Relatório Técnico

Núcleo de Computação Eletrônica

Hamiltonian Problems for Reducible Flowgraphs

Oswaldo Vernet
NCE/UFRJ
Lilian Markenzon
PUC/RJ

NCE - 02/97

Universidade Federal do Rio de Janeiro

Hamiltonian Problems for Reducible Flowgraphs

Oswaldo Vernet
Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, Brasil
oswaldo@nce.ufrj.br

Lilian Markenzon
Pontifícia Universidade Católica
Rio de Janeiro, RJ, Brasil
lilian@nce.ufrj.br

Abstract. In this paper, we discuss hamiltonian problems for reducible flowgraphs. The main result is finding, in linear time, the unique hamiltonian cycle, if it exists. In order to obtain this result, two other related problems are solved: finding the hamiltonian path starting at the source vertex and finding the hamiltonian cycle given the hamiltonian path.

Keywords. Flowgraphs, reducibility, hamiltonian paths, hamiltonian cycles.

1 – Introduction

Reducible flowgraphs are digraphs that model the control flow of computers programs. In this paper we discuss hamiltonian problems for this family.

Some path problems have already been studied for flowgraphs. Gabow, Maheshwari and Osterweil [GM76] show an efficient algorithm to solve the multiple node constrained path problem; they also show that the impossible pairs constrained path problem is NP-complete, even for an acyclic digraph. Related to impossible pairs is the notion of must pairs; Ntafos and Hakimi [NH79, NH81] show that finding a path that does not violate any must pairs is an NP-complete problem, even for acyclic D-structured flowgraphs.

Recently, dynamic algorithms have been developed for reducible flowgraphs. [RR94] and [CR88] have solved the problem of incrementally maintaining the dominator tree of a reducible flowgraph under edge insertions and deletions.

The main result of this paper is finding, if it exists, the hamiltonian cycle of a reducible flowgraph; we prove that this cycle is unique. In order to obtain this result two other related problems are solved: the hamiltonian path in which the starting point is the source of the flowgraph and the hamiltonian cycle, given the hamiltonian path. All the problems studied here are known to be NP-complete if G is a general digraph [GJ79].

We solve these problems for reducible flowgraphs in linear time. These results have been developed in [V97].

In section 2 we present the background concepts needed. Section 3 presents the linear algorithm for the hamiltonian path starting at the source vertex of the reducible flowgraph; section 4 presents the solution, also in linear time, for the hamiltonian cycle problem in reducible flowgraphs.

2 - Background and Notation

A *digraph* is a pair of finite sets $D = (V, E)$, where V is the set of *vertices*, E is the set of *edges* and $E \subseteq V \times V$. A *path of length* $k \geq 0$ *from* vertex v *to* vertex w in D is a sequence of vertices $[v = x_0, x_1, \ldots, x_k = w]$ where $(x_i, x_{i+1}) \in E$, $0 \leq i < k$. A path $[v]$ of length 0 is a *trivial* path. A *cycle* is a non-trivial path from a vertex to itself. A digraph is *acyclic* if it contains no cycles. A *hamiltonian path* is a path in which every vertex in V appears exactly once. A *hamiltonian cycle* is a cycle in which every vertex appears exactly once, except the first and last vertices that are the same.

A *flowgraph* is a triple $G = (V, E, s)$, where (V, E) is a digraph, $s \in V$ is a distinguished *source vertex*, and there is a path from s to every other vertex in V. A *directed rooted tree* is a flowgraph $T = (V, E, r)$ with $|E| = |V| - 1$.

Given $v, w \in V$ in a flowgraph $G = (V, E, s)$, v *dominates* w if v lies on every path from s to w. Thus, every vertex dominates itself and the source vertex s dominates all vertices in V. For every pair of vertices $v, w \in V$, the greatest common dominator $\mathrm{GCD}\{v, w\}$ is defined as the unique vertex z such that z dominates v and w and every other common dominator of v and w dominates z.

