Figure 1 shows a typical example.

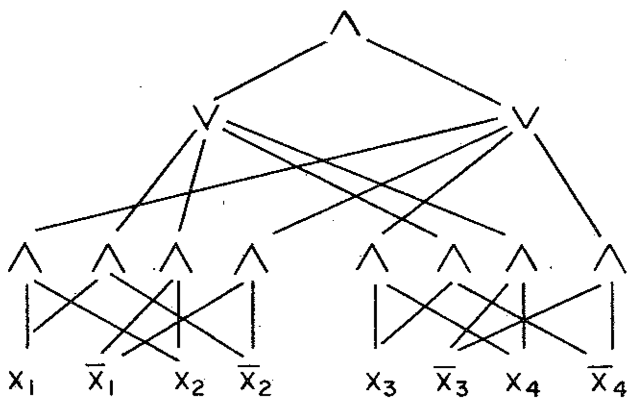
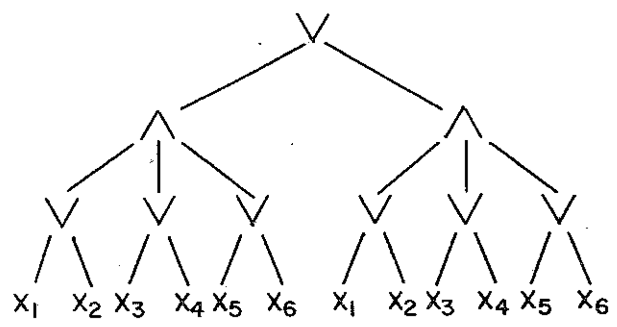


Figure 1

Without loss of generality we can assume that negations occur only as negated input variables. If negations appear higher up in the circuit we can move them down to the inputs using DeMorgan's laws which at most doubles the size of the circuit. Observe that we have alternating levels of AND and OR gates, since two adjacent gates of the same type can be collapsed into one gate.

The crucial parameters for a circuit are its depth and its size. *Depth* is defined as the length of the longest path from an input to the output and

As an example, let us convert the following depth-3 circuit into a circuit of depth 2.



x_1 x_2 x_3 x_4 x_5 x_6 x_1 x_2 x_3 x_4 x_5 x_6

Figure 2

Take any gate at distance two from the inputs. It represents a sub-circuit of depth 2. In this case this circuit will be an AND of ORs. Now observe that any function can be written either as an AND or ORs or as and OR of ANDs. Thus we can change this depth 2 circuit to and OR of ANDs which computes the same function. Thus we have the following circuit computing the same function.

