Theory of Computation
Chapter 8

Guan-Shieng Huang

Apr. 28, 2003

Reduction

To reduce Problem A to Problem B, we mean if B is solved, then A is solved.

x: an instance of Problem A

$\mathcal{R}$: transformation from A to B

$\mathcal{R}(x)$: an instance of B

We require $\mathcal{R}(x) \in B$ iff $x \in A$.
Hence B is solved implies that A is solved.
Or, B is at least as hard as A.

For computational problems, we say language L_1 is reducible to L_2 if there is a log-space reduction $\mathcal{R}$ such that

$$x \in L_1 \text{ if and only if } \mathcal{R}(x) \in L_2$$

for any string x as the input of decision problem for L_1.

Propositional 8.1

If $\mathcal{R}$ is a log-space reduction, then $\mathcal{R}$ is a polynomial-time reduction.

1. There are at most $O(nc^{k \lg n})$ possible configurations where c and k are constants..
2. If a computation for a Turing machine is terminated, each configuration can appear at most once.
3. Hence, $\mathcal{R}$ uses at most polynomial steps.

Reducing Hamilton Path (HP) to SAT

(Example 8.1)

HP: Given a graph, whether there is a path that visits each node exactly once.

$$G \text{ has an HP iff } \mathcal{R}(G) \text{ is satisfiable.}$$

$x_{i,j}$: node j is the ith node in the HP.

$$\mathcal{R}(G) = \begin{cases} (x_{1,j} \vee x_{2,j} \vee \cdots \vee x_{n,j}) & \text{for } 1 \leq j \leq n \\ (\neg x_{i,j} \vee \neg x_{k,j}) & \text{for } 1 \leq i, j \neq k \leq n \\ (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,n}) & \text{for } 1 \leq i \leq n \\ (\neg x_{k,i} \vee \neg x_{k+1,j}) & \text{for each pair } (i,j) \text{ not in } G \end{cases}$$

Reducing Reachability To SAT

(Example 8.2)

Given a graph G labeled from 1 to n, is there a path from node 1 to node n in G?

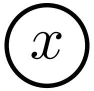
$g_{i,j,k}$: there is a path from node i to node j and this path passes through nodes with indices at most k.

$$\mathcal{R}(G) = \begin{cases} g_{i,j,k} \Leftrightarrow (g_{i,k,k-1} \wedge g_{k,j,k-1}) \vee g_{i,j,k-1}, & \text{for } 1 \leq i, j, k \leq n \\ g_{i,j,0}, & \text{if } (i,j) \text{ is an edge in } G. \end{cases}$$

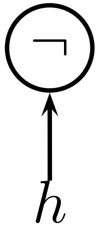
Then node 1 can reach node n in G if and only if $\mathcal{R}(G)$ is satisfiable.

Reducing Circuit SAT to SAT

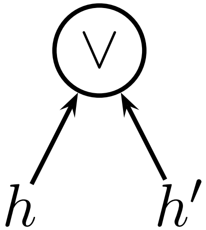
(Example 8.3)



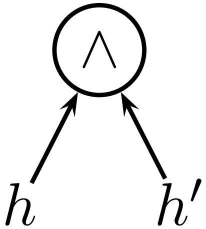
$\Longrightarrow \neg g \vee x, g \vee \neg x$ $(g \Leftrightarrow x)$



$\Longrightarrow \neg g \vee \neg h, g \vee h$ $(g \Leftrightarrow \neg h)$



$\Longrightarrow \neg h \vee g, \neg h' \vee g, h \vee h' \vee \neg g$ $(g \Leftrightarrow h \vee h')$



$\Longrightarrow \neg g \vee h, \neg g \vee h', \neg h \vee \neg h' \vee g$ $(g \Leftrightarrow h \wedge h')$

$$(x_3 \wedge \neg((x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))) \vee (\neg x_3 \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2)) \equiv x_1 + x_2 + x_3$$

