Graph Composition and Some Forbidden Subgraph Problems

David Fernández-Baca* *and* Mark A. Williams†
Department of Computer Science
Iowa State University
Ames, Iowa
U.S.A. 50011

Abstract. Let G be a graph with $r \geq 0$ special vertices, $b_1, \ldots, b_r$, called pins. G can be composed with another graph H by identifying each b_i with another vertex a_i of H. The resulting graph is denoted $H \circ G$. Let Π denote a decision problem on graphs. We consider the problem of constructing a "small" minor G^b of G that is "equivalent" to G with respect to the problem Π. Specifically, G^b should satisfy the following:

(C1) G^b has the same pins as G.
(C2) $\Pi(H \circ G^b) = \Pi(H \circ G)$ for every H for which $H \circ G$ is defined.
(C3) $|V(G^b)| + |E(G^b)| \leq c \cdot p$, where p is the number of pins of G, and c is a constant depending only on Π.
(C4) G^b is a minor of G.

We provide linear-time algorithms for constructing such graphs when Π stands for either series-parallelness or outer-planarity. These algorithms lead to linear-time algorithms that determine whether a hierarchical graph is series-parallel or outer-planar and to linear-space algorithms for generating a forbidden subgraph of a hierarchical graph, when one exists.

1. Introduction

Graph composition has been recently studied by several researchers [3, 6, 8, 9, 11, 12, 13]. The following definition is adapted from [9, 11, 13]. Let G denote a graph with $r \geq 0$ distinguished vertices, $b_1, \ldots, b_r$, called *pins*. The pins of G are used to *glue* G onto other graphs as follows:

Definition 1: Let H be a graph, and let $L = [(a_1, b_1), \ldots, (a_r, b_r)]$ be a list of pairs where $a_1, \ldots, a_r$ are distinct vertices (not necessarily pins) of H. L is called a *gluing list*, and H and L are said to be *compatible* with G. The graph $J = H \circ_L G$ is the result of gluing G *onto* H using L. If L is empty, J is the disjoint union of G and H. Otherwise, J is obtained by identifying a_i and b_i for $1 \leq i \leq r$. The pins of J are those of H. We will omit gluing lists in situations where no confusion can arise.

For any graph G, let Env(G), the *environment* of G, be the set of all graph and gluing list pairs (H, L) such that H and L are compatible with G. Clearly, two graphs have the same environment if and only if they have the the same pins.

*Supported in part by NSF Grant CCR-8909626
†Supported by an IBM Graduate Fellowship

Let Π denote a decision problem on graphs. In this paper, we study the problem of finding, for a given graph G, another graph G^b that satisfies the following properties:

(C1) $\text{Env}(G^b) = \text{Env}(G)$.

(C2) For each $(H, L) \in \text{Env}(G), \Pi(H \circ_L G^b) = \Pi(H \circ_L G)$.

(C3) $|V(G^b)| + |E(G^b)| \leq c \cdot p$, where p is the number of pins of G, and c is a constant that depends only on Π.

(C4) G^b is a minor of G.

Conditions Cl and C2 state that G and G^b are *replaceable* [13] with respect to the problem Π. Replaceability is an equivalence relation, and is denoted $G^b \sim_\Pi G$. Condition C3 states that, in some sense, G^b is a small replacement for G. The ability to find a small replacement for G in polynomial time has been exploited by Lengauer [8, 9] and Lengauer and Wanke [13] to obtain polynomial-time algorithms for various problems on *hierarchical graphs*, a model for succinctly representing graphs.

Condition C4 arises from our interest in problems of the form: Given a finite family $\mathcal{F}$ of *forbidden graphs*, $\Pi_\mathcal{F}(G)$ is true if and only if G contains no subgraph homeomorphic from a member of $\mathcal{F}$ (see Section 2 for definitions). Thus, if conditions Cl–C3 hold and $\Pi_\mathcal{F}(H \circ G)$ is false, then $H \circ G^b$ must contain a subgraph F^b homeomorphic from a member of $\mathcal{F}$. Furthermore, if C4 holds, then F^b must be a minor of some subgraph F of $H \circ G$.

The case where $\mathcal{F} = \{K_5, K_{3,3}\}$ (planar graphs) was studied earlier by Lengauer [9]. Lengauer showed that it is possible to construct in linear time a graph G^b satisfying properties Cl–C3, leading to a linear time algorithm for planarity testing on hierarchical graphs. However, since the graph G^b constructed by Lengauer's method will not, in general, satisfy C4, his approach is not suitable for generating forbidden subgraphs, if they exist.

In this paper, we consider two families of forbidden graphs, $\{K_4\}$ and $\{K_4, K_{2,3}\}$, which define, respectively, series-parallel and outer-planar graphs. For $\mathcal{F} = \{K_4\}$, we denote the problem $\Pi_\mathcal{F}$ by SP, and for $\mathcal{F} = \{K_4, K_{2,3}\}$, we denote $\Pi_\mathcal{F}$ by OP. Lengauer's ideas [9] can be combined with techniques by Asano [2] and Liu and Geldmacher [14] to obtain linear-time algorithms that, given G, generate a graph G^b satisfying Cl–C3 for problem SP or OP. These results lead to linear-time tests for series-parallelness and outer-planarity of hierarchical graphs. The main contribution of this paper is a proof that it is possible to construct in linear time a graph G^b that satisfies condition C4, in addition to Cl–C3, for problems SP and OP. These algorithms can be used to obtain linear-space hierarchical algorithms for generating subgraphs homeomorphic from K_4 and $K_{2,3}$, when they exist.

This paper is organized as follows. Section 2 presents the basic terminology used throughout the rest of the paper. Section 3 presents preliminary results on series-parallel and outer-planar graphs, edge contractions, and edge deletions. In Section 4, we describe how to obtain a graph G^b satisfying C1–C4 when G is biconnected. These results are then built upon in Section 5 to obtain G^b in the general case. We also describe implementations of these algorithms that operate in time linear in the size of the input graph. Section 6 summarizes our results, and discusses their application to hierarchical graphs as well as possible extensions and open problems.

2. Terminology

We use $V(G)$ and $E(G)$ to denote, respectively, the vertex and edge sets of an undirected graph G. A *loop* is an edge whose endpoints are not distinct. We allow graphs to have loops, and also allow multiple edges with the same endpoints. The latter are called *parallel* edges. An edge is *redundant* if it is parallel to another edge. A vertex $v \in V(G)$ is *isolated* if its degree is zero, a *leaf* if its degree is one, and a *series vertex* if its degree is two.

An edge $e = (u, v) \in E(G)$ is *contracted* by deleting e and identifying its endpoints to create a new vertex denoted uv. The resulting graph is denoted G/e. A *minor* of G is a graph obtained from a subgraph of G by a sequence of edge deletions and contractions. Let Π be a graph problem. Since we are interested in maintaining replaceability as edges are deleted and contracted, we introduce the following notation:

Definition 2: An edge $e \in E(G)$ is Π*-contractible* if $G/e \sim_\Pi G$, and is Π*-removable* if $G - e \sim_\Pi G$.

