UNIVERSITÄT DUISBURG ESSEN

ABTEILUNG FÜR INFORMATIK UND ANGEWANDTE KOGNITIONSWISSENSCHAFT

FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN

INFORMATIK

Technischer Bericht Nr. 2012-01

Applying the Graph Minor Theorem to the Verification of Graph Transformation Systems

Salil Joshi
Barbara König

20.4.2012

ISSN 1863-8554

IMPRESSUM:

Technische Berichte der Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen

ISSN 1863-8554

Herausgeber:

Abteilung für Informatik und Angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen
Campus Duisburg
47048 Duisburg

http://duepublico.uni-duisburg-essen.de/informatik/berichte.xml

Applying the Graph Minor Theorem to the Verification of Graph Transformation Systems*

Salil Joshi[1] and Barbara König[2]

[1] Indian Institute of Technology, Delhi, India
[2] Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Germany

Abstract. We show how to view certain subclasses of (single-pushout) graph transformation systems as well-structured transition systems, which leads to decidability of the covering problem via a backward analysis. As the well-quasi order required for a well-structured transition system we use the graph minor ordering. We give an explicit construction of the backward step and apply our theory in order to show the correctness of a leader election protocol.[3]

1 Introduction

In a series of seminal papers Robertson and Seymour have shown that graphs are well-quasi-ordered with respect to the minor ordering [7, 8]: in any (infinite) sequence of graphs $G_0, G_1, G_2, \ldots$ there are always two indices $i < j$ such that G_i is a minor of G_j. This means that G_i can be obtained from G_j by deleting and contracting edges and by deleting isolated nodes.

The theorem has far-reaching consequences. It guarantees that every set of graphs that is upward-closed with respect to the minor ordering can be represented by a finite number of minimal graphs. Similarly, any downward-closed set of graphs (e.g., planar graphs, forests, graphs embeddable in a torus) can be characterized by a finite set of forbidden minors. A well-known special case are (undirected) planar graphs which are characterized by two forbidden minors: the complete graph with five nodes (K_5) and the complete bipartite graph with six nodes ($K_{3,3}$), a fact which is known as Kuratowski's theorem.

Well-quasi-orders (wqo's) also play a fundamental role in the analysis of a class of (infinite-state) transition systems, so called well-structured transition systems (WSTS) [4]. States in a WSTS are well-quasi-ordered and the standard analysis method shows whether some state in an upward-closed set is reachable from an initial state by performing backward analysis. The well-quasi-ordering guarantees that upward-closed sets are finitely representable, that the set of predecessors is also upward-closed and that the technique terminates after finitely many steps.

One important example for WSTS are Petri net transition system, where a marking m_1 is considered larger than or equal to m_2 if it contains at least as many tokens in every place. Other examples are string rewrite systems, basic process algebra and communicating finite state machines. A transition system that can not be naturally viewed as a WSTS can often be turned into one by introducing some notion of "lossiness". For instance an unreliable channel may lose messages and a

* Research partially supported by the DFG project SANDS.

[3] This paper is the full version of a paper published in the CAV '08 proceedings. Furthermore it corrects two errors which were present in the original version: there was a mistake in the definition of the minor relation for hypergraphs (Definition 11), and furthermore Fig. 4 was wrong.

suitable wqo considers the content c_1 of a channel as greater than c_2 if c_2 can be obtained from c_1 by dropping some messages.

The graph minor ordering fits well with this intuition of "lossiness" and seems to be applicable to networks where edges (connections or processes) may disappear—possibly due to faults—and where edges can be contracted. The latter phenomenon appears if a process leaves a network by connecting its predecessor and successor, something which typically happens in rings.

Here we show how to view certain graph transformation systems (GTS) as WSTS with respect to the minor ordering. GTS are an intuitive formalism, well-suited to model concurrent and distributed systems. In general GTS are Turing-complete and due to undecidability issues it is hard to imagine a useful wqo for the general case. However, if the GTS exhibits features as described above it can be successfully verified.

GTS are typically defined by means of category theory, which makes the definition of rewriting steps less tedious. Graph rewriting is defined via pushouts in a suitable category of graph morphisms and in the rest of this paper we will exploit certain well-known properties of pushouts. The relation of a graph G to its minor H can be represented by a partial graph morphism with specific properties. Since the theory requires the handling of partial morphisms, we have decided to work in the single-pushout approach (SPO) which uses partial morphisms [5, 3].

The paper is organized as follows: Section 2 introduces the basic definitions. In Section 3 we consider classes of GTS that can be seen as WSTS, and introduce the techniques for their analysis. In Section 4 we will look at a leader election protocol and show how the analysis method works in practice.

2 Preliminaries

Here we introduce some of the basic notions needed in the paper, especially well-quasi-orders, well-structured transition systems, graph transformation systems and minors.

2.1 Well-quasi-order

Definition 1 (wqo). *A* well-quasi-order *(wqo) is any quasi-ordering*[4] $\leq$ *(over some set* X*) such that, for any infinite sequence* $x_0, x_1, x_2, \ldots$ *in* X*, there exist indices* $i < j$ *with* $x_i \leq x_j$*.*

An upward-closed set *is any set* $I \subseteq X$ *such that* $y \geq x$ *and* $x \in I$ *entail* $y \in I$*. A* downward-closed set *can be analogously defined.*

For an element $x \in I$*, we define* $\uparrow x = \{y \mid y \geq x\}$*. Then, a* basis *of an upward-closed set* I *is a set* I^b *such that* $I = \bigcup_{x \in I^b} \uparrow x$*.*

Lemma 2.

1. *If* $\leq$ *is a well-quasi-ordering then any upward-closed* I *has a finite basis.*
2. *If* $\leq$ *is a wqo and* $I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots$ *is an infinite increasing sequence of upward-closed sets, then there exists an index* $k \in \mathbb{N}$ *such that* $I_k = I_{k+1} = I_{k+2} = \ldots$

2.2 Well-Structured Transition Systems

Definition 3 (WSTS). *A* well-structured transition system *(WSTS) is a transition system* $T = (S, \Rightarrow, \leq)$*, where* S *is a set of states and* $\Rightarrow \subseteq S \times S$*, such that the following conditions hold:*

[4] Note that a quasi-order is the same as a preorder.

1. ***Well quasi ordering:*** *$\leq$ is a well-quasi-ordering on S.*
2. ***Compatibility:*** *For all $s_1 \leq t_1$ and a transition $s_1 \Rightarrow s_2$, there exists a sequence $t_1 \Rightarrow^* t_2$ of transitions such that $s_2 \leq t_2$.*

$$\begin{array}{ccc} t_1 & \stackrel{*}{\Longrightarrow} & t_2 \\ \vee\!| & & \vee\!| \\ s_1 & \Longrightarrow & s_2 \end{array}$$

Given a set $I \subseteq S$ of states we denote by $Pred(I)$ the set of direct predecessors of I, i.e., $Pred(I) = \{s \in S \mid \exists s' \in I \colon s \Rightarrow s'\}$. Furthermore $Pred^*(I)$ is the set of all predecessors.

Let $(S, \Rightarrow, \leq)$ be a WSTS. Consider a set of states $I \subseteq S$. Backward reachability analysis involves the computation of $Pred^*(I)$ as the limit of the sequence $I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots$ where $I_0 = I$ and $I_{n+1} = I_n \cup Pred(I_n)$. However, in general this may not terminate. For WSTS, if I is upward-closed then it can be shown that $Pred^*(I)$ is also upward-closed (compatibility condition) and that termination is guaranteed (Lemma 2).

Definition 4 (Effective pred-basis). *A WSTS has an* effective pred-basis *if there exists an algorithm accepting any state $s \in S$ and returning $pb(s)$, a finite basis of $\uparrow Pred(\uparrow s)$.*

Now assume that T is a WSTS with effective pred-basis. Pick a finite basis I^b of I and define a sequence $K_0, K_1, K_2, \ldots$ of sets with $K_0 = I^b$ and $K_{n+1} = K_n \cup pb(K_n)$. Let m be the first index such that $\uparrow K_m = \uparrow K_{m+1}$. Such an m must exist by Lemma 2 and we have $\uparrow K_m = Pred^*(I)$. Finally, note that due to Lemma 2 every set K_n can be represented by a finite basis.

The *covering problem* is to decide, given two states s and t, whether starting from a state s it is possible to cover t, i.e. to reach a state t' such that $t' \geq t$. From the previous argument follows the decidability of the covering problem.

