

# On the Inference of Non-Confluent NLC Graph Grammars

Robert Brijder<sup>1\*</sup> and Hendrik Blockeel<sup>2,3</sup>

<sup>1</sup> Hasselt University and Transnational University of Limburg,  
Belgium, [robert.brijder@uhasselt.be](mailto:robert.brijder@uhasselt.be)

<sup>2</sup> Leiden Institute of Advanced Computer Science,  
Universiteit Leiden, The Netherlands

<sup>3</sup> Department of Computer Science, Katholieke Universiteit Leuven,  
Belgium, [hendrik.blockeel@cs.kuleuven.be](mailto:hendrik.blockeel@cs.kuleuven.be)

**Abstract.** Grammar inference deals with determining (preferably simple) models/grammars consistent with a set of observations. There is a large body of research on grammar inference within the theory of formal languages. However, there is surprisingly little known on grammar inference for graph grammars. In this paper we take a further step in this direction and work within the framework of node label controlled (NLC) graph grammars. Specifically, given a graph  $G$  and a set  $\mathcal{S}$  of disjoint and isomorphic subgraphs of  $G$ , we characterize whether or not there is an graph grammar consisting of one production such that  $G$  may be derived from  $G_0$ , the graph obtained from  $G$  by “contraction” of each subgraph in  $\mathcal{S}$  to a node labelled by  $N$ . This generalizes a previous result which assumes boundary NLC graph grammars, and leads one to consider the more involved “non-confluent” graph grammar rules.

**Keywords:** Graph grammars, Grammar inference

## 1 Introduction

Grammar inference, also called grammar induction, is a general line of research where one is concerned with determining a “simple” grammar that is consistent with a given set of possible and impossible outcomes. Hence, one “goes back” in the derivation: instead of determining the generative power of a grammar, one determines the grammar given the generated output. This topic is well-studied for formal languages, especially with respect to regular languages, see e.g. [8, 6], however, relatively little is known for graph grammars.

The topic of inference of graph grammars is considered in [7] and uses their so-called Subdue scheme developed in [3]. Moreover, in [2] a rigorous approach of grammar inference within the framework of node label controlled (NLC) graph

---

\* This research is performed while at Leiden Institute of Advanced Computer Science, Universiteit Leiden, The Netherlands.

grammars [4, 5], a natural and well-studied class of graph grammars, is initiated. In this paper, we build on the work in [2].

Given a graph  $G$  and a set  $\mathcal{S}$  of disjoint and isomorphic subgraphs of  $G$ , we characterize whether or not there is a graph grammar consisting of one production (we extend this in Section 8 to multiple productions) such that  $G$  may be derived from the graph  $G_0$ , where  $G_0$  is the graph obtained from  $G$  by “contraction” of each subgraph  $S_i \in \mathcal{S}$  to a node  $v_i$  labelled by  $N$ . In [2] such a characterization is given assuming a boundary NLC (BNLC) grammar: any two nodes labelled by  $N$  in  $G_0$  are required to be non-adjacent. In this way, we generalize the result in [2]. Such a generalization requires one to deal with a number of issues. Most notably, one has to deal with non-confluency issues: the generated graph depends on the order in which subgraphs are generated.

An extended abstract containing selected results without proofs was presented at CiE’09 [1].

## 2 Notation and Terminology

We consider (simple) graphs  $G = (V, E)$ , where  $V$  is a finite set of nodes and  $E \subseteq \{\{x, y\} \mid x, y \in V, x \neq y\}$  is the set of edges — hence no loops or parallel edges are allowed. We denote  $V(G) = V$  and  $E(G) = E$ . For  $S \subseteq V$ , the induced subgraph of  $G$  is the graph  $(S, E')$  where  $E' \subseteq E$  and for each  $e \in E$  we have  $e \in E'$  iff  $e \subseteq S$ . We consider only induced subgraphs, and therefore we often write just “subgraph” instead of induced subgraph. The neighborhood of  $S \subseteq V$  in  $G$ , denoted by  $N_G(S)$ , is  $\{v \in V \setminus S \mid \{s, v\} \in E \text{ for some } s \in S\}$ . If  $S = \{x\}$  is a singleton, then we also write  $N_G(x) = N_G(S)$ . A labelled graph is a triple  $G = (V, E, l)$  where  $(V, E)$  is a graph and  $l : V \rightarrow L$  is a node labelling function, where  $L$  is a finite set of labels. As usual, graphs are considered isomorphic if they are identical modulo the identity of the nodes. It is important to realize that for labelled graphs, nodes identified by an isomorphism have identical labels. In graphical depictions of labelled graphs we always represent the nodes by their labels.

Subgraphs  $S_1$  and  $S_2$  are called *disjoint* if  $V(S_1)$  and  $V(S_2)$  are disjoint. They are called *touching* if they are disjoint and there is an edge  $e \in E(G)$  with one node in  $S_1$  and the other in  $S_2$ .

Define, as usual, for  $W_1, W_2 \subseteq V$ ,  $W_1 \times W_2 = \{(x_1, x_2) \mid x_1 \in W_1, x_2 \in W_2\}$ . Define  $u((x_1, x_2))$  to be  $\{x_1, x_2\}$ , and define  $\pi_i((x_1, x_2)) = x_i$  for  $i \in \{1, 2\}$ . Often, for a function  $f : X \rightarrow Y$  we write  $f(D) = \{f(x) \mid x \in D\}$  for  $D \subseteq X$ . Also, if  $x_1, x_2 \in X$ , we write  $f((x_1, x_2)) = (f(x_1), f(x_2))$ .

## 3 NLC Graph Grammars

Typically, a graph grammar transforms a graph  $G$  by replacing an (induced) subgraph  $H$  by another graph  $H'$  where  $H'$  is embedded in the remaining part  $G \setminus H$  of the original graph in a way prescribed by a so-called graph grammar embedding relation. The node label controlled (NLC) graph grammars are the

simplest class of these grammars, where  $H$  is a single node. Note that for the grammars the exact identities of the nodes are not important as multiple copies of  $H'$  may be inserted. Hence, we consider labelled graphs where the embedding relation is defined w.r.t. node labels instead of nodes. In this section we recall informally the notions and definitions concerning NLC grammars used in this paper, and refer to [4] for a gentle and more detailed introduction to these grammars.

A NLC graph grammar is a system  $Q$  consisting of a set of node labels  $L$ , an *embedding relation*  $E \subseteq L^2$ , and a set of *productions*  $P$  where a production is of the form  $N \rightarrow S$  where  $N \in L$  and  $S$  is a (labelled) graph. For clarity of exposition, we focus in this paper first on the case  $|P| = 1$ , and subsequently generalize it to the general case in Section 8. Thus, assuming from now on  $|P| = 1$ ,  $Q$  can be denoted as a *rule*  $r = N \rightarrow S/E$  (if  $L$  is understood from the context of considerations). Given a graph  $G$ ,  $r$  can be applied to any node  $v$  labelled by  $N$ . The result of applying  $r$  to  $v$  in  $G$  is the graph  $G'$  obtained from  $G$  by (1) removing  $v$  along with the edges adjacent to  $v$ , (2) adding an isomorphic copy  $S'$  of  $S$ , and (3) adding an edge  $e = \{x, y\}$  iff  $x \in V(S)$ ,  $y \in N_G(v)$  and  $(l(x), l(y)) \in E$  (recall that  $l$  is the labelling function). We write  $(G, v) \rightarrow_r (G', S')$  (note that  $S'$  is a subgraph of  $G'$ ). To avoid confusion with embedding relations, the set of edges of a graph  $G$  are written in the remainder as  $E(G)$  and not as  $E$ .

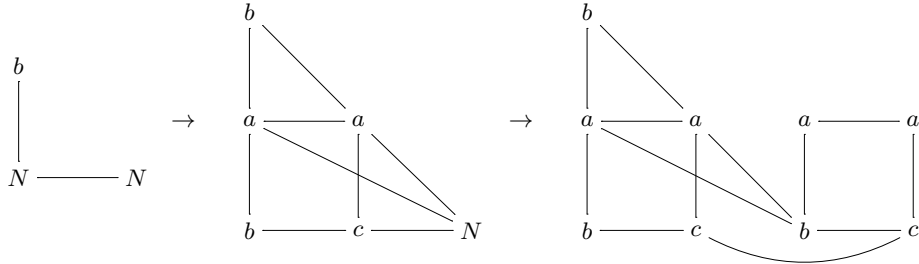
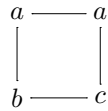


Fig. 1: The derivation of graph  $G'$  (right-hand side) from  $G$  (left-hand side) in Example 1.

