ELSEVIER

Information Processing Letters 51 (1994) 149–153

Information Processing Letters

Enumerating the kernels of a directed graph with no odd circuits

Jayme L. Szwarcfiter [a,*], Guy Chaty [b]

[a] Universidade Federal do Rio de Janeiro, Instituto de Matematica and Núcleo de Computação Electrônica, Caixa Postal 2324, 20001 Rio de Janeiro, RJ, Brazil

[b] Université de Paris-Nord, Laboratoire d'Informatique de Paris-Nord, Av. Jean-Baptiste Clément, 93430 Villetaneuse, France

Received 7 October 1993
Communicated by T. Lengauer

Abstract

The problems of generating and counting the number of kernels of a directed graph G with no odd circuits are considered. An algorithm is described for generating all distinct kernels of G. Its complexity is $O(nm(k+1))$, where n, m and k are the number of vertices, edges and kernels of G. In contrast, we show that the problem of determining the number of kernels of G is #P-complete, even if the longest circuit of G has length 2. A special case in which the counting problem can be solved in polynomial time is also presented.

Key words: #P-complete problems; Algorithms; Computational complexity; Directed graphs; Kernels

1. Introduction

We employ the term *graph* with the meaning of a directed graph, unless otherwise stated. A kernel of a graph G is a set of non-adjacent vertices S, such that every vertex of G reaches S by a directed path of length at most 1. We consider the problems of generating and counting the number of distinct kernels of a graph G, with no odd circuits. If G is an arbitrary graph, deciding whether it has a kernel is well known to be NP-complete [3]. However, if G has no odd circuits, then it necessarily has one [12]. We describe an algorithm for generating all kernels of G, in this case. The complexity for generating each new kernel of the collection is the product of the number of vertices and edges of G. In contrast, we prove that the corresponding counting problem is #P-complete. It remains so, even if the length of the longest circuit of G is 2. The method employed implies that the number of kernels of graphs with no odd circuits is always a power of 2. Finally, for a special subclass of these graphs, we describe an algorithm which computes the number of kernels in polynomial time.

Kernels of graphs with no odd circuits have been of interest for some time [1,2,4–6,8–12]. An algorithm for generating all kernels of an arbitrary graph has been described in [7]. Its complexity is exponential per kernel. A motivation for formulating an algorithm for graphs with no odd circuits is that the complexity per kernel decreases to polynomial time.

* Corresponding author. The work has been written while this author was visiting the Laboratoire d'Informatique de Paris-Nord, Université de Paris-Nord, Villetaneuse, France.

Elsevier Science B.V.
SSDI 0020-0190(94)00072-7

The vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, respectively. G is *trivial* when $|V(G)| = 1$. The *subjacent* graph of G is its underlying undirected graph. For each $v \in V(G)$, let

$$N^-(v) = \{w \in V(G) \mid (w, v) \in E(G)\}$$

and

$$N^+(v) = \{w \in V(G) \mid (v, w) \in E(G)\}.$$

For $S \subset V(G)$, $N^-(S) = \bigcup_{v \in S} N^-(v)$ and $N^+(S) = \bigcup_{v \in S} N^+(v)$. Write $N^-[S] = N^-(S) \cup S$ and $N^+[S] = N^+(S) \cup S$. If G contains a path from vertex v to w then v is an *ancestor* of w, and w a *descendant* of v. An *ancestor* (*descendant*) of $S \subset V(G)$ is a vertex which is an ancestor (descendant) of some $w \in S$. G is *strongly connected* when for every pair of vertices, each one is an ancestor of the other. The *strongly connected components* of G are the maximal strongly connected subgraphs C_i of G. C_i is *mimimal* when $N^+(V(C_i)) \setminus V(C_i) = \emptyset$. $S \subset V(G)$ is an *independent set* when $S \cap N^+(S) = \emptyset$. If, additionally, $V(G) \setminus S \subset N^-(S)$ then S is a *kernel* of G.

Denote by $\Psi(G)$ the set of kernels of G, $n = |V(G)|$, $m = |E(G)|$ and $k = |\Psi(G)|$.

