FERENCES

AN, J. D. (1974), "The Design and Analysis of
y, Reading, Mass.
C. P., McAuliffe, K. P., Rudolp, L., and Sn
—Designing a MIND shared memory parall
), 175–189.
lgorithms for edge coloring bipartite graphs and
117–129.
lgorithms in numerical linear algebra, *SIAM Rev.*
machine organization and programming, *Comput.*
G. (1981), A fast parallel algorithm for routing in
nput. **C30** (2), 93–100.
3), Granularity of parallel memories, TR-89,
ant Institute, New York Univ., New York; for an
shared memory in parallel computation, in
eoretic Concepts in Computer Science (WG-83),
nabruck, June, in press.
" Academic Press, New York.
H. (1983), Parallel dictionaries on 2–3 trees, in
tes in Computer Science, Vol. 154, pp. 597–609,
t. efficient parallel algorithms for some graph
-691.
), Simulation of parallel random access machines
Research Center, Yorktown Heights, New York;
computation—A survey, TR-71, Department of
w York Univ., New York.
ation of certain arithmetic expressions, *J. Assoc.*

INFORMATION AND CONTROL **56**, 183–198 (1983)

Succinct Representations of Graphs

HANA GALPERIN*

Mini Systems Inc., Herzlia, Israel

AND

AVI WIGDERSON*,†

Computer Science Division, University of California, Berkeley, California 94720

For a fixed graph property Q, the complexity of the problem: Given a graph G, does G have property Q? is usually investigated as a function of $|V|$, the number of vertices in G, with the assumption that the input size is polynomial in $|V|$. In this paper the complexity of these problems is investigated when the input graph is given by a succinct representation. By a succinct representation it is meant that the input size is polylog in $|V|$. It is shown that graph problems which are approached this way become intractable. Actually, no "nontrivial" problem could be found which can be solved in polynomial time. The main result is characterizing a large class of graph properties for which the respective "succinct problem" is NP-hard. Trying to locate these problems within the P-Time hierarchy shows that the succinct versions of polynomially equivalent problems may not be polynomially equivalent.

1. INTRODUCTION

The design of efficient algorithms for graph theoretic problems is a major research area in recent years. The word "efficient" generally means that the amount of computing resources is minimized. One of the ways considered frequently is the use of complex data structures in algorithms, while the assumption is made that the input is given by some conventional representation. Traditionally, graphs are represented by either adjacency matrices or adjacency lists with representation size of $O(|V|^2)$ and $O(|E|)$, respectively. For graphs that are relatively small this is perfectly acceptable, but when we deal with graphs that have a huge number of vertices the conventional representations are quite costly. In the areas of architectural design systems

* This research was conducted when the authors were in the EECS Department at Princeton University, Princeton, N. J.

† The author was partially supported by NSF Grant ENG76-16808 and DARPA Contract N0039-82-C-0235.

0019-9958/83 $3.00

and very large scale integrated circuitry (VLSI) design systems the graphs dealt with could have millions of elements. This motivates us to develop succinct graph representation (i.e., represent a graph G in space $o(|V|)$). The goals one would like to achieve by using a succinct representation are:

(1) Reduce the amount of space required to store the graph.

(2) Improve the complexity of certain graph algorithms.

In this paper we deal with a specific succinct representation—the small circuit representation (SCR). While certain graphs can be represented in logarithmic space using the SCR model, checking simple graph properties for graphs represented this way is very difficult.

In Section 2 we prove some simple properties of the SCR model, which are helpful in proving that certain graphs have such a representation. Then we illustrate the difficulty of checking simple graph properties on this representation by proving in details a typical theorem.

Our results are listed in Table I.

Sections 3–5 are devoted to the proofs of these results. In Section 3 we characterize a large class of graph properties for which the respective problems are NP-hard. In Section 4 we improve this lower bound to Σ_2/Π_2-hardness for some of the problems. Section 5 shows how to obtain upper bounds for these problems, when given upper bounds on the complexity of the respective predicates for a non-succinct representation (e.g., adjacency matrix) of the input graph.

TABLE I

Problem	Upper Bound	Lower Bound
(1) Has a triangle	NP	NP
(2) Has a k-cycle	NP	NP
(3) Has a k-path	NP	NP
(4) $\Delta(G) \geqslant k$	NP	NP
(5) $\delta(G) \leqslant k$	Σ_2	Σ_2
(6) Has a cycle	DSPACE(n)	NP
(7) Has an Euler circuit	NSPACE(n)	NP
(8) Has an $s-t$ path	NSPACE(n)	Π_2
(9) Connectivity	NSPACE(n)	Π_2
(10) Perfect matching	Exp.-DTIME	Π_2
(11) Hamiltonian circuit	Exp.-NTIME	Π_2
(12) Planar	Exp.-DTIME	Σ_2
(13) Bipartite	Exp.-DTIME	Σ_2
(14) k-colorable	Exp.-NTIME	Σ_2

Note. G is a simple undirected graph, Δ and δ denote the maximum and minimum degree, respectively, and k is a fixed integer.

In the last section we suggest further research directions, and state some open problems.

2. The Small Circuit Representation

Let $G(V, E)$ be a graph with $m \leqslant 2^n$ vertices $v_0, v_1, \ldots, v_{m-1}$. We can encode the names of vertices with n-bit strings. Denote the binary representation of a number x by $\bar{x}$.