*Example 1.* We assume that the set of labels  $L$  is  $\{a, b, c, N\}$ . Let  $G$  be the graph on the left-hand side of Figure 1. Consider the grammar rule  $r = N \rightarrow S/E'$ , where  $S$  is the graph



and  $E' = \{(a, b), (b, a), (c, c), (a, N), (c, N)\}$ . (Note that formally we have only defined  $S$  up to isomorphism, however, as the behaviour of graph grammars

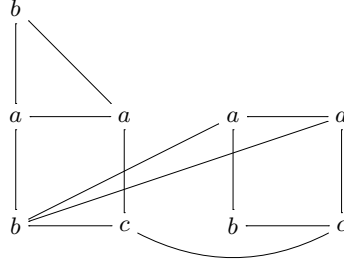


Fig. 2: Graph  $G''$  obtainable from  $G$  in Example 1.

is independent on the exact identity of the nodes of the graph, this is not an objection.) Then Figure 1 depicts one possible derivation from  $G$  to a graph  $G'$  (on the right-hand side of the figure) for which no rule is applicable anymore. Note that there is one other possible derivation to a “terminal” graph (i.e., a graph without nodes labelled by  $N$ ). This terminal graph  $G''$ , different from  $G'$ , is given in Figure 2. To obtain  $G''$  we choose first the node labelled by  $N$  on the right-hand side (the one not connected to the node labelled by  $b$ ) in  $G$  in the derivation. This example will be our running example of this paper.

In [2] the inference of NLC grammars with exactly one rule  $r = N \rightarrow S/E$  are studied where moreover  $S$  does not contain a node labelled by  $N$  and  $E$  does not contain a tuple containing  $N$ . This is sufficient for the case where the subgraphs isomorphic to  $S$  are disjoint and non-touching. To consider the case where the subgraphs are only disjoint, we allow  $E$  to contain tuples containing  $N$ . However, we do require that  $S$  remains without nodes labelled by  $N$ . Therefore there is no “real” recursion: no nodes labelled by  $N$  can be introduced in any derivation.

#### 4 Known results: Non-touching graphs

In this section we recall some notions and a result from [2] which we will need in subsequent sections. First we define in this context the notion of compatibility.

**Definition 2.** Let  $G$  be a graph and  $S$  be a subgraph of  $G$ . We say that  $E \subseteq L \times L$  is *compatible* with  $S$  (in  $G$ ) if there is a graph  $F$ , a node  $v$  of  $F$ , and a NLC grammar rule  $r = N \rightarrow S'/E$  such that  $(F, v) \rightarrow_r (G, S)$ .

Note that  $S'$  is necessarily isomorphic to  $S$  in Definition 2.

*Example 3.* Reconsider our running example. Hence we again let  $G'$  be the graph at the right-hand side of Figure 1. Moreover we let  $S_1$  and  $S_2$  be the subgraphs of  $G'$  of isomorphic to  $S$  where  $S_1$  is the one connected to the node labelled by  $b$  and  $S_2$  is the other one. Note that  $S_1$  and  $S_2$  are disjoint and touching in  $G'$ . We have that, e.g.,  $E_1 = \{(b, a), (c, c)\}$ ,  $E_2 = \{(b, a), (c, c), (a, b)\}$  or  $E_3 = \{(b, a), (c, c), (c, b), (b, b)\}$  is compatible with  $S_2$  in  $G'$ . The middle graph

of the figure is a graph  $F$  such that an application of the NLC grammar rule  $N \rightarrow S/E$  to  $F$  “creates”  $S_2$  and obtains graph  $G'$ .

To characterize the notion of compatibility, the notions of *inset* and *outset* for arbitrary  $Q \subseteq V^2$  (where  $V$  is the set of nodes of  $G$ ) are crucial.

**Definition 4.** Let  $Q \subseteq V^2$ . We define the *inset* of  $Q$ , denoted by  $I_Q$ , as the set  $\{(l(x), l(y)) \mid \{x, y\} \in E(G), (x, y) \in Q\}$ , and *outset* of  $Q$ , denoted by  $O_Q$ , as the set  $\{(l(x), l(y)) \mid \{x, y\} \notin E(G), (x, y) \in Q\}$ .

*Remark 5.* Note that the inset and outset are precisely the sets  $l(In_Q)$  and  $l(Out_Q)$  respectively, where  $\{In_Q, Out_Q\}$  is the partition of  $Q$  where, for  $q \in Q$ ,  $q \in In_Q$  iff  $u(q) \in E(G)$ .  $\square$

Let  $S$  be a subgraph of  $G$ . Then the *inset* (*outset*, resp.) of  $S$ , denoted by  $I_S$  ( $O_S$ , resp.), is defined to be the inset (outset, resp.) of  $Q = V(S) \times N_G(V(S))$ .

The following lemma from [2, Section 4.1] characterizes compatibility for a single graph  $S$  in terms of the inset and outset of  $S$ : the inset are tuples that *should* be in  $E$ , while the outset are tuples that *should not* be in  $E$ .

**Lemma 6.** Let  $S$  be a subgraph of  $G$ , and let  $E \subseteq L \times L$ . Then  $E$  is compatible with  $S$  iff  $I_S \subseteq E \subseteq L^2 \setminus O_S$  (i.e.,  $E$  separates  $I_S$  from  $O_S$ ).

Hence, there is an  $E$  compatible with  $S$  in  $G$  iff  $I_S \cap O_S = \emptyset$ .

*Example 7.* Reconsider again our running example. Then  $I_{S_2} = \{(b, a), (c, c)\}$  and  $O_{S_2} = \{(a, a), (a, c), (c, a), (b, c)\}$  (w.r.t.  $G'$ ). Since  $I_{S_2} \cap O_{S_2} = \emptyset$ , there is an  $E$  compatible with  $S_2$  in  $G'$ . We have that  $I_{S_2} \subseteq E \subseteq L^2 \setminus O_{S_2}$  holds for, e.g.,  $E_1$ ,  $E_2$  and  $E_3$  in Example 3.

We consider now sequences of subgraphs to be generated by a single graph rule. Note that these graphs must necessarily be mutually isomorphic.

**Definition 8.** Let  $G$  be a graph and  $S_1, S_2, \dots, S_n$  be subgraphs of  $G$  isomorphic to  $S$ . We say that  $E \subseteq L \times L$  is *compatible* with  $(S_1, S_2, \dots, S_n)$  (in  $G$ ) if there are graphs  $G_0, \dots, G_n$  such that  $G_n = G$  and for each  $i \in \{1, \dots, n\}$ ,  $G_i$  we have  $(G_{i-1}, v) \rightarrow_r (G_i, S_i)$  for some node  $v$  of  $G_{i-1}$  and NLC grammar rule  $r = N \rightarrow S/E$ .

Note that, in general, the order of the elements  $(S_1, S_2, \dots, S_n)$  is important. E.g. a given  $E$  may be compatible with  $(S_1, S_2)$  while it is incompatible with  $(S_2, S_1)$  (we will see such an example in the next section). However, for a set of mutually non-touching and isomorphic subgraphs  $S_i$  for  $i \in \{1, \dots, n\}$  of  $G$ , the order of the elements is not important. Thus,  $E \subseteq L \times L$  being compatible with  $C = (S_1, S_2, \dots, S_n)$  implies that  $E$  is compatible with any permutation of  $C$ . In fact we have that,  $E \subseteq L \times L$  is compatible with  $S_1$ , with  $S_2, \dots$ , and with  $S_n$  iff it is compatible with  $C$  (or any permutation of  $C$ ). Therefore, as noted in [2], in this case Lemma 6 can be trivially generalized as follows.

**Lemma 9.** Let  $C = (S_1, S_2, \dots, S_n)$  be a sequence of mutually disjoint, non-touching, and isomorphic subgraphs of a graph  $G$ . Then  $E \subseteq L \times L$  is compatible with  $C$  iff  $\bigcup_i I_{S_i} \subseteq E \subseteq L^2 \setminus (\bigcup_i O_{S_i})$ .

## 5 Two touching graphs

In this section and the next we consider the case where a single NLC grammar rule  $N \rightarrow S/E$  generates disjoint subgraphs which can (possibly) touch each other. As a consequence, we generalize Lemma 9 by removing the non-touching condition. To this aim we allow the non-terminal  $N$  to be present in tuples of the embedding relation  $E$  of the NLC grammar rule  $N \rightarrow S/E$ . This introduces the issue of non-confluency: the order in which non-terminals are replaced by subgraphs influences the obtained graph. Example 1 illustrates this as the different graphs  $G'$  and  $G''$  can both be obtained from the original graph  $G$ .

