Problem

Prove that if A is a regular language, a family of branching programs (B_1,B_2, . . .) exists wherein each B_n accepts exactly the strings in A of length n and is bounded in size by a constant times n.

Step-by-step solution

Step 1 of 4

A formal language, which can be expressed using a regular expression, is called as a **regular language**. In other words, it can be defined as a language which is recognized by a finite automation. All the languages which are finite are regular.

• Now consider a **regular language** A**,** then a family of branching program $(B_1, B_2, \ldots)$ in which a string, of length n in language A, is accepted by each B_n and is **bounded by a fixed time** n **in size.** It can be achieved by a way which is given below.

Comment

Step 2 of 4

Now consider a branching program**. A branching program** is known as "**a directed acyclic graph** where labels of all the nodes are maintained by the variables, except for two output nodes which are labeled as 1 or 0.

• All the nodes **whose labels are maintained by the variables** are called **query nodes**.

• Every query nodes consists of two outgoing edges: one is labeled 1 and another one is labeled 0. Both output nodes doesn't consists outgoing edges.

Comment

Step 3 of 4

So, from the definition of **branching program** and **regular language** A as defined above "the n-input function or a finite regular language can be computed by a branching program that consist a constant $O(n)$ size.

• A **bubble-sort** can be implemented as a circuit n. A set of **branching programs** is taken in such a way that each branching program accepts exactly the strings in A of length n.

• It is used to compare two bits and after comparing, reordering them if necessary is rather easy. The inputs can be called as x_1, x_2 and the outputs can be called as y_1, y_2.

• A sub-circuit can be written which accomplishes this as $y_1 = OR(x_1, x_2)$ and $y_2 = AND(x_1, x_2)$. **This circuit contains a size of two**.

• Now, the action of the bubble-sort algorithm can be mimicked on an array. It can be implemented one step at position to be the n input, n-output sub-circuit that passes through all the inputs taken as $< k \; and \geq k+1$ are unchanged.

• Now, the compare-swap sub-circuit, which is described above, on $< k \; and \geq k+1st$ input can be used to generate kth and $k+1st$ output. This still has size two. Now, **a pass** can be implemented as the serial concatenation of steps for each of $k = 1, 2, \ldots, n-1$, which has a size $(n-1)*2$.

• A bubble-sort can be Proceed to implement as the serial concatenation of one passes. Therefore, this gives a size $1(\mathbf{n-1})*\mathbf{2} = \mathbf{O(n)}$.

Comment

Step 4 of 4

Hence, from the above explanation it can be said that "**a family of branching program** $(B_1, B_2, \ldots)$ **in which a string, of length** n **in language** A**, is accepted by each** B_n **and is bounded by a fixed time** n **in size if the language** A **is regular**.