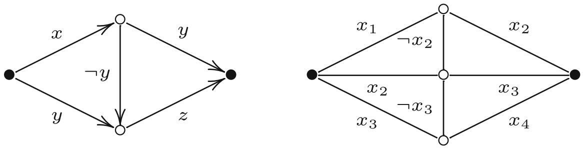


Fig. 1.5 A nondeterministic branching program computing the majority function $\mathrm{Maj}_3(x, y, z) = 1$ iff $x + y + z \geq 2$, and a non-monotone switching network computing the threshold function $\mathrm{Th}_2^4(x_1, x_2, x_3, x_4) = 1$ iff $x_1 + x_2 + x_3 + x_4 \geq 2$

consistent with a, that is, along which all wires are switched On by a. That is, each input switches the wires on or off, and we accept that input if and only if after that there is a nonzero conductivity between the nodes s and t (see Fig. 1.5). Note that we can have many paths consistent with one input vector a; this is why a program is nondeterministic.

An n.b.p. is *monotone* if it does not have negated contacts, that is, wires labeled by negated variables. It is clear that every such program can only compute a monotone boolean function. For a monotone boolean function f, let $\mathrm{NBP}_+(f)$ denote the minimum size of a monotone n.b.p. computing f, and let $\mathrm{NBP}(f)$ be the non-monotone counterpart of this measure. Let also $l(f)$ denote the minimum length of its minterm, and $w(f)$ the minimum length of its maxterm.

Theorem 1.8. (Markov 1962) *For every monotone boolean function f,*

$$\mathrm{NBP}_+(f) \geq l(f) \cdot w(f) .$$

Proof. Given a monotone n.b.p. program, for each node u define $d(u)$ as the minimum number of variables that need to be set to 1 to establish a directed path from the source node s to u. In particular, $d(t) = l(f)$ for the target node t.

For $0 \leq i \leq l(f)$, let S_i be the set of nodes u such that $d(u) = i$. If u is connected to v by an unlabeled wire (i.e., not a contact) then $d(u) \geq d(v)$, hence there are no unlabeled wires from S_i to S_j for $i < j$. Thus for each $0 \leq i < l(f)$, the set E_i of contacts out of S_i forms a cut of the branching program. That is, setting these contacts to 0 disconnects the graph, and hence, forces the program output value 0 regardless on the values of the remaining variables. This implies that the set $X(E_i)$ of labels of contacts in E_i must contain a maxterm of f, hence $|X(E_i)| \geq w(f)$ distinct variables. □

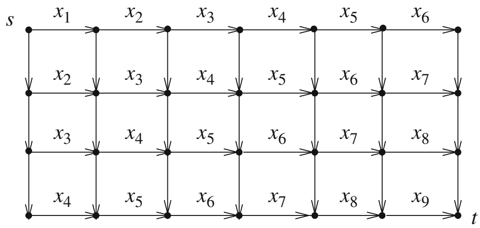
For the threshold function Th_k^n we have $l(\mathrm{Th}_k^n) = k$ and $w(\mathrm{Th}_k^n) = n - k + 1$, so every monotone n.b.p. has at least $k(n - k + 1)$ contacts. Actually, this bound is tight, as shown in Fig. 1.6. Thus we have the following surprisingly tight result.

Corollary 1.9. (Markov 1962) $\mathrm{NBP}_+(\mathrm{Th}_k^n) = k(n - k + 1)$.

In particular, $\mathrm{NBP}_+(\mathrm{Maj}_n) = \Theta(n^2)$.

It is also worth noting that the famous result of Szelepcsényi (1987) and Immerman (1988) translates to the following very interesting simulation: there

Fig. 1.6 The naive monotone n.b.p. for Th_k^n has $k(n-k+1)$ contacts; here $n = 9, k = 6$



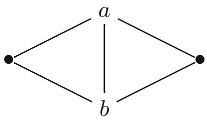


Fig. 1.7 A graph which is *not* parallel-serial: it has a "bridge" $\{a, b\}$ which is traversed in different directions

exists a constant c such that for every sequence (f_n) of boolean functions,

$$\mathrm{NBP}(\neg f_n) \le \mathrm{NBP}(f_n)^c .$$

This is a "NP = co-NP" type result for branching programs.

A *parity branching program* is a nondeterministic branching program with the "counting" mode of acceptance: an input vector a is accepted iff the number s-t paths consistent with a is odd.

Switching networks A *switching network* (also called a *contact scheme*) is defined in the same way as an n.b.p. with the only difference that now the underlying graph is *undirected*. Note that in this case unlabeled wires (rectifiers) are redundant since we can always contract them.

A switching network is a *parallel-serial network* (or π*-scheme*) if its underlying graph consists of parallel-serial components (see Fig. 1.8). Such networks can be equivalently defined as switching networks satisfying the following condition: it is possible to direct the wires in such a way that every s-t path will turn to a directed path from s to t; see Fig. 1.7 for an example of a switching network which is *not* parallel-serial.

It is important to note that switching networks include DeMorgan formulas as a special case!

Proposition 1.10. *Every DeMorgan formula can be simulated by a* π*-scheme of the same size, and vice versa.*

Proof. This can be shown by induction on the leafsize of a DeMorgan formula F. If F is a variable x_i or its negation $\neg x_i$, then F is equivalent to a π-scheme consisting of just one contact. If $F = F_1 \wedge F_2$ then, having π-schemes S_1 and S_2 for subformulas F_1 and F_2, we can obtain a π-scheme for F by just identifying the target node of S_1 with the source node of S_2 (see Fig. 1.8). If $F = F_1 \vee F_2$ then,