Theorem 5 (Covering problem). *The covering problem is decidable for a WSTS with an effective pred-basis and a decidable wqo $\leq$.*

Thus, if T is a WSTS and the "error states" can be represented as an upward-closed set I, then it is decidable whether any element of I is reachable from the start state.

2.3 Graphs and Graph Transformation

Definition 6 (Hypergraph). *Let Λ be a finite sets of edge labels and $ar \colon \Lambda \to \mathbb{N}$ a function that assigns an arity to each label. A (Λ-)*hypergraph *is a tuple (V_G, E_G, c_G, l_G^E) where V_G is a finite set of nodes, E_G is a finite set of edges, $c_G \colon E_G \to V_G^*$ is a connection function and $l_G^E \colon E_G \to \Lambda$ is an edge labelling function. We require that $|c_G(e)| = ar(l_G^E(e))$ for each edge $e \in E_G$*

An edge e is called adjacent *to a node v if v occurs in $c_G(e)$.*

Directed labelled graphs are a special case of hypergraphs where every sequence $c_G(e)$ is of length two.

A path in a hypergraph is a sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n$ of nodes such that for every index $i \in \{1, \ldots, n\}$ both nodes v_{i-1} and v_i are adjacent to e_i.

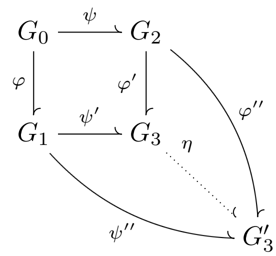
Definition 7 (Partial hypergraph morphism). *Let G, G' be (Λ-)hypergraphs. A* partial hypergraph morphism *(or simply* morphism*) $\varphi \colon G \rightharpoonup G'$ consists of a pair of partial functions $(\varphi_V : V_G \rightharpoonup V_{G'}, \varphi_E : E_G \rightharpoonup E_{G'})$ such that for every $e \in E_G$ it holds that $l_G(e) = l_{G'}(\varphi_E(e))$ and $\varphi_V(c_G(e)) = c_{G'}(\varphi_E(e))$ whenever $\varphi_E(e)$ is defined. Furthermore if a morphism is defined on an edge, it must be defined on all nodes adjacent to it. (This condition need not hold in the other direction.)*

Total morphisms *are denoted by an arrow of the form $\to$.*

In the following we will drop the subscripts and write φ instead of φ_V and φ_E.

Gluing of graphs along a common subgraph is done via pushouts in the category of partial graph morphisms.

Definition 8 (Pushout).
Let $\varphi: G_0 \rightharpoonup G_1$ and $\psi: G_0 \rightharpoonup G_2$ be two partial graph morphisms. The pushout *of φ and ψ consists of a graph G_3 and two graph morphisms $\psi': G_1 \rightharpoonup G_3$, $\varphi': G_2 \rightharpoonup G_3$ such that $\psi' \circ \varphi = \varphi' \circ \psi$ and for every other pair of morphisms $\psi'': G_1 \rightharpoonup G_3'$, $\varphi'': G_2 \rightharpoonup G_3'$ such that $\psi'' \circ \varphi = \varphi'' \circ \psi$ there exists a unique morphism $\eta: G_3 \rightharpoonup G_3'$ with $\eta \circ \psi' = \psi''$ and $\eta \circ \varphi' = \varphi''$.*



It is known that pushouts of partial graph morphisms always exist, that they are unique up to isomorphism and that they can be constructed as follows. The intuition behind the construction is that G_1, G_2 are glued together along a common interface G_0 and that an element is deleted if it is deleted by either φ or ψ.

Proposition 9 (Construction of pushouts). *Let $\varphi: G_0 \rightharpoonup G_1$, $\psi: G_0 \rightharpoonup G_2$ be partial hypergraph morphisms. Furthermore let $\equiv_V$ be the smallest equivalence on $V_{G_1} \cup V_{G_2}$ and $\equiv_E$ the smallest equivalence on $E_{G_1} \cup E_{G_2}$ such that $\varphi(x) \equiv \psi(x)$ for every element x of G_0.*

An equivalence class of nodes is called valid *if it does not contain the image of a node x for which $\varphi(x)$ or $\psi(x)$ are undefined. Similarly a class of edges is* valid *if the analogous condition holds and furthermore all nodes adjacent to these edges are contained in valid equivalence classes.*

Then the pushout G_3 of φ and ψ consists of all valid equivalence classes $[x]_\equiv$ as nodes and edges, where $l_{G_3}([e]_\equiv) = l_{G_i}(e)$ and $c_{G_3}([e]_\equiv) = [v_1]_\equiv \dots [v_k]_\equiv$ if $e \in E_{G_i}$ and $c_{G_i}(e) = v_1 \dots v_k$.

It can be seen that the pushout of two total morphisms (in the category of partial morphisms) always results in two total morphisms. Furthermore it is equal to their pushout in the category of total morphisms. However φ total and ψ partial does not necessarily imply that φ' is total. This is due to so-called *deletion/preservation conflicts* where two elements x_0, x_0' of G_0 are mapped to the same element of G_1, i.e., $\varphi(x_0) = \varphi(x_0')$, while $\psi(x_0)$ is defined, whereas $\psi(x_0')$ is undefined. The construction above suggests that then $\varphi'(\psi(x_0))$ must be undefined, i.e., φ' is not total. If no such elements x_0, x_0' can be found, then φ is said to be *conflict-free* with respect to ψ and in this case φ' is always total.

Definition 10 (Graph rewriting). *A* rewriting rule *is a partial morphism $r: L \rightharpoonup R$, where L is called left-hand side and R right-hand side.*

A match *(of r) is a total morphism $m: L \to G$ which is conflict-free wrt. r.*

Given a rule and a match, a rewriting step *or an application of the rule to the graph G, resulting in H, is a pushout diagram as shown in Fig. 1 on the left. In this case we write $G \Rightarrow H$.*

Intuitively, we can think of this as follows: L is a subgraph of G, all items of L whose image is undefined under r are deleted, the new items of R are added and connected as specified by r. Note that whenever a node is deleted, all adjacent edges will be deleted as well.

Fig. 1 shows two examples for graph rewriting steps. In the middle pushout a binary hyperedge generates another (unary) hyperedge, whereas in the right pushout

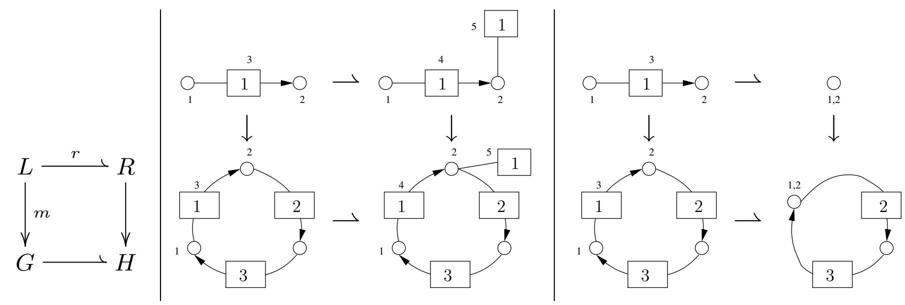


Fig. 1: Single-pushout graph rewriting (pushout diagram and example rewriting steps).

an edge is contracted. The way in which the morphisms map nodes and edges is indicated by the small numbers next to the edges. These specific rewriting rules will also play a role in our application (see Section 4).

In the context of this paper a *graph transformation system (GTS)* consists of a finite set $\mathcal{R}$ of rewriting rules. Sometimes we will fix an *initial graph* or *start graph*.

2.4 Minors and Minor Morphisms

We will now review the notion of a graph minor.

Definition 11 (Minor). *A graph $\hat{G}$ is a* minor *of a (hyper-)graph G, if $\hat{G}$ can be obtained from G by (repeatedly) performing the following operations on G:*

1. *Deletion of an edge.*
2. *Contraction of an edge. In this case we remove the edge, choose an arbitrary partition on the nodes connected to the edge and merge the nodes as specified by the partition. (This includes edge deletion as a special case.)*
3. *Deletion of an isolated node.*

The *Robertson-Seymour Theorem* [7] says that the minor order is a well-quasi-order. In fact, this theorem is true even if the edges and vertices of the graphs are labelled from a well-quasi-ordered set, and also for hypergraphs and directed graphs (see [8]). In fact, we here use a slightly different minor ordering than the one in [8], but we will prove in Appendix A (Proposition 22) that our variant also gives rise to a well-quasi order.