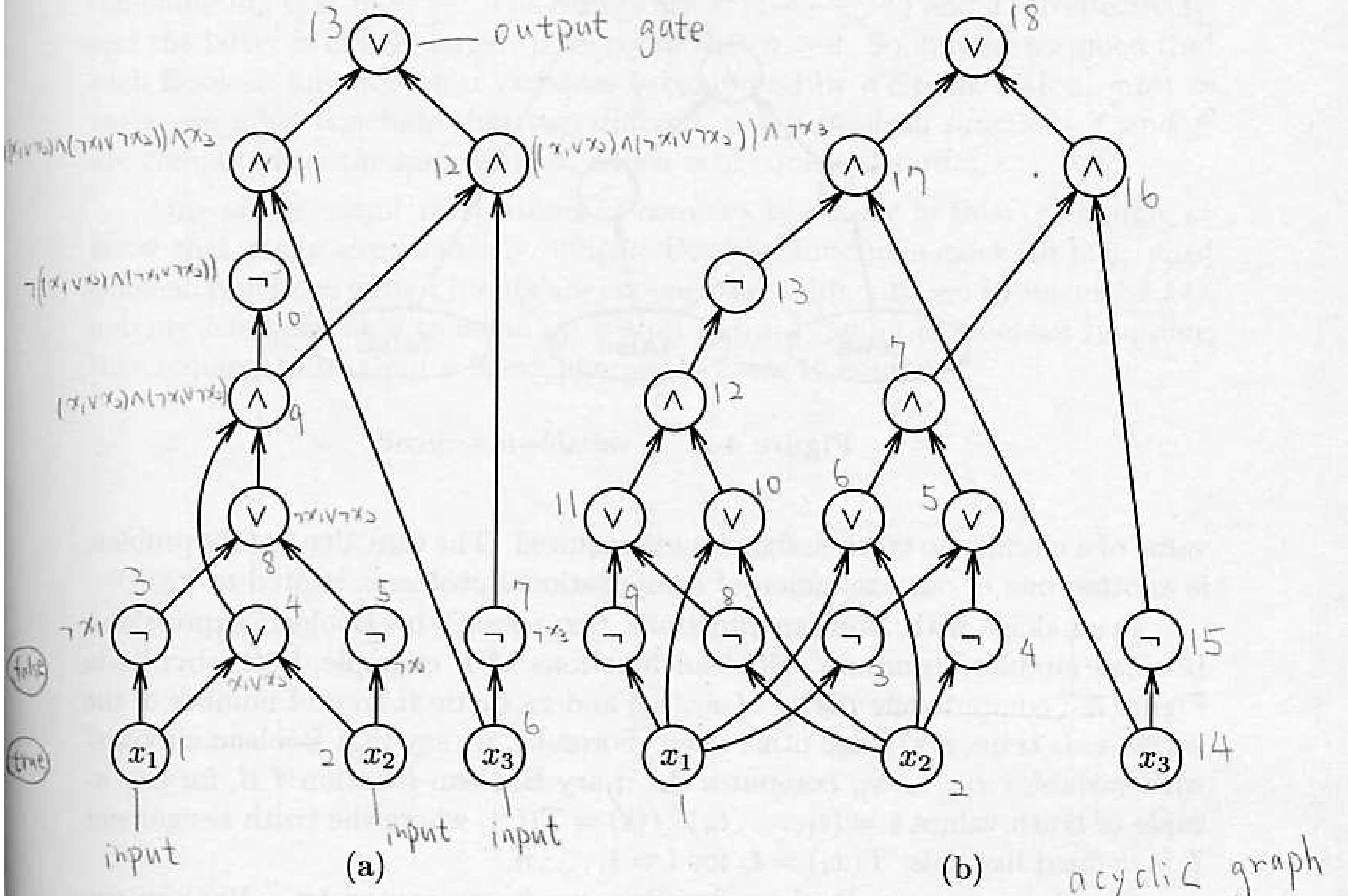


Figure 4-2. Two circuits.

Reducing Circuit Value to Circuit SAT

Reduction by generalization.

Proposition 8.2

If $\mathcal{R}$ is a reduction from L_1 to L_2 and $\mathcal{R}'$ is a reduction from L_2 to L_3, then there is a reduction from L_1 to L_3.