We define C_G to be an SCR of G if the following hold:

(1) C_G is a combinatorial circuit (i.e., a circuit without memory).

(2) C_G has two inputs of n bits each.

(3) C_G has r gates, $r = O(n^k)$ for some integer k.

(4) The output of C_G is given by

$$\begin{aligned} C_G(\bar{i}, \bar{j}) &= ? && \text{if } v_i \notin V \text{ or } v_j \notin V, \\ &= 0 && (v_i, v_j) \notin E, \\ &= 1 && (v_i, v_j) \in E. \end{aligned}$$

Note. This representation can be used for directed and undirected graphs. However, since for an undirected graph $C_G(\bar{i}, \bar{j}) = C_G(\bar{i}, \bar{j})$, we define it only for $i < j$.

Next we derive two basic lemmas concerning SCR which will be used in Section 3.

Lemma 2.1. *Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two graphs that have* SCRs *such that $V_2 \subseteq V_1$. Then $G(V_1, E_1 \cup E_2)$ has an* SCR.

Proof. Let C_{G_1}, C_{G_2} be the small circuits that represent G_1, G_2, respectively. Then we define C_G, the circuit that represents $G(V_1, E_1 \cup E_2)$ as

$$\begin{aligned} C_G(\bar{i}, \bar{j}) &= ? && \text{if } C_{G_1}(\bar{i}, \bar{j}) = ?, \\ &= 1 && \text{if } C_{G_1}(\bar{i}, \bar{j}) = 1 \text{ or } C_{G_2}(\bar{i}, \bar{j}) = 1, \\ &= 0 && \text{if } C_{G_1}(\bar{i}, \bar{j}) = 0 \text{ and } C_{G_2}(\bar{i}, \bar{j}) = 0. \end{aligned}$$

Since $|V_2| \leqslant |V_1|$ and C_{G_1}, C_{G_2} are small also C_G is small. ∎

Definition 2.1. SAT is the following problem:

Input. F, a Boolean CNF formula s.t. $|F| = O(p(n))$, where n is the number of variables in F and p is some polynomial.

Question. Is F satisfiable?
SAT is well known to be NP-complete (Cook, 1971).

DEFINITION 2.2. Let F be an instance of SAT with n variables. We define the *graph of* F, $G_F(V_F, E_F)$ by

$$V_F = \{v_0, v_1, ..., v_{2^n-1}, w = v_{2^n}\}, \qquad E_F = \{(v_i, w) \mid i < 2^n, F(\bar{i}) = 1\}.$$

In words, v_i and w are adjacent iff $\bar{i}$ satisfies F.

LEMMA 2.2. *G_F has an* SCR.

Proof. We construct a circuit C_{G_F}, that represents G_F. It has two inputs of $n+1$ bits each.
The outputs are:

$$\begin{aligned} C_{G_F}(\bar{i}, \bar{j}) &= ? && \text{if} \quad i > 2^n \text{ or } j > 2^n, \\ &= 1 && \text{if} \quad i < 2^n, j = 2^n \text{ and } F(\bar{i}) = 1, \\ &= 0 && \text{otherwise.} \end{aligned}$$

C_{G_F} is a SCR since the number of connectives (*or*, *and*, $\neg$) in F, which dominates the number of gates in C_{G_F}, is polynomially bounded by n. ∎

Given an SCR of a graph G, it is difficult to check if G has certain graph properties. This will be shown true for a large class of such properties in Section 3. We illustrate it here by proving that it is NP-complete to test if a graph has a triangle.
Define the problem TRIANGLE by

Input. C_G, an SCR of an undirected graph $G(V, E)$.

Question. Does G have a triangle?

THEOREM 2.1. TRIANGLE *is* NP-*complete.*

Proof. (a) TRIANGLE $\in$ NP. We guess the three vertices and feed every pair of vertices into the circuit to verify that the edges exist.

(b) SAT $\propto$ TRIANGLE. Let F be an instance of SAT with n variables. Define $G_1(V_1, E_1)$ as

$$V_1 = \{v_0, v_1, ..., v_{2^n-1}, w = v_{2^n}, a = v_{2^n+1}\}, \qquad E_1 = \{(v_i, a) \mid 0 \leqslant i \leqslant 2^n\}.$$

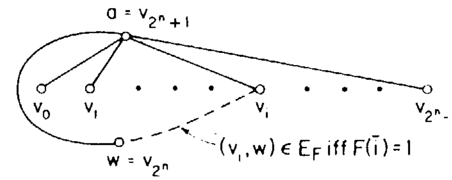


FIGURE 2.1

The following small circuit C_{G_1} represents G_1:

$$\begin{aligned} C_{G_1}(\bar{i}, \bar{j}) &= ? && \text{if} \quad i > 2^n + 1 \text{ or } j > 2^n + 1, \\ &= 1 && \text{if} \quad i \leqslant 2^n, j = 2^n + 1, \\ &= 0 && \text{otherwise.} \end{aligned}$$

Let $G_F(V_F, E_F)$ be the graph of F. By Lemma 2.2, G_F has a SCR. Also, V_F is contained in V_1. (Intentionally we used the same names for the vertices). The graph $G(V_1, E_1 \cup E_F)$ is shown in Fig. 2.1. We construct C_G, the SCR of G as in Lemma 2.1.