A graph H is *homeomorphic from* G, denoted $G \leq_h H$, if H can be obtained from G by a sequence edge subdivisions [5]. We use $G \subseteq_h H$ to denote that H contains a subgraph homeomorphic from G. H is *homeomorphic to* G if H and G are homeomorphic from some common graph.

For $u, v \in V(G)$, $G+(u, v)$ is the graph obtained by adding an edge connecting u and v, even if such an edge already exists. If, however, one or both of u and u are not vertices of G, then $G + (u, v)$ is obtained from G by adding any appropriate new vertices, and then adding an edge connecting u and v.

A vertex v of G is a *cutpoint* if $G - v$ has more connected components than G. G is said to be *biconnected* if it is connected and has no cutpoints. A *biconnected component* (*block*) of G is a maximally biconnected subgraph of G. The *biconnectivity forest* (bc-forest) [13] of G, denoted $bc(G)$, is a forest with two types of nodes: b-nodes and c-nodes. Corresponding to each b-node b is a unique block $\beta(b)$ of G, and corresponding to each c-node c is a unique cutpoint $\kappa(c)$ of G. Nodes u and v of $bc(G)$ are adjacent if and only if u is a b-node, v is a c-node, and $\kappa(v) \in V(\beta(u))$. The blocks, cutpoints, and biconnectivity forest of a graph G can be found in linear time [15].

The following definitions are from Hopcroft and Tarjan [7]. Let G be a biconnected graph and $S = \{a, b\}$ a pair of vertices of G. Partition $E(G)$ into equivalence classes $E1, E2, \ldots, E_m$, called *separation classes*, such that two edges are in the same class if and only if they lie on a common path in G containing no vertex of S except as an endpoint. If $m \geq 2$, the pair S is called a *separation pair* unless (1) $m = 2$ and one class contains a single edge, or (2) $m = 3$ and each class contains a single edge. G is *triconnected* if it is biconnected and contains no separation pairs.

Let $S = \{a, b\}$ be a separation pair of G. Split G into graphs G_1 and G_2, each having at least two edges, such that $E(G_i)$ is the union of some of the separation classes of G, $E(G_1) \cap E(G_2) = \emptyset$, $V(G_1) \cap V(G_2) = S$, and $V(G_1) \cup V(G_2) = V(G)$. Let G_i' be obtained by adding *virtual edge* (a, b) to G_i. The two virtual edges are called a *companion pair*, and G_1' and G_2' are called *split graphs* of G with respect to S. The splitting process is continued separately on G_1' and G_2' until no more splits are possible. The resulting graphs are called the *split components* of G. Each split component is a triconnected graph, a triangle, or a triple bond (two vertices connected by three parallel edges).

Two split components sharing a companion pair are *merged* by deleting the companion edges and appropriately identifying their endpoints. The *triconnected components* of G are obtained by merging all polygons (i.e., cycles) that share companion pairs, and merging all bonds that share companion pairs. Thus, the triconnected components of G consist of tri-connected graphs, polygons, and multiple bonds. Each edge of G belongs to exactly one triconnected component, and is called a *real* edge of that component. The triconnected components of G have a total of at most $3 \cdot |E(G)| - 6$ edges [7]. The *triconnectivity tree* [7] of G, denoted $tc(G)$, is a tree in which each vertex v corresponds to a unique triconnected component $\tau(v)$ of G. Two nodes u and v of $tc(G)$ are adjacent if and only if $\tau(u)$ and $\tau(v)$ share a companion pair. The triconnected components and the triconnectivity tree of G can be found in $O(|V(G)| + |E(G)|)$ time [7].

3. Preliminary Results

Here we shall prove various results to be used in subsequent sections. We first dispose of the cases where G is not series-parallel or not outer-planar.

Theorem 3.1. *If G is not series-parallel, then in linear time we can construct a graph G_b satisfying C1–C4.*

Proof: Liu and Geldmacher's $O(max(|V(G)|, |E(G)|))$ algorithm [14] can be used to test if G is series-parallel. If it is not, the same procedure will find a subgraph J of G that is homeomorphic from K_4. Now, delete from G all edges in $E(G) - E(J)$ and then delete all isolated non-pin vertices in the remaining graph. Call the resulting graph H. Finally, G^b is obtained from H by repeatedly contracting edges incident on non-pin series vertices, until no such edges exist. This process can be implemented in linear time. It is clear that G^b satisfies C1, C2, and C4. G^b also satisfies C3, since it has at most $p + 4$ vertices and $p + 6$ edges, were p is the number of pins of G. ∎

Theorem 3.2. *If G is not outer-planar, then in linear time we can construct a graph G_b satisfying C1–C4.*

Proof: Analogous to the proof of the preceding result. In this case, we may use Asano's algorithm [2] to find a subgraph of G homeomorphic from $K_{2,3}$ when G has no subgraph homeomorphic from K_4. ∎

Note that it is possible to test whether a graph is series-parallel or outer-planar in linear time (Liu and Geldmacher [14], Asano [2]). From now on, we shall concentrate on the cases where G is either series-parallel or outer-planar.

The proofs of the following two lemmas are straightforward.

Lemma 3.1. *Let $e = (u, v)$ be and edge in a series-parallel graph G. Then, e is SP-contractible if u or v is a non-pin series vertex.*

Lemma 3.2. *Let $e = (u, v)$ be and edge in a series-parallel graph G. Then, e is OP-contractible if u or v are non-pin series vertices.*

Lemma 3.3. *Let $\mathcal{F}$ be any family of graphs having no redundant edges. Then, any redundant edge e of a graph G is $\Pi_{\mathcal{F}}$-removable. In addition, $G - e$ is biconnected if G is biconnected.*

Proof: Clearly $G - e$ is biconnected if G is biconnected. Consider any $(H, L) \in \text{Env}(G)$. Since e is redundant in G, e is redundant in $H \circ_L G$. Then, since no member of $\mathcal{F}$ has a redundant edge, $\Pi_{\mathcal{F}}((H \circ_L G) - e) = \Pi_{\mathcal{F}}(H \circ_L G)$. The lemma follows since $(H \circ_L G) - e = H \circ_L (G - e)$. ∎

The next lemmas examine replaceability among connected graphs having two pins.

Lemma 3.4. *Let $\mathcal{G}$ be the set of all connected graphs with two pins u and v. The following are the $\sim_{\text{SP}}$-equivalence classes of $\mathcal{G}$:*

- (SP1) *Graphs G such that $K_4 \subseteq_h G$.*
- (SP2) *Graphs G such that $K_4 \not\subseteq_h G$, but $K_4 \subseteq_h G + (u, v)$.*
- (SP3) *Graphs G such that $K_4 \not\subseteq_h G + (u, v)$.*

Proof: Figure 1 shows representatives from each equivalence class. It is clear that SP1, SP2, and SP3 partition $\mathcal{G}$, and that graphs from different sets of the partition are not replaceable. It remains to show that any two graphs from the same set of the partition are replaceable.