As we will see the inset and outset between the nodes of two touching graphs turns out to be crucial. We will now define the inset and outset of two subgraphs in general.

**Definition 10.** Let  $S_1$  and  $S_2$  be subgraphs of  $G$ . For  $Q_1 = V(S_2) \times (V(S_1) \cap N_G(S_2))$ , we denote  $I_{Q_1}$  and  $O_{Q_1}$  by  $I_{(S_1, S_2)}$  and  $O_{(S_1, S_2)}$ , respectively. Moreover, for  $Q_2 = V(S_2) \times V(S_1)$ , we denote  $I_{Q_2}$  and  $O_{Q_2}$  by  $I_{((S_1, S_2))}$  and  $O_{((S_1, S_2))}$ , respectively.

Notice that these in- and outsets, e.g.  $I_{(S_1, S_2)}$ , are concerned with the tuples going *from*  $S_2$  *to*  $S_1$ . This is because these tuples are important in the second step in the derivation to  $G$  which first creates  $S_1$  (first step) followed by the creation of  $S_2$  (second step).

We now state some basic properties of the insets and outsets of Definition 10. Note first that  $I_{(S_1, S_2)} = I_{((S_1, S_2))}$ . In fact, it is equal to the inset of

$$(V(S_2) \cap N_G(S_1)) \times (V(S_1) \cap N_G(S_2)).$$

Also note that, for node labels  $x$  and  $y$ , we have  $(x, y) \in I_{((S_1, S_2))}$  iff  $(y, x) \in I_{((S_2, S_1))}$ . This holds similarly for  $O_{((S_1, S_2))}$ , however, this does not hold in general for  $O_{(S_1, S_2)}$ . Moreover note that  $O_{(S_1, S_2)} \subseteq O_{((S_1, S_2))}$  (as they are the outsets of  $Q_1$  and  $Q_2$  resp., where  $Q_1 \subseteq Q_2$ ), and

$$O_{((S_1, S_2))} \setminus O_{(S_1, S_2)} = l(V(S_2) \times (V(S_1) \setminus N_G(S_2))).$$

Finally note that  $\pi_2(I_{(S_1, S_2)}) = l(V(S_1) \cap N_G(S_2))$ . We will use these basic properties frequently in the remainder of this paper.

*Example 11.* In our running example, we have  $I_{(S_1, S_2)} = \{(b, a), (c, c)\}$ ,  $O_{(S_1, S_2)} = \{(a, a), (a, c), (b, c), (c, a)\}$ , and  $O_{((S_1, S_2))} = L'^2 \setminus I_{(S_1, S_2)}$  with  $L' = \{a, b, c\}$ . Moreover, we have  $I_{(S_2, S_1)} = \{(a, b), (c, c)\}$ ,  $O_{(S_2, S_1)} = \{(a, c), (b, b), (b, c), (c, b)\}$ , and  $O_{((S_2, S_1))} = L'^2 \setminus I_{(S_2, S_1)}$ .

We now adapt the definition of inset and outset for a graph  $S$ , by incorporating the issues related to touching graphs.

**Definition 12.** Let  $S_1, \dots, S_n$  be subgraphs of  $G$ , let  $T = \bigcup_{j \in \{1, \dots, n\}} V(S_j)$ , and let  $Q = \bigcup_{i \in \{1, \dots, n\}} (V(S_i) \times (N_G(V(S_i)) \setminus T))$ . We denote  $I_Q$  and  $O_Q$  by  $I_{[S_1, \dots, S_n]}$  and  $O_{[S_1, \dots, S_n]}$ , respectively.

Note that  $I_{[S_1, S_2]} = I_{[S_2, S_1]}$  and if  $S_1$  and  $S_2$  are disjoint and non-touching, we have  $I_{[S_1, S_2]} = I_{S_1} \cup I_{S_2}$ .

*Example 13.* In our running example, we have  $I_{[S_1, S_2]} = \{(a, b)\}$ , and  $O_{[S_1, S_2]} = \{(b, b), (c, b)\}$ .

Definitions 10 and 12 are to separate three types of insets and outlets. Roughly speaking, the two types of insets and outlets of Definition 10 deal with the tuples between  $S_1$  and  $S_2$ , while the type of inset and outlet of Definition 12 deals with the tuples from  $S_1$  to the “outside world” (the nodes in the neighborhood of  $S_1$  which do not belong to  $S_2$ ) *plus* the tuples from  $S_2$  to the “outside world” (the nodes in the neighborhood of  $S_2$  which do not belong to  $S_1$ ).

We now characterize the embedding relations  $E$  such that  $E$  is compatible with  $(S_1, S_2)$  where  $S_1$  and  $S_2$  are isomorphic and touching subgraphs of  $G$ .

**Lemma 14.** *Let  $S_1$  and  $S_2$  be isomorphic and touching subgraphs of  $G$ . Then  $E \subseteq L \times L$  is compatible with  $(S_1, S_2)$  iff the following conditions hold:*

1.  $I_{(S_1, S_2)} \subseteq E$ ,
2.  $\{(x, N) \mid x \in \pi_2(I_{(S_1, S_2)})\} \subseteq E$ ,
3. If  $e \in O_{((S_1, S_2))}$ , then  $(\pi_2(e), N) \notin E$  or  $e \notin E$  (or both), and
4.  $I_{[S_1, S_2]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, S_2]})$ .

Moreover, if this is the case, then we have  $E \cap O_{(S_1, S_2)} = \emptyset$ .

*Proof.* In the case where there are no edges between  $S_1$  and  $S_2$ , we have, by Lemma 6, that  $E \subseteq L \times L$  is compatible with  $(S_1, S_2)$  iff  $I_{S_1} \cup I_{S_2} \subseteq E \subseteq L^2 \setminus (O_{S_1} \cup O_{S_2})$  — this is equivalent to condition (4) (since  $S_1$  and  $S_2$  are non-touching).

Now, since edges between  $S_1$  and  $S_2$  introduce *additional* constraints on  $E$  (i.e., not less constraints), we may consider only the graph  $F$  equal to an edge having both nodes labelled by  $N$ , and check the necessary and sufficient (additional) constraints on  $E$  to transform the graph in two steps where  $S_1$  appears first and then  $S_2$  such that the edges between  $S_1$  and  $S_2$  are identical to those between  $S_1$  and  $S_2$  in  $G$ .

Now, let  $x$  be a node of  $S_1$  labelled by  $b$ , and  $y$  be a node of  $S_2$  labelled by  $a$ . Assume first that  $x$  is connected to  $y$  in  $G$ . Now, if we apply the NLC rule to create  $S_1$ , then  $x$  should be connected to the node labelled by  $N$  — thus we need  $(b, N) \in E$ . Indeed, without this rule  $x$  will not be connected to any node of  $S_2$  (after applying the NLC rule to create  $S_2$ ). Now, if we subsequently apply the NLC rule to create  $S_2$ , then  $y$  should be connected to  $x$  and hence we need  $(a, b) \in E$ . Hence  $(a, b) \in E$  and  $(b, N) \in E$  results in an edge between  $x$  and  $y$ . Conversely, if  $(b, N) \notin E$  or  $(a, b) \notin E$ , then  $x$  is not connected to  $y$ . Consequently, both  $(a, b) \in E$  and  $(b, N) \in E$  iff there is an edge between every node labelled by  $b$  in  $S_1$  and every node labelled by  $a$  in  $S_2$  (equivalently, we may replace “every node” by “a node”).

Thus,  $E$  is compatible with  $(S_1, S_2)$  iff 1)  $I_{((S_1, S_2))} \cup \{(x, N) \mid x \in \pi_2(I_{((S_1, S_2))})\} \subseteq E$  and 2) both  $(a, b) \in E$  and  $(b, N) \in E$  implies  $(a, b) \notin O_{((S_1, S_2))}$ . Recall that  $I_{((S_1, S_2))} = I_{(S_1, S_2)}$ , and the iff statement holds.

Finally, we have in this case  $E \cap O_{(S_1, S_2)} = \emptyset$ . Indeed, if  $e \in E \cap O_{(S_1, S_2)}$ , then  $e \in O_{((S_1, S_2))}$  and  $e \in E$  and therefore, by condition (3),  $(\pi_2(e), N) \notin E$ . Now,  $\pi_2(O_{(S_1, S_2)}) \subseteq \pi_2(I_{(S_1, S_2)}) = l(V(S_1) \cap N_G(S_2))$  cf. Definition 10. Consequently, by condition (2),  $(\pi_2(e), N) \in E$  — a contradiction.  $\square$

Note that Lemma 14 does *not* hold when  $S_1$  and  $S_2$  are disjoint non-touching subgraphs of  $G$ . Indeed, in this case,  $E$  is compatible with  $(S_1, S_2)$  iff  $I_{[S_1, S_2]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, S_2]})$  (i.e., condition (4) holds). Moreover, condition (1) and (2) hold trivially as  $I_{(S_1, S_2)} = \emptyset$ . However, condition (3) may not hold in this case.