A flowgraph can be traversed according to predefined rules, such as those of a *depth first search* (DFS). DFS can be implemented as a recursive procedure that automatically maintains a search path stack. Vertices are numbered according to the order in which they are stacked ($dfsin(v), \forall v \in V$) and unstacked ($dfsout(v), \forall v \in V$) during the search.

As a result of performing a DFS on a flowgraph starting at the source vertex, the set of edges is divided into four disjoint subsets, the *tree*, *forward*, *back* and *cross* edges, respectively. The set of tree edges determines a directed tree rooted at s, called a *depth first tree*. A description of DFS can be found in [Mc90], for instance.

Let $G = (V, E, s)$ be a flowgraph and B, the set of back edges resulting from a DFS on G. The acyclic flowgraph $\mathrm{dag}(G) = (V, E - B, s)$ is called the *directed acyclic graph* (*dag*) *associated to* G.

The following are equivalent definitions for a reducible flowgraph $G = (V, E, s)$ [HU72, HU74]:

(i) Any DFS of G starting at s determines the same set B of back edges.

(ii) G does not contain the forbidden subflowgraph $\mathrm{SP}(s, x, y, z)$, shown in Figure 1.

(iii) For every back edge $(v, w) \in B$, w dominates v.

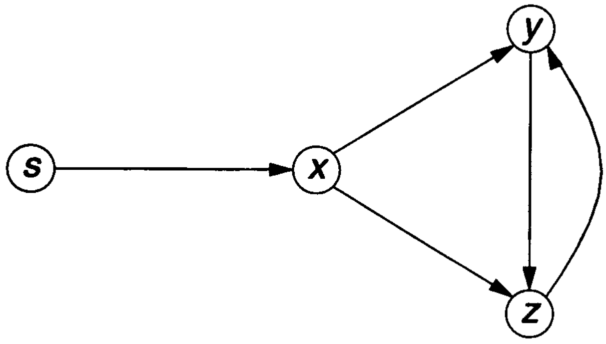

Figure 1: The forbidden subflowgraph (edges represent paths)

A topological ordering τ of an acyclic flowgraph $G = (V, E, s)$ is a bijective numbering $\tau: V \rightarrow \{1, \ldots, |V|\}$ of its vertices such that $\forall (v, w) \in E$, $\tau(v) < \tau(w)$. An $O(|E|)$-time algorithm for finding this ordering is to carry out a depth first search and order the vertices in decreasing order as they are unstacked from the search path stack (i.e., $\tau(v) = |V| + 1 - \mathit{dfsout}(v), \forall v \in V$) [Tar83].

A flowgraph G is supposed to be represented through its set of adjacency lists $A_G(v) = \{w \mid (v, w) \in E\}$, one for each $v \in V$.

3 - The Hamiltonian Path

The first problem that we consider is to find a hamiltonian path starting at the source vertex s of a reducible flowgraph $G = (V, E, s)$. Some results must be presented:

Lemma 1: Let $G = (V, E, s)$ be a reducible flowgraph. If G has a hamiltonian path starting at s, then this path is unique.

Proof:

Let us suppose, by contradiction, that G has two distinct hamiltonian paths P_1 and P_2 starting at s. So, there must be vertices $x, y \in V$ such that x precedes y in P_1 and y precedes x in P_2:

$$P_1 = [s, \ldots, x, \ldots, y, \ldots] \quad \text{and} \quad P_2 = [s, \ldots, y, \ldots, x, \ldots]$$

Hence, neither x dominates y nor y dominates x. Let $z = \mathrm{GCD}\{x, y\}$. Clearly, z precedes x in P_1, z precedes y in P_2 and there are subpaths $[z, \ldots, x] \subset P_1$ and $[z, \ldots, y] \subset P_2$ having only vertex z in common.