2. Generating the kernels

Lemma 1. *Let C be a minimal strongly connected component of a directed graph G and S a kernel of C. Then S' is a kernel of $G \setminus N^-[S]$ iff $S \cup S'$ is a kernel of G.*

Proof. ($\Rightarrow$) If $G = C$ there is nothing to prove. Otherwise, suppose S, S' are kernels of C and $G \setminus N^-[S]$, respectively. $S \cup S'$ must be an independent set. Otherwise there exist vertices $v \in S$ and $v' \in S'$, such that either (v, v') or (v', v) is an edge of G. In the former case, C is not minimal, which is a contradiction. In the latter, $v' \in N^-[S]$. But this contradicts $v' \in S'$, since S' is a subset of vertices of $G \setminus N^-[S]$. Hence $S \cup S'$ is an independent set. Suppose it is not a kernel of G. Then there exists a vertex w such that $w \notin N^-[S] \cup N^-[S']$. Clearly, w is not a vertex neither from C nor from $V(G) \setminus N^-[S]$, because they admit S and S' as kernels, respectively. Hence $w \in N^-[S] \setminus V(C)$. Since $V(G) \neq V(C)$, this contradicts our assumption. Consequently, $S \cup S'$ is a kernel of G.

($\Leftarrow$) Let S, S^* be kernels of C and G, such that $S \subset S^*$. We show that the independent set $S^* \setminus S$ is a kernel of $G \setminus N^-[S]$. Suppose the contrary. Then $G \setminus N^-[S]$ has a vertex w such that $w \notin N^-[S^* \setminus S]$. Since S^* is a kernel of G, $w \in N[S^*]$. Consequently, $w \in N^-[S]$, contradicting w to be a vertex of $G \setminus N^-[S]$. □

Lemma 2. *Let G be a strongly connected directed graph. Then G contains no odd circuit iff its subjacent undirected graph is bipartite.*

Proof. ($\Rightarrow$) Let G be a strongly connected graph with no odd circuits. Suppose its subjacent graph G_u is not bipartite. Then it contains an odd undirected cycle Y_u. For each undirected edge (v_i, v_j) of Y_u, let p_{ij} and p_{ji} denote the shortest paths in G from v_i to v_j and from v_j to v_i, respectively. Clearly, p_{ij} or p_{ji} has length one, as (v_i, v_j) or (v_j, v_i) is an edge of G. Hence all p_{ij} and p_{ji} paths must be of odd length. Otherwise, if p_{ij} is of even length then it forms with the edge $(v_j, v_i) \in E(G)$ an odd circuit, a contradiction. Similarly for p_{ji}.

Next, traverse Y_u in any fixed direction and obtain a directed circuit Y'. Let P be the digraph constructed from Y', by replacing each directed edge $(v_i, v_j) \in E(Y')$ by the corresponding shortest path p_{ij} in G. Then P is an eulerian directed subgraph of G, having an odd number of edges. Let Y be a circuit of P. If $Y = P$ then the lemma holds, since $|Y|$ would be odd, a contradiction. When $Y \neq P$ remove Y from P. The remaining digraph is still eulerian, but its number of circuits has been decreased.

($\Leftarrow$) Clear. □

Corollary 3. *Let G be a directed graph containing no odd circuits, C a minimal strongly connected component of it and $\Psi(G)$ the set of kernels of G. Let $V_0 \cup V_1 = V(C)$ be a bipartition of the subjacent graph of C. Then $S \in \Psi(G)$ iff $S = V_i \cup S'$, where*

(i) $i \in \{0, 1\}$,
(ii) $V_i \neq 0$, *and*
(iii) $S' \in \Psi(G \setminus N^-[V_i])$.

Proof. ($\Rightarrow$) Let $S \in \Psi(G)$. By Lemma 1, $S = S'' \cup S'$, where S'' is a kernel of C and S' a kernel of $G \setminus N^-[S'']$. Since C is strongly connected, $V_0 \cup V_1 = V(C)$ and V_i is an independent set, it follows $V(C) \setminus V_i \subset N^-(V_i)$, provided $V_i \neq 0$, $i \in \{0, 1\}$. Hence, V_i is a kernel of C and the theorem holds.

($\Leftarrow$) Let V_i,S' satisfying (i)–(iii) above and $S = V_i \cup S'$. Since $V_i \neq 0$, by a similar argument as above, we conclude that V_i is a kernel of C. Then applying Lemma 1, it follows $S \in \Psi(G)$. □

The algorithm can now be described. The input is a directed graph G. A recursive procedure KERNEL(H, S) is employed. H is the subgraph of G in consideration, and S the current kernel of G being constructed. S contains solely vertices of $V(G) \setminus V(H)$. The procedure is invoked by the external call KERNEL(G, $\emptyset$). The algorithm either finds all kernels of G or reports that G has an odd circuit.

