverses left using the transitional function

$$\delta(q_2, B) \rightarrow (q_4, B, L)$$

Now, it replaces all the $'1'$ by B and traverses left to find B. The transitional functions are

$$\delta(q_4, 1) \rightarrow (q_4, B, L)$$
$$\delta(q_4, 0) \rightarrow (q_4, 0, L)$$

On getting B, it replaces it by $'0'$ and halts.

$$(q_2, B) \rightarrow (q_6, B, L)$$

If $x < y$, all $'0'$ representing x will finish before y. In the state q_0, the machine gets 1. It changes $'1'$ by B and traverses right with the state change.

$$\delta(q_0, 1) \rightarrow (q_5, B, R)$$

478 | Introduction to Automata Theory, Formal Languages and Computation

The state q_5 replaces all $'0'$ and $'1'$ by B to make the tape empty, and upon getting B it halts.

$$\delta(q_5, 0) \rightarrow (q_5, B, R)$$
$$\delta(q_5, 1) \rightarrow (q_5, B, R)$$
$$\delta(q_5, B) \rightarrow (q_6, B, H)$$

Example 9.8 Compute the function

$$f(x) = x - 2 if x > 2$$
$$= 0 \text{ if } x \leq 2.$$

Solution: The input tape is 0x. The calculation process is

i) Traverse x number of $'0'$ up to $'B'$.

ii) On getting B, change the state and traverse left.

iii) Replace the first $'0'$ by B with the state change and the second $'0'$ by B with the state change and halt.

The transitional diagram of the function is given in Fig. 9.11.

Solution: The input tape is 0x. The calculation process is

i) Traverse x number of $'0'$ up to $'B'$.

ii) On getting B, change the state and traverse left.

iii) Replace the first $'0'$ by B with the state change and the second $'0'$ by B with the state change and halt.

The transitional diagram of the function is given in Fig. 9.11.

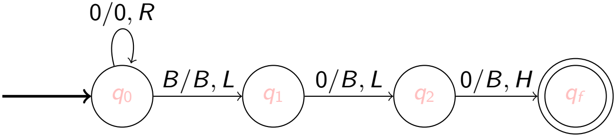


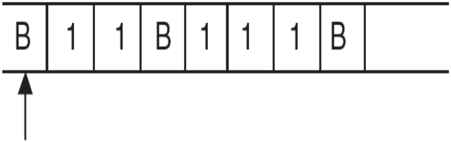
Fig. 9.11

Example 9.9 Multiplication of two integers $f(x, y) = x * y$.

Solution: Let $x = 2$ and $y = 3$. The input tape is in the form

Example 9.9 `Multiplication of two integers` $f(x, y) = x * y$.

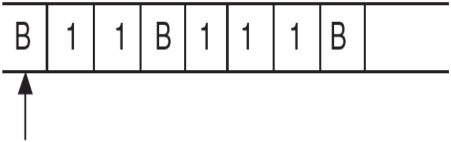
Solution: Let $x = 2$ and $y = 3$. The input tape is in the form



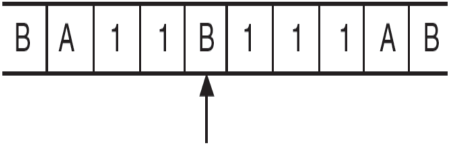
Make the tape in the following form

Example 9.9 Multiplication of two integers $f(x, y) = x * y$.

Solution: Let $x = 2$ and $y = 3$. The input tape is in the form



Make the tape in the following form



by the transitional functions

$$\delta(q_0, B) \rightarrow (q_1, A, R)\ \delta(q_1, 1) \rightarrow (q_1, 1, R)\ \delta(q_1, B) \rightarrow (q_2, B, R)$$
$$\delta(q_2, 1) \rightarrow (q_2, 1, R)\ \delta(q_2, B) \rightarrow (q_3, A, L)\ \delta(q_3, 1) \rightarrow (q_3, 1, L)$$

Here, A denotes the beginning of the first number and the end of the second number.

The transitional diagram for multiplication operation is given in Fig. 9.12.

The transitional diagram for multiplication operation is given in Fig. 9.12.

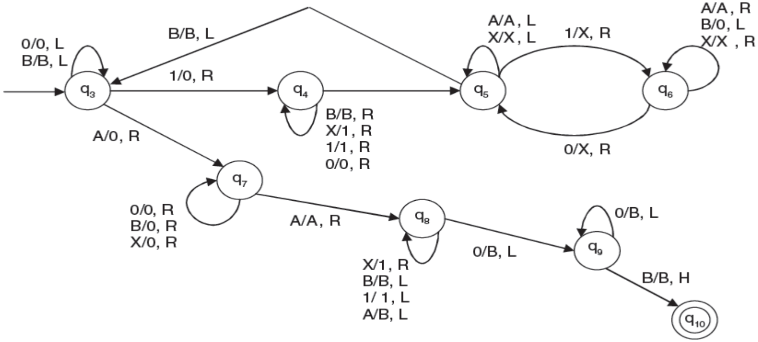


Fig. 9.12

Example 9.10 Th:e remainder after dividing one integer number by 2. $f(x,y) = x\%2$.

Solution: The remainder of the integer division 2 is either 1 or 0. The input tape is 0^x. The calculation process is as follows.

Example 9.10 `Th:e remainder after dividing one integer number by 2.` $f(x, y) = x\%2$.

Solution: The remainder of the integer division 2 is either 1 or 0. The input tape is 0^x. The calculation process is as follows.

i) Traverse the string from left to right. Getting $'B'$, traverse left with a state change.

ii) Replace all the $'0'$ of the tape by $'B'$ with alternating change of state and traverse left.

iii) Getting $'B'$, either halt on the final state or traverse the right by replacing one $'B'$ by $'0'$ and halt depending on the state.

The transitional functions are

$\delta(q_0, B) \rightarrow (q_1, B, R)$

$\delta(q_1, 0) \rightarrow (q_1, 0, R)$ // traverse right

$\delta(q_1, B) \rightarrow (q_2, B, L)$ // end of the string, so traverse left

$\delta(q_2, 0) \rightarrow (q_3, B, L)$] alternating change of state with replacement of 0 by B

$\delta(q_3, 0) \rightarrow (q_2, B, L)$

$\delta(q_3, B) \rightarrow (q_4, B, R)$ // number is odd, so traverse right

$\delta(q_4, B) \rightarrow (q_5, 0, H)$ // replace the 'B' by '0'

$\delta(q_2, B) \rightarrow (q_5, B, H)$ // number is even, so halt

Example 9.11 `Square of an integer.` $f(x) = x^2$.

Solution: The square of an integer means the multiplication of the number by the same number. The multiplication function is already described.

The input tape is in the form 1^X. It has to be made in the form 1^X B 1^X. To make this, the transitional functions are

$\delta(q_0, B) \rightarrow (q_1, X, R)$ // replace the first $'1'$ by X.

$\delta(q_1, 1) \rightarrow (q_1, 1, R)$ // traverse right to find $'B'$

$\delta(q_1, B) \rightarrow (q_2, B, R)$

$\delta(q_2, 1) \rightarrow (q_2, 1, R)$ // need from the second traversal

$\delta(q_2, B) \rightarrow (q_3, 1, L)$ // replace one $'B'$ after the end marker $'B'$ by $'1'$

$\delta(q_3, 1) \rightarrow (q_3, 1, L)$ // need from the second traversal

$\delta(q_3, B) \rightarrow (q_4, B, L)$ // traverse left

$\delta(q_4, 1) \rightarrow (q_4, 1, L)$ // traverse left to find the replaced X

$\delta(q_4, X) \rightarrow (q_0, X, R)$

$\delta(q_0, B) \rightarrow (q_5, B, L)$ // if all the 1 are replaced by $'X'$

$\delta(q_5, X) \rightarrow (q_5, 1, L)$ // replace all $'X'$ by $'1'$

$\delta(q_5, B) \rightarrow$ enter the Turing machine for multiplication

$\delta(q_0, B) \rightarrow (q_5, B, L)$ // if all the 1 are replaced by $'X'$

$\delta(q_5, X) \rightarrow (q_5, 1, L)$ // replace all $'X'$ by $'1'$

$\delta(q_5, B) \rightarrow$ enter the Turing machine for multiplication

9.3 Universal Turing Machine

From the discussions in the previous sections, it is clear that the Turing machine can perform a large number of tasks. The Turing machine can even perform any computational process carried out by the present day's computer. What is the difference between a Turing machine and real computer?

The answer is very simple. The Turing machine is designed to execute only one program but real computers are reprogrammable. A Turing machine is called a universal Turing machine if it simulates the behaviour of a digital computer. A digital computer takes input data from user and produces the output by using an algorithm. A Turing machine can be said to be a universal Turing machine if it can accept (a) the input data and (b) the algorithm for performing the task.

Each task performed by a digital computer can be designed by a Turing machine. So, a universal Turing machine can simulate all the Turing machines designed for each separate task.

How to design a universal Turing machine? The simple answer is to add all the Turing machines designed for each separate task. But, in reality, it is a complex one. We can do this by

- Increasing the number of read–write heads (like multiple head TM)
- Increasing the number of input tapes (like multiple tape TM)
- Increasing the dimension of moving the read–write head (k-dimensional TM)
- Adding special purpose memory like stack.

9.4 Linear-Bounded Automata (LBA)

An LBA is a special type of Turing machine with restricted tape space. The name 'linear bounded' suggests that the machine is linearly bounded. If we compare a LBA with a TM, then we see that the difference is in the operational tape space. For TM, the input tape is virtually infinite in both directions whereas the allowable input tape space of LBA is the linear function of the length of input string. The limitation of tape space makes LBA a real model of computer that exists, than TM in some respect. The diagram of an LBA is shown in Fig. 9.13.