Now, if we could show that a GTS satisfies the compatibility condition of Definition 3 (with respect to the minor ordering), we could analyze it using the theory of WSTS. But before we characterize such GTS we first need the definition of minor morphisms and their properties. A *minor morphism* is a partial morphism that identifies a minor of a graph.

Definition 12 (Minor morphism). *A partial morphism $\mu : G \rightharpoonup \hat{G}$ is a minor morphism (written $\mu : G \mapsto \hat{G}$) if*

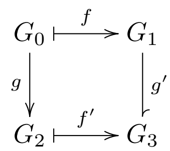
1. *it is surjective,*
2. *it is injective on edges* and
3. *whenever $\mu(v) = \mu(w) = z$ for some $v, w \in V_G$ and $z \in V_{\hat{G}}$, there exists a path between v and w in G where all nodes on the path are mapped to z and μ is undefined on every edge on the path.*

In [8] a different way to characterize minors is proposed: a function, going in the opposite direction, mapping nodes of $\hat{G}$ to subgraphs of G. This however can not be seen as a morphism in the sense of Definition 7 and we would have problems integrating it properly into the theory of graph rewriting.

One can show the following facts about minor morphisms.

Lemma 13. *$\hat{G}$ is a minor of G iff there exists a minor morphism $\mu : G \mapsto \hat{G}$.*

Lemma 14. *Pushouts preserve minor morphisms in the following sense: If $f : G_0 \mapsto G_1$ is a minor morphism and $g : G_0 \to G_2$ is total, then the morphism f' in the pushout diagram below is a minor morphism.*



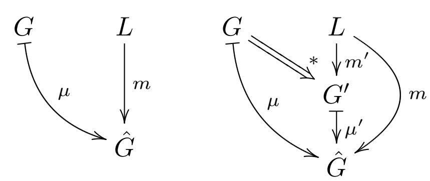
3 GTS as WSTS!

As observed earlier, a GTS can be seen as a WSTS with the minor relation as the well-quasi-ordering, provided the GTS satisfies the compatibility condition introduced in Definition 3.

3.1 Characterization

We will first give a sufficient condition that allows us to view a GTS as a WSTS. Note that the fundamental problem is that whenever a minor of G contains a left-hand side, then G might contain a "disconnected" copy of the left-hand side.

Proposition 15 (GTS as WSTS). *Let $\mathcal{R}$ be a GTS that satisfies the following condition: For every rule $(r : L \rightharpoonup R) \in \mathcal{R}$, every minor morphism $\mu : G \mapsto \hat{G}$ and every match $m : L \to \hat{G}$ (see diagram on the left) there exists a graph G' such that $G \Rightarrow^* G'$, there is a minor morphism $\mu' : G' \mapsto \hat{G}$ and there exists a match $m' : L \to G'$ such that $m = \mu' \circ m'$ (see commuting diagram below on the right). Then $\mathcal{R}$ is a WSTS.*



With this characterization we can now identify suitable types of GTS that are WSTS:

- Context-free graph grammars, where the left-hand side of every rule consists of a single hyperedge. Here G must always contain a match of L that makes the above diagram commute and no intermediate graph G' is needed.
- GTS where the left-hand sides of the rules consist of disconnected edges. The argument is analogous to the case above.
- Any arbitrary GTS can be transformed into a WSTS with the addition of all proper edge contraction rules for every edge label (the contraction rule that does not contract any nodes, but only deletes the edge, can be omitted). Now, if $\hat{G}$ contains a subgraph which is isomorphic to a left-hand side, the pre-image of this subgraph under μ is present in G, but it might possibly be disconnected. The minor morphism μ makes the elements of L adjacent by contracting paths and the same can be done by applying the additional edge contraction rules.

3.2 Backward Analysis

Let $\mathcal{R}$ be a set of graph transformation rules which satisfies the compatibility condition. Now we consider the question of performing a backward reachability analysis on $\mathcal{R}$ which requires a method for computing an effective pred-basis $pb(S)$ for a given graph S.

Our method will involve the backwards application of an SPO rewriting rule. This requires the completion of a diagram of the form $L \rightharpoonup R \rightarrow H$ by a graph G and morphisms $L \rightarrow G \rightharpoonup H$ such that the square is a pushout. Then G is a so-called *pushout complement*. Pushout complements are well-studied for total morphisms since they are an essential ingredient in double-pushout rewriting. For partial morphisms they have been studied to a lesser extent.

We will first demonstrate some issues that can arise with pushout complements: for instance, the two total morphisms $L \rightharpoonup R \rightarrow H$ shown in Fig. 2 (left) (edges and nodes are unlabelled, morphisms are indicated by numbers $1, 2$) have five different pushout complements. Note also that each pair of total morphisms has only finitely many pushout complements (up to isomorphism).

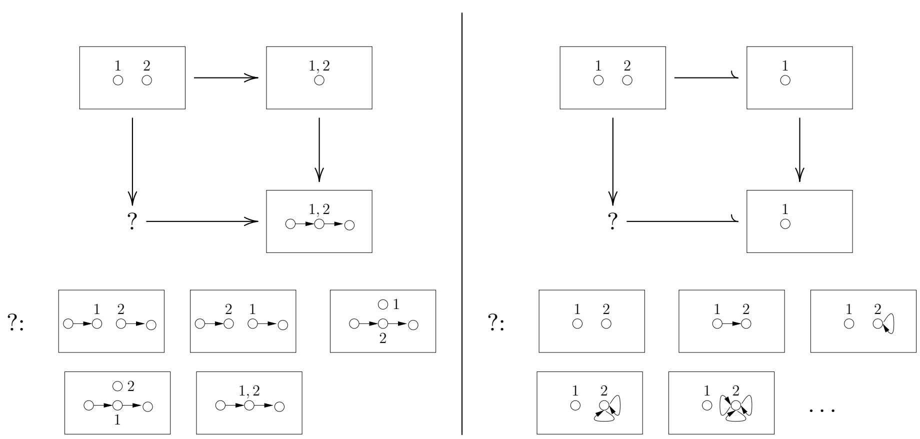


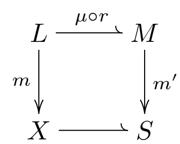
Fig. 2: Left: Two total morphisms with five pushout complements. Right: A partial and a total morphism with infinitely many pushout complements.

While the existence of multiple pushout complements is a feature that will be needed to determine the pred-basis, the situation for partial morphisms is more involved. Consider the diagram in Fig. 2 (right) where the morphism from L to R is partial. Here we have infinitely many pushout complements. Note however that the first graph is a minor of all other pushout complements. This suggests that only the computation of minimal pushout complements is needed.

Now we will give a high-level description of the procedure for computing $pb(S)$ for a given graph S. A more detailed account will be given in Section 3.3 where we will also argue that the procedure is indeed effective.

1. For each rule $(r : L \rightharpoonup R) \in \mathcal{R}$, let $\mathcal{M}_R$ be the (finite) set of all minor morphisms with source R.
2. For each $(\mu: R \mapsto M) \in \mathcal{M}_R$ consider the rule $\mu \circ r: L \rightharpoonup M$.

3. For each total match $m' : M \to S$ compute all minimal[5] pushout complements X such that $m : L \to X$ below is total and conflict-free wrt. r.



4. The set $pb(S)$ contains all graphs X obtained in this way.

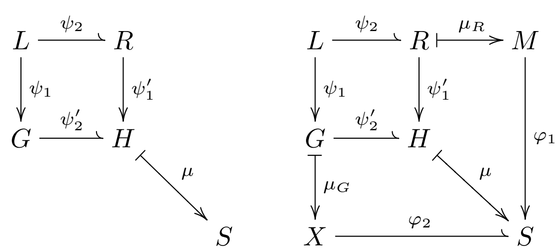
That is, we use all minors of R as right-hand sides for the backward step. This is needed since S represents an upward-closed set and not all items of R must be present in S itself. We can now show the correctness of the procedure $pb(S)$, where the proof depends crucially on Lemma 14.

Theorem 16. *The procedure $pb(S)$ computes a finite subset of $Pred(\uparrow S)$.*

In order to prove that $pb(s)$ generates every member of the pred-basis, we first prove a general result in the category of graphs and partial morphisms.

Lemma 17. *Let $\psi_1 : L \to G$ be total and conflict-free wrt. ψ_2. If the diagram below on the left is a pushout and $\mu : H \mapsto S$ a minor morphism, then there exist minors M and X of R and G respectively, such that*

1. the diagram below on the right commutes and the outer square is a pushout.
2. the morphisms $\mu_G \circ \psi_1 : L \to X$ and $\varphi_1 : M \to S$ are total and $\mu_G \circ \psi_1$ is conflict-free wrt. ψ_2.