Given any x (either $x \notin L_1$ or $x \in L_1$), we have

$$x \in L_1 \text{ iff } \mathcal{R}(x) \in L_2 \text{ iff } \mathcal{R}'(\mathcal{R}(x)) \in L_3.$$

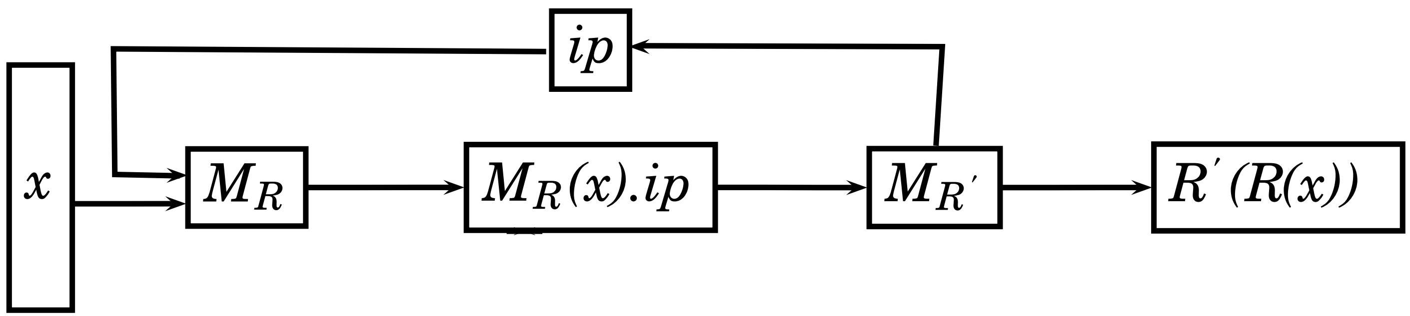
Thus, we have a reduction s.t. $x \in L_1$ iff $\mathcal{R}'(\mathcal{R}(x)) \in L_3$.

However, we cannot implement the composition $\mathcal{R}' \circ \mathcal{R}$ as

1. Compute $\mathcal{R}(x)$;

2. Compute $\mathcal{R}'(\mathcal{R}(x))$.

This is because we may need polynomial spaces in order to store $\mathcal{R}(x)$ in Step 1.



Complete Problems

(Definition 8.2)

$\mathcal{C}$: complexity class

L: a language in C

We say L is $\mathcal{C}$-complete if any language $L' \in \mathcal{C}$ can be reduced to L.

Examples:

NP-complete, P-complete, PSPACE-complete, NL-complete

Definition A class $\mathcal{C}'$ is closed under reductions if whenever L is reducible to L' and $L' \in \mathcal{C}'$, then also $L \in \mathcal{C}'$.

Remark

1. A complete problem is the least likely among all problems in $\mathcal{C}$ to belong in a weaker class $\mathcal{C}' \subseteq \mathcal{C}$.
2. If it does, then the whole class $\mathcal{C}$ coincides with the weaker class $\mathcal{C}'$, as long as $\mathcal{C}'$ is closed under reduction.

Proposition 8.3

P, NP, coNP, L, NL, PSPACE, and EXP are all closed under log-space reductions.

Remark:

If an NP-complete problem is in P, then P=NP.

Proposition 8.4

If two classes $\mathcal{C}$ and $\mathcal{C}'$ are both closed under reductions, and there is a language L which is complete for both $\mathcal{C}$ and $\mathcal{C}'$, then $\mathcal{C} = \mathcal{C}'$.

Observe that $\mathcal{C} \subseteq \mathcal{C}'$ and $\mathcal{C}' \subseteq \mathcal{C}$, and thus $\mathcal{C} = \mathcal{C}'$.

Cook's Theorem (Theorem 8.2)

SAT is NP-complete.