The computation of KERNEL(H, S) is as follows. When $V(H) = \emptyset$, S is a kernel of G and can be reported. Otherwise, find a minimal strongly connected component C of H. By Lemma 2, the subjacent graph C_u of C is bipartite. Hence a bipartition $V_0 \cup V_1 = V(C)$ can be determined, $V_0 \neq \emptyset$. Clearly, $V_1 = 0$ *iff* C is trivial. Then for $i = 0,1$, such that $V_i \neq \emptyset$, V_i is a kernel of C and recursively compute KERNEL($H \setminus N^-[V_i]$,$S \cup V_i$). At the end of KERNEL(H, S), all kernels of G containing S have been generated.

The following formulation describes the process.

procedure KERNEL(H, S)

if $V(H) = \emptyset$ **then output** kernel S of G
else find a minimal strongly connected component C of H
  choose $v \in V(C)$
  $V_0 := \{v\}$; $V_1 := \emptyset$; $Z := V(C)$;
  while $Z \neq \emptyset$ **do**
    choose $z \in Z \cap (V_0 \cup V_1)$
    for $w \in N^+(z)$ **do**
      if $z \in V_0$ **then** $V_1 := V_1 \cup \{w\}$
      else $V_0 := V_0 \cup \{w\}$
    $Z := Z \setminus \{z\}$
  if $V_0 \cap V_1 \neq \emptyset$ **then stop**: G has an odd circuit
  for $i = 0, 1$ **do**
    if $V_i \neq \emptyset$ **then** KERNEL($H \setminus H^-[V_i]$, $S \cup V_i$)

All the operations involved in the computation of each KERNEL(H, S) can be performed in time $O(n + m)$. There can be $O(n)$ calls of the procedure before a new kernel is generated. The time bound is $O(nm(k + 1))$.

3. The counting problem

The proposition below follows from Corollary 3.

Corollary 4. *Let G be a graph with no odd circuits. Then G has exactly 2^p kernels, $0 \leqslant p \leqslant \lfloor n/2 \rfloor$.*

In addition, for each n, there is a graph G with no odd circuits and 2^p kernels, for all $0 \leqslant p \leqslant \lfloor n/2 \rfloor$.

Theorem 5. *Counting the number of distinct kernels of a graph with no odd circuits is #P-complete, even if the length of the longest circuit of the graph is 2.*

Proof. The proof is a transformation from the problem of counting the number of truth assignments of a boolean expression E, in conjunctive normal form. Let $v_1, \ldots, v_p$ and $Z_1, \ldots, Z_q$ be the variables and clauses of E, respectively. Each variable v_i corresponds to the literals v_i and $\bar{v}_i$, where v_i is *true* iff $\bar{v}_i$ is *false*. We construct a directed graph G with no odd circuits, with the following property. Knowing the number of distinct kernels of G leads to determining, by an easy computation, the number of distinct truth assignments of E, a known #P-complete problem [13,14]. The construction of G can be done in polynomial time, with respect to the size of the satisfiability problem.

G has $q + 2p + 4$ vertices and $q + 2p + l + 5$ edges, l the total number of occurrences of literals in E. Each variable v_j of E corresponds to two vertices v_{j0}, v_{j1} of G. Each clause Z_i of E is associated to a vertex z_i of G. In addition, G

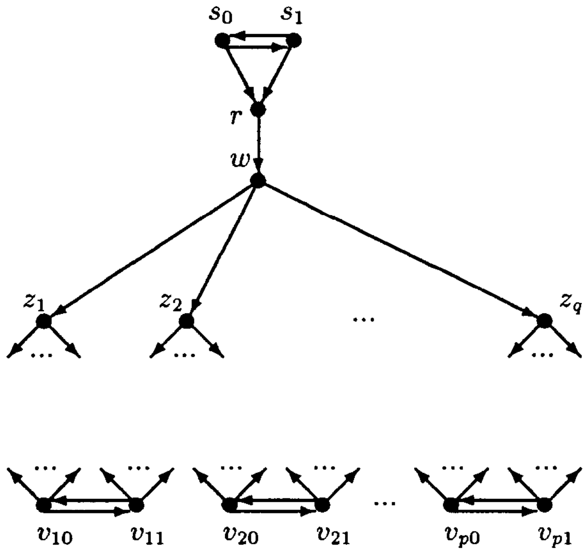

Fig. 1.

contains the vertices s_0, s_1, r and w. The edges of G are defined as follows. For each variable v_j, G contains the directed edges (v_{j0}, v_{j1}) and (v_{j1}, v_{j0}). For each occurrence of the literal v_j or $\bar{v}_j$ in Z_i, G contains the edge (z_i, v_{j0}) or (z_i, v_{j1}), respectively. For each clause Z_i, G contains the edge (w, z_i). The remaining edges of G are (r, w), (s_0, r), (s_1, r), (s_0, s_1) and (s_1, s_0). This completes the description of G. See Fig. 1.