Let us consider now the subpaths $S_1 = [x, \ldots, y] \subset P_1$ and $S_2 = [y, \ldots, x] \subset P_2$. Let $S = \{v \in V | v \in S_1 \wedge v \in S_2\} - \{x, y\}$. If $S = \emptyset$, then G contains the forbidden subflowgraph $\mathrm{SP}(s, z, x, y)$. If S is not empty, let $t \in S$ such that t precedes, in S_1, any other vertex in S. Then, G contains $\mathrm{SP}(s, z, x, t)$. In both cases, G is not reducible, as shown in Figure 2. ∎

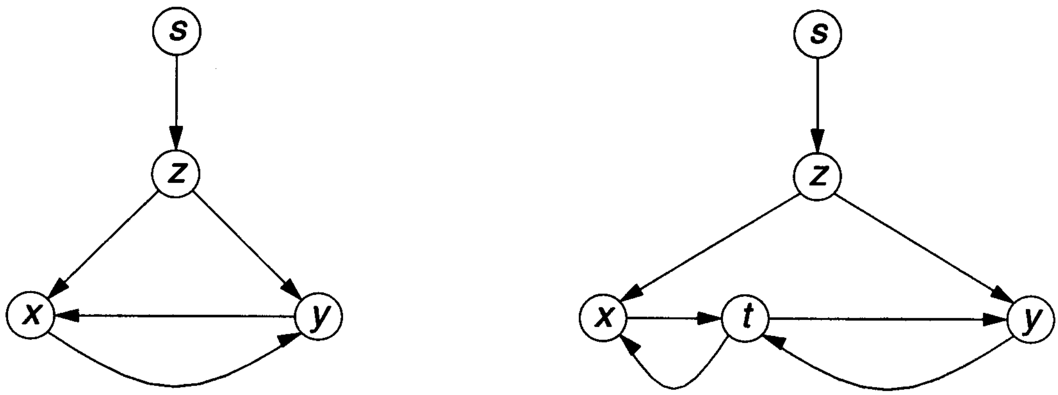

Figure 2: Proof of lemma 1 (edges represent paths)

Lemma 2: Let $G = (V, E, s)$ be a reducible flowgraph. If G has a hamiltonian path starting at s, then this path lies on $\mathrm{dag}(G)$.

Proof:

Let $P = [s, \ldots, v, w, \ldots]$ be the unique hamiltonian path of G starting at s. Let us suppose, by contradiction, that (v, w) is a back edge. As G is reducible, w dominates v and w must precede v in any path starting at s. Hence, w appears twice in P, and P is not hamiltonian. ∎

Lemma 3: Let $G = (V, E, s)$ be a reducible flowgraph. If G has a hamiltonian path starting at s, then *dfsout*$(v), \forall v \in V$ is the same, no matter the depth first tree.

Proof:

As the hamiltonian path lies on $\mathrm{dag}(G)$, this dag admits a unique topological ordering τ, where $\tau(v)$ corresponds to the position of vertex v in the hamiltonian path. Thus, *dfsout*(v) does not depend on the depth first tree, since *dfsout*$(v) = |V| + 1 - \tau(v)$, $\forall v \in V$. ∎

By lemmas 1, 2 and 3, we are able to present a linear algorithm to determine whether a reducible flowgraph $G = (V, E, s)$ has a hamiltonian path starting at the source vertex s. It is sufficient to verify whether $\text{dag}(G)$ has such a path. The main steps are:

step 1: compute a topological numbering τ for the vertices of $\text{dag}(G)$.

step 2: for all $v \in V$, test if there exists $w \in A_G(v)$ such that $\tau(w) = \tau(v) + 1$.

Steps 1 and 2 may be performed simultaneously during a DFS on G, based on the computation of $dfsout(v), v \in V$ (section 1). As no new computations are added, the algorithm complexity is the same of the depth first search for connected digraphs, $O(|E|)$.

It is important to observe that the results presented are only valid if the source vertex s is taken as the starting point of the hamiltonian path. Figure 3 shows a reducible flowgraph; $[d, e, f, b, c, s, a]$ and $[d, e, f, s, a, b, c]$ are distinct hamiltonian paths, both starting at vertex d.