Let J be any graph, and let $L = [(u', u), (v', v)]$, where $u' \neq v'$ are vertices of J. For $G \in$ SP1, $J \circ_L G$ is never series-parallel. For $G \in$ SP2, $K_4 \subseteq_h J \circ_L G$ if and only if $K_4 \subseteq_h J$, or u' and v' are connected in J. For $G \in$ SP3, $K_4 \subseteq_h J \circ_L G$ if and only if $K_4 \subseteq_h J$ or $K_4 \subseteq_h J + (u', v')$. Thus, SP1, SP2, and SP3 are the $\sim_{\text{SP}}$-equivalence classes of $\mathcal{G}$. ∎

Lemma 3.5. *Let $\mathcal{G}$ be the set of all connected graphs G having two pins u and v. Let s and t be vertices not in any $G \in \mathcal{G}$. The following are the $\sim_{OP}$-equivalence classes of $\mathcal{G}$:*

(OP1) *Graphs that are not outer-planar.*

(OP2) *Outer-planar graphs G such that $G + (u, v)$ is not outer-planar.*

(OP3) *Graphs G such that $G + (u, v)$ is outer-planar, but $G + \{(u, s), (s, v)\}$ is not outer-planar.*

(OP4) *Graphs G such that $G + (u, v)$ and $G + \{(u, s), (s, v)\}$ are outer-planar, but $G + \{(u, s), (s, v), (u, t), (t, v)\}$ is not outer-planar.*

(OP5) *Outer-planar graphs G whose only paths from u to v are single edges.*

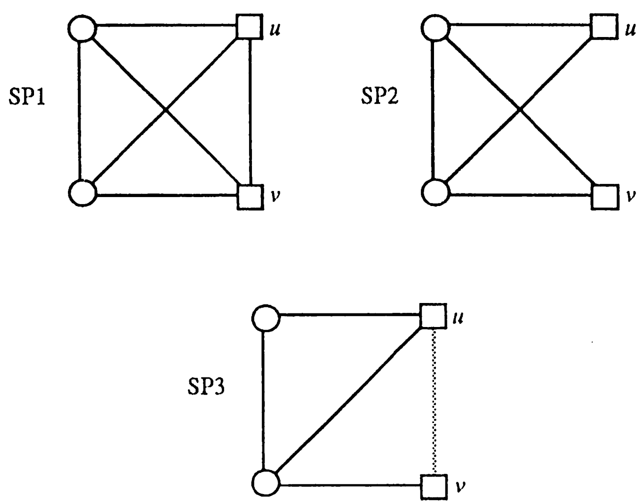

Figure 1: Equivalence classes for Lemma 3.4.
Gray lines indicate edges outside of G.

Proof: The proof is similar to that of Lemma 3.4, and is left to the reader. Figure 2 shows representatives from each equivalence class. ∎

Finally, we prove a technical lemma. Suppose G is a biconnected graph separated by $\{u, v\} \subseteq V(G)$. If G has no parallel edges with endpoints u and u, then there exist distinct vertices $x, y \in V(G) - \{u, v\}$ such that every path connecting x and y passes through either u or v. We say that $\{u, v\}$ *separates* x from y.

Lemma 3.6. *Let G be a biconnected graph with a non-redundant edge $e = (u, v)$ such that $\{u, v\}$ does not separate G. Then, G/e is biconnected. Furthermore, for* *any graph A, if G/e contains a subgraph homeomorphic from A, then so does G.*

Proof: Clearly, if G/e contains a subgraph homeomorphic from A then so does G. G/e can have no cutpoint other than the vertex uv. Since G is biconnected and e is not redundant, G/e has no loops. Let $x, y \in V(G) - \{u, v\}$. Since $\{u, v\}$ does not separate G, x and y are connected by a path in G avoiding u and v. This path avoids uv in G/e. Thus, G/e is biconnected. ∎

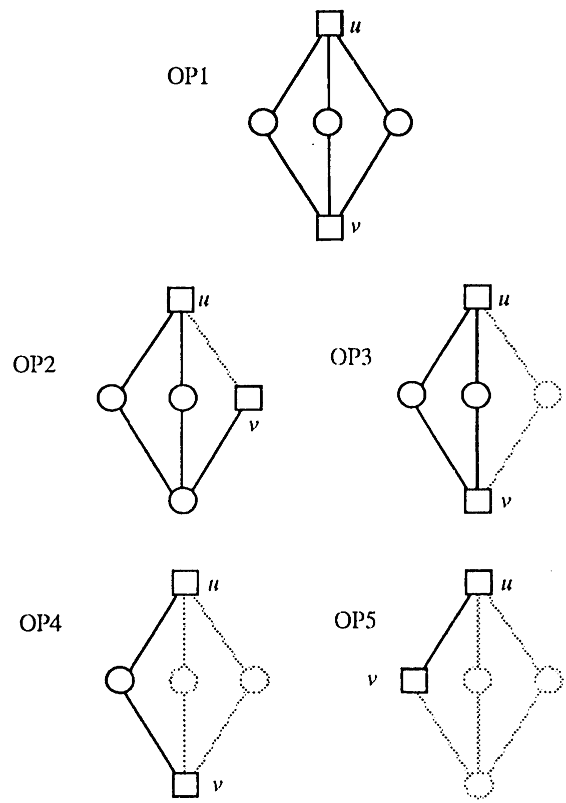

Figure 2: Equivalence classes for Lemma 3.5.
Gray lines indicate edges and vertices outside of G.

4. Contractible Edges in Biconnected Graphs

We now consider the case where the graph G is biconnected. This section is devoted to proving the following results:

Theorem 4.1. *Let G be a biconnected series-parallel graph, and let $e = (u, v) \in E(G)$ be a non-redundant edge such that u and v are non-pins and $\{u, v\}$ does not separate G. Then, e is an SP-contractible edge of G.*

Theorem 4.2. *Let G be a biconnected outer-planar graph, and let $e = (u, v) \in E(G)$ be a non-redundant edge such that u and v are non-pins and $\{u, v\}$ does not separate G. Then, e is an OP-contractible edge of G.*

We shall then use these results to obtain algorithms for problems SP and OP that reduce a biconnected graph G to a biconnected graph G^b satisfying Cl–C4.