We show below that G has k kernels iff E admits $k - 2^p$ truth assignments.

Let f be a boolean assignment of the variables of E. That is, for each literal v_j, $f(v_j)$ is either *true* or *false*. Write $f_G(v_j) = v_{j0}$ if $f(v_j) = true$, or $f_G(v_j) = v_{j1}$, otherwise. Suppose E has the value *true* for f. Let $S_0,S_1 \subset V(G)$ be defined as follows:

$$S_0 = \{f_G(v_1), \dots, f_G(v_p), w, s_0\},$$

$$S_1 = \{f_G(v_1), \dots, f_G(v_p), w, s_1\}.$$

We show that S_0 is a kernel of G. Clearly, S_0 is an independent set. In addition, s_1, $w \in N^-(S_0)$. Finally, since E has the value *true*, each Z_i contains a *true* literal. That is, for each $z_i \in V(G)$, there is an edge $(z_i, f_G(v_j)) \in E(G)$, for some $1 \leqslant j \leqslant p$. Hence S_0 is a kernel of G. Similarly, S_1 is a kernel of G.

Next, suppose E has the value *false* for f. Let Z' be the set of clauses of E, having the value *false* for f. Clearly, $Z' \neq \emptyset$. Let z' be the corresponding subset of vertices of G, i.e. $Z_i \in Z'$ iff $z_i \in z'$. Define $S \subset V(G)$ as follows

$$S = \{f_G(v_1), \dots, f_G(v_p), r\} \cup z'.$$

We show that S is a kernel of G. S is an independent set because $(z_i, f_G(v_j)) \notin E(G)$, for each $z_i \in z'$ and all $j = 1, \dots, p$, otherwise clause $Z_i \in Z'$ has the value *true*, a contradiction. In addition, $w \in N^-(S)$ because $(w, z_i) \in E(G)$, for $z_i \in z'$. Also, $s_0,s_1 \in N^-(S)$. Finally, for $i = 0,1$, $v_{ji} \neq f_G(v_j)$ implies $(v_{ji}, f_G(v_j)) \in E(G)$ and therefore $v_{ji} \in N^-(S)$. Hence S is a kernel of G. Moreover, whenever E has the value *false* for the assignment f, S is the only kernel of G containing $\{f_G(v_1), \dots, f_G(v_p)\}$. Because in this case, the vertices of z' must all belong to S, which implies $w \notin S$, $r \in S$ and $s_0,s_1 \notin S$.

Consequently, for each assignment f, G has exactly one kernel whenever E has the value *false* for f, and two kernels if E is *true*. Since there are 2^p possible assignments, we conclude that G has k kernels iff E is satisfied by $k - 2^p$ assignments. In addition, the construction of G as well as the representation of the value k can be done in time polynomial in p, q. The theorem follows. □

Next, consider the following special class of graphs.

Let G be a directed graph and $v,w \in V(G)$. A path from v to w is *forbidden* when v, w belong to distinct non-trivial strongly connected compo-

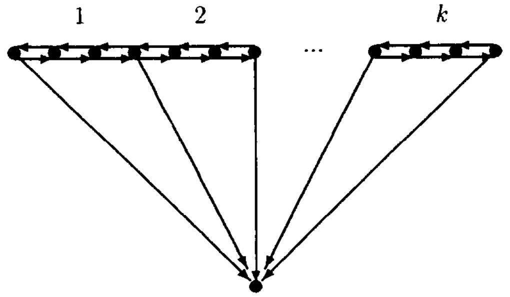

Fig. 2.

nents of G. The following proposition shows how to compute the number of kernels of a directed graph with no odd circuits nor forbidden paths.