Table Method

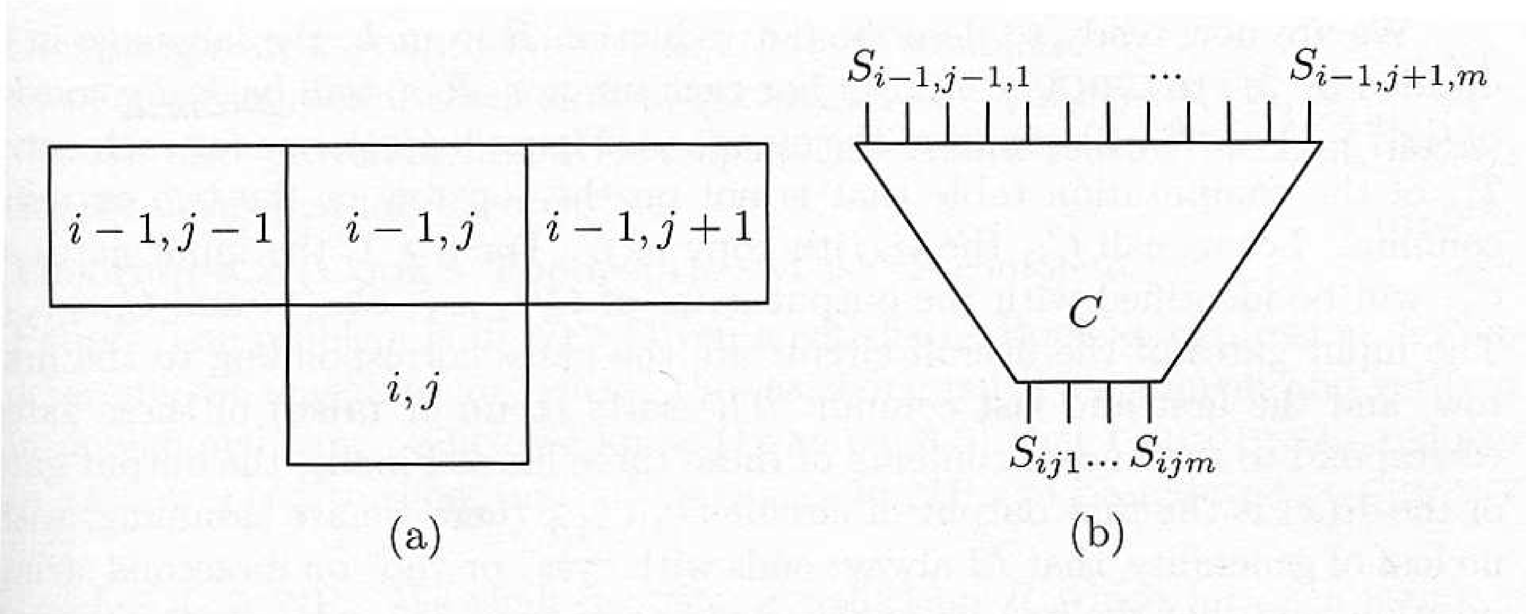
$\triangleright$	0_s	1	1	0	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1_{q_0}	1	0	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1	1_{q_0}	0	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1	1	0_{q_0}	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1	1	0	$\sqcup_{q_0}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1	1	$0_{q_0'}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1	1_q	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1_q	1	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright_q$	1	1	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	1_s	1	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright$	1_{q_1}	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright$	1	$\sqcup_{q_1}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright$	$1_{q_1'}$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright_q$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright$	$\sqcup_s$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$
$\triangleright$	$\triangleright$	$\triangleright$	"yes"	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$

Figure 8.3. Computation table.