4.1. Series-parallel graphs

We shall first prove Theorem 4.1. We need the following result:

Lemma 4.1. *Let G be a biconnected graph with a non-redundant edge $e = (u, v)$ such that $\{u, v\}$ does not separate G. For $1 \leq i \leq m$, let H_i be a connected graph with pins x_i and y_i, and assume that for any $v \in V(H_i)$, H_i has a simple path from x_i to v to y_i. Let $L_i = [(a_i, x_i), (b_i, y_i)]$ be a gluing list for gluing H_i onto G, and assume $a_i, b_i \notin \{u, v\}$. Then, for each i:*

1. *$F_i = (\cdots((G \circ_{L_1} H_1) \circ_{L_2} H_2) \circ_{L_3} \cdots \circ_{L_i} H_i)$ is biconnected and not separated by $\{u, v\}$,*
2. *$F_i' = (\cdots(((G/e) \circ_{L_1} H_1) \circ_{L_2} H_2) \circ_{L_3} \cdots \circ_{L_i} H_i$ is biconnected, and*
3. *$SP(F_i) = SP(F_i')$ if G is series-parallel.*

Proof (induction on i):
<u>$i = 1$</u>: By Lemma 3.6, G/e is biconnected. It follows from the properties of H_1 that F_1 and F_1' are biconnected. Since H_1 is connected and $\{u, v\}$ does not separate G, $\{u, v\}$ does not separate F_i. Thus, parts 1 and 2 hold.

Assume G is series-parallel. By Lemma 3.6, G/e is also series-parallel. We can rewrite F_1 as $H_1 \circ_{N_1} G$ and F_1' as $H_1 \circ_{N_1} (G/e)$, where $N_1 = [(x_1, a_1), (y_1, b_1)]$. Thus, we need only show that G and G/e belong to the same $\sim_{SP}$-equivalence classes (assuming their pins are a_1 and b_1). Since G and G/e are series-parallel, neither belongs to SP1. We show $G \in$ SP2 if and only if $G/e \in$ SP2.

Suppose $G \notin$ SP2. Then, $G \in$ SP3 and, hence, $G + (a_1, b_1)$ is series-parallel. Clearly, $G/e + (a_1, b_1)$ is also series-parallel. Thus, $G/e \notin$ SP2.

Suppose $G \in$ SP2. Then, $K_4 \subseteq_h G + (a_1, b_1)$. Since G is biconnected and series-parallel, it must contain a subgraph F as shown in Figure 3, where wavy lines represent paths of one or more edges. Suppose the path in F labeled p consists only of the edge e. Then, since $\{u, v\}$ does not separate G, there is a path in G connecting a_1 and b_1 that avoids u and v.

Thus, $K_4 \subseteq_h G$, a contradiction. Therefore $p \neq e$, and hence $K_4 \subseteq_h F/e + (a_1, b_1)$. Since $F/e \subseteq G/e$, it follows that $G/e \in$ SP2.

Therefore, G and G/e belong to the same $\sim_{SP}$-equivalence classes, and hence, $SP(F_1) = Sp(F_1')$

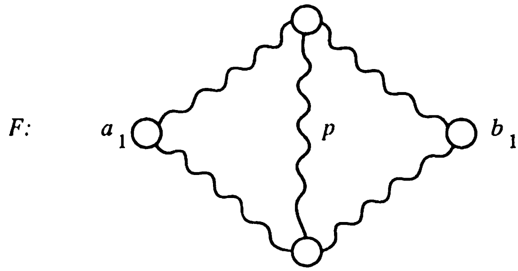

Figure 3: Example used in the proof of Lemma 4.1

<u>$i > 1$</u>: By the induction hypothesis, F_{i-1} and F'_{i-1} are biconnected and F_{i-1} is not separated by $\{u, v\}$. Then, by the same argument as used in the basis, $F_i = F_{i-1} \circ_{L_i} H_i$ and $F'_i = F'_{i-1} \circ_{L_i} H_i$ are biconnected and F_i is not separated by $\{u, v\}$.

Assume G is series-parallel. Then, by the induction hypothesis, $\text{SP}(F_{i-1}) = \text{SP}(F'_{i-1})$. If F_{i-1} and F'_{i-1} are not series-parallel, then neither are F_i and F'_i. Suppose F_{i-1} and F'_{i-1} are series-parallel. Since $a_i, b_i \notin \{u, v\}$, we can express F_i as $H_i \circ_{N_i} F_{i-1}$ and F'_i as $H_i \circ_{N_i} F'_{i-1}$, where $N_i = [(x_i, a_i), (y_i, b_i)]$. By applying the same argument as used in the basis, it follows that F_{i-1} and F'_{i-1} are in the same $\sim_{\text{SP}}$-equivalence class (assuming their pins are a_i and b_i). Thus, $\text{SP}(F_i) = \text{SP}(F'_i)$. ∎

We can now prove our main result concerning contraction and replaceability.

Proof of Theorem 4.1: Suppose G has $r > 0$ pins. We shall show that for every $(H, L) \in \text{Env}(G), \text{SP}(H \circ_L G) = \text{SP}(H \circ_L (G/e))$. Recall that u and v are the endpoints of e.

If H is not series-parallel, then neither are $H \circ G$ and $H \circ (G/e)$. Assume H is series-parallel. Suppose $r = 1$. Then, G is a block of $H \circ G$. By Lemma 3.6, G/e is biconnected and series-parallel, and hence, is a block of $H \circ (G/e)$. Since K_4 is biconnected, a graph is series-parallel if and only if each of its blocks is series parallel. Thus, $\text{SP}(H \circ G) = \text{SP}(H \circ (G/e)) = \text{TRUE}$.

Assume $r > 1$. G is a subgraph of some block A of $H \circ G$, and every other block of $H \circ G$ is a block of H. It can be easily verified that $\{u, v\}$ does not separate A. Then, by Lemma 3.6, A/e is a block of $H \circ (G/e)$ and, hence, G/e is a subgraph of A/e. Since we need only consider the blocks A and A/e, we can assume $H \circ G$ is biconnected without loss of generality.

Let R be the set of pins of G. Assume that $H \circ G$ is formed by identifying each pin x of G with a vertex x' of H. Let R' be those vertices of H to be identified with pins of G. Consider the tree T in Figure 4(a) whose leaves, x', y', and z', are the only vertices of R' in T. We consider two cases, depending on whether or not H contains a subgraph such as T.

Case 1: H contains T.

Since G is biconnected, it contains one one of the two graphs shown in Figure

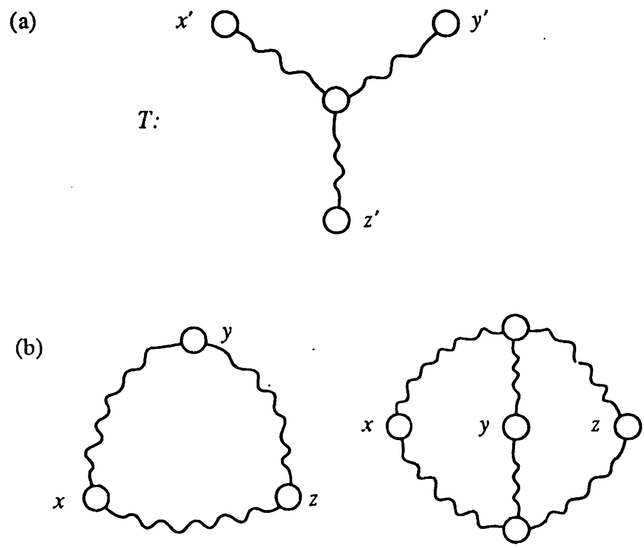

Figure 4: Examples used in the proof of Theorem 4.1

4(b). Then, $K_4 \subseteq_h H \circ G$. Since $u, v \notin R, G/e$ also contains one of these two graphs, and therefore, $K_4 \subseteq_h H \circ G/e$.