Claim. F is satisfiable iff G has a triangle.

Proof. only if Suppose there exists an i such that $F(\bar{i}) = 1$. Then $\{a, w, v_i\}$ form a triangle in G.

if Suppose G has a triangle. Since (v_i, v_j) is not in $E_1 \cup E_F$, $0 \leqslant i < j \leqslant 2^n - 1$, then a and w must be two of the vertices in the triangle. Suppose the triangle consists of $\{a, w, v_i\}$, then $(v_i, w) \in E$ which implies $F(\bar{i}) = 1$. ∎

3. NP-HARDNESS

Let Q_S be defined for every graph property Q by:

Input. C_G, an SCR of a graph G.

Question. "$Q(G)$?" (Does G have property Q?).

This whole paper is concerned with the complexity of Q_S for various (undirected) graph properties Q. In this section we will generalize the idea of Theorem 2.1, to characterize a class of graph properties Q for which Q_S is NP-hard. Then we show that many nontrivial graph properties are in this class.

DEFINITION 3.1. A graph $G(V, E)$ is called *t-critical w.r.t. a property* Q if the following hold:

(1) $V = \{v_0, v_1, \ldots, v_{t-1}, w = v_t, v_{t+1}, \ldots, v_{|V|-1}\}$. $|V| = O(t)$.

(2) Let $M = \{(v_i, w) \mid 0 \leqslant i \leqslant t-1\}$. Then $M \cap E = \varnothing$.

(3) $\neg Q(G(V, E))$. (G does not have property Q).

(4) Let M' be any nonempty subset of M. Then $Q(G'(V, E \cup M'))$ (if we add at least one edge of M to G, the resulting graph G' has property Q).

If (1)–(4) hold, G is denoted by G_t^Q.

THEOREM 3.1. *Let Q be a graph property, such that for every positive integer t:*

(1) *There exists a t-critical graph w.r.t. Q, G_t^Q.*

(2) *G_t^Q has an* SCR, *C_t.*

Then Q_S is NP-*hard.*

Proof. We show that SAT $\propto Q_S$. Let F be an instance of SAT with n variables. The graph $G_F(V_F, E_F)$ has an SCR by Lemma 2.1. The graph $G_{2^n}^Q$ exists and has an SCR by the conditions in the theorem. Also note that V_F is contained in V. Therefore, by Lemma 2.1, we can construct C_G, a small circuit that represents $G(V, E \cup E_F)$. Since $|V| = O(2^n)$, constructing C_G takes polynomial time in n.

Claim. F is satisfiable iff $Q(G)$.

Proof. *if* If F is not satisfiable, then $E_F = \varnothing$, and $G(V, E \cup E_F)$ is in fact the graph $G_{2^n}^Q$. From Definition 3.1(3), $\neg Q(G_{2^n}^Q)$ holds, and therefore $\neg Q(G)$ holds.

only if If F is satisfiable, then E_F is a nonempty subset of M (Definition 3.1(2)). Therefore $Q(G)$ holds (Definition 3.1(4)). ∎

It seems in order to prove that Q_S (for some property Q) is NP-hard using Theorem 3.1, substantial work should be done. We have to come up with an infinite list of critical graphs w.r.t. Q, each having an SCR. However, for all the properties we considered, it is easy to construct "uniform" critical graphs, i.e., graphs with the same structure for every t. The procedure is as follows:

(1) Find a 1-critical graph w.r.t. Q, G_1^Q.

(2) Replicate v_0 in G_1^Q t times to get G_t^Q.

The symmetric structure of G_t^Q guarantees that it has an SCR.

COROLLARY 3.1. *Let G be an undirected graph and k a fixed integer. If Q is one of the properties in the following then Q_S is* NP-*hard.*

(1) *G has an edge,*

(2) *G is connected,*

(3) *G has a triangle (a k-path, a k-cycle),*

(4) *G has a cycle,*

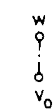
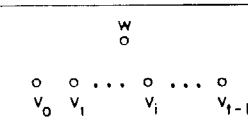
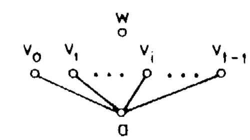
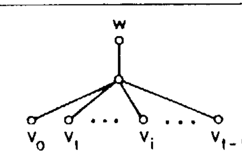
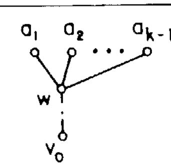
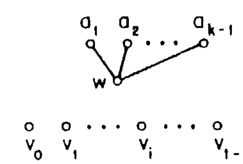
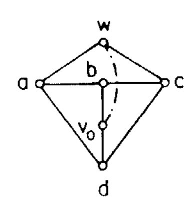
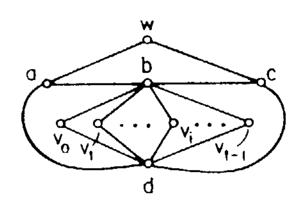
(5) *G is not bipartite (not k-colorable),*

(6) *$\Delta(G) \geqslant k$. ($\Delta(G)$ is the maximum degree in G),*

(7) *G is not planar.*

Proof. The critical graphs for these properties are shown in Table II. ∎

TABLE II

Q	G_1^Q	G_t^Q
1		
2		
3 4 5		
6		
7		

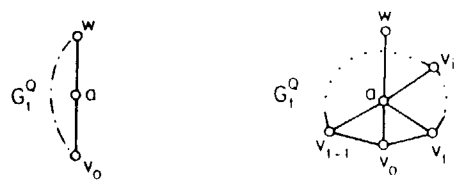


FIGURE 3.2

Sometimes it is not sufficient just to replicate v_0 t times, and we need to build a simple structure on $v_0, v_1, \dots, v_{t-1}$, such as clique, cycle, or path.