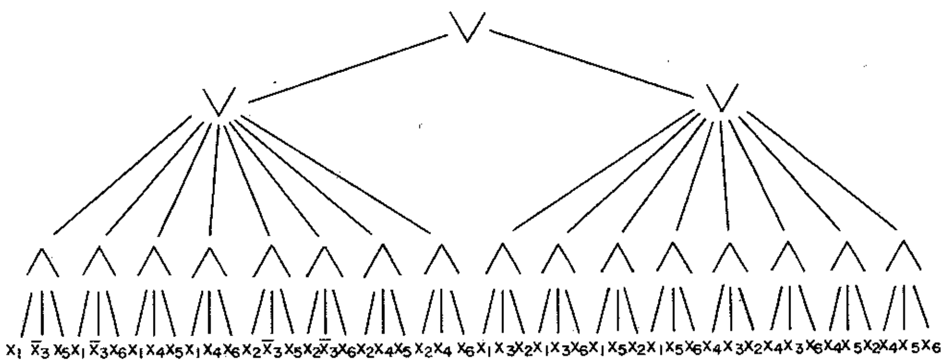


Figure 3

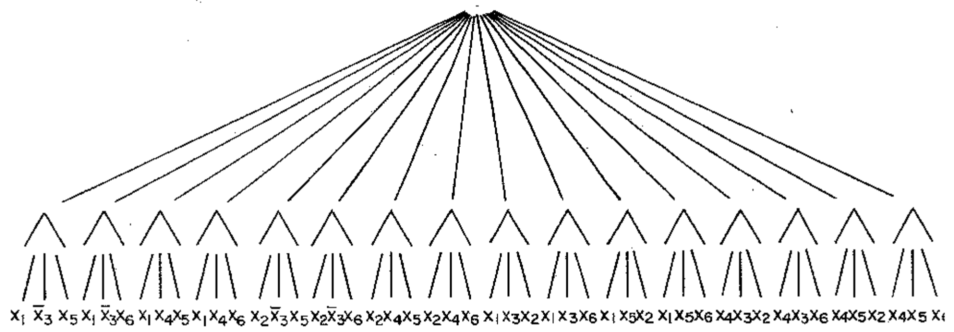


Figure 4

There is an obvious drawback with the above procedure. When we convert an AND of ORs to an OR of ANDs the size of the circuit will in general increase considerably. Thus we have converted a *small* depth k circuit to a *large* depth $k-1$ circuit and hence we fail to achieve (2).

3.2 Restrictions

The way around this problem was introduced in [FSS] and works as follows. If we assign values to some of the variables we can simplify the circuit. In particular if we assign the value 1 to one of the inputs of an OR gate, the output of the OR gate will be 1 no matter what the other inputs are. In the same way we need only know that one of the inputs to an AND gate is 0 to decide that it outputs 0. This means that the value of any specific gate on the bottom level can be forced by assigning a suitable value to one of its inputs. However there are many more gates than inputs and we have to do something more sophisticated. Let us first make formal what we mean by fixing some variables.

A picture of G which is good to keep in mind is the following.

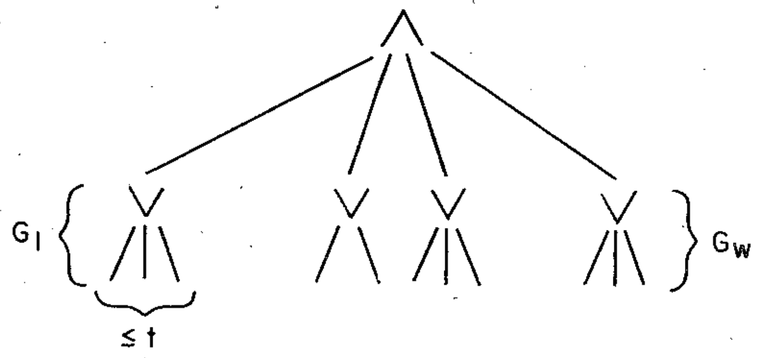


Figure 5

If $w = 0$ the lemma is obvious ($G \equiv 1$). For the induction step let us study what happens to G_1, the first OR in the circuit. We have two possibilities, either it is forced to be 1 or it is not. We estimate these two probabilities separately. We have

$$Pr[min(G) \geq s \mid F\lceil_\rho \equiv 1] \leq$$

$$\max(Pr[min(G) \geq s \mid F\lceil_\rho \equiv 1 \wedge G_1\lceil_\rho \equiv 1],$$
$$Pr[min(G) \geq s \mid F\lceil_\rho \equiv 1 \wedge G_1\lceil_\rho \not\equiv 1])$$

The first term is

$$Pr[min(G) \geq s \mid (F \wedge G_1)\lceil_\rho \equiv 1]$$

Thus by definition f_k^m is a function of $m^{k-1}\sqrt{k/2}$ variables.