Intuitively, condition (4) of Lemma 14 deals with the edges from both  $S_1$  and  $S_2$  to the “outside world”, while conditions (1) to (3) deal with the edges between  $S_1$  and  $S_2$ . For the latter, conditions (1) and (2) state the tuples that must necessarily be in  $E$ , while condition (3) states requirements on which tuples must not (together) be in  $E$ .

Since  $E \cap O_{(S_1, S_2)} = \emptyset$  by Lemma 14, we may modify conditions (1) and (3) of the previous lemma as follows:

- 1'.  $I_{(S_1, S_2)} \subseteq E \subseteq L^2 \setminus O_{(S_1, S_2)}$ ,
- 3'. If  $e \in O_{((S_1, S_2))} \setminus O_{(S_1, S_2)} = l(V(S_2) \times (V(S_1) \setminus N_G(S_2)))$ , then  $(\pi_2(e), N) \notin E$  or  $e \notin E$  (or both).

However, in this way the condition  $E \cap O_{(S_1, S_2)} = \emptyset$  is explicitly assumed and not part of the result as stated in the lemma.

*Remark 15.* Note that by condition (4) of the lemma, we may go even further and instead state “ $e \in l(V(S_2) \times (V(S_1) \setminus N_G(S_2))) \setminus O_{[S_1, S_2]}$ ” in condition (3').  $\square$

Also note that we have, for  $e \in I_{[S_1, S_2]} \cup I_{(S_1, S_2)}$  (and hence  $e \subseteq E$ ),  $e \in O_{((S_1, S_2))}$  implies  $(\pi_2(e), N) \notin E$ .

*Example 16.* We continue our running example. As we have seen, an  $E \subseteq L \times L$  compatible with  $(S_1, S_2)$  in  $G'$  allows, given the graph  $G$  on the left-hand side of Figure 1, for the generation of the middle graph (in the figure) and subsequently the generation of  $G'$ . We now determine, using Lemma 14 and the modified conditions below the lemma, the constraints on  $E$  for it to be compatible with  $(S_1, S_2)$ .

Recall that  $I_{(S_1, S_2)} = \{(b, a), (c, c)\}$ ,  $O_{(S_1, S_2)} = \{(a, a), (a, c), (b, c), (c, a)\}$ ,  $I_{[S_1, S_2]} = \{(a, b)\}$ , and  $O_{[S_1, S_2]} = \{(b, b), (c, b)\}$ . Moreover, we have  $\pi_2(I_{(S_1, S_2)}) = l(V(S_1) \cap N_G(S_2)) = \{a, c\}$  and thus  $\{(x, N) \mid x \in \pi_2(I_{(S_1, S_2)})\} = \{(a, N), (c, N)\}$ . Hence, by conditions (1'), (2), and (4) of Lemma 14 we have  $\{(a, b), (b, a), (c, c), (a, N), (c, N)\} \subseteq E$  and  $E \cap \{(a, a), (a, c), (b, b), (b, c), (c, a), (c, b)\} = \emptyset$ . Now,  $O_{((S_1, S_2))} \setminus O_{(S_1, S_2)} = l(V(S_2) \times (V(S_1) \setminus N_G(S_2))) = \{a, b, c\} \times \{b\} = \{(a, b),$



$(b, b), (c, b)$ . Hence by condition (3')  $(b, N) \notin E$  or  $(a, b) \notin E$ . The latter is a contradiction, hence  $(b, N) \notin E$ . Consequently,

$$E = \{(a, b), (b, a), (c, c), (a, N), (c, N)\}$$

is compatible with  $(S_1, S_2)$ , in fact, in this case,  $E$  is unique with the property of being compatible with  $(S_1, S_2)$  in  $G'$ . Note that since  $E$  is unique, the intermediate graph in Figure 1 in the derivation of  $G'$  from  $G$  is also unique. Note that  $E$  is *not* compatible with  $(S_2, S_1)$  in  $G'$ . Finally note that e.g.  $E_2 = E \cup \{(b, N)\}$  is incompatible with  $(S_1, S_2)$  — the generated graph would then have edges between the node labelled  $b$  in  $S_1$  and the two nodes labelled  $a$  in  $S_2$ .

Using Lemma 14, the *existence* of an embedding relation  $E$  is elegantly characterized, as shown in the next lemma.

**Lemma 17.** *Let  $S_1$  and  $S_2$  be isomorphic and touching graphs. There is an  $E \subseteq L \times L$  compatible with  $(S_1, S_2)$  iff*

$$\begin{aligned} (I_{[S_1, S_2]} \cup I_{(S_1, S_2)}) \cap O_{[S_1, S_2]} &= \emptyset, \\ \pi_2(I_{(S_1, S_2)}) \cap \pi_2(I_{[S_1, S_2]} \cap O_{((S_1, S_2))}) &= \emptyset, \text{ and} \\ I_{(S_1, S_2)} \cap O_{((S_1, S_2))} &= \emptyset. \end{aligned}$$

Moreover, if this is the case, then  $(I_{[S_1, S_2]} \cup I_{(S_1, S_2)}) \cap O_{(S_1, S_2)} = \emptyset$ .

*Proof.* Assume first that that right-hand side of the iff statement of the lemma holds. Then take  $E' = I_{[S_1, S_2]} \cup I_{(S_1, S_2)}$ , take  $F' = \pi_2(I_{(S_1, S_2)})$ , and let  $E = E' \cup \{(x, N) \mid x \in F'\}$ . Now, conditions (1), (2), and (4) of Lemma 14 hold trivially. Finally to prove condition (3), we need to show that  $e \in O_{((S_1, S_2))} \cap E$  implies  $(\pi_2(e), N) \notin E$ . Let  $e \in O_{((S_1, S_2))} \cap E$ . We have, by definition of  $E$ , that  $e \in I_{[S_1, S_2]}$  or  $e \in I_{(S_1, S_2)}$ . The latter is a contradiction of  $I_{(S_1, S_2)} \cap O_{((S_1, S_2))} = \emptyset$ . The former implies, by the second equality of this lemma, that  $\pi_2(e) \notin \pi_2(I_{(S_1, S_2)}) = F'$ . Consequently,  $(\pi_2(e), N) \notin E$ .

Now, we prove the reverse implication. If there is such compatible  $E$ , then, by Lemma 14,  $(I_{[S_1, S_2]} \cup I_{(S_1, S_2)}) \cap O_{[S_1, S_2]} = \emptyset$ . Assume  $I_{(S_1, S_2)} \cap O_{((S_1, S_2))} \neq \emptyset$ , and let  $e \in I_{(S_1, S_2)} \cap O_{((S_1, S_2))}$ . Since  $e \in I_{(S_1, S_2)}$ , we have, by condition (1) in Lemma 14,  $e \in E$ , and we have by condition (2)  $(\pi_2(e), N) \in E$ . Now since  $e \in O_{((S_1, S_2))}$  we have a contradiction by condition (3). Finally, assume  $\pi_2(I_{(S_1, S_2)}) \cap \pi_2(I_{[S_1, S_2]} \cap O_{((S_1, S_2))}) \neq \emptyset$  and let  $x \in \pi_2(I_{(S_1, S_2)}) \cap \pi_2(I_{[S_1, S_2]} \cap O_{((S_1, S_2))})$ . Then, by condition (2),  $(x, N) \in E$ , and by condition (3),  $(x, N) \notin E$  — a contradiction.

By Lemma 14, we have in this case  $(I_{[S_1, S_2]} \cup I_{(S_1, S_2)}) \cap O_{(S_1, S_2)} = \emptyset$ , since  $I_{[S_1, S_2]} \cup I_{(S_1, S_2)} \subseteq E$  and  $E \cap O_{(S_1, S_2)} = \emptyset$  for any compatible  $E$ .  $\square$

Recall that  $I_{(S_1, S_2)} = I_{((S_1, S_2))}$ , hence the third equality of Lemma 17 may be rephrased more symmetrically as “ $I_{((S_1, S_2))} \cap O_{((S_1, S_2))} = \emptyset$ ”. Notice that the case  $N_G(V(S_1) \cup V(S_2)) = \emptyset$  (roughly) corresponds to the situation where the original graph  $F$  that generates  $G$  has a connected component equal to an

edge where both nodes are labelled by  $N$ . In this case, by Lemma 17, there is an  $E \subseteq L \times L$  compatible with  $(S_1, S_2)$  iff  $I_{((S_1, S_2))} \cap O_{((S_1, S_2))} = \emptyset$  (since  $I_{[S_1, S_2]} = O_{[S_1, S_2]} = \emptyset$ ).

