Problem

a. Let *ADD* = { $\langle x, y, z \rangle$ I x, y, z > 0 are binary integers and x+ y = z}. Show that *ADD* ? L.

b. Let *PAL-ADD* = { $\langle x, y \rangle$ I x, y > 0 are binary integers where x + y is an integer whose binary representation is a palindrome}. (Note that the binary representation of the sum is assumed not to have leading zeros. A palindrome is a string that equals its reverse.) Show that *PAL-ADD* ? L.

Step-by-step solution

Step 1 of 2

The class L: L is the class of languages that are decidable in logarithmic space on a deterministic truing machine.

That is, $L = SPACE(\log n)$

(a) Given Language is

$ADD = \{\langle x, y, z\rangle \mid x, y, z > 0 \text{ are binary integers and } x + y = z\}$

We have to show that $ADD \in L$.

That means, we have to construct a deterministic Turing machine (DTM) that decides *ADD* in logarithmic space.

Let M_1 be the *DTM* that decides *ADD* in logarithmic space.

The construction of M_1 is as follows:

$M_1 =$"On input $\langle x, y, z\rangle$:

1. If either of the three strings is not a binary number in the sense defined above then reject.
2. Initialize three binary counters i, j, k pointing to the right – most of x, y and respectively.
3. Perform binary addition (bit – wise long addition) using i, j, k and *a* carry flag.
4. If any discrepancy arises between the calculated next bit of *z* and the actual next bit of *z*, then reject.
5. If the end of all three number representations is reached and no error detected, then accept. So clearly M_1 runs in log space (it uses 3 counter only) and decides *ADD"*.

Thus *ADD* is in *L*. So, $ADD \in L$.

Comment

Step 2 of 2

(b) Given language is

$PAL\_ADD =$ { $\langle x, y\rangle \mid x, y > 0$ are binary integers where $x + y$ is an integer whose binary representation is a palindrome}

We have to show that $PAL\_ADD \in L$.

That means, we have to construct a *DTM* that decides PAL_ADD in logarithmic space.

Let M_2 be the *DTM* that decides PAL_ADD in logarithmic space.

The construction of M_2 is as follows.

$M_2 =$"On input $\langle x, y\rangle$:

1. If either of the two strings is not a binary number in the sence defined above, then reject.
2. Compute the length *l* of $x + y$ in binary.
3. For $i = 1$ to $l/2$.

(i) Compute the i^{th} bit a of $x + y$

(ii) Compute the $(l-i+1)^{th}$ bit *b* of $x+y$

(iii) if $\neq b$ then reject.

4. otherwise, accept"

Clearly M_2 runs in logspace and decides PAL_ADD.

Thus PAL_ADD is in L. So, $PAL\_ADD \in L$

Comment