Case 2: H contains no such subgraph T.
Recall that H is assumed to be series-parallel. We show that H can be decomposed into series-parallel graphs $h_1, \ldots, h_s$ such that for each $1 \leq i \leq s$:

- (Pl) h_i contains two vertices of R', say x'_i and y'_i,
- (P2) h_i and $h_j (i \neq j)$ have at most one common vertex, which must be a vertex of R'.
- (P3) for each $v \in V(h_i)$, h_i has a simple path from x'_i to v to y'_i.

Consider any $x', y' \in R'$. Suppose H has a simple path p that connects x' and y' and contains no vertices of R' except for x' and y'. Let $h_{x',y'}$ be the subgraph of H containing x', y', and all such paths p. Together, all subgraphs $h_{x',y'}$ of this form constitute a decomposition of H satisfying Pl and P3. If $V(H) = R'$, then P2 is also satisfied. Assume $v \in V(H)$ is not in R'. Since $H \circ G$ is biconnected, there exist $x', y' \in R'$ such that $v \in V(h_{x',y'})$. However, since H does not contain the tree T, every path from v to any $z' \in R' - \{x', y'\}$ must contain x' or y'. Thus, P2 is satisfied.

Let $h_1, \ldots, h_s$ be the decomposition of H as specified. We can rewrite $H \circ G$ as $G \circ h_1 \circ \cdots \circ h_s$, where each h_i is glued onto G by its two unique vertices of R'. Then, by Lemma 4.1, it follows that SP($H \circ G$) = SP($H \circ (G/e)$). ∎

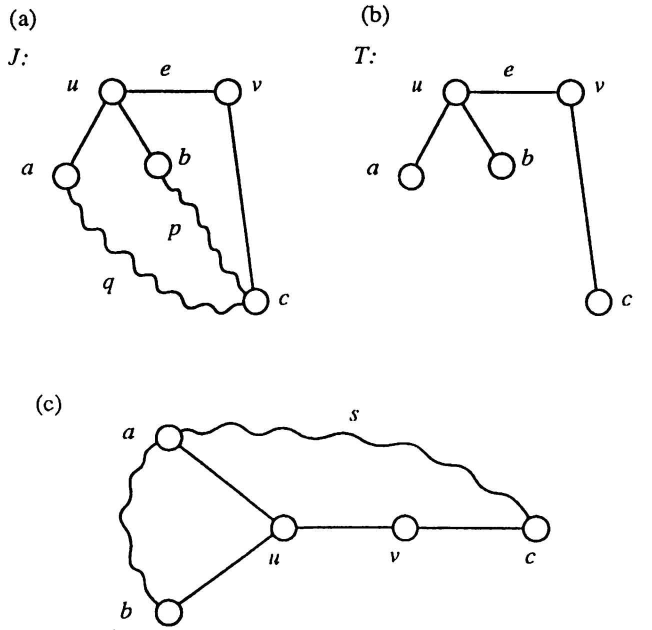

Figure 5: Examples used in the proof of Theorem 4.2

4.2. Outer-planar graphs

Proof of Theorem 4.2: Since every outer-planar graph is also series-parallel, given our results for K_4, we need only concern ourselves with $K_{2,3}$. Assume G contains $r > 0$ pins. We prove that for every $(H, L) \in \text{Env}(G)$, $\text{OP}(H \circ_L G) = \text{OP}(H \circ_L (G/e))$.

If H is not outer-planar, the result follows. Assume H is outer-planar. By an argument similar to that given in the proof of Theorem 4.1, the result holds when $r = 1$. Assume $r > 1$. If $H \circ (G/e)$ is not outer-planar, then neither is $H \circ G$. Suppose $H \circ G$ is not outer-planar, but $H \circ (G/e)$ is outer-planar. By Theorem 4.1, we may assume $K_{2,3} \subseteq_h H \circ G$ but $K_4 \not\subseteq_h H \circ G$. We shall arrive at a contradiction. Since $K_{2,3}$ is biconnected and H is outer-planar, we may assume that $H \circ G$ is biconnected without loss of generality. By Lemma 3.6, G/e is biconnected, and, hence, so is $H \circ (G/e)$.

Consider $J \subseteq H \circ G$ homeomorphic from $K_{2,3}$. We can assume such a J exists for which $e \in E(J)$, for otherwise $K_{2,3} \subseteq_h (H \circ G)/e = H \circ (G/e)$, a contradiction. J has two degree 3 vertices that we call *corners*. By assumption, $K_{2,3} \not\leq_h J/e$. Thus, an endpoint of e (assume u) is a corner of J, and J must be of the form shown in Figure 5(a). Since u and v are not pins of G, the graph T shown in Figure 5(b) must be a subgraph of G. Thus, both corners of J are vertices of G.

G is biconnected, outer-planar, and not separated by $\{u, v\}$. Although G can be biconnected in many ways, the reader can easily verify that in order for G to satisfy all three properties, it must contain a subgraph of the form shown in Figure 5(c) (symmetrically, the path s could connect b and c instead of a and c). Since s connects a and c, the path p in J cannot be entirely contained in G, for otherwise G is not outer-planar. Thus, H completes a path from b to c that avoids u, v, and a. Therefore, $K_4 \subseteq_h H \circ G$, a contradiction. ∎

4.3. Contracting a biconnected graph

We consider series-parallelness first. Let G be a biconnected graph. If G is not series-parallel, a graph G^b satisfying C1–C4 is constructed as in Theorem 3.1. Otherwise, G^b is constructed by applying the following operation:

> **SP-Reduce:** Repeat the following step until it no longer applies:
>
> If G contains a redundant edge, delete it. If G contains an edge $e = (u, v)$ such that (i) u or v is a non-pin series vertex, or (ii) e is not redundant, u and v are non-pin vertices, and $\{u, v\}$ does not separate G, then contract e.

Lemma 4.2. $G \sim_{SP} G^b$.

Proof: Follows from Lemmas 3.1, 3.3, and Theorem 4.1, since the edges deleted are SP-removable and the edges contracted are SP-contractible. ∎

Lemma 4.3. *$|V(G^b)| + |E(G^b)| \leq c \cdot r$, where r is the number of pins of G, and c is a constant independent of G.*

Proof: Let $T = tc(G^b)$. Each $v \in V(T)$ corresponds to a triconnected component $\tau(v)$ of G^b. By assumption, G^b is series-parallel. Thus, every triconnected component of G^b is either a bond or a polygon [2]. Let P be the vertices of T that correspond to polygons. Given the definition of SP-Reduce, G^b contains no redundant edges and no contractible edges. Therefore, it follows that:

1. Every leaf of T corresponds to a polygon.
2. Every non-pin of a polygon is the endpoint of a virtual edge.
3. Every real edge of a polygon has an endpoint that is a pin.
4. Every edge of G^b whose endpoints are not pins belongs to a bond.