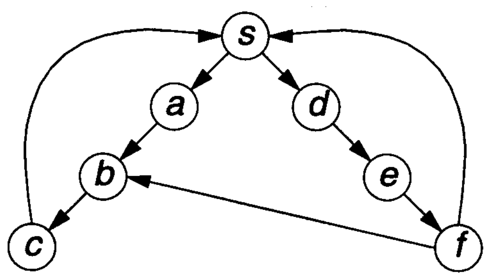

Figure 3: Two distinct hamiltonian paths in a reducible flowgraph

4 - The Hamiltonian Cycle

For a general digraph $D = (V, E)$, the problem of verifying if D has a hamiltonian cycle remains NP-complete even if a hamiltonian path is given as part of the instance [GJ79]. However, for reducible flowgraphs, the knowledge of the hamiltonian path starting at the source vertex is fundamental, as stated in the next theorem.

Theorem 1: If a reducible flowgraph $G = (V, E, s)$ has a hamiltonian cycle, this cycle is unique and consists of a hamiltonian path starting at the source vertex s and a back edge (v, s), where v is the last vertex in the path.

Proof:

Let us suppose, by contradiction, that G has two distinct hamiltonian cycles

$$C_1 = [s, v_1, \ldots, v_{|V|-1}, s] \quad \text{and} \quad C_2 = [s, w_1, \ldots, w_{|V|-1}, s].$$

By removing the last vertex (s) in both cycles, we obtain two distinct hamiltonian paths starting at s. Then, by lemma 1, G is not reducible. ∎

Theorem 1 leads us to a linear-time algorithm for finding the hamiltonian cycle of a reducible flowgraph $G = (V, E, s)$:

step 1: find the hamiltonian path of G starting at s, using the algorithm presented in the previous section;

step 2: if such a path exists, test if $(v, s) \in E$, where v is last vertex in the path.

5 - Conclusions

When dealing with paths in flowgraphs, the source vertex is often given an special treatment. This fact motivated the solution of the hamiltonian path problem for reducible flowgraphs starting at the source vertex. As it happens this path is shown to have two important properties: if it exists, it is unique and can be determined in linear time; furthermore, the knowledge of this path leads to the solution, also in linear time, of the hamiltonian cycle problem.

References

[CR88] Carrol, M. D., Ryder, B. G., Incremental Data Flow Analysis via Dominator and Attribute Updates, *Proceedings of the 15th Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp 274-284, 1988.

[GJ79] Garey, M. R., Johnson, D. S., *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.

[GM76] Gabow, H. N., Maheshwari, S. N., Osterweil, L. J., On Two Problems in the Generation of Program Test Paths, *IEEE Transactions on Software Engineering*, vol SE-2, n 3, pp 227-231, 1976.

[HU72] Hecht, M. S.., Ullman, J. D., Flow Graph Reducibility, *SIAM Journal on Computing*, vol 2, n 2, pp 188-202, 1972.

[HU74] Hecht, M. S., Ullman, J. D., Characterizations of Reducible Flow Graphs *Journal of the ACM*, vol 21, n 3, pp 367-375, 1974.

[Mc90] McHugh, J. A., *Algorithmic Graph Theory*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[NH79] Ntafos, S. C., Hakimi, S. L., On Path Cover Problems in Digraphs and Applications to Program Testing, *IEEE Transactions on Software Engineering*, vol SE-5, n 5, pp 520-529, 1979.

[NH81] Ntafos, S. C., Hakimi, S. L., On Structured Digraphs and Program Testing, *IEEE Transactions on Computers*, vol C-30, n 1, pp 67-77, 1981.

[RR94] Ramalingam, G., Reps, T., An Incremental Algorithm for Maintaining the Dominator Tree of a Reducible Flowgraph, *Conference Record of the 21st ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, pp 287-296, 1994.

[Tar83] Tarjan, R. E., *Data Structures and Network Algorithms*, SIAM, Philadelphia, 1983.

[V97] Vernet, O., Maximalidade em Grafos de Fluxo Redutíveis, *D.Sc. Thesis*, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 1997.