The lemma above says that whenever S is a minor of H and G is a predecessor of H, then we can make a backwards step for S and obtain X, a minor of G. Using this lemma we can now state the completeness of the procedure $pb(S)$.

Theorem 18. *The set generated by $pb(S)$ is a pred-basis of S.*

3.3 Computing Minimal Pushout Complements

Now we consider the question of how to construct pushout complements when some (but not all) of the morphisms involved may be partial. Hence consider a diagram $L \overset{\varphi}{\rightharpoonup} \tilde{L} \to \tilde{X}$. The idea is to split $L \overset{\varphi}{\rightharpoonup} \tilde{L} = L \rightharpoonup dom(\varphi) \to \tilde{L}$ where $dom(\varphi) \to \tilde{L}$ is total and $L \rightharpoonup dom(\varphi)$ is an inverse injection, i.e., a morphism which is injective, surjective, but not necessarily total. Now the task of computing pushout complements can be divided into two subtasks.

Lemma 19. *Let L and $\tilde{L}$ be graphs, $\varphi_1 : L \rightharpoonup \tilde{L}$ be an inverse injection, and $\psi_2 : \tilde{L} \to \tilde{X}$ be a total morphism. Now construct a specific pushout complement X' with morphisms $\psi'_1 : L \to X'$, $\varphi'_2 : X' \rightharpoonup \tilde{X}$ as follows:*

[5] "Minimimal" means "minimal wrt. the well-quasi ordering $\leq$".

1. *Take a copy of the graph $\tilde{X}$, and let ψ_1' be $\psi_2 \circ \varphi_1$. The morphism φ_2' is the identity.*
2. *Let Y be the set of elements of L the image of which is undefined under φ_1. Add a copy of Y to this copy of $\tilde{X}$, and extend ψ_1' by mapping Y into this set. Furthermore φ_2' is undefined on all elements of the copy of Y.*
3. *Now merge these new elements (originally contained in Y) in all possible combinations, i.e., factor through all appropriate[6] equivalences. The morphisms ψ_1' and φ_2' are modified accordingly.*

The set of graphs obtained in this way is denoted by $\mathcal{P}$. Each element X' of $\mathcal{P}$ is a pushout complement of φ_1, ψ_2 and the corresponding morphisms $\psi_1' : L \to X'$ are total. Any other pushout complement X where $\psi_1 : L \to X$ is total (see diagram on the right) has some graph $X' \in \mathcal{P}$ as a minor.

$$\begin{array}{ccc} L & \xrightharpoonup{\varphi_1} & \tilde{L} \\ \downarrow{\scriptstyle \psi_1} & & \downarrow{\scriptstyle \psi_2} \\ X & \xrightharpoonup{\varphi_2} & \tilde{X} \end{array}$$

Finally, if $\psi_1 : L \to X$ is conflict-free wrt. to a rule $r : L \rightharpoonup R$, then there exists a pushout complement $X' \in \mathcal{P}$ with $\psi_1' : L \to X'$ conflict-free wrt. r, such that $X' \leq X$.

In order to do backwards application of rules in order to obtain $pb(s)$, we construct pushout complements (with total conflict-free matches) as follows:

Proposition 20. *Let $r : L \rightharpoonup R$ be a fixed rule. Furthermore let L, M and S be graphs, with a partial morphism $\varphi_1 : L \rightharpoonup M$ and a total morphism $\psi_2 : M \to S$. Then, if we apply the following procedure we only construct pushout complements X' of φ_1, ψ_2 and any other pushout complement X (with $\psi_1 : L \to X$ where ψ_1 is total and conflict-free wrt. r) has one of them as a minor.*

1. *Split φ_1 into two morphisms as follows: let $\varphi_1' : L \rightharpoonup dom(\varphi_1)$ be an inverse injection and let $\varphi_1'' : dom(\varphi_1) \to M$ be total.*
2. *Now consider the total morphisms $\varphi_1'' : dom(\varphi_1) \to M$, and $\psi_2 : M \to S$. Construct all their pushout complements as usual for total morphisms.*[7]
3. *Let $\tilde{X}$ be any such pushout complement with $\eta : dom(\varphi_1) \to \tilde{X}$.*
4. *For φ_1', η use the construction of Lemma 19 in order to obtain the minimal pushout complements X' (with total and conflict-free ψ_1').*
5. *Finally, from all such pushout complements X' take the minimal ones.*

The situation is depicted in the diagram below.

$$\begin{array}{ccccc} L & \xrightharpoonup{\varphi_1'} & dom(\varphi_1) & \xrightarrow{\varphi_1''} & M \\ \downarrow{\scriptstyle \psi_1'} & & \downarrow{\scriptstyle \eta} & & \downarrow{\scriptstyle \psi_2} \\ X' & \rightharpoonup & \tilde{X} & \longrightarrow & S \end{array}$$

4 Example: Leader Election

As an example, we shall apply this technique to a typical leader election protocol, to verify its correctness. The rules for this leader election protocol are shown in Fig. 3. We start with a ring containing processes, each with a unique natural number as ID. These processes can generate messages containing their ID, which are forwarded whenever the ID of the message is smaller than the ID of the process which receives it. A process becomes the leader if it receives a message containing its own ID. Non-leader processes may also choose to leave the system at any time, connecting

[6] Here "appropriate" means that whenever two edges are in the equivalence relation, all their adjacent nodes must be pairwise equivalent.

[7] We do not describe this construction here, but it is well-known that there are only finitely many such pushout complements and that they can be constructed effectively.

its predecessor and successor. We will prove that such a system can never create two leaders in the ring.

It can be seen that these rules satisfy the compatibility condition. The rule for edge contraction can be interpreted as a process leaving the system. Note that we do not need to add a rule for contracting messages (since messages are unary hyperedges), or for edge deletion in order to ensure compatibility.

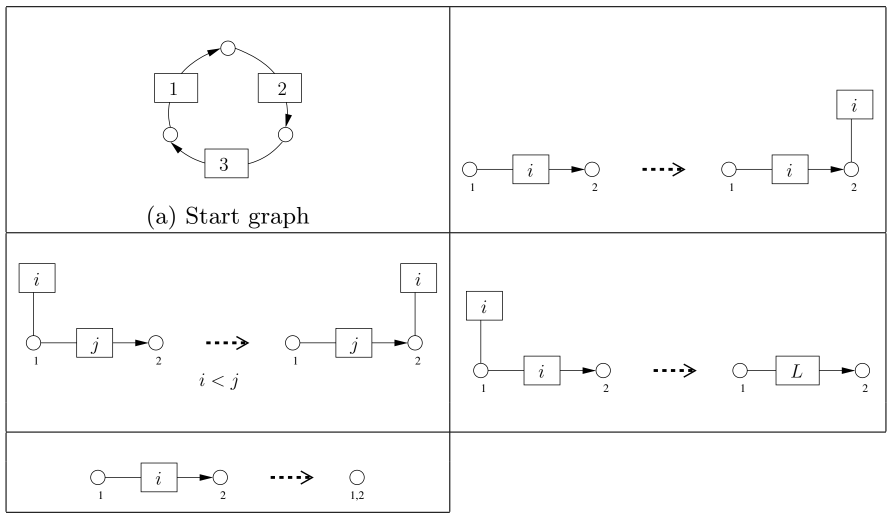


Fig. 3: Leader election (start graph and rewriting rules).

All forbidden minors (which we computed manually) are shown in Fig. 4. We start with the first of these as the error state, and performing the backward analysis we obtain the rest of the forbidden minors. We consider natural numbers up to a certain bound, in order to keep the label and rule sets finite. Here, i, j or k as a label indicates "any number" (except where a constraint is indicated). Thus, the entire process has been fully parametrized, so that these forbidden minors are valid for a start graph with an arbitrarily large number of processes in the ring. Since the given start graph does not have any of these forbidden graphs as a minor, we can conclude that the leader election protocol is correct, i.e., it can never create two leaders.

Note that since our technique can handle infinite state spaces, we could use the expressive power of graph transformation to extend the example in such a way that the ring is extended by new processes during runtime.

5 Conclusion

We have shown how to view subclasses of graph transformation systems as WSTS which gives us a decision algorithm for the covering problem. Currently we are working on an implementation which will help us to get a better insight into efficiency issues. Specifically it will help us to answer how many backward steps usually have to be taken and how many forbidden minors are generated. Although the worst case behaviour of this technique will certainly be bad, it might be feasible for many practical applications. We are also working on a more extended case study involving a termination detection protocol.