The two vertices of each bond are repeated in each of its neighboring polygons. Thus, $|V(G^b)|$ is bounded above by the sum of $|V(\tau(v))|$ over all $v \in P$.

Let $v \in P$, and assume $\tau(v)$ has I_v virtual edges. The companion of a virtual edge in a polygon is a virtual edge in a bond, and vice-versa. Thus, since T has one edge for each companion pair, the sum of I_v over all $v \in P$ equals $|E(T)|$. Let S_v be the pins of G^b in $\tau(v)$ whose two incident edges are real. These vertices belong to no other polygons and no other bonds. Thus, the sum of $|S_v|$ over all $v \in P$ is at most r. Notice that every vertex in $V(\tau(v)) - S_v$ must be the endpoint of a virtual edge of $\tau(v)$. Thus, $|V(\tau(v))| \leq |S_v| + 2 \cdot I_v$. Therefore, $\sum_{v \in P} |V(\tau(v))| \leq r + 2 \cdot |E(T)|$.

It remains to show that T has $O(r)$ edges. Let R be the set of pins of G^b. Let v be any leaf of T. Then, since $\tau(v)$ is a polygon with only one virtual edge, it must have two real edges that share an endpoint, say x. Since no edge of G^b is incident on a non-pin series vertex, x is a pin. Furthermore, x belongs to no other polygon or bond. Let R_l consist of all such $x \in R$, and let $R_s = R - R_l$. Then, each leaf of T can be matched with a unique member of R_l.

Consider a path $q = v_1, v_2, \ldots, v_m (m > 0)$ of series vertices in T. One of v_1 and v_2 belongs to P. Let it be v_i. Since $\tau(v_i)$ has only two virtual edges, it contains a real edge, and therefore, a vertex in R_s. If $i + 4 > m$, then q has at most 5 vertices, one of which can be matched with a unique vertex in R_s. Assume $i + 4 \leq m$. Consider polygons $\tau(v_i)$, $\tau(v_{i+2})$, and $\tau(v_{i+4})$. Each contains a real edge with an endpoint in R_s. Since their real edges are distinct, at least two of their endpoints in R_s must be distinct. Thus, a unique vertex in R_s can be matched with at least every fifth vertex on q. Therefore, $|V(T)| \leq 2 \cdot |R_l| + 5 \cdot |R_s| \leq 5r$, and hence, $|E(T)| < 5r$. ∎

Next, consider outer-planarity. If G is not outer-planar, then G^b is constructed using Theorem 3.2. Otherwise, construct G^b by using the following operation:

> **OP Reduce:** Repeat the following step until it no longer applies:
>
> If G contains a redundant edge, delete it. If G contains and edge $e = (u, v)$ such that u and v are not pins, and either (i) u and v are series vertices, or (ii) e is not redundant and $\{u, v\}$ does not separate G, then contract e.

Lemma 4.4. $G \sim_{OP} G^b$.

Proof: Similar to the proof of Lemma 4.2 using Lemmas 3.2, 3.3, and Theorem 4.2. ∎

Lemma 4.5. $|V(G^b)| + |E(G^b)| \leq c \cdot r$, *where* r *is the number of pins of* G *and* c *is a constant independent of* G.

Proof: If $r = 1$, G^b has at most two vertices. Assume $r > 1$. Then, G^b is biconnected, outer-planar, and contains no OP-removable or OP-contractible edges. Let H be the result of applying SP-Reduce to G. Every vertex of G^b that is not a vertex of H is a non-pin series vertex adjacent to two distinct pins. Let $S = V(G^b) - V(H)$, and let R be the set of pins of G^b. We show that $|S| \leq |R| = r$. On the contrary, suppose $|S| > r$.

Let J be the bipartite subgraph of G^b that has bipartition R and S and contains as many edges as possible. J has at least $2r + 1$ vertices. Since each vertex of S is adjacent to two vertices of R, J has at least $2(r + 1)$ edges. Therefore, J contains a cycle C. Since J is bipartite, C has an even number of vertices, and since G^b has no redundant edges, $|V(C)| \geq 4$.

Suppose C contains all vertices of R. Since we assume $|S| > r$, there is some $v \in S - V(C)$ adjacent to distinct pins x and y of C. But, since x and y are not adjacent on C, $J \subseteq G^b$ is homeomorphic from $K_{2,3}$, a contradiction.

Suppose R has a vertex v not on C. Since G^b is biconnected, there are at least two paths that connect v to C that are vertex disjoint except for v. These paths must also end on pins of C since the non-pins in C are series vertices of G^b. Thus, G^b is not outer-planar, a contradiction.

Therefore, G^b has at most r more vertices than H, which, by Lemma 4.3, has $O(r)$ vertices. ∎

5. Contracting Non-Biconnected Graphs

We are now ready to tackle the general case where G may not be biconnected. All graphs are assumed to be either series-parallel or outer-planar, depending on the problem at hand.

We often find it convenient to rewrite a composition of graphs in the following fashion. Let A be a block of a graph G. The *coupling* vertices of A are its vertices that are pins or cutpoints of G. Notice that for any $(H, L) \in \mathrm{Env}(G)$, the coupling vertices of A attach it to the rest of $H \circ_L G$. Thus, $H \circ_L G$ can be rewritten as $J \circ A$, where J is the result of splitting off A from $H \circ_L G$, and A is glued onto J via its coupling vertices. In this situation, the coupling vertices of A can be viewed as its pins.

Definition 3: A block A of G is *useless* if A has no coupling vertices, or its only coupling vertex is a cutpoint of G.

Lemma 5.1. *Let $\mathcal{F}$ be a family of biconnected graphs, none of which is a single edge, and let G be a graph such that $\Pi_{\mathcal{F}}(G)$ is true (i.e., no subgraph of G is homeomorphic to a member of $\mathcal{F}$). Let G' denote the minor of G obtained by first contracting all edges in useless blocks of G and then removing all isolated non-pin vertices. Then, $G' \sim_{\Pi_{\mathcal{F}}} G$.*

Proof: Trivial. ∎

The preceding result enables us to restrict our analysis to graphs with no useless blocks. We shall consider series-parallelness and outer-planarity separately. In both cases, we use the following definition:

Definition 4: A block-path of G is a sequence $p = A_1, A_2, \ldots, A_m$ of blocks of G such that (1) each A_i has exactly two cutpoints c_{i-1} and c_i of G, and (2) $V(A_i) \cap V(A_{i+1}) = \{C_i\}$ for each $i < m$. The block-path p is *compressible* if $m > 1$ and the only pins contained in $A_1, A_2, \ldots, A_m$, if any, are c_0 and c_m.

5.1. Series-parallel graphs

Let $p = A_1, \ldots, A_m$ be a compressible block-path of G. Consider the following operation:

SP-Compress: Find an i such that $A_i \in$ SP2, where c_{i-1} and C_i are considered the pins of A_i. If no such i exists, let $i = 1$. For each $j \neq i$, contract all edges in the block A_j.