COROLLARY 3.2. *If Q is the predicate "G is Hamiltonian" or the predicate "G is not Eulerian" then Q_S is* NP-*hard.*

Proof. The t-critical graphs of the two predicates are given in Figs. 3.2 and 3.3, respectively. ∎

Note that in the proof of Theorem 3.1 we use only the t-critical graphs for t values that are powers of 2. Therefore, it is sufficient to present t-critical graphs for any sequence of integers that contains $\{2^i\}_{i=0}^{\infty}$.

COROLLARY 3.3. *If Q is the predicate "G has a perfect matching," then Q_S is* NP-*hard.*

Proof. We construct G_t^Q for all even integers $t = 2r$. G_{2r}^Q is shown in Fig. 3.4. ∎

The above list of graph properties for which Q_S is NP-hard is by no means exhaustive. One can easily construct critical graphs for many other properties, using the same method. Also, it is not difficult to create a similar list for properties of directed graphs.

We conclude this section by noting that we proved the lower bounds for problems (1)–(4), (6), and (7) in Table I. Since checking if a graph has a triangle, a k-path, a k-cycle or a vertex of degree at least k (k fixed) amounts only to guessing a fixed number of edges and verifying their existence using C_G, we have also the upper bounds on problems (1)–(4) in the table.

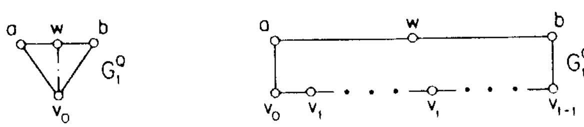


FIGURE 3.3

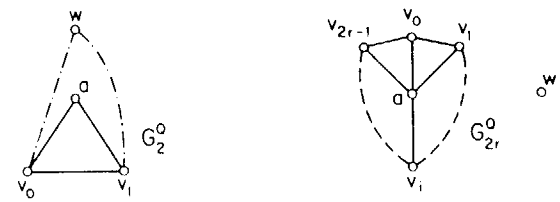


FIGURE 3.4

4. Σ_2- AND Π_2-HARDNESS

In this section we improve the lower bounds of Section 3 for several problems. We first review some known facts and introduce some notation which will be used in this section.

DEFINITION. $C_2 = \{R(X, Y) \mid R(X, Y)$ is a Boolean formula, and for all X there exist Y s.t. $R(X, Y) = 1\}$.

The following useful theorems are proved in Stockmeyer (1977).

THEOREM A. *C_2 is log-complete in Π_2^p.*

THEOREM B. *For a problem PR,*

$$PR \text{ is } \Pi_2\text{-hard (complete)} \Leftrightarrow \neg PR \text{ is } \Sigma_2\text{-hard (complete)}.$$

Let F be $\forall X\, \exists Y\, R(X, Y)$, where $X = \{x_1, \dots, x_r\}$ and $Y = \{y_1, \dots, y_s\}$. By assigning i to X, where $0 \leqslant i \leqslant 2^r - 1$, we mean that we take the binary representation of i, $\bar{i}$, padded with zeros to the left so that $|\bar{i}| = r$, and we assign the kth bit of $\bar{i}$ to x_k. Assigning j to Y has the same meaning. We denote the assignment by $R(\bar{i}, \bar{j})$.

The rest of the section contains the proofs of the lower and upper bound on problem (5), and the lower bounds for problems (8)–(14) in Table I. In the following theorems we polynomially reduce C_2 to Q_S for the property Q under consideration. For every instance $F = R(X, Y)$ (with $|X| = r$ and $|Y| = s$) of C_2 we construct a graph G, s.t. $F \in C_2$ iff G has property Q. Following similar arguments as in Section 3, the graphs constructed have an SCR, so we will not go into the boring details of those small circuits.

THEOREM 4.1. *For Q: "$\delta(G) > k$", where $\delta(G)$ is the minimum degree of G and k is some fixed constant, Q_S is Π_2-complete.*

Proof. (a) $Q_S \in \Pi_2$. Let C_G be an SCR of G. Then Q_S can be represented by the Boolean formula $\forall x\, \exists y_1, \dots, y_k$ $(\bigwedge_{i=1}^{k} C_G(\bar{x}, \bar{y}_i) = 1$, $\bigwedge_{1 \leqslant i < j \leqslant k} y_i \neq y_j)$, where $x, y_1, \dots, y_k$ are the codes of the vertices.

(b) Let Q_{S0} be Q_S with $k=0$. Define $G(V, E)$ (Fig. 4.1) by

$$V=\{x_i \mid 0 \leqslant i \leqslant 2^r-1\} \cup \{y_j \mid 0 \leqslant j \leqslant 2^s-2\},$$
$$E=\{(y_j, y_{j+1}) \mid 0 \leqslant j \leqslant 2^s-2\} \cup \{(x_i, y_j) \mid R(\bar{i}, \bar{j})=1\}.$$

Claim. $F \in C_2 \Leftrightarrow \delta(G)>0$.