Another issue is the treatment of negative application conditions that have so far posed many problems in the analysis of GTSs. As already observed in [9] backward

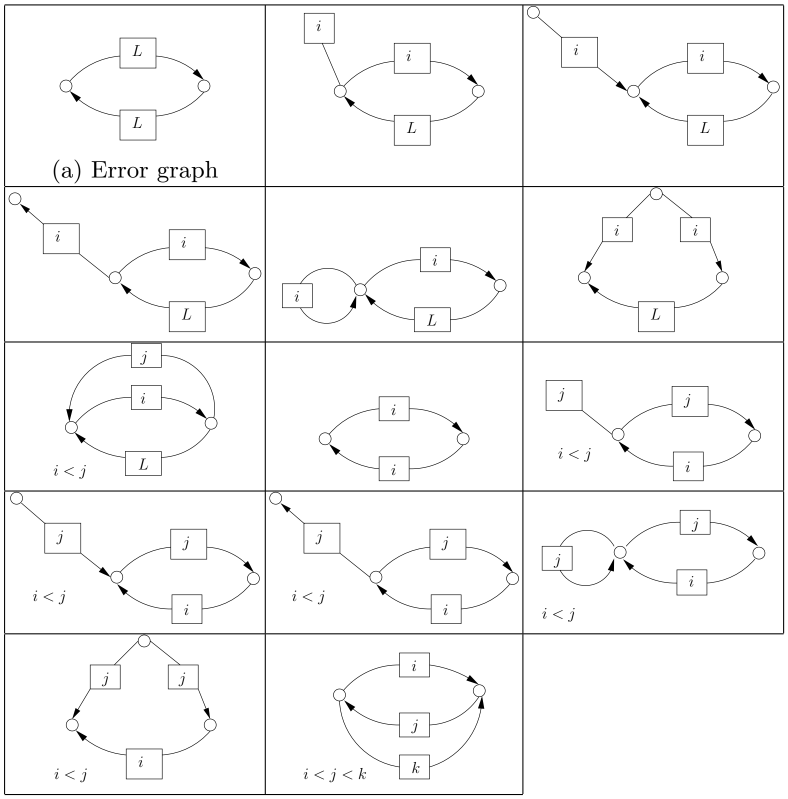


Fig. 4: Leader election (forbidden minors).

techniques seem to have fewer problems with negative application conditions than forward techniques which have so far mainly been studied. We also believe that such application conditions can be integrated with our technique.

Additional future work will be the investigation of partial order techniques (as in [1]) and the combination with (approximative) forward techniques (as described in [2, 6]) in order to eliminate states which are not reachable from the start graph early on. In addition we work on a related technique which allows to show whether certain invariants (represented by forbidden minors) are preserved by graph transformation rules.

Acknowledgements: We would like to thank Javier Esparza for his suggestion to explore the relation between WSTS and graph transformation.

References

1. Parosh Aziz Abdulla, Bengt Jonsson, Mats Kindahl, and Doron Peled. A general approach to partial order reductions in symbolic verification. In *Proc. of CAV '98*, pages 379–390. Springer, 1998. LNCS 1427.
2. Paolo Baldan, Andrea Corradini, and Barbara König. A static analysis technique for graph transformation systems. In *Proc. of CONCUR '01*, pages 381–395. Springer-Verlag, 2001. LNCS 2154.
3. H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, and A. Corradini. Algebraic approaches to graph transformation—part II: Single pushout approach and comparison with double pushout approach. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol.1: Foundations*, chapter 4. World Scientific, 1997.

4. Alain Finkel and Phillipe Schnoebelen. Well-structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.
5. Michael Löwe. Algebraic approach to single-pushout graph transformation. *Theoretical Computer Science*, 109:181–224, 1993.
6. Arend Rensink and Dino Distefano. Abstract graph transformation. In *Proc. of SVV '05 (3rd International Workshop on Software Verification and Validation)*, volume 157.1 of *ENTCS*, pages 39–59, 2005.
7. Neil Robertson and Paul Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.
8. Neil Robertson and Paul Seymour. Graph minors. XXIII. Nash-Williams' immersion conjecture, 2006. Submitted for publication. `http://www.math.princeton.edu/~pds/papers/GM23/GM23.pdf`.
9. Mayank Saksena, Oskar Wibling, and Bengt Jonsson. Graph grammar modeling and verification of ad hoc routing protocols. In *Proc. of TACAS '08*, pages 18–32. Springer, 2008. LNCS 4963.

A Proofs

Proposition 9 (Construction of pushouts). *Let $\varphi: G_0 \rightharpoonup G_1$, $\psi: G_0 \rightharpoonup G_2$ be partial hypergraph morphisms. Furthermore let $\equiv_V$ be the smallest equivalence on $V_{G_1} \cup V_{G_2}$ and $\equiv_E$ the smallest equivalence on $E_{G_1} \cup E_{G_2}$ such that $\varphi(x) \equiv \psi(x)$ for every element x of G_0.*

An equivalence class of nodes is called valid *if it does not contain the image of a node x for which $\varphi(x)$ or $\psi(x)$ are undefined. Similarly a class of edges is* valid *if the analogous condition holds and furthermore all nodes adjacent to these edges are contained in valid equivalence classes.*

Then the pushout G_3 of φ and ψ consists of all valid equivalence classes $[x]_\equiv$ as nodes and edges, where $l_{G_3}([e]_\equiv) = l_{G_i}(e)$ and $c_{G_3}([e]_\equiv) = [v_1]_\equiv \dots [v_k]_\equiv$ if $e \in E_{G_i}$ and $c_{G_i}(e) = v_1 \dots v_k$.

Proof. First, note that G_3 is a well-defined graph due to the removal of dangling edges.

Now let $\psi': G_1 \rightharpoonup G_3$, $\varphi': G_2 \rightharpoonup G_3$ be the morphisms that map every item x of G_1 respectively G_2 to its equivalence class $[x]_\equiv$ in G_3 if this class is valid. It is easy to show that $\psi' \circ \varphi = \varphi' \circ \psi$, i.e., the diagram commutes.

Now let G_3' with morphisms $\psi'': G_1 \rightharpoonup G_3'$, $\varphi'': G_2 \rightharpoonup G_3'$ be another pair of morphisms with $\psi'' \circ \varphi = \varphi'' \circ \psi$. We have to show that there exists a unique morphism $\eta: G_3 \rightharpoonup G_3$ with $\eta \circ \psi' = \psi''$ and $\eta \circ \varphi' = \varphi''$.

Define η as follows: let $[x]_\equiv$ be an equivalence class of G_3. Whenever x is an element of G_1 define $\eta([x]_\equiv) = \psi''(x)$. Otherwise define $\eta([x]_\equiv) = \varphi''(x)$. It can be shown that η is well-defined.

The most complicated part of the proof is to show that the two triangles commute. Take an item x_1 of G_1: if $\psi'(x_1)$ is defined, then by definition $\eta(\psi'(x_1)) = \eta([x_1]_\equiv) = \psi''(x_1)$. Now assume that $\psi'(x_1)$ is undefined, which implies that the equivalence class of x_1 is invalid. That is, there exists a sequence $y_1, \dots, y_n$ of elements of G_0 such that $x_1 = \varphi(y_1)$, $\psi(y_1) = \psi(y_2)$, $\varphi(y_2) = \varphi(y_3)$, …, and either $\varphi(y_n)$ or $\psi(y_n)$ is undefined. Assume without loss of generality that $\varphi(y_n)$ is undefined. By post-composing every equation with either ψ'' or φ'' and taking commutativity into account we obtain that $\psi''(x_1) = \psi''(\varphi(y_1)) = \varphi''(\psi(y_1)) = \varphi''(\psi(y_2)) = \dots = \psi''(\varphi(y_n))$, which is undefined. Hence we conclude that $\psi''(x_1)$ is also undefined.

And due to its construction η is the unique morphism that makes the triangles commute. □

Lemma 21. *The class of minor morphisms is closed under composition.*

Proof. Let $\mu\colon G \mapsto H$, $\mu'\colon H \mapsto J$ be two minor morphisms. Obviously the composition $\mu' \circ \mu$ is surjective and injective on edges. It remains to show that the third property is also satisfied.

So let v, w be two nodes of G with $\mu'(\mu(v)) = \mu'(\mu(w)) = z$. Since μ' is a minor morphism there exists a path between $\mu(v)$ and $\mu(w)$ in H consisting of edges $e'_1, \dots, e'_n$, where e'_i is adjacent to nodes v'_i, w'_i. Furthermore $\mu(v) = v'_1$, $w'_i = v'_{i+1}$, $w'_n = \mu(w)$, all nodes are mapped to z by μ' and the image of all edges is undefined.

