Problem

A ***Boolean formula*** is a Boolean circuit wherein every gate has only one output wire. The same input variable may appear in multiple places of a Boolean formula. Prove that a language has a polynomial size family of formulas iff it is in NC^1. Ignore uniformity considerations.

Step-by-step solution

Step 1 of 2

A Boolean formula is defined as a Boolean circuit which consist only a single output wire for every input gate. The Boolean formula may consists the same input variable at many places. Here, it can be shown that **a polynomial size family of formulas can be used to compute all the languages in** $\mathbf{NC^1}$.

A normal induction hypothesis on d is used to show that "a formula, whose size is less than $O(2^d)$ is similar to every Boolean circuit of depth d". For each step of the induction, the circuit's output gate is considered in such a way that the maximum fan-in value acquired is 2. The induction hypothesis can also be applied to each input gate.

• The nth circuit C_n has depth $O(\log n)$ in an $\mathbf{NC^1}$ circuit family. Therefore, the equivalent formula has size $2^{O(\log n)} = n^{O(1)}$ that is polynomial in size.

Comment

Step 2 of 2

To prove its converse, first it need to proof that every tree with $h \geq 2$ leaves consist a sub-tree with between $h/3$ and $2h/3$ leaves.

• Suppose a binary tree is denoted by B with $h \geq 2$ leaves. Beginning at the parent (root) of B, and traverse towards the child's (leaves), always taking a sub-tree with minimum half of the number of the existing sub-tree.

• Finally stop this iteration when a sub-tree $B^{'}$ is reached which consists at most $2h/3$ leaves. Then, $B^{'}$ will contain minimum of $h/3$ leaves as the previous sub-tree consists more than $2h/3$ leaves. Thus, the desired sub-tree is $B^{'}$.

Thus, from the above explanation, it can be said that **a polynomial size family of formulas can be used to compute all the languages in** $\mathbf{NC^1}$.

.

Comment