*Example 18.* We continue Example 16 (our running example). Recall that  $I_{[S_1, S_2]} \cup I_{(S_1, S_2)} = \{(a, b), (b, a), (c, c)\}$  and  $O_{[S_1, S_2]} = \{(b, b), (c, b)\}$  — hence these sets are disjoint. Also,  $\pi_2(I_{(S_1, S_2)}) = \{a, c\}$  and  $\pi_2(I_{[S_1, S_2]} \cap O_{((S_1, S_2))}) = \pi_2(\{(a, b)\}) = \{b\}$ , and therefore these sets are also disjoint. Finally,  $I_{(S_1, S_2)} \cap O_{((S_1, S_2))} = \emptyset$ . Consequently, by Lemma 17, there is an  $E$  compatible with  $(S_1, S_2)$  — such an  $E$  is given in Example 16.

Finally note that by Lemma 17 one requires precisely the sets  $I_{[S_1, S_2]}$ ,  $O_{[S_1, S_2]}$ ,  $I_{(S_1, S_2)}$ , and  $O_{((S_1, S_2))}$  to determine the existence of a compatible  $E$ .

## 6 Set of touching graphs

Let  $\mathcal{S}$  be a set of mutually isomorphic and disjoint subgraphs of  $G$ . In this section we turn to the question of whether or not there is an  $E \subseteq L \times L$  and a linear ordering  $C = (S_1, S_2, \dots, S_n)$  of  $\mathcal{S}$  such that  $E$  is an embedding relation compatible with  $C$ .

The following result is easily obtained from Lemma 14.

**Lemma 19.** *Let  $G$  be a graph,  $E \subseteq L \times L$ , and  $C = (S_1, \dots, S_n)$  be a linear ordering of  $\mathcal{S}$ . Then  $E$  is compatible with  $C$  iff (1)  $I_{[S_1, \dots, S_n]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, \dots, S_n]})$  and (2) for each two touching  $S_i$  and  $S_j$  with  $i < j$ , we have that the first three conditions of Lemma 14 hold w.r.t.  $E$  and  $(S_i, S_j)$ .*

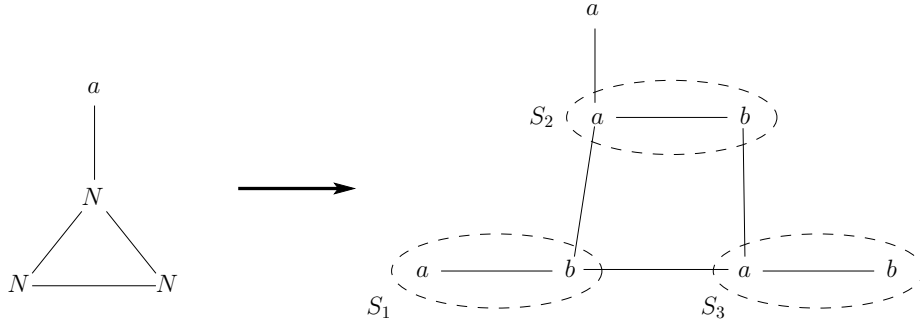


Fig. 3: Graphs  $G$  (left-hand side) and  $G'$  (right-hand side) of Example 20.

*Example 20.* Let  $G$  be the graph on the left-hand side of Figure 3. Moreover, let  $S$  be the graph consisting of precisely one edge  $e$  where the endpoints of  $e$  are nodes labelled by  $a$  and  $b$ , and let  $G'$  be the graph on the right-hand side

of Figure 3 where we have identified three occurrences  $S_1$ ,  $S_2$ , and  $S_3$  of  $S$  in  $G'$ . Let  $\mathcal{S} = \{S_1, S_2, S_3\}$ . The set of labels  $L$  is  $\{a, b, N\}$ . In this new running example, we examine the (possible) derivation of  $G'$  from  $G$  through the rule  $r = N \rightarrow S$ .

Let  $C$  be the ordering  $(S_1, S_2, S_3)$  of  $\mathcal{S}$ . By Lemma 19,  $E \subseteq L \times L$  is compatible with  $C$  iff  $I_{[S_1, S_2, S_3]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, S_2, S_3]})$  and, for  $1 \leq i < j \leq 3$ , the first three conditions of Lemma 14 hold w.r.t.  $E$  and  $(S_i, S_j)$ . We have  $I_{[S_1, S_2, S_3]} = \{(a, a)\}$ ,  $O_{[S_1, S_2, S_3]} = \{(b, a)\}$ , and  $I_{(S_1, S_2)} = I_{(S_2, S_3)} = I_{(S_1, S_3)} = \{(a, b)\}$ . Therefore, assuming  $E$  is compatible with  $C$ , we have (1)  $\{(a, a), (a, b), (b, N)\} \subseteq E$  and (2)  $(b, a) \notin E$ . Moreover, we have  $O_{((S_1, S_2))} = O_{((S_2, S_3))} = O_{((S_1, S_3))} = \{(a, a), (b, a), (b, b)\}$ . Hence, (3)  $(a, N) \notin E$  or  $(a, a) \notin E$ , and (4)  $(b, N) \notin E$  or  $(b, a), (b, b) \notin E$ . Conversely, when these four conditions hold,  $E \subseteq L \times L$  is compatible with  $C$  (by Lemma 19). Consequently, precisely when both  $\{(a, a), (a, b), (b, N)\} \subseteq E$  and  $(b, a), (b, b), (a, N) \notin E$ , we have that  $E$  is compatible with  $C$ . Note that, in this case,  $E$  is unique with this property.

As we have seen in Section 4, the compatibility of  $E$  for the case where the elements of  $\mathcal{S}$  are mutually non-touching is much less involved: it does not depend on the ordering  $C$  of  $\mathcal{S}$  — it only depends on  $\mathcal{S}$ . For touching graphs, the situation is different as the conditions in Lemma 14 are not symmetric: e.g.  $I_{(S_i, S_j)}$  and  $I_{(S_j, S_i)}$  generally differ. Hence, we must choose a linear ordering in a “compatible” way. First, we focus on the question whether or not there exists an  $E$  compatible with a *given* linear ordering  $C$  of  $\mathcal{S}$ .

We characterize the existence by generalizing Lemma 17 for the case where more than two graphs can (possibly) touch each other.

To this aim consider the following graph that represents, w.r.t. an ordering  $C$  of  $\mathcal{S}$ , whether or not subgraphs  $S_i$  and  $S_j$  in  $\mathcal{S}$  touch.

**Definition 21.** Let  $G$  be a graph and let  $C = (S_1, S_2, \dots, S_n)$  be an ordering of  $\mathcal{S}$ . The *touching graph* of  $G$  w.r.t.  $C$ , is the directed graph  $(\mathcal{S}, \{(S_i, S_j) \mid S_i \text{ and } S_j \text{ touch, and } i < j\})$ .

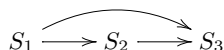


Fig. 4: Touching graph  $D$  of  $G'$  w.r.t.  $C = (S_1, S_2, S_3)$  of Example 20.

*Example 22.* We continue Example 20 (our new running example). Let again  $C$  be the ordering  $(S_1, S_2, S_3)$  of  $\mathcal{S}$ . The touching graph  $D$  of  $G'$  w.r.t.  $C$  is given in Figure 4.

For  $e = (S_i, S_j) \in E(D)$  we write  $O_e = O_{(S_i, S_j)}$  and  $O_{(e)} = O_{((S_i, S_j))}$  (and similarly for the insets  $I_e$  and  $I_{(e)}$ ).

Roughly speaking, by Lemma 19, the requirement for embedding relation  $E$  to be compatible with  $C = (S_1, S_2, \dots, S_n)$  is the “union” of the requirements

of  $E$  to be compatible with any  $e = (S_i, S_j)$  where  $S_i$  and  $S_j$  are touching and  $i < j$ . Therefore the generalization of Lemma 17 for arbitrary  $C$  is obtained in a straightforward way by replacing e.g. the inset  $I_{(S_1, S_2)}$  and outset  $O_{(S_1, S_2)}$  in Lemma 17 by the union of the insets  $I_e$  and outsides  $O_e$  for these  $e$ . However some care must be taken for  $I_{[S_1, \dots, S_n]}$  and  $O_{[S_1, \dots, S_n]}$ , and therefore we prove this result directly instead of relying on Lemma 17.