It is obvious that the degree of all the y-vertices is greater than 0. For an x_i to be connected to another vertex, there should exist some j for which $(x_i, y_j) \in E$ or in other words $R(\bar{i}, \bar{j})=1$. So $\delta(G)>0 \Leftrightarrow \forall i\, \exists j (x_i, y_j) \in E \Leftrightarrow \forall i\, \exists j\, R(\bar{i}, \bar{j})=1 \Leftrightarrow F \in C_2$.

This proves that Q_{S0} is Π_2-complete. This idea is generalized for every k by adding $k-1$ vertices that are connected to all x_i, y_j. Hence, Q_S is Π_2-complete.

THEOREM 4.2. *For Q, "G is connected" Q_S is Π_2-hard.*

Proof. Let $G(V, E)$ be the graph in Theorem 4.1 (Fig. 4.1). It is easily seen that G is connected iff $F \in C_2$. ∎

THEOREM 4.3. *For Q, "G has a path connecting a and b," Q_S is Π_2-hard.*

Proof. Define $G(V, E)$ (Fig. 4.2) by

$$V=\{a, b\} \cup \{x_i \mid 0 \leqslant i \leqslant 2^r-1\} \cup \{y_{i,j} \mid 0 \leqslant i \leqslant 2^r-1, 0 \leqslant j \leqslant 2^s-1\},$$
$$E=\{(a, x_0)\} \cup \{(y_{i,0}, x_{i+1}) \mid 1 \leqslant i \leqslant 2^r-1\} \cup \{(y_{2^r-1,0}, b)\}$$
$$\cup \{(y_{i,j}, y_{i,j+1}) \mid 0 \leqslant i \leqslant 2^r-1, 0 \leqslant j \leqslant 2^s-2\}$$
$$\cup \{(x_i, y_{i,j}) \mid R(\bar{i}, \bar{j})=1\}.$$

Claim. $F \in C_2 \Leftrightarrow G$ has a path connecting a and b.

In order for a and b to be connected by a path there must exist an edge $(x_i, y_{i,j})\, \forall i$. For all i there exists an edge $(x_i, y_{i,j}) \Leftrightarrow \forall i\, \exists j$ such that $R(\bar{i}, \bar{j})=1 \Leftrightarrow F \in C_2$. ∎

THEOREM 4.4. *For Q, "G is planar," Q_S is Σ_2-hard.*

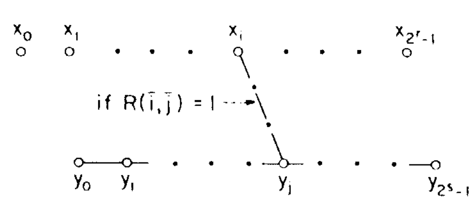


FIGURE 4.1

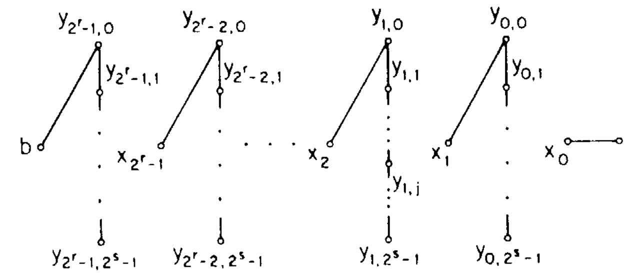


FIGURE 4.2

Proof. We show that $\neg Q_S$ is Π_2-hard. Define $G(V, E)$ by

$$V=\{a, b, c, d, e\} \cup \{x_i \mid 0 \leqslant i \leqslant 2^r-1\}$$
$$\cup \{y_{i,j} \mid 0 \leqslant i \leqslant 2^r-1, 0 \leqslant j \leqslant 2^s-1\},$$
$$E=\{(a, c), (a, d), (a, e), (b, c), (b, d), (b, e), (c, d), (d, e), (c, e),$$
$$(a, x_0), (y_{2^r-1,0}, b)\}$$
$$\cup \{(y_{i,j}, y_{i,j+1}) \mid 0 \leqslant i \leqslant 2^r-1, 0 \leqslant j \leqslant 2^s-2\}$$
$$\cup \{(x_i, y_{i,j}) \mid R(\bar{i}, \bar{j})=1\} \cup \{(y_{i,0}, x_{i+1}) \mid 1 \leqslant i \leqslant 2^r-2\}.$$

This is essentially a complete graph on $\{a, b, c, d, e\}$, except that the edge (a, b) is replaced by the graph of Fig. 4.2. Therefore it is clear that G is nonplanar iff there is a path from a to b, which by the previous theorem happens iff $F \in C_2$. Since $\neg Q_S$ is Π_2-hard, Q_S is Σ_2-hard. ∎