Lemma 6. *Let G be a graph with no odd circuits nor forbidden paths, $\mathscr{C}$ the family of non-trivial strongly connected components C_i of G, Z the set of ancestors of $\bigcup_{C_i \in \mathscr{C}} V(C_i)$ and S a kernel of $G \setminus Z$. Then G has exactly 2^p kernels, p the number of non-trivial strongly connected components of $G \setminus N^-[S]$.*

Proof. We first show that S' is a kernel of $G \setminus N^-[S]$ iff $S \cup S'$ is a kernel of G. Let S' be a kernel of $G \setminus N^-[S]$. Then $S \cup S'$ is an independent set. Let $v \in V(G) \setminus (S \cup S')$. If $v \in V(G) \setminus Z$ then $v \in N^-(S)$, since S is a kernel of $G \setminus Z$. Otherwise, $v \in Z \setminus S'$, and $v \in N^-[S]$ or $v \in N^-[S']$. Hence $S \cup S'$ is a kernel of G. Conversely, let S^* be a kernel of G. Since $G \setminus Z$ is acyclic, S is unique and $S \subset S^*$. Hence $S^* \setminus S$ must be a kernel of $G \setminus N^-[S]$. Therefore $|\Psi(G)| = |\Psi(G \setminus N^-[S])|$. Since G has no forbidden paths, each non-trivial strongly connected component C_i' of $G \setminus N^-[S]$ is minimal, $1 \leqslant i \leqslant p$. By Lemma 1, every kernel S' of $G \setminus N^-[S]$ must contain a kernel of C_i', $1 \leqslant i \leqslant p$. Because G has no odd circuits, each C_i' has exactly two distinct kernels. Every choice is possible. Therefore $|\Psi(G \setminus N^-[S])| \geqslant 2^p$. Let S'' be a kernel of $\cup C_i'$. Then $N^-[S'']$ is acyclic. That is, each S'' extends to a kernel of $G \setminus N^-[S]$ and conversely. Hence $|\Psi(G \setminus N^-[S])| = 2^p$. □

By standard techniques, it can be can decided if a graph has no forbidden paths nor odd circuits, in time $O(nm)$. A procedure for counting the number of kernels, within this same complexity, can be described using Lemma 6. Note that graphs of this class may have an exponential number of kernels. For example, the graph of Fig. 2 has $4k$ vertices and 2^k kernels, $k > 1$.

References

[1] C. Berge and P. Duchet, Recent problems and results about kernels in directed graphs, *Discrete Math.* **86** (1990) 27–31.

[2] G. Chaty and F. Levy, Default logic and kernels in digraphs, Rapport LIPN 91-9, Université de Paris-Nord, Villetaneuse, France, 1991.

[3] V. Chvátal, On the computational complexity of finding a kernel, Rapport CRM-300, Centre des Recherches Mathèmatiques Université de Montreal, Montreal, Canada, 1973.

[4] P. Duchet, Graphes noyau-parfaits, in: *Annals of Discrete Mathematics* **9** (North-Holland, Amsterdam, 1980). 93–101.

[5] P. Duchet and H. Meyniel, A note on kernel-critical graphs, *Discrete Math.* **33** (1981) 103–105.

[6] P. Duchet and H. Meyniel, Une généralisation du théorème de Richardson sur l'existence de noyaux dans le graphes orientés, *Discrete Math.* **43** (1983) 21–27.

[7] C.N. Frangakis, A. backtracking algorithm to generate all kernels of a Directed Graph, *Internat. J. Comput. Math.* **10** (1981) 35–41.

[8] H. Galeana-Sanchez and V. Neuman-Lara, A counterexample to a conjecture of Meyniel on kernel-perfect graphs, *Discrete Math.* **41** (1982) 105–107.

[9] H. Galeana-Sanchez and V. Neuman-Lara, On kernels and semikernels of digraphs, *Discrete Math.* **48** (1984) 67–76.

[10] F. Maffray, Une étude structurelle des graphes parfaits, Thèse de Doctorat, Université Joseph Fourier, Grenoble, France, 1992.

[11] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior* (Princeton University Press, Princeton, NJ, 1944).

[12] M. Richardson, Solutions of irreflexive relations, *Ann., of Math.* **58** (1953) 573–580.

[13] I. Simon, On some central problems in computational complexity, Ph.D. Thesis, Dept. of Computer Science, Cornell University, Ithaca, NY, 1975.

[14] L.G. Valiant, The complexity of computing the permanent, Tech. Rept. CSR-14-77, University of Edinburgh, Edinburgh, 1977.