The morphism μ is surjective, hence there exist edges $e_1, \dots, e_n$ in G such that $\mu(e_i) = e'_i$, where e_i is adjacent to nodes v_i, w_i with $\mu(v_i) = v'_i$ and $\mu(w_i) = w'_i$. Hence v_i and w_{i+1} are connected by a path $f^i_1, \dots, f^i_{m_i}$. Furthermore, since $\mu(v_1) = v'_1 = \mu(v)$ there exists a path $f^0_1, \dots, f^0_{m_0}$ from v to v_1 and analogously a path $f^{n+1}_1, \dots, f^{n+1}_{m_{n+1}}$ from w_n to w. Also, the image of all these edges under μ is undefined.

So the combined path $f^0_1, \dots, f^0_{m_0}, e_1, f^1_1, \dots, f^n_{m_n}, e_n, f^{n+1}_1, \dots, f^{n+1}_{m_{n+1}}$ connects v and w and it satisfies all the requirements of Definition 12. □

Lemma 13. *$\hat{G}$ is a minor of G iff there exists a minor morphism $\mu : G \mapsto \hat{G}$.*

Proof. If $\hat{G}$ is a minor of G, then $\hat{G}$ can be obtained from G by deleting edges and isolated nodes, and contracting edges as specified in Definition 12. Clearly each of these operations can be separately specified by a minor morphism. If such operations are applied repeatedly the result follows from the fact that minor morphisms are closed under composition.

Conversely, let $\mu : G \mapsto \hat{G}$ be a minor morphism. Now perform the following operations on G. First, determine all nodes in $\hat{G}$ which have more than one preimage under μ. Since all preimages have to be connected by paths in G, where μ is undefined on the edges in the paths, we can contract all such edges, resulting in a graph G' with a minor morphism $G \mapsto G'$, where all nodes in a preimage have been merged. Afterwards, if an edge in G has no image under μ, then we can delete it from G. If a node has no image under μ, then, since μ is a morphism, it is clear that all edges adjacent to μ also do not have an image under μ. Hence, we can delete these edges from G, leaving us with an isolated node, which can then be deleted. Continuing this process we will obtain $\hat{G}$, and since we have restricted ourselves only to the "allowed" operations, it is clear that $\hat{G}$ is a minor of G. □

Proposition 22. *The minor ordering is a well-quasi order.*

Proof. The result is a corollary of the results in [8]. Assume that we have a sequence $G_1, G_2, G_3, \dots$ of graphs. We will use Theorem (1.6) from [8] that requires a sequence of hypergraphs where each edge is connected to a sequence of nodes which are all distinct and for which we have a well-quasi order on the label set.

In order to make sure that the nodes attached to an edge are all distinct, we transform graphs as follows: let Λ be the label alphabet and for each label $\ell \in \Lambda$ with $ar(\ell) = k$ we enumerate all partitions on the set $\{1, \dots, k\}$. For each such partition we fix an arbitrary order on the equivalence classes. The new label set Λ' now consists of pairs $(\ell, E_1 \dots E_n)$ where $E_1 \dots E_n$ is one of the chosen sequences of equivalence classes. We set $ar((\ell, E_1 \dots E_n)) = n$. Now transform a graph G into a graph G' by replacing every edge with label ℓ of arity k where $n = |\{v \in V_G \mid v \text{ adjacent to } e\}|$ by a corresponding edge e' with label $(\ell, E_1 \dots E_n)$. Here $E_1, \dots, E_n$ are the equivalence classes induced by the equivalence $i \equiv j \iff [c_G(e)]_i = [c_G(e)]_j$. The new edge e' is attached to a node sequence $v'_1 \dots v'_n$, where $v'_i = [c_G(e)]_j$ for an arbitrary index $j \in E_i$. Note that two graphs G, H are isomorphic if and only if their transformed graphs G', H' are isomorphic.

Concerning the second requirement (well-quasi order on the labels): since we have only finitely many labels in Λ the set Λ' is finite as well and we can choose the identity as well-quasi order.

We now consider the transformed sequence $G'_1, G'_2, G'_3, \ldots$. According to Proposition (1.6) there exists indices $i < j$ such that there is a collapse of G'_j to G'_i. More precisely, there exists a function η with domain $V_{G'_i} \cup E_{G'_i}$ such that:

1. $\eta(v)$ is a non-empty connected subgraph of $K_{V_{G'_j}}$ (where K_V is the undirected complete graph on the node set V) and the graphs $\eta(u), \eta(v)$ are pairwise disjoint for distinct $u, v \in V_{G'_i}$.
2. $\eta(e) \in E_{G'_j}$ for all $e \in E_{G'_i}$ and η is injective on edges and label-preserving.
3. For $e \in E_{G'_i}$ if $c_{G'_i}(e) = v_1 \ldots v_n$, then $c_{G'_j}(\eta(e)) = u_1 \ldots u_n$ and u_i is contained in the subgraph $\eta(v_i)$ for every $i \in \{1, \ldots, n\}$.
4. For each $v \in V_{G'_i}$ and each (undirected) edge f in $\eta(v)$, connecting $x, y \in V_{G'_j}$, there exists an edge $e \in E_{G'_j}$ which is adjacent to x, y. Furthermore e is not in the image of η. (The latter can be assumed since our label alphabet is finite and each label is associated with an arity. Hence every edge is bounded, i.e., has a finite neighbourhood.)

Now define a minor morphism $\mu: G'_j \mapsto G'_i$ as follows:

- An edge e' of G'_j is mapped to e in G'_i whenever $\eta(e) = e'$. If no such edge exists $\mu(e)$ is undefined. This is well-defined since η is injective on edges (Condition 2). Furthermore μ is injective and surjective on edges and preserves labels.
- Whenever a node v' of G'_j is contained in a subgraph $\eta(v)$ we map v' to v. Otherwise $\mu(v')$ is undefined. Clearly due to Condition 1 μ obtained in this way is well-defined and surjective on nodes.

We next verify that μ is a partial morphism. Assume that $\mu(e') = e$ with $c_{G'_i}(e) = v_1 \ldots v_n$ and $c_{G'_j}(e') = u_1 \ldots u_n$: then $\eta(e) = e'$ and u_i is contained in $\eta(v_i)$ (Condition 3). Hence u_i is mapped to v_i, which means that the image of all nodes is defined and the map μ is structure-preserving.

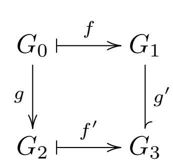
Finally assume that $\mu(v') = \mu(w') = z$. This means that v', w' are both contained in $\eta(z)$. Since the subgraph $\eta(z)$ is connected there exists a path from v' to w' in $\eta(z)$. Let us denote this path by $v' = v'_0, f_1, v'_1, \ldots, v'_{n-1}, f_n, v'_n$. By Condition 4 we can require that there exists edges $e'_k \in E_{G'_j}$, which are not in the image of η and adjacent to v'_{k-1}, v'_k. This implies the existence of a path $v'_0, e'_1, v'_1, \ldots, v'_{n-1}, e'_n, v'_n$ such that $\mu(e'_k)$ is undefined and $\mu(v'_k) = z$ (since all nodes $v'_0, \ldots, v'_n$ are within the subgraph $\eta(z)$ and are hence mapped to z).

This means that $\mu: G'_j \mapsto G'_i$ is a minor morphism. It is now left to transform G'_i, G'_j back to G_i, G_j. It is easy to check that there exists a minor morphism $\mu: G_i \mapsto G_j$.

Note that the collapse relation of [8] is finer than the minor ordering of Definition 12. Especially a minor morphism might map straight edges to loops, which is not allowed in the collapse. However, this only means that we might "miss" some pairs of related graphs, but we will always find one.

□

Lemma 14. *Pushouts preserve minor morphisms in the following sense: If $f : G_0 \mapsto G_1$ is a minor morphism and $g : G_0 \to G_2$ is total, then the morphism f' in the pushout diagram below is a minor morphism.*



Proof. Let G_3 be the pushout of G_0 along f and g. Let $v, w \in V_{G_2}$ be two nodes that are mapped to the same node $z \in V_{G_3}$ via the morphism $f' : G_2 \to G_3$. But this means that v and w are in the same equivalence class, and thus necessarily have pre-images v' and w' in G_0.