THEOREM 4.5. *For Q, "G is bipartite," Q_S is Σ_2-hard.*

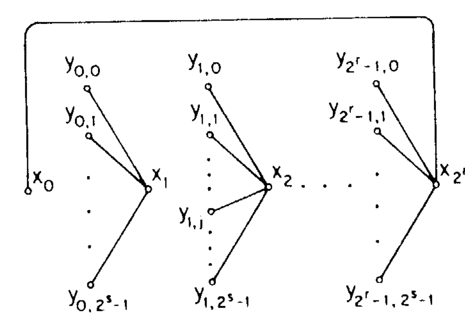


FIGURE 4.3

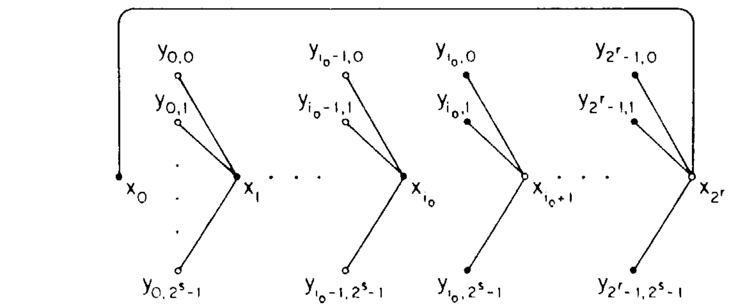


FIGURE 4.4

Proof. We show that $\neg Q_S$ is Π_2-hard. Define $G(V, E)$ (Fig. 4.3) by

$$V = \{x_i \mid 0 \leqslant i \leqslant 2^r\} \cup \{y_{i,j} \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\},$$

$$E = \{(y_{i,j}, x_{i+1}) \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\} \cup \{(x_i, y_{i,j}) \mid R(\bar{i}, \bar{j}) = 1\} \cup \{(x_0, x_{2^r})\}.$$

Claim. $F \in C_2 \Leftrightarrow G$ is not bipartite.

$\Rightarrow$ Suppose $F \in C_2$. Let $j(i)$ be any y-value for which $R(\bar{i}, \overline{j(i)}) = 1$ $(0 \leqslant i \leqslant 2^r - 1)$. Then $\{x_0, y_{0,j(0)}, x_1, y_{1,j(1)}, \ldots, x_{2^r-1}, y_{2^r-1,j(2^r-1)}, x_{2^r}, x_0\}$ is an odd cycle in G and G is not bipartite.

$\Leftarrow$ Suppose $F \notin C_2$, then there exist i_0 such that $\forall j$, $R(\bar{i}_0, \bar{j}) = 0$ so the vertices of G can be colored Black and White (Fig. 4.4) in the following way:

$$\text{Black} = \{x_i \mid 0 \leqslant i \leqslant i_0\} \cup \{y_{ij} \mid i_0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\},$$

$$\text{White} = V\text{-Black}. \quad \blacksquare$$

COROLLARY 4.1. *For Q, "G is k-colorable," Q_S is Σ_2-hard.*

Proof. Connect every vertex of the graph in Fig. 4.3 to all vertices of a $(k-2)$-clique. The new graph is k-colorable iff the original is bipartite. $\blacksquare$

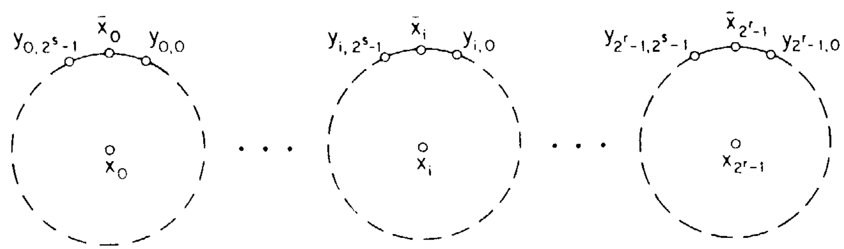


FIGURE 4.5

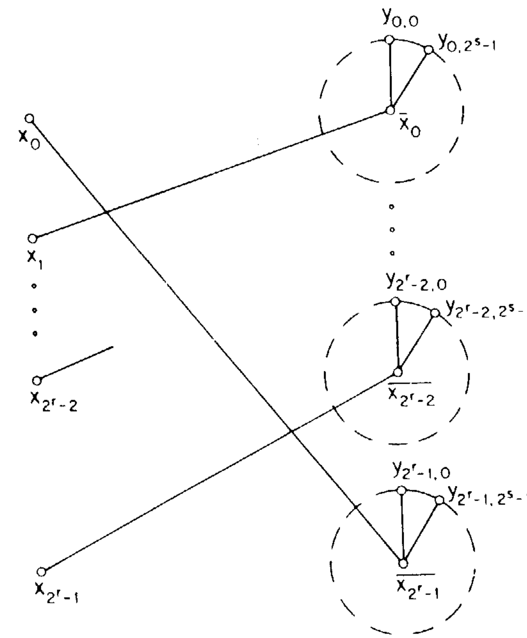


FIGURE 4.6

THEOREM 4.6. *For Q, "G has a perfect matching," Q_S is Π_2-hard.*

Proof. Define $G(V, E)$ (Fig. 4.5) by

$$V = \{x_i, \bar{x}_i \mid 0 \leqslant i \leqslant 2^r - 1\} \cup \{y_{ij} \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\},$$

$$E = \{(y_{i,j}, y_{i,j+1})\} \cup \{(y_{i,0}, \bar{x}_i) \mid 0 \leqslant i \leqslant 2^r - 1\} \cup \{(y_{i,2^s-1}, \bar{x}_i) \mid 0 \leqslant i \leqslant 2^r - 1\} \cup \{(x_i, y_{i,j}) \mid R(\bar{i}, \bar{j}) = 1\}.$$

Claim. $F \in C_2 \Leftrightarrow G$ has a perfect matching.