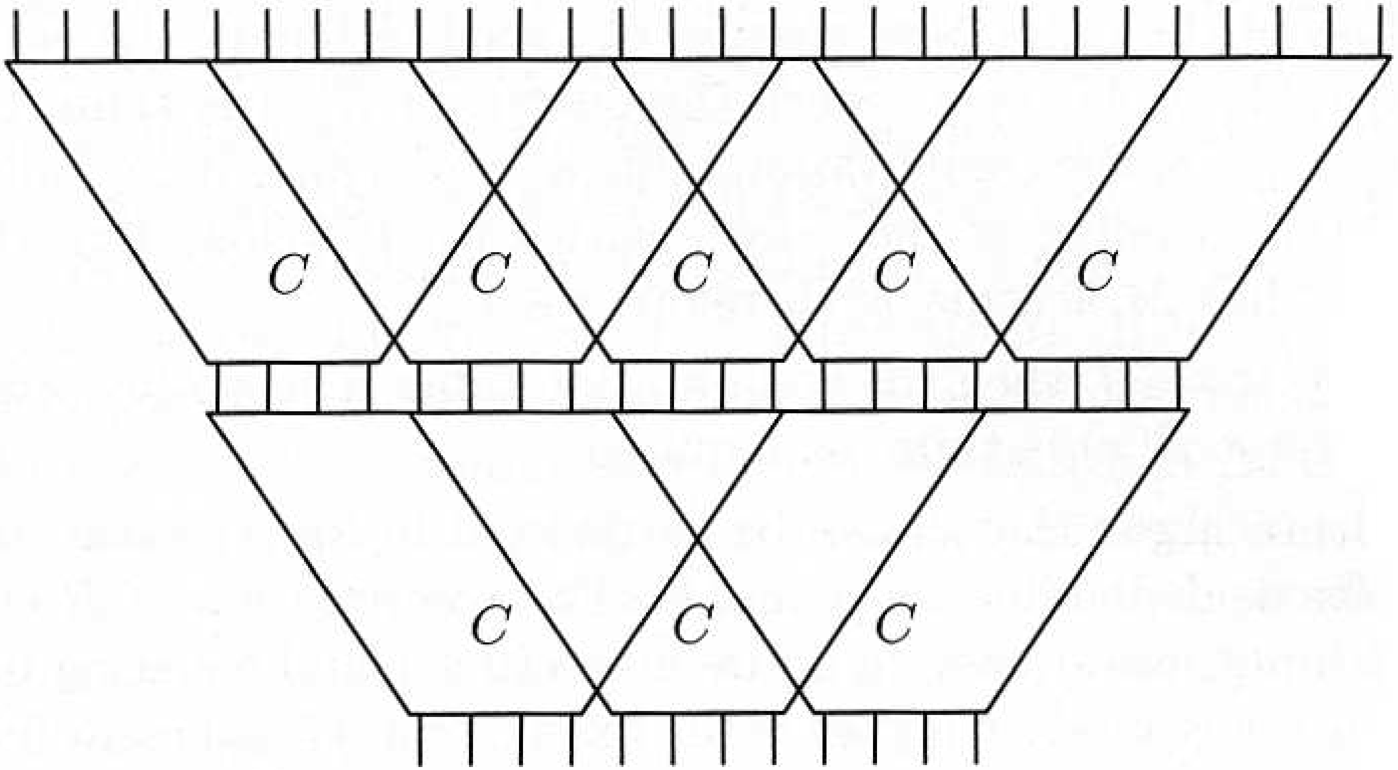
Theorem 8.1

Circuit Value is P-complete.

$p(|x|) \times p(|x|)$ size computation table where p is the time bound for the algorithm.



(a) (b)



(c)

Corollary: MONOTONE CIRCUIT VALUE is P-complete.

Cook's Theorem

SAT is NP-complete.

To standardize the behavior of non-determinism:

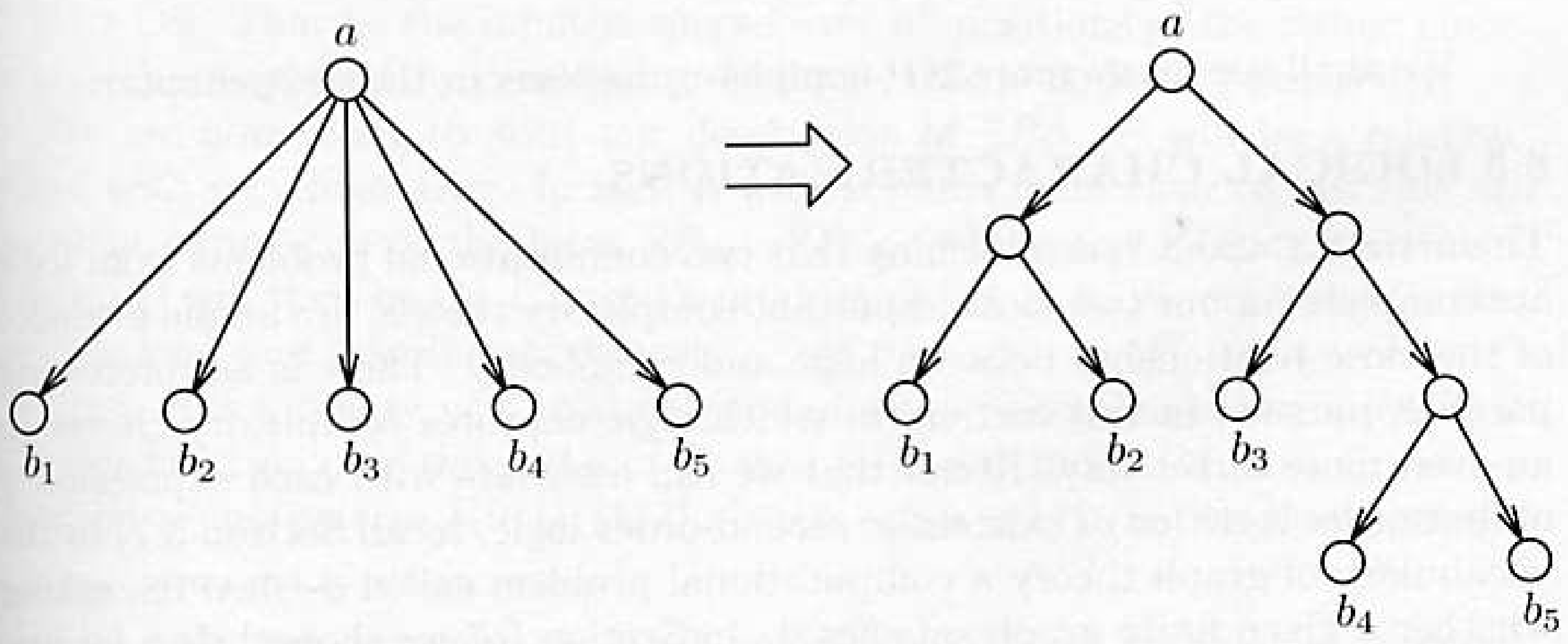


Figure 8-5. Reducing the degree of nondeterminism.

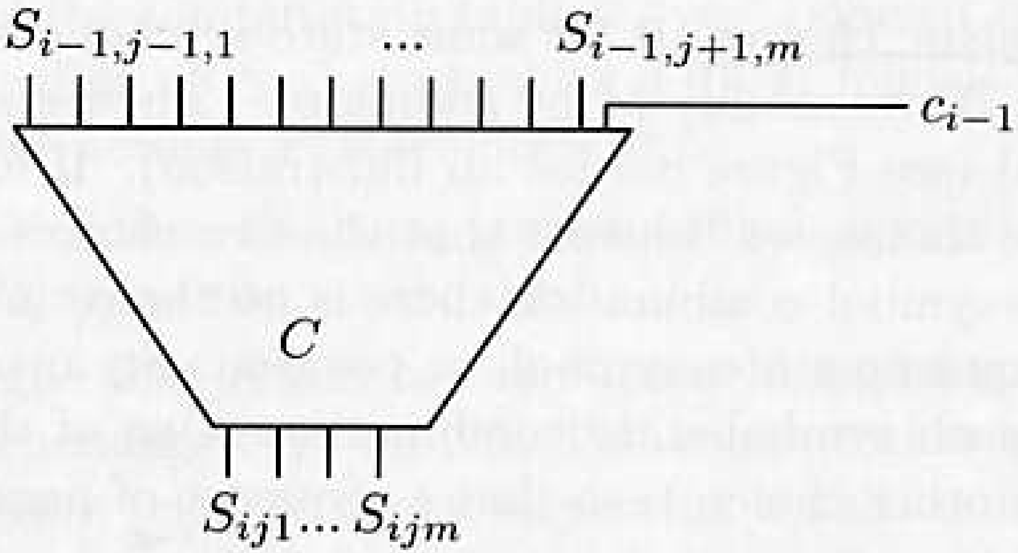


Figure 8-6. The construction for Cook's theorem.