**Theorem 23.** *Let  $G$  be a graph, let  $C = (S_1, S_2, \dots, S_n)$  be an ordering of  $\mathcal{S}$ , and let  $D$  be the touching graph of  $G$  w.r.t.  $C$ . There is an  $E \subseteq L \times L$  compatible with  $C$  iff*

$$\begin{aligned} (I_{[S_1, \dots, S_n]} \cup (\bigcup_{e \in E(D)} I_e)) \cap O_{[S_1, \dots, S_n]} &= \emptyset, & (1) \\ \pi_2(\bigcup_{e \in E(D)} I_e) \cap \pi_2(I_{[S_1, \dots, S_n]} \cap (\bigcup_{e \in E(D)} O_e)) &= \emptyset, \text{ and} & (2) \\ (\bigcup_{e \in E(D)} I_e) \cap (\bigcup_{e \in E(D)} O_e) &= \emptyset. & (3) \end{aligned}$$

Moreover, if this is the case, then

$$(I_{[S_1, \dots, S_n]} \cup (\bigcup_{e \in E(D)} I_e)) \cap (\bigcup_{e \in E(D)} O_e) = \emptyset. \quad (4)$$

*Proof.* This proof is in the same spirit as the proof of Lemma 17.

Assume first that that right-hand side of the iff statement holds. Then take  $E' = I_{[S_1, \dots, S_n]} \cup (\bigcup_{e \in E(D)} I_e)$ , and take  $F' = \pi_2(\bigcup_{e \in E(D)} I_e)$ . Now, let  $E = E' \cup \{(x, N) \mid x \in F'\}$ . Hence, by the first assumed equality, we have  $I_{[S_1, \dots, S_n]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, \dots, S_n]})$ . Thus by Lemma 19 it suffices to show that for each two touching  $S_i$  and  $S_j$  with  $i < j$ , the first three conditions of Lemma 14 hold w.r.t.  $E$  and  $r = (S_i, S_j)$ . Now, conditions (1) and (2) of Lemma 14 hold trivially. Finally to prove condition (3) of Lemma 14, we show that  $e \in O_{(r)} \cap E$  implies  $(\pi_2(e), N) \notin E$ . Let  $e \in O_{(r)} \cap E$ . We have, by definition of  $E$ , that  $e \in I_{[S_1, \dots, S_n]}$  or  $e \in I_{(S_{k_1}, S_{k_2})}$  for some  $k_1$  and  $k_2$ . The latter is a contradiction of  $I_{(S_{k_3}, S_{k_4})} \cap O_{(r)} = \emptyset$  (the third assumed equality). The former implies, by the second assumed equality, that  $\pi_2(e) \notin \pi_2(\bigcup_{e \in E(D)} I_e) = F'$ . Consequently,  $(\pi_2(e), N) \notin E$ .

Now, we prove the reverse implication. Assume that there is an  $E \subseteq L \times L$  compatible with  $C$ . Then by Lemma 19, (1)  $I_{[S_1, \dots, S_n]} \subseteq E \subseteq L^2 \setminus (O_{[S_1, \dots, S_n]})$  and (2) for each two touching  $S_i$  and  $S_j$  with  $i < j$ , we have that the first three conditions of Lemma 14 hold w.r.t.  $E$  and  $(S_i, S_j)$ . Hence, by Lemma 14,  $(I_{[S_1, \dots, S_n]} \cup (\bigcup_{e \in E(D)} I_e)) \cap O_{[S_1, \dots, S_n]} = \emptyset$ . Assume now that  $I_{f_1} \cap O_{(f_2)} \neq \emptyset$  for some  $f_1, f_2 \in E(D)$ , and let  $e \in I_{f_1} \cap O_{(f_2)}$ . Since  $e \in I_{f_1}$ , we have, by condition (1) in Lemma 14,  $e \in E$ , and we have by condition (2)  $(\pi_2(e), N) \in E$ . Now since  $e \in O_{(f_2)}$  we have a contradiction by condition (3). Finally, assume  $\pi_2(I_{f_1}) \cap \pi_2(I_{[S_1, \dots, S_n]} \cap O_{(f_2)}) \neq \emptyset$  and let  $x \in \pi_2(I_{f_1}) \cap \pi_2(I_{[S_1, \dots, S_n]} \cap O_{(f_2)})$ . Then, by condition (2),  $(x, N) \in E$ , and by condition (3),  $(x, N) \notin E$  — a contradiction.

Finally, we have  $E \cap O_e = \emptyset$  for all  $e \in E(D)$ , by the same reasoning as in the proof of Lemma 14, and we have both  $I_{[S_1, \dots, S_n]} \subseteq E$  and  $I_e \subseteq E$  for all  $e \in E(D)$  by Lemma 19. Consequently,  $(I_{[S_1, \dots, S_n]} \cup (\bigcup_{e \in E(D)} I_e)) \cap (\bigcup_{e \in E(D)} O_e) = \emptyset$ .  $\square$

Note that, in Theorem 23,  $\bigcup_{e \in E(D)} I_e$  equals  $\bigcup_{i < j} I_{(S_i, S_j)}$  as  $I_{(S_i, S_j)} = \emptyset$  whenever  $S_i$  and  $S_j$  are non-touching. However, note that we may *not* replace, e.g.,  $\bigcup_{e \in E(D)} O_{(e)}$  by  $\bigcup_{i < j} O_{(S_i, S_j)}$ .

Hence, by Theorem 23, for a given set  $\mathcal{S}$  of disjoint subgraph *and* an ordering  $C$  of  $\mathcal{S}$ , it is computationally efficient to determine whether or not a compatible embedding relation  $E$  exists. Indeed, we ‘only’ need to determine the sets  $I_{[S_1, \dots, S_n]}$ ,  $O_{[S_1, \dots, S_n]}$ ,  $\bigcup_{e \in E(D)} I_e$ , and  $\bigcup_{e \in E(D)} O_{(e)}$ , where  $D$  is the touching graph of  $G$  w.r.t.  $C$ .

*Example 24.* We continue the running Example 20. Recall that for  $C = (S_1, S_2, S_3)$ , we have  $I_{[S_1, S_2, S_3]} = \{(a, a)\}$ ,  $O_{[S_1, S_2, S_3]} = \{(b, a)\}$ ,  $\bigcup_{e \in E(D)} I_e = \{(a, b)\}$ , and  $\bigcup_{e \in E(D)} O_{(e)} = \{(a, a), (b, a), (b, b)\}$ , where  $D$  is the touching graph of  $G'$  w.r.t.  $C$  given in Figure 4. By Theorem 23 we easily (cf. the deduction in Example 20) see that there is an embedding relation  $E$  compatible with  $C$ : we have (in this order)  $(\{(a, a)\} \cup \{(a, b)\}) \cap \{(b, a)\} = \emptyset$ ,  $\pi_2(\{(a, b)\}) \cap \pi_2(\{(a, a)\}) \cap \{(a, a), (b, a), (b, b)\} = \{b\} \cap \{a\} = \emptyset$ , and  $\{(a, b)\} \cap \{(a, a), (b, a), (b, b)\} = \emptyset$ .

## 7 Determining compatible sequences of subgraphs

In the previous section we described, below Theorem 23, a method to determine an  $E \subseteq L \times L$  compatible with a *given* ordering  $C$  of a set  $\mathcal{S}$  of disjoint and isomorphic subgraphs. In this section we discuss a method to determine, given  $\mathcal{S}$ , an ordering  $C$  of  $\mathcal{S}$  for which there exists a compatible  $E$ . A naïve method would simply check the conditions of Theorem 23 for every possible ordering  $C$  until one such  $C$  satisfies the conditions. However, as we will show, we can do much better.

First we state the following corollary to Theorem 23 — the corollary is better suited to computationally determine a valid  $C$  for given  $\mathcal{S}$ .

**Corollary 25.** *Let  $G$  be a graph and  $C = (S_1, \dots, S_n)$  be a sequence of subgraphs of  $G$  and let  $D$  be the touching graph of  $G$  w.r.t.  $C$ . There is an  $E \subseteq L \times L$  compatible with  $C$  iff*

$$I_{[S_1, \dots, S_n]} \cap O_{[S_1, \dots, S_n]} = \emptyset, \quad (5)$$

$$I_{[S_1, \dots, S_n]} \cap O_e = \emptyset \text{ and } I_e \cap O_{[S_1, \dots, S_n]} = \emptyset \text{ for all } e \in E(D), \quad (6)$$

$$I_e \cap O_{(f)} = \emptyset \text{ and } \pi_2(I_e) \cap \pi_2(O_{(f)}) = \emptyset \text{ for all } e, f \in E(D). \quad (7)$$

*Proof.* The corollary is obtained by iteratively applying the distributivity property (i.e.,  $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ ) of union and disjunction on each of the four conditions of Theorem 23.