The ith component of G has a perfect matching iff x_i is connected to any of the y_{ij} or in other words if $\exists j$ such that $R(\bar{i}, \bar{j}) = 1$. So G has a perfect matching $\Leftrightarrow \forall i \, \exists j \, R(\bar{i}, \bar{j}) = 1 \Leftrightarrow F \in C_2$.

THEOREM 4.7. *For Q, "G has a Hamiltonian circuit" Q_S is Π_2-hard.*

Proof. Define $G(V, E)$ (Fig. 4.6) by

$$V = \{x_i, \bar{x}_i \mid 0 \leqslant i \leqslant 2^r - 1\} \cup \{y_{i,j} \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\},$$

$$E = \{(\bar{x}_i, x_{i+1}) \mid 0 \leqslant i \leqslant 2^r - 1\} \cup \{(\bar{x}_i, y_{i,j}) \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\} \cup \{(y_{i,j}, y_{i,j+1}) \mid 0 \leqslant i \leqslant 2^r - 1, 0 \leqslant j \leqslant 2^s - 1\} \cup \{(x_i, y_{i,j}) \mid R(\bar{i}, \bar{j}) = 1\},$$

$F \in C_2 \Leftrightarrow G$ has a Hamiltonian circuit.

$\Leftarrow$ Suppose $F \notin C_2$, then $\exists i_0$ such that $\forall j$, $R(\bar{i_0}, \bar{j}) = 0$. In this case x_{i_0} is connected only to $\overline{x_{i_0-1}}$ and could not be included in a cycle. $\Rightarrow G$ does not have a Hamiltonian circuit.

$\Rightarrow$ Suppose $F \in C_2$. Let $j(i)$ be any y-value for which $R(\bar{i}, \overline{j(i)}) = 1$ $(0 \leqslant i \leqslant 2^r - 1)$. Then $\forall i\, \exists j$ such that $R(\bar{i}, \overline{j(i)}) = 1$. $\Rightarrow \{x_0, y_{0j(0)}, y_{0,j(0)+1}, \ldots, y_{0,j(0)-1}, \bar{x}_0, x_1, y_{1j(1)}, \ldots, y_{2^r-1,j(2^r-1)}, \overline{x_{2^r-1}}, x_0\}$ is a Hamiltonian circuit. ▮

5. Upper Bounds

Define Q_N to be the problem of deciding whether a graph, given by its adjacency matrix, has property Q or not. In this section we show how to convert any algorithm for Q_N into an algorithm for Q_S. This yields simple time and space upper bounds for Q_S. The model of computation we assume is the RAM (Aho *et al.*, 1979).

Let n be the size of an SCR of a graph on m vertices. From the definition of the SCR we have that $n \leqslant c \log^k m$ for fixed constants c and k. Also note that $n \geqslant 2 \log m$ since there are $2 \log m$ input lines in the SCR.

Lemma 5.1. *Given an* SCR *of a graph* $G(V, E)$, *we can construct the adjacency matrix of* G *in time* $O(n2^{2n})$, *where* n *is the size of the* SCR.

Proof. There are $|V|^2 = O(2^{2n})$ entries in the matrix. For each entry we input the binary encoding of the two vertices into the SCR, and fill the entry according to the result. Since this is a combinatorial circuit, the processing time is bounded by the size of the circuit, so computing each entry takes time $O(n)$. The total time is therefore $O(n2^{2n})$. ▮

Theorem 5.1. *Let* A *be an algorithm that solves* Q_N *in time* $T_A(m)$ *for any graph on* m *vertices. There is an algorithm* B *that solves* Q_S *in time* $T_B(n) = O(n2^{2n} + T_A(2^n))$, *where* n *is the size of the* SCR.

Proof. Algorithm B first constructs the adjacency matrix of the input graph from the given SCR. By the lemma, it requires $O(n2^{2n})$ steps. Then it feeds the matrix to algorithm A, which runs in time $T_A(m) = O(T_A(2^n))$ since $m \leqslant 2^n$. Therefore the total number of steps required is $O(n2^{2n} + T_A(2^n))$. ▮

Corollary 5.1.

If $Q_N \in$ P-DTime *then* $Q_S \in$ Exp-DTime.
If $Q_N \in$ P-NTime *then* $Q_S \in$ Exp-NTime.

Since testing whether a graph is planar, bipartite or has a perfect matching is in P, and testing for a Hamiltonian circuit or k-colorability is in NP, the upper bounds 12-16 in Table 1.1 follow from Corollary 5.1.

Theorem 5.2. *Let* A *be an algorithm that solves* Q_N *in space* $S_A(m)$ *for graphs on* m *vertices, where* $S_A(m) \geqslant \log m$. *Then there is an algorithm* B *that solves* Q_S *in space* $S_B(n) \leqslant S_A(2^n)$, *where* n *is the input size of* Q_S.

Proof. Let C_G be the input to Q_S. Algorithm B mimics algorithm A except when A consluts the adjacency matrix, B consults C_G. This is possible since $S_B(n) = S_B(m) \geqslant \log m \geqslant n$. Therefore $S_B(n) = S_A(m) \leqslant S_A(2^n)$. ▮

Corollary 5.2. *For any integer* r:

If $Q_N \in$ DSPACE$(\log^r n)$ *then* $Q_S \in$ DSPACE(n^r).
If $Q_N \in$ NSPACE$(\log^r n)$ *then* $Q_S \in$ NSPACE(n^r).

