Problem

Show that the majority function with n inputs can be computed by a branching program that has O(n^2) nodes.

Step-by-step solution

Step 1 of 2

A branching programis defined as "**a directed acyclic graph** where the variables are used to label all the nodes except only two output nodes which is labeled 0 and 1". The **query nodes** are defined as all the nodes whish are labeled by the variables. All the query nodes consists two outgoing edges, labeled as 0 and 1. Both output nodes doesn't consists outgoing edges.

Now, consider about the **function majority** $\left(majority_n : \{0,1\}^n \rightarrow \{0,1\}\right)$, that is defined as:

$$\begin{aligned} majority_n\left(x_1, x_2, ..., x_n\right) &= 0 \ \ if \sum x_i < n/2; \\ &= 1 \ \ if \sum x_i \geq n/2. \end{aligned}$$

The computation of the **majority function** can be done by using **a branching program**, which consist $O\left(n^2\right)$nodes.

Comment

Step 2 of 2

Now, suppose the number of inputs taken is n. **A bubble-sort can be implemented as a circuit**. 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 $\mathbf{y_1 = OR\left(x_1, x_2\right)}$ and $\mathbf{y_2 = AND\left(x_1, x_2\right)}$. This will be act as a part of **branching program**. 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 $k = 1, 2, ..., n-1$, which has a size $(n-1)*2$.

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

Here, AND gates and OR gates are used to construct the branching program. Therefore, it can be said that "a branching program with $O\left(n^2\right)$ nodes can be used to compute the majority function with n inputs.

Comment