The equalities obtained from Equality 4,  $I_{[S_1, \dots, S_n]} \cap O_e = \emptyset$  for  $e \in E(D)$  and  $I_e \cap O_{(f)} = \emptyset$  for  $e, f \in E(D)$ , are used as follows. The former is included in Equality 6, while the latter is omitted as  $O_f \subseteq O_{(f)}$  for all  $f \in E(D)$ : hence  $I_e \cap O_{(f)} = \emptyset$  implies  $I_e \cap O_f = \emptyset$ .  $\square$

Note that Equality 5 (in the corollary) is computationally much easier to check compared to Equality 6, while Equality 6 in turn is much easier to check compared to Equality 7. As we will see, the reason for including the superfluous (see the proof of Corollary 25) equality  $I_{[S_1, \dots, S_n]} \cap O_e = \emptyset$  in Equality 6 is to speed up the evaluation of Equality 7.

For touching graphs  $S_1$  and  $S_2$ , we call  $e = (S_1, S_2)$  *admissible* (w.r.t.  $\mathcal{S}$ ) if both  $I_{[S_1, \dots, S_n]} \cap O_e = \emptyset$  and  $I_e \cap O_{[S_1, \dots, S_n]} = \emptyset$ . Equality 6 (in the previous corollary) thus states that all edges of  $D$  are admissible.

**Definition 26.** The *admissible touching graph* of  $G$  w.r.t.  $\mathcal{S}$  is the directed graph  $A = (\mathcal{S}, \{e \in \mathcal{S}^2 \mid \text{the nodes of } e \text{ are touching graphs and } e \text{ is admissible}\})$  (we allow anti-parallel edges).

*Example 27.* We continue the running Example 20. Recall that  $I_{[S_1, S_2, S_3]} = \{(a, a)\}$ ,  $O_{[S_1, S_2, S_3]} = \{(b, a)\}$  and  $I_{(S_1, S_2)} = I_{(S_2, S_3)} = I_{(S_1, S_3)} = \{(a, b)\}$ . Moreover we have  $O_{(S_1, S_2)} = O_{(S_2, S_3)} = O_{(S_1, S_3)} = \{(b, b)\}$  and thus  $(S_1, S_2)$ ,  $(S_2, S_3)$ , and  $(S_1, S_3)$  are admissible. However, as  $I_{(S_2, S_1)} = I_{(S_3, S_2)} = I_{(S_3, S_1)} = \{(b, a)\}$ ,  $(S_2, S_1)$ ,  $(S_3, S_2)$ , and  $(S_3, S_1)$  are *not* admissible. Hence, as it turns out, the admissible touching graph of  $G'$  w.r.t.  $\mathcal{S}$  is precisely the touching graph  $D$  of  $G'$  w.r.t.  $C = (S_1, S_2, S_3)$  given in Figure 4.

The general outline of the algorithm is as follows: (1) check Equality 5, (2) construct the admissible touching graph  $A = (\mathcal{S}, E_A)$ , and check that any two distinct  $v_1, v_2 \in \mathcal{S}$  where  $v_1$  and  $v_2$  touch in  $G$  have an admissible edge, and (3) find a linear ordering  $C$  of  $\mathcal{S}$  such that for the touching graph  $T = (\mathcal{S}, E_T)$  of  $G$  w.r.t.  $C$  we have that both  $E_T \subseteq E_A$  holds *and* Equality 7 holds (i.e., it is possible to remove edges of  $A$  which have an anti-parallel counterpart such that the obtained graph has a topological ordering that satisfies Equality 7). Only if checks (1) and (2) are fulfilled *and* a suitable  $C$  in (3) is found, then the obtained  $C$  is an ordering of  $\mathcal{S}$  for which there is a compatible  $E$ .

One can see now that including equality  $I_{[S_1, \dots, S_n]} \cap O_e = \emptyset$  in the definition of admissible is an optimization: the less number of edges in  $A$ , the less number of possible orderings  $C$  for which Equality 7 needs to be checked.

This algorithm is additionally optimized by further reducing the number of edges of  $A$  as follows. If there are edges  $e$  and  $f$  of  $A$  such that the Equality 7 does not hold for  $e$  and  $f$ , then they cannot both appear in any  $E(D)$  where  $D$  is the touching graph of any suitable  $C$ . Since any two touching subgraphs must have an admissible edge, if edge  $e = (x, y)$  does not have the anti-parallel edge  $(y, x)$  and  $e$  and  $f$  together do not satisfy Equality 7, then  $f$  can be removed from  $A$ . Clearly, if  $f$  also does not have an anti-parallel edge, then we can stop and there is no suitable  $C$ .

Now, the algorithm to determine an ordering  $C$  of  $\mathcal{S}$  such that there is an  $E \subseteq L \times L$ , where  $E$  is compatible with  $C$ , is as follows.

- 1: **procedure** FINDORDERING( $G, \mathcal{S}$ )
- 2:     **if** Eq. 5 (of Corollary 25) does not hold **then**                     ▷ Checking Eq. 5

```

3:         stop and there is no such  $C$ 
4:     end if
5:     construct the admissible touching graph  $A$  of  $G$  w.r.t.  $S$ .
6:     if there are vertices  $v_1, v_2$  of  $A$  without any edge between them and the
subgraphs  $v_1$  and  $v_2$  of  $G$  touch then                                ▷ Checking Eq. 6
7:         stop and there is no such  $C$ 
8:     end if
9:      $Q = \{e \in E_A \mid e \text{ has no anti-parallel counterpart}\}$ 
10:    while  $Q \neq \emptyset$  do                                          ▷ Optimization
11:        let  $e \in Q$  and  $Q \leftarrow Q \setminus \{e\}$ 
12:        for all  $f \in E_A$  do
13:            if  $e \neq f$  and Eq. 7 does not hold for  $e$  and  $f$  then
14:                if  $f$  has an anti-parallel counterpart  $f'$  in  $A$  then
15:                    remove  $f$  from  $A$ 
16:                     $Q \leftarrow Q \cup \{f'\}$ 
17:                else
18:                    stop and there is no such  $C$ 
19:                end if
20:            end if
21:        end for
22:    end while
23:    if there is a graph  $B$  obtainable from  $A$  by removing antiparallel edges
such that Eq. 7 holds for  $B$  then                                    ▷ Checking Eq. 7
24:        return  $C$ 
25:    end if
26:    stop and there is no such  $C$ 
27: end procedure

```

Notice that lines 9 to 22 represent the above described optimization to reduce the number of edges of  $A$  by detecting local incompatibilities of Equality 7.

## 8 Grammars with Multiple Productions

In this section we demonstrate that the results of this paper can be generalized in a straightforward way to the case where a NLC graph grammar  $Q$  can have more than one production. Although we could have stated our results directly for the general case, for clarity of exposition however, we choose to restrict ourselves to the case of one production and subsequently show ways to generalize it.

Thus let  $Q$  have now a *set*  $P$  of productions:  $P = \{N_i \rightarrow S_i \mid i \in \{1, \dots, n\}\}$ . Recall that the embedding relation  $E$  by definition is always fixed for given NLC graph grammar. We do again assume that the labels of  $S_i$  are terminals, i.e., none of the labels are equal to  $N_i$  for any  $i$ . This is the only restriction we make in this section compared to NLC graph grammars in general.

In the special case where all  $N_i$  are identical, say to  $N$ , it is easy to see that the results of the previous sections carry over essentially unchanged. Indeed, the

restriction that all the elements of  $\mathcal{S}$  are isomorphic to some graph  $S$  is not relevant in any of the results and proofs — the restriction was only needed to allow it to be generated by only one production  $N \rightarrow S$ .

We now consider the case where we have different non-terminals  $N_i$ . As is illustrated in the next example, we do require some modifications in the results to incorporate this case. As it turns out the required changes are only moderate.

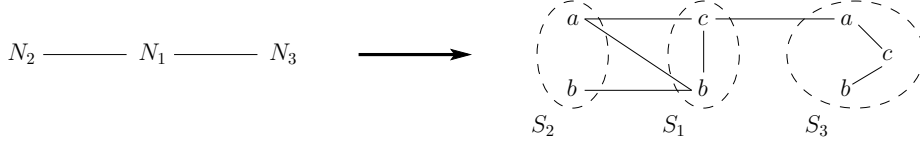


Fig. 5: Graphs  $G$  (left-hand side) and  $G'$  (right-hand side) of Example 28.