Given the adjacency matrix of a graph, testing it for an $s-t$ path or connectivity are known to be in NSPACE$(\log |V|)$. Testing for an Eulerian circuit is in the same complexity class, since it is merely a connectivity test plus verifying that all vertices have even degrees, which is easily done in $\log |V|$ space. Therefore the upper bounds (7)-(9) in Table I follow Corollary 5.2.

We are left to prove that testing whether a graph (given by an SCR) has a cycle, takes only $O(n)$ space on a deterministic Turing machine. Hong (1980) gives an algorithm with this upper bound for a certain class of succinctly representable graphs. His algorithm is easily seen to perform similarly when the input graph is given by an SCR.

6. Further Research and Open Problems

Our major motivation in studying succinct representation of graphs comes from the VLSI world. The new technology makes it possible to place on one chip tens of thousands of elements. The layout of a chip forms a graph, whose description by an adjacency matrix would be horrible. Also, those circuits usually have a "uniform" structure which gives rise to hope that they can be represented succinctly. To find out if this idea is practical we investigated the difficulty in testing graph properties on a succinct represen tation. The lower bounds obtained in this paper seem to discourage this idea. However, those results were obtained only for an SCR, which is only one type of succinct representation. In fact, another succinct representation which yields more "positive" results is analyzed in Galperin (1983). Other forms of succinct representation should be examined. These even may be "special purpose" representations designed especially for the types of graphs we find on VLSI chips.

Other Open Problems

(1) Theorem 3.1 gives sufficient conditions for a graph predicate to be NP-hard. We conjecture that for every "nontrivial" graph property Q, the relevant decision problem on succinct input Q_S, is NP-hard. The term "nontrivial" graph property should be defined. A possible definition could be a property that has infinitely many critical graphs. Note that we do not require that those critical graphs be succinctly representable.

(2) Table II leaves a lot of room for improvement. One can try to improve the upper and lower bounds for predicates in the table, or work on other properties. One of the difficulties we could not overcome in proving lower bounds, was to show that a problem is hard for Π_i or Σ_i, $i \geqslant 3$. This may require different techniques then those we developed to probe NP-hardness and Σ_2/Π_2-hardness.

(3) Characterize classes of graphs that can be represented succinctly.

RECEIVED: August 5, 1983; ACCEPTED: September 22, 1983

ACKNOWLEDGMENTS

We are grateful to Professor Richard Lipton for guiding us in this research. We also thank a careful referee for his comments.

REFERENCES

AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. (1979), "The Design and Analysis of Computer Algorithms," Addison–Wesley, Reading, Mass.

COOK, S. A. (1971), The complexity of theorem proving procedures, *in* "Proceedings, Third Annual STOC," pp. 151–158.

GALPERIN, H. (1982), "Succinct Representations of Graphs," Ph.D. thesis, Department of Electrical Engineering and Computer Science, Princeton University, Princeton, N.J., August.

HONG, J.-W. (1980), On some deterministic space complexity problems, *in* "Proceedings of SIGACT Conference," pp. 310–317.

HOPCROFT, J. E., AND ULLMAN J. D. (1979), "Introduction to Automata Theory, Languages and Computation," Addison–Wesley, Reading, Mass.

STOCKMEYER, L. J. (1977), The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3**, 1–22.

INFORMATION AND CONTROL 56, 199–211 (1983)

On Storage Media with Aftereffects

H. S. WITSENHAUSEN AND A. D. WYNER

Bell Laboratories, Murray Hill, New Jersey 07974

Successive independent batches of data are written into a memory and then read out. The behavior of a memory cell in each read/write cycle is dependent (this is the "aftereffect") on its previous history. Each cell can be modelled as an automaton. Using a large number of identical cells and coding/decoding across the memory, what is the maximum throughput that can be achieved in N cycles with negligible error probability?

This problem is equivalent (by a time/space interchange) to finding the total capacity of a certain multiuser interference channel.

Exact answers are obtained for the cell-model in which the output of a cell is the exclusive *OR* of the two most recent inputs. For the (more realistic) inclusive *OR*, lower and upper bounds are determined.

The increase in throughput obtainable by delaying some of the read cycles is also determined or bounded.

1. INTRODUCTION

We consider a medium made up of a large number N of independent cells, each capable of staring one bit of information. The medium is used in the following way: For $t = 1, 2, \ldots, T$, data from a source S_t, possibly encoded, is stored at time t and read at time $t + \theta_t$ by a user who seeks to recover the data from S_t with arbitrarily small error probability. This user does not care about the sources S_τ $(\tau \neq t)$. The sources are independent.

If the cells function perfectly and $0 < \theta_t < 1$, then N bits of data can be transmitted this way from each source. This is a rate of 1 bit per cell at each usage cycle.

Now consider the case of imperfect cells, each cell being a copy of a certain stochastic or deterministic finite state machine. (In the stochastic case, the randomness in each cell is assumed independent of that in all other cells.) Then, the aftereffect on the cells of usage in prior time periods introduces errors in the current storage-retrieval cycle. As the data of previous cycles is only known in distribution, not in realization, one is faced with a noisy channel and error-correcting codes must be used. As N is large, the capacity of this channel will indicate the maximum possible rate of transmission.

0019-9958/83 $3.00