Lemma 5.2. *Let G be series-parallel and let G' be the result of applying SP-Compress to a compressible block-path $p = A_1, \ldots, A_m$ of G. Then, $G' \sim_{SP} G$.*

Proof: Let A_i be block chosen by SP-Compress, and let G_p be the subgraph of G represented by p. G_p is attached to the rest of G by the cutpoints c_0 and C_m. G' has a block A isomorphic to A_i that is attached to the rest of G' by c_0 and C_m. The graphs $G - V(G_p)$ and $G' - V(A)$ are identical. Thus, we can view SP-Compress as the substitution of A for G_p in G. Consider any $(H, L) \in \text{Env}(G)$. $H \circ_L G$ and $H \circ_L G'$ can be rewritten, respectively, as $J \circ_N G_p$ and $J \circ_N A$, where $N = [(u, c_0), (v, c_m)]$ for some distinct $u, v \in V(J)$. Thus, we need only show that G_p and A belong to the same $\sim_{SP}$-equivalence class, assuming their pins are c_0 and c_m.

Since G is series-parallel, $G_p \notin$ SP1. $G_p \in$ SP2 if and only if $A_j \in$ SP2 for some $1 \leq j \leq m$. By the choice of the block A_i, it follows that G_p and A both belong to either SP2 or SP3. ∎

In addition to contracting useless blocks and compressing block-paths, we also apply SP-Reduce to blocks of G. However, when we apply SP- Reduce to a block A of G, the coupling vertices of A are considered to be its pins.

Lemma 5.3. *Let G be a series-parallel graph and let A be any block of G. If G' denotes the result of applying SP-Reduce to A, then $G' \sim_{SP} G$.*

Proof: Let $(H, L) \in \text{Env}(G)$. We show that SP($H \circ_L G$) = SP($H \circ_L G'$). Let A' be the block of G' corresponding to A. Since A' and A have the same coupling vertices, $H \circ_L G$ and $H \circ_L G'$ can be written, respectively, as $J \circ A$ and $J \circ A'$, where A and A' are glued onto J via their coupling vertices. Thus, we need only show that $A' \sim_{SP} A$. This follows immediately from Lemma 4.2. ∎

We can now show how to construct G^b from G. This process is implemented by a function B_{SP}, called the *series-parallel burner*, shown in Figure 6. The result of applying B_{SP} to G is denoted by $B_{SP}(G)$

Theorem 5.1. *$G^b = B_{SP}(G)$ satisfies conditions Cl–C4.*

Proof: That G^b satisfies Cl, C2, and C4 follows from Theorem 3.1 and Lemmas 5.1, 5.2, and 5.3. It remains to show that G^b satisfies C3. Let p denote the number of pins of G. If G is not series-parallel, then by Theorem 3.1, G^b satisfies C3. Assume G is series-parallel. Since G^b satisfies Cl and C2, G^b must also be series-parallel, and hence $|E(G^b)| = O(|V(G^b)|)$. Thus, we need only show that $|V(G^b)|$ is $O(p)$.

Let F be the bc-forest of G^b. An upper bound on $|V(G^b)|$ is the sum of $|V(\beta(x))|$ over all b-nodes x in F. By Lemma 4.3, $|V(\beta(x))| = O(s_x + t_x)$, where s_x is the number of pins in $\beta(x)$ that are not cutpoints of G^b, and t is the number of vertices of $\beta(x)$ that are cutpoints of G^b. The number of times that a cutpoint of G^b is repeated in blocks of G^b is exactly the degree in F of its corresponding c-node. The sum of the degrees of the c-nodes in F is equal to $|E(F)|$. Thus, $|V(G^b)| = O(p + |E(F)|)$. Every leaf of F corresponds to a block of G^b containing a pin that is not a cutpoint of G^b, and at least every fourth node on any path of series vertices in F corresponds to a block that contains a pin or to a cutpoint that is a pin. Thus, $|E(F)|$ is $O(p)$ and, hence, so is $|V(G^b)|$. Therefore, G^b satisfies C3. ∎

```
function B_SP (G: graph):graph
begin
    G^b := G;
    if G^b has a subgraph J homeomorphic from K_4 then
        Delete from G^b every edge not in J;
        Delete from G^b every isolated non-pin vertex;
        while G^b has an edge e incident on a non-pin series vertex do
            G^b := G^b/e
    else
        Contract all useless blocks from G^b;
        Apply SP-Compress to every maximal-length
        compressible block-path of G^b;
        Apply SP-Reduce to every block of G^b
    end;
    return G^b
end
```

Figure 6: The series-parallel burner.

5.2. Outer-planar graphs

The procedure to construct G^b is virtually identical to that used for problem SP. The case in which G is not outer-planar is handled using Theorem 3.2. Assume G is outer-planar. First, all useless blocks are contracted. Then, the following operation is applied to each maximal-length compressible path $p = A_1, \ldots, A_m$ of G (assuming the pins of each A_i are the cutpoints c_{i-1} and c_i):

> **OP-Compress:** Find an i such that $A_i \in$ OP2. If none exists, find an i such that $A_i \in$ OP3. If none exists, find an i such that $A_i \in$ OP4. If none exists, then for each $j \geq 3$, contract all edges of block A_j. Otherwise, for each $j \neq i$, contract all edges in block A_j.

Finally, OP-Reduce is applied to every block of G (again, assuming the pins of a block are its coupling vertices). The result is the graph G^b.

Theorem 5.2. *G^b satisfies C1–C4.*

Proof: Regarding condition C3, the proof is almost identical to that of Theorem 5.1 (using Lemma 4.5). Condition C4 holds by construction. The proof of replaceability follows along the lines of Lemmas 5.2 and 5.3 using Lemma 5.1. ∎

5.3. Implementation

Let G be any graph. We shall show that a graph G^b satisfying C1–C4 can be constructed in linear time for both the SP and OP problems. We limit our discussion to series-parallelness since the procedure for outer-planarity is entirely analogous.

By Theorem 3.1, we need only consider the case where G is series-parallel. Depth-first search [15] can be used to compress all useless blocks of G. Depth-first search can also be used locate maximal length compressible block-paths. The time required to compress a block-path is linear in the size of the corresponding subgraph of G. Thus, the time required to compress all compressible block-paths is linear in the size of G.

Applied to a block A of G, SP-Reduce deletes SP-removable edges and contracts SP-contractible edges until no such edges exist. Let T be the triconnectivity tree of A. Since A is series-parallel, it has at most $2 \cdot |V(A)| - 3$ edges [2]. Thus, T has size $O(|V(A)|)$ and can be found in $O(|V(A)|)$ time. We show that a single post-order traversal of T is sufficient to implement SP-Reduce.

Every redundant edge of A belongs to a bond corresponding to leaf of T, and every contractible edge belongs to a polygon. Let v be the vertex of T currently being visited. We consider whether or not $\tau(v)$ is a bond or a polygon.