*Example 28.* Let  $G$  be the graph on the left-hand side of Figure 5, and let the labels  $N_1, N_2, N_3$  of  $G$  be distinct. Moreover, let  $G'$  be the graph on the right-hand side of Figure 5 having the disjoint subgraphs  $S_1, S_2$ , and  $S_3$  as depicted in the figure, and let  $r_i = N_i \rightarrow S_i$  for  $i \in \{1, 2, 3\}$  be rules of the graph grammar  $Q$ . We examine for which embedding relation  $E, G'$  can be generated from  $G$  by  $Q$ , replacing  $N_1, N_2, N_3$  by  $S_1, S_2, S_3$  in this order.

By Lemma 14 (more precisely, by the trivial generalization of this result),  $E$  is compatible with  $(S_1, S_2)$  in  $G'$  iff  $\{(a, c), (a, b), (b, b), (c, N_2), (b, N_2)\} \subseteq E$  and  $E \cap \{(b, c)\} = \emptyset$  (replacing  $N_2$  by  $S_2$  in the derivation). Also, by Lemma 14,  $E$  is compatible with  $(S_1, S_3)$  in  $G'$  iff  $\{(a, c), (c, N_3)\} \subseteq E$ ,  $E \cap \{(b, c), (c, c)\} = \emptyset$ , and  $(b, N_3) \notin E$  or  $(a, b), (b, b), (c, b) \notin E$  (replacing  $N_3$  by  $S_3$  in the derivation). We conclude, by Lemma 19, that  $E$  is compatible with  $(S_1, S_2, S_3)$  in  $G'$  iff  $\{(a, c), (a, b), (b, b), (c, N_2), (b, N_2), (c, N_3)\} \subseteq E$  and  $E \cap \{(b, c), (c, c), (b, N_3)\} = \emptyset$ . Thus such an  $E$  compatible with  $(S_1, S_2, S_3)$  in  $G'$  exists as we assume that  $N_2 \neq N_3$ . However, if  $N_2 = N_3$ , then there is no compatible  $E$  as we would have  $(b, N_2) \in E$  and  $(b, N_2) = (b, N_3) \notin E$  — a contradiction.

The previous example illustrates that by increasing the number of non-terminals, we have more freedom/independence to find a compatible  $E$ . Indeed, we may have the requirements  $(a, N_i) \in E$  and  $(a, N_j) \notin E$ , and these requirements can only both be fulfilled when  $N_i \neq N_j$ .

Assume now the most general case: each  $S_i \in \mathcal{S}$  is generated by a unique non-terminal in the derivation (thus this requires at least  $|\mathcal{S}|$  non-terminals and productions). Then, we see, e.g., that the requirement “ $\pi_2(I_e) \cap \pi_2(I_{[S_1, \dots, S_n]} \cap O_{(f)}) = \emptyset$  for all  $e, f \in E(D)$ ” in Eq. 7 in Corollary 25 must now hold only for the case where  $e$  and  $f$  have a common outgoing node  $S_j$  (i.e.,  $\pi_2(e) = \pi_2(f) = S_j$ ). Since the other requirements of the corollary are not concerned with tuples of  $E$  having a non-terminal, this is the only modification we need to incorporate. Hence we straightforwardly obtain the following generalization of Corollary 25.



**Corollary 29.** *Let  $G$  be a graph and  $C = (S_1, \dots, S_n)$  be a sequence of disjoint (but not necessarily isomorphic) subgraphs of  $G$  and let  $D$  be the touching graph of  $G$  w.r.t.  $C$ . There is an  $E \subseteq L \times L$  compatible with  $C$  iff*

$$\begin{aligned} I_{[S_1, \dots, S_n]} \cap O_{[S_1, \dots, S_n]} &= \emptyset, \\ I_{[S_1, \dots, S_n]} \cap O_e &= \emptyset \text{ and } I_e \cap O_{[S_1, \dots, S_n]} = \emptyset \text{ for all } e \in E(D), \\ I_e \cap O_{(f)} &= \emptyset \text{ and } \pi_2(I_e) \cap \pi_2(I_{[S_1, \dots, S_n]} \cap O_{(f)}) = \emptyset \\ &\text{for all } e, f \in E(D) \text{ with } \pi_2(e) = \pi_2(f). \end{aligned}$$

Thus, except for modifying Eq. 7 in Corollary 25 as described above, the algorithm of Section 7 remains unchanged. Note that the checks on lines 13 and 23 in the algorithm should thus only be done on edges having the same outgoing node (as we require  $\pi_2(e) = \pi_2(f)$ ).

*Remark 30.* Recall that we are determining whether or not a given set  $\mathcal{S}$  of subgraphs of  $G$  is “compressible” by graph rules to a graph  $I$  having vertices labelled by  $N$  instead of the subgraphs in  $\mathcal{S}$  (note however that, in general,  $G$  is only uniquely reproducible from  $I$  when also the order in which the vertices labelled by  $N$  are used in a production are stored). Although allowing  $\mathcal{S}$  to contain non-isomorphic graphs does not significantly increase the complexity of our problem under consideration, the problem of *finding* an optimal (in some measure) set  $\mathcal{S}$  of subgraphs of  $G$  that is compressible by graph rules is computationally significantly harder when  $\mathcal{S}$  may have non-isomorphic subgraphs.  $\square$

## 9 Discussion

In this paper we considered the problem of graph grammar inference for the case where one is given a set  $\mathcal{S}$  of disjoint and isomorphic subgraphs to be generated by a single rule  $r = N \rightarrow S/E$ , where the embedding relation  $E$  is allowed to contain tuples containing  $N$ . Moreover, we have considered ways to extend the results for sets of rules instead of a single rule. In this way we generalize results in [2]. The results of this paper are to be seen as a further step towards a systematic account of NLC graph grammar inference.

Formally, we characterized, given a  $\mathcal{S}$ , the existence of an ordering  $C$  of  $\mathcal{S}$  and an  $E \subseteq L \times L$  such that  $E$  is compatible with  $C$ . Moreover, if such a  $C$  exists, then it is shown to be a topological ordering of a suitable graph that identifies admissible pairs of touching subgraphs. The efficiency of the resulting algorithm depends significantly on the cardinality of  $\mathcal{S}$  — for small  $\mathcal{S}$  the algorithm seems practical, however this has yet to be verified.

Finding a graph  $S$ , such that the set  $\mathcal{S}$  of subgraphs of  $G$  isomorphic to  $S$  is (1) “compressible”, i.e., there is an embedding relation compatible with an ordering of  $\mathcal{S}$ , and (2) optimal (either in cardinality of  $\mathcal{S}$ , or in some other measure) remains to be investigated.

Also, it is natural to consider the case where we allow non-terminals as labels of nodes of  $S$  on the right-hand side of productions  $p = N \rightarrow S$ . This would

introduce “real” recursion where graphs may be “nested” in various ways — in this way the restriction of disjointness of the graphs in  $\mathcal{S}$  considered here is loosened.

### Acknowledgements

We thank the referee for comments and corrections on the paper. As a consequence, the paper is significantly improved. This research is supported by the Netherlands Organization for Scientific Research (NWO), project “Annotated graph mining”.

### References

1. H. Blockeel and R. Brijder. Learning non-confluent NLC graph grammar rules. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *5th Conference on Computability in Europe (CiE 2009), Mathematical Theory and Computational Practice, Abstract Booklet*, pages 60–69, 2009.
2. H. Blockeel and S. Nijssen. Induction of node label controlled graph grammar rules. In *Proceeding of the 6th International Workshop on Mining and Learning with Graphs (MLG 2008)*, 2008.
3. D.J. Cook and L.B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.
4. J. Engelfriet and G. Rozenberg. Graph grammars based on node rewriting: An introduction to NLC graph grammars. In *Graph-Grammars and Their Application to Computer Science*, pages 12–23, 1990.
5. J. Engelfriet and G. Rozenberg. Node replacement graph grammars. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation. Vol. I: Foundations*, chapter 1, pages 1–94. World Scientific, Singapore, 1997.
6. K.S. Fu and T.L. Booth. Grammatical inference: introduction and survey - part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:343–359, 1986.
7. I. Jonyer, L. Holder, and D. Cook. MDL-based context-free graph grammar induction and applications. *International Journal on Artificial Intelligence Tools*, 13:65–79, 2004.
8. E. Vidal. Grammatical inference: An introductory survey. In *Grammatical Inference and Applications*, volume 862 of *Lecture Notes in Computer Science*, pages 1–4, 1994.