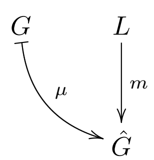
Now, since they are in the same equivalence class, there exists a sequence of nodes $y_1, y_2, \ldots y_n \in V_{G_0}$ with $y_1 = v'$ and $y_n = w'$ such that $f(y_1) = f(y_2), g(y_2) = g(y_3), \ldots$ Since f is a minor morphism there exists a path (in G_0) from y_i to $y_{i+1}, i \in \{1, 3, \ldots\}$ such that all nodes on the path are mapped to $f(y_i)$.

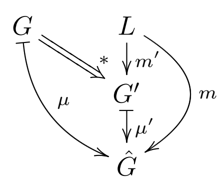
Since g is total, there also exists a path from $g(y_i)$ to $g(y_{i+1}), i \in \{1, 3, \ldots\}$ in G_2. Now, due to commutativity, all nodes on such a path (in G_2) will be mapped to the same node in G_3. Also due to commutativity, the images of all the edges in this path are undefined, since the equivalence class is not valid (due to f being a minor morphism). Further, since $g(y_i) = g(y_{i+1})$ for $i \in \{2, 4, \ldots\}$, there is a path from $g(y_1) = v$ to $g(y_n) = w$ such that all nodes on that path are mapped to the same node, $z \in V_{G_3}$, and none of the edges in this path lie in the domain of f'. Also, surjectivity (on nodes and edges) and injectivity (on edges) is preserved by the pushout construction.

Thus, f' is a minor morphism. □

Note that in the proof above we did not require g' to be total, i.e., the lemma also holds in the presence of deletion/preservation conflicts. It is however necessary to demand that g is total.

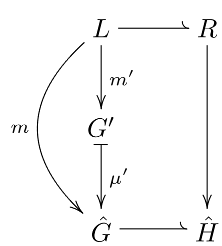
Proposition 15 (GSTS as WSTS). *Let $\mathcal{R}$ be a GTS that satisfies the following condition: For every rule $(r: L \rightharpoonup R) \in \mathcal{R}$, every minor morphism $\mu: G \mapsto \hat{G}$ and every match $m: L \to \hat{G}$ (see diagram on the left) there exists a graph G' such that $G \Rightarrow^* G'$, there is a minor morphism $\mu': G' \mapsto \hat{G}$ and there exists a match $m': L \to G'$ such that $m = \mu' \circ m'$ (see commuting diagram below on the right). Then $\mathcal{R}$ is a WSTS.*

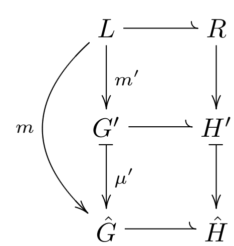




Proof. Let $\hat{G}$ be a graph that is rewritten to $\hat{H}$ via $r: L \rightharpoonup R$ and an injective match $m: L \to \hat{G}$. Assume that $\hat{G} \leq G$. That is, according to Lemma 13 there exists a minor morphism $\mu: G \to \hat{G}$.

This means that m can be factored as specified in the assumption, leading to the diagram below on the left where the square is a pushout. Note also that whenever m is conflict-free, then m' must be conflict-free.



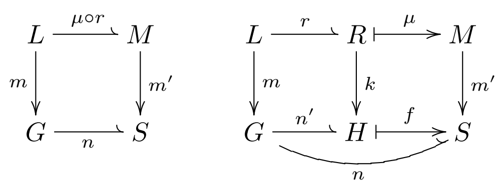


The pushout square can be split into two pushouts via standard pushout splitting (see diagram on the right) and from a variation of Lemma 14 it follows that the resulting morphism $H' \mapsto \hat{H}$ is a minor morphism. (Note that the minor morphism $\mu'\colon G' \mapsto \hat{G}$ does not contract any edges on which $G' \rightharpoonup H'$ is undefined, hence those contractions can also be performed in H'.)

Hence G can first be rewritten to G' and then G' is rewritten in one step to H', where $\hat{H} \leq H'$. This concludes the proof. □

Theorem 16. *The procedure $pb(S)$ computes a finite subset of $Pred(\uparrow S)$.*

Proof. Let G be any graph that has been obtained in an iteration of the procedure $pb(S)$. Then there must be a Rule $r : L \rightharpoonup R$ and a minor M of R (identified by the minor morphism μ), such that the diagram on the left below is a pushout.



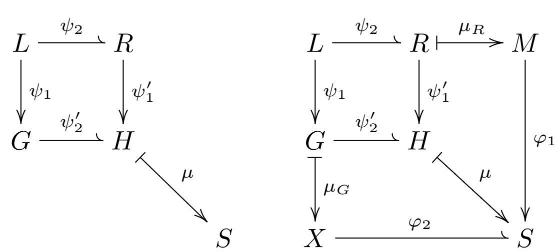
Now, construct the pushout H of the morphisms r and m. Then, there exists a unique morphism $f : H \rightharpoonup S$ (see diagram on the right).

This diagram commutes. Further, the outer rectangle is a pushout and the inner left square is a pushout (by construction). This implies that the inner right square is also a pushout.

Then, since m is conflict-free wrt. r (by construction), the rule is applicable and G rewrites to H. Furthermore k is total and hence from Lemma 14 we know that f is a minor morphism. Therefore $H \geq S$ and $G \in Pred(\uparrow S)$, for each $G \in pb(S)$. □

Lemma 17. *Let $\psi_1\colon L \to G$ be total and conflict-free wrt. ψ_2. If the diagram below on the left is a pushout and $\mu\colon H \mapsto S$ a minor morphism, then there exist minors M and X of R and G respectively, such that*

1. *the diagram below on the right commutes and the outer square is a pushout.*
2. *the morphisms $\mu_G \circ \psi_1 : L \to X$ and $\varphi_1 : M \to S$ are total and $\mu_G \circ \psi_1$ is conflict-free wrt. ψ_2.*



Proof. From R, construct a minor M (and simultaneously a minor morphism μ_R) as follows:

1. First, let M be simply a copy of R
2. For $e \in E_R$, if the image of e in H under ψ'_1 is contracted to construct S, then contract the corresponding edge in M. In this case e is undefined under μ_R and its adjacent nodes are mapped to the merged node in M. If e is deleted (without contracting the nodes), delete it in M as well, and leave e undefined under μ_R.

3. Now, let $v \in V_R$ be such that its image in H is deleted in constructing S. This implies that the image of v in H is either an isolated node, or all its adjacent edges were deleted. So since we deleted corresponding edges in M, we can safely delete v in M, and leave it undefined under μ_R. (Note that it is not possible for R to have an edge that is not mapped to an edge in H, since $L \to G$ is total and conflict-free and hence ψ'_1 must be total).

Now, M is a minor of R, because the construction involved only the "allowed" operations. Further, due to its construction, μ_R is a minor morphism.

Perform a similar construction for X, with one difference: For $x \in G$ such that x has a pre-image in L, do not contract/delete it in X, even if x had an image in H that was contracted/deleted in constructing S. (The intuition for this is that it is enough for an item to be contracted/deleted by one of the minor morphisms for it to be contracted/deleted in the pushout graph.) The rest of the construction is as before. Again, X is a minor of G and $\mu_G : G \mapsto X$ is a minor morphism. Further $\mu_G \circ \psi_1$ is total since ψ_1 is total and μ_G is defined for all elements with a pre-image in L.

Also, $\mu_G \circ \psi_1$ is conflict-free with respect to ψ_2. To see this suppose there exist nodes $v_1, v_2 \in L$ such that $(\mu_G \circ \psi_1)(x_1) = (\mu_G \circ \psi_1)(x_2)$. Whenever x_1, x_2 are edges, then $\psi_1(x_1) = \psi_1(x_2)$, since μ_G does not merge edges. In this case by assumption ψ_2 is either undefined on both or defined on both. A similar argument applies whenever x_1, x_2 are nodes and $\psi_1(x_1) = \psi_1(x_2)$. So now assume that x_1, x_2 are nodes and $y_1 = \psi_1(x_1) \neq \psi_1(x_2) = y_2$. Then, $\mu_G(y_1) = \mu_G(y_2)$ implies that y_1 and y_2 are nodes and have distinct images in H with a path connecting them which is contracted while constructing S. Hence, $\psi_2(x_1)$ and $\psi_2(x_2)$ are both defined and distinct. Thus there cannot be a deletion/preservation conflict.