Suppose $\tau(v)$ is a bond. If v is an interior vertex of T, then $\tau(v)$ is not changed. If v is a leaf, we delete all but one of the real edges of $\tau(v)$, and then merge it with its neighboring polygon.

Suppose $\tau(v)$ is a polygon. We traverse its edges in cyclic order, contracting a real edge e if either (1) the next edge f in $\tau(v)$ is real and e and f share an endpoint that is not a pin, or (2) both endpoints of e are not pins. If v is a leaf or a series node of T (i.e. $\tau(v)$ has at most two virtual edges), then these contractions may reduce $\tau(v)$ to a bond. Assume that is the case. If v is a leaf, it is handled as any other leaf whose triconnected component is a bond. If v is a series node, we merge $\tau(v)$ and its two neighboring bonds, and delete any redundant real edges of the resulting bond.

Therefore, SP-Reduce has an implementation that operates on $O(|V(A)|)$ time and, hence, G^b can be constructed in linear time. Thus, we have:

Theorem 5.3. *For both series-parallelness and outer-planarity, a graph G^b satisfying conditions C1–C4 can be constructed in linear time from G.*

6. Discussion

We have presented linear-time algorithms that, given a graph G containing pins, find small replacements that preserve series-parallelness or outer-planarity. Our algorithms rely on the existence of rules that allow us to determine which edges of G can be contracted (deleted) so that for any $(H, L) \in \mathrm{Env}(G)$, $\Pi(H \circ_L G) = \Pi(H \circ_L (G/e))(\Pi(H \circ_L G) = \Pi(H \circ_L (G-e)))$, where Π is SP or OP.

We now briefly discuss the application of our results to *hierarchical graphs* [11,13]. A hierarchical graph $\Gamma = (G_1, \ldots, G_n)$ is a finite list of simple graphs (i.e., graphs having no loops and no parallel edges) called *cells*. For each i, $V(G_i)$ is partitioned into *pins*, *terminals*, and *nonterminals*. G_i has p_i pins whose names are the integers $1, \ldots, p_i$. Each nonterminal of G_i has a *type*, which is a symbol in the set $\{G_1, \ldots, G_{i-1}\}$. A nonterminal of type G_j has degree p_j, and each incident edge is labeled with the name of a unique pin of G_j. Edges between nonterminals are not permitted. Γ is the description of a potentially much larger graph $X(\Gamma)$, referred to as the *expansion* of Γ. $X(\Gamma)$ is defined recursively as follows. For $1 \leq i \leq n$, let $\Gamma_i = (G_1, \ldots, G_i)$. Γ_i is a hierarchical graph, whose expansion $X(\Gamma_i)$ is constructed according to the following rules:

- If G_i has no nonterminals, then $X(\Gamma_i) = G_i$.
- If G_i has nonterminals, then $X(\Gamma_i)$ is the graph that results from doing the following operation for each nonterminal v of G_i:

  Suppose v is of type G_j. Let $v(1), \ldots, v(p_j)$ be the vertices of G_i adjacent to v where, for each m, edge $(v, v(m))$ has label m. Then, set G_i equal to the result of the composition $(G_i - v) \circ_L X(\Gamma_j)$, where $L = [(v(1), 1), \ldots, (v(p_j), p_j)]$.

$X(\Gamma)$ is the same as $X(\Gamma_n)$. The attractive feature of hierarchical graphs is that the graph $X(\Gamma)$ can be exponentially larger than its description Γ thus enabling hierarchical graphs to model certain facets of VLSI design [12]. The results of the previous sections can be used together with Lengauer's techniques to provide linear-time algorithms to test the series-parallelness or outer-planarity of the expansions of hierarchical graphs (note that the running times are linear in Γ, not in $X(\Gamma)$). We omit the details of these algorithms, since they are quite similar to those presented in [9]. Given that the replacement graphs that our algorithms produce satisfy condition (C4), we are also able to provide linear-space algorithms to generate forbidden subgraphs, if they exist.

Series-parallel and outer-planar graphs are *partial 2-trees* (Arnborg et al. [1]). A problem of interest is to determine the contractible / removable edges of k-trees for $k \geq 3$. Another open problem is to determine which edges are contractible/removable for planar graphs, and whether such edges can be found efficiently. Lengauer's hierarchical planarity test [9] determines in linear time whether the expansion of a hierarchical graph is planar, but does not rely solely on the construction of minors.

Acknowledgement

The first author thanks Anand Medepalli for his assistance with the figures.

References

1. Arnborg, S., A. Proskurowski, and D. G. Corneil, *Complexity of finding embeddings in a k-tree*, SIAM J. Alg. Disc. Methods **8** (1987), 277–284.
2. Asano, T., *An approach to the subgraph homeomorphism problem*, Theoretical Computer Science **38** (1985), 249–267.
3. Bern, M. W., E. L. Lawler, and A. L. Wong, *Linear-time computation of optimal subgraphs of decomposable graphs*, Journal of Algorithms **8** (1987), 216–235.
4. Dufffn, R. J., *Topology of series-parallel networks*, Journal of Mathematical Analysis and Applications **10** (1965), 303–318.
5. F. Harary, *Graph theory*, Reading, MA: Addison Wesley.
6. Hare E., S. Hedetniemi, R. Laskar, K. Peters, and T. Wimer, *Linear-time computablility of combinatorial problems on generalized-series-parallel graphs*, in "Discrete Algorithms and Complexity", Academic Press, Boston, 1987, pp. 437–457.
7. Hopcroft, J., and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM Journal on Computing **2** (1973), 135–158.
8. Lengauer, T., *Efficient algorithms for finding minimum spanning forests of hierarchically defined graphs*, Journal of Algorithms **8** (1987), 260–284.
9. Lengauer, T., *Hierarchical planarity testing algorithms*, Journal of the ACM **32** (1989), 474–509.
10. Lengauer, T., and K. Wagner, *The correlation between the complexities of the nonhierarchical and hierarchical versions of graph properties*, in "Proceedings of STACS 87, Springer LNCS 247", Springer-Verlag, Berlin, 1987.
11. Lengauer, T., and E. Wanke, *Efficient analysis of graph properties on context-free graph languages*, in "Proceedings of ICALP 88, Springer LNCS 317", Springer-Verlag, Berlin, 1988.
12. Lengauer, T., and E. Wanke, *Efficient processing of hierarchical graphs for engineering design*, EATCS Bulletin **35** (1988), 143–157.
13. Lengauer, T., and E. Wanke, *Efficient solution of connectivity problems on hierarchically defined graphs*, SIAM Journal on Computing **17** (1989), 1063–1080.
14. Liu, P. C., and R. C. Geldmacher, *An O(max(m, n)) algorithm for finding a subgraph homeomorphic to* K_4, Congressus Numerantium **29** (1980), 597–609.
15. Tarjan, R. E., *Depth-first search and linear graph algorithms*, SIAM Journal on Computing **1** (1972), 146–160.
16. Williams, M. A., *Efficient processing of hierarchical graphs*, Ph.D. Dissertation, Iowa State University, (1990).