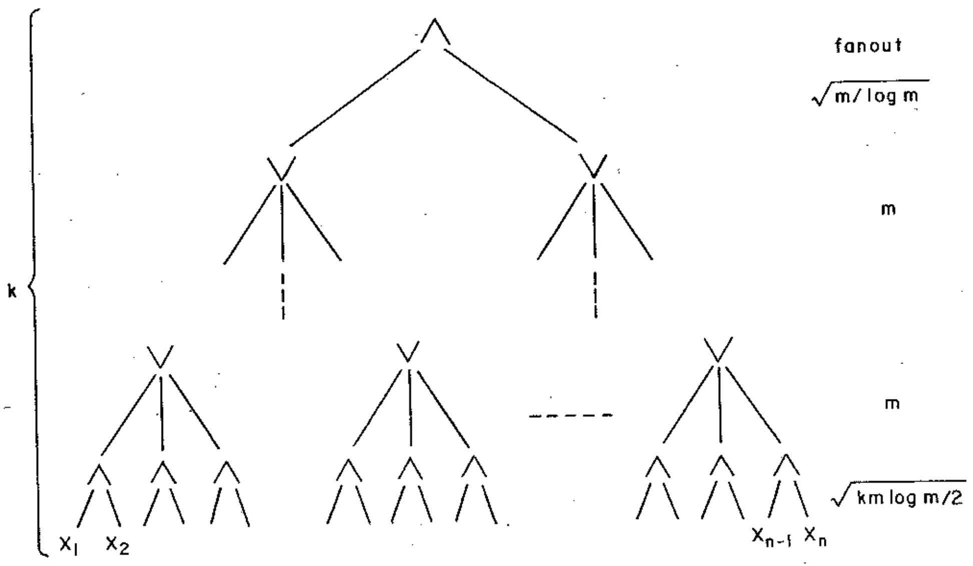


Figure 6

Yao has claimed exponential lower bounds for these functions, but the proof has not yet appeared. We have the following results for the functions f_k^m.

Theorem 6.1: *Depth $k-1$ circuits computing f_k^m are of size at least $2^{\frac{1}{12\sqrt{2k}}\sqrt{\frac{m}{\log m}}}$ for $m > m_1$, where m_1 is some absolute constant.*

As an immediate corollary we get.

Corollary 6.2: *Polynomial size circuits of depth $f(n)$ are more powerful than polynomial size circuits of depth $f(n)-1$ if $f(n) < \frac{\log n}{3 \log\log n} -$*

Proof: The fact that k is odd implies that the two lower levels look like:

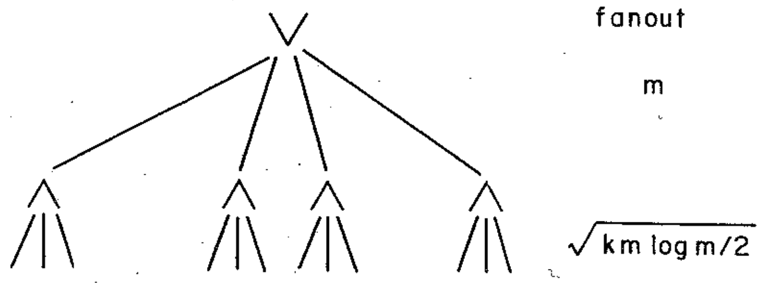


Figure 7

We establish a series of facts.

Fact 1: The AND gate corresponding to block B_i takes the value s_i for all i, with probability at least $\frac{5}{6}$ for $m > m_0$.

The AND gate corresponding to block B_i takes the values s_i precisely when not only ones are given to the block. The probability of this happening is $(1-q)^{|B_i|} = (1 - \sqrt{\frac{2k \log m}{m}})^{\sqrt{\frac{km}{2 \log m}}} < e^{-k \log m} = \frac{1}{6} m^{-k}$. Thus the probability that this happens for any block B_i is bounded by $\frac{1}{6}$ for sufficiently large m

Fact 2 With probability at least $\frac{5}{6}$ at least $\sqrt{(k-1)m \log m/2}$ inputs given the value $*$ by $\rho g(\rho)$ to each OR gate at level $k-1$. Again this is true only for sufficiently large m.

The expected number of such inputs is $\sqrt{2km \log m}$ and the fact follows from known estimates using $\frac{m}{\log m} \geq 100k$. For completeness let us include a very elementary proof.

Let p_i be the probability that an OR gate has as input exactly i AND gates which take the value $*$. Then