Now, we construct the morphisms $\varphi_1 : M \to S$ and $\varphi_2 : X \rightharpoonup S$ (see diagram above). For any $x \in R$ We define φ_1 as:

$$\varphi_1(\mu_R(x)) \stackrel{def}{=} \mu(\psi'_1(x))$$

To see that this is valid, first note that if $\mu_R(x)$ is undefined, then $\mu(\psi'_1(x))$ will also be undefined because of the construction of μ_R. Then, if there exist x_1, x_2 such that $\mu_R(x_1) = \mu_R(x_2)$, then x_1 and x_2 must be nodes and not edges, because μ_R is injective on edges. Secondly, we must have $\mu(\psi'_1(x_1)) = \mu(\psi'_1(x_2))$. Thus the above definition is valid.

Further, ψ'_1 is total (since ψ_1 is total and conflict-free), $\mu(\psi'_1(x))$ is undefined iff μ is undefined at $\psi'_1(x)$, and in that case $\mu_R(x)$ will also be undefined. This, combined with the fact that μ_R is surjective, implies that φ_1 is total. Also, the relevant part of the diagram commutes, due to the definition of φ_1.

Similarly, φ_2 is defined as:

$$\varphi_2(\mu_G(x)) \stackrel{def}{=} \mu(\psi'_2(x))$$

By essentially the same argument as in the case of μ_R, we can show that this definition is valid and the relevant part of the above diagram commutes. However, in this case φ_2 may be partial, because for a node x with a pre-image in L, $\psi'_2(\mu(x))$ may undefined but $\mu_G(x)$ will still be defined. For such a node, φ_2 will be left undefined.

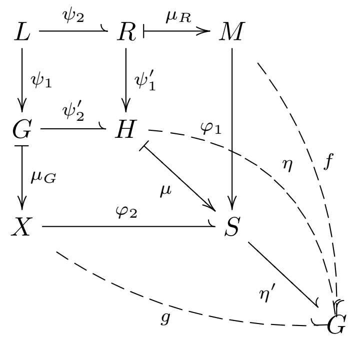
Furthermore it can be straightforwardly shown that φ_1, φ_2 satisfy all properties of a morphism. And finally, since each of the parts of the above diagram commute, the diagram as a whole also commutes.

Now, let there be some graph G and two morphisms $f : M \to G$ and $g : X \to G$, such that $(f \circ \mu_R) \circ \psi_2 = (g \circ \mu_G) \circ \psi_1$, then, since H is a pushout, there exists $\eta : H \to G$ such that η is the unique morphism with $f \circ \mu_R = \eta \circ \psi'_1$ and $g \circ \mu_G = \eta \circ \psi'_2$.

For $x \in H$, define a morphism $\eta' : S \to G$ as follows:

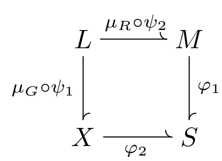
$$\eta'(\mu(x)) \overset{def}{=} \eta(x)$$

Since μ is surjective, every element $y \in S$ has a pre-image $x \in H$, hence η' can in principle be defined for every such y by the above definition. The current situation is depicted below:



It is left to show that η' is well-defined and unique. Now, if there exist x_1, x_2 such that $\mu(x_1) = \mu(x_2)$, then x_1 and x_2 must be nodes (since μ is a minor morphism), and further, there must be a path connecting them, such that all nodes on this path are also mapped to the same node. Then, if $\eta(x_1) \neq \eta(x_2)$, there exist y_1, y_2 which lie on this path from x_1 to x_2, such that they are adjacent, and $\mu(y_1) = \mu(y_2)$ but $\eta(y_1) \neq \eta(y_2)$. Let e be the edge connecting them. It must have a pre-image in either R or G (or both). Suppose e has a pre-image e' in R with the pre-images of y_1 and y_2 being y'_1 and y'_2 respectively. Then, if e is contracted in S it holds that e' is contracted in M, and hence $\mu_R(y'_1) = \mu_R(y'_2)$. But then, $f \circ \mu_R = \eta \circ \psi'_1$ implies that $\eta(y_1) = \eta(y_2)$, which leads us to a contradiction. Now, suppose e has a pre-image e'' in G instead of R. If e'' does not have a pre-image in L, then we arrive at a contradiction by a similar argument as before. On the other hand, if e'' has a pre-image in L, then e must have a pre-image in R (since H is a pushout), hence the previous argument applies. Hence, such x_1, x_2 cannot exist, and η' is well-defined.

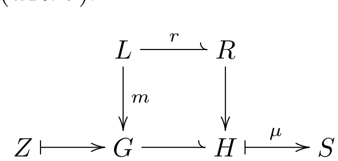
This gives us an $\eta' : S \to G$ such that $\eta' \circ \mu = \eta$. This implies $f \circ \mu_R = (\eta \circ \mu) \circ \psi'_1$ and $(g \circ \mu_G) = (\eta \circ \mu) \circ \psi'_2$. Now η and therefore $\eta' \circ \mu$ is the *unique* morphism with this property. Since μ is fixed and surjective, this means that η' is the unique morphism such that $f = \varphi_1 \circ \eta'$ and $g = \varphi_2 \circ \eta'$. Thus, the following diagram is a pushout.



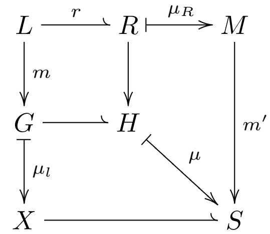
Further, φ_1 and $\mu_G \circ \psi_1$ are both total, and $\mu_G \circ \psi_1$ is conflict-free with respect to ψ_2. ☐

Theorem 18. *The set generated by $pb(S)$ is a pred-basis of S.*

Proof. Let Z be a member of $\uparrow Pred(\uparrow S)$. Then, there exists $G \in Pred(\uparrow S)$ and G is a minor of Z. Hence there is a rule $r : L \rightharpoonup R$ which can be applied to G to get some $H \in \uparrow S$, i.e., S is a minor of H. Hence we have the following situation, where m is a conflict-free match (wrt. r):



Now, by Lemma 17 we can construct minors of R and G (denoted by M and X) such that the following is a pushout (with $\mu_l \circ m$ total and conflict-free wrt. r and m' total):



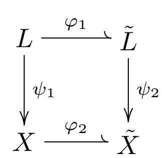
Hence X is a pushout complement for which the match is conflict-free wrt. r. The procedure $pb(S)$ as described above computes the minimal pushout complements with such properties. Hence there exists $X' \in pb(S)$ with $X' \leq X \leq G \leq Z$.

This entire argument holds for *every* such Z, which implies that $pb(s)$ generates a suitable pred-basis. □

Lemma 19. *Let L and $\tilde{L}$ be graphs, $\varphi_1 : L \rightharpoonup \tilde{L}$ be an inverse injection, and $\psi_2 : \tilde{L} \to \tilde{X}$ be a total morphism. Now construct a specific pushout complement X' with morphisms $\psi_1' \colon L \to X'$, $\varphi_2' \colon X' \rightharpoonup \tilde{X}$ as follows:*

1. *Take a copy of the graph $\tilde{X}$, and let ψ_1' be $\psi_2 \circ \varphi_1$. The morphism φ_2' is the identity.*
2. *Let Y be the set of elements of L the image of which is undefined under φ_1. Add a copy of Y to this copy of $\tilde{X}$, and extend ψ_1' by mapping Y into this set. Furthermore φ_2' is undefined on all elements of the copy of Y.*
3. *Now merge these new elements (originally contained in Y) in all possible combinations, i.e., factor through all appropriate[8] equivalences. The morphisms ψ_1' and φ_2' are modified accordingly.*

The set of graphs obtained in this way is denoted by $\mathcal{P}$. Each element X' of $\mathcal{P}$ is a pushout complement of φ_1, ψ_2 and the corresponding morphisms $\psi_1' \colon L \to X'$ are total. Any other pushout complement X where $\psi_1 : L \to X$ is total (see diagram on the right) has some graph $X' \in \mathcal{P}$ as a minor.



Finally, if $\psi_1 \colon L \to X$ is conflict-free wrt. to a rule $r : L \rightharpoonup R$, then there exists a pushout complement $X' \in \mathcal{P}$ with $\psi_1' \colon L \to X'$ conflict-free wrt. r, such that $X' \leq X$.

Proof. First, it is clear that the graphs produced by the above construction will in fact be pushout complements. Also, ψ_1' will be total.

Let X be any pushout complement with ψ_1 total. Since φ_1 is an inverse injection and ψ_1, ψ_2 are total, φ_2 must also be an inverse injection. So the entire graph $\tilde{X}$ has a pre-image in X, or we can say that X contains an "exact copy" of $\tilde{X}$.

Now, if v is a node in X, then either it belