Problem

For any positive integer x, let x^R be the integer whose binary representation is the reverse of the binary representation of x. (Assume no leading 0s in the binary representation of x.) Define the function $\mathcal{R}^+ : \mathcal{N} \longrightarrow \mathcal{N}$ where $\mathcal{R}^+(x) = x + x^{\mathcal{R}}$.

a. Let $A_2 = \{\langle x, y\rangle \mid \mathcal{R}^+(x) = y\}$. Show $A_2 \in$ L.

b. Let $A_3 = \{\langle x, y\rangle \mid \mathcal{R}^+(\mathcal{R}^+(x)) = y\}$. Show $A_3 \in$ L.

Step-by-step solution

Step 1 of 1

Given:

x is one of the positive integer and the reverse of the integer x is donated by the symbol x^R in the binary representation. But the condition is that there is no zero's in the binary representation of the number *x*.

Function R^+ need to be defined in such a way that the integer x is a natural number.

$R^+(x) = x + x^R$

Suppose, the positive integer x is 15, binary representation of $(15)_{10}$ is written as $(1111)_2$.

The reverse of binary representation x^R is represented as $(0000)_2$.

Inverse of the binary representation is done by converting the ones with zeros and zeros with ones.

Proof:

Consider the Turing machine *M* for every positive integer *x* , inverse of *x* is computed. When the computation is performed on the integer *x* and the inverse of the integer *x* then result will also be *x.*

This is because when a binary number 1 is added to 0 then result is always 1. Here, *x* positive integer is converted into binary representation but the condition is that the binary representation of *x* should not contain 0.

Turing machine will accept only those values of *x* whose binary representation is 1. Even Turing machine will accept those values of inverse of *x* whose binary representation is 0.

After that machine perform the computation between *x* and x^R.

Hence, it is proved that language $A_2 \in L$.

Consider the Turing machine *M* for every positive integer *x* , inverse of *x* is computed. When the computation is performed on the integer *x* and the inverse of the integer *x* then result will also be x.

This is because when a binary number 1 is added to 0 then result is always 1. Here, *x* positive integer is converted into binary representation but the condition is that the binary representation of *x* should not contain 0.

After performing the computation Turing machine once again call the same function. After calling the same function also the result after computation will be the value equal to *x*.

Turing machine will accept only those values of *x* whose binary representation is 1. Even Turing machine will accept those values of inverse of *x* whose binary representation is 0.

After that machine perform the computation between *x* and x^R.

Hence, it is proved that language $A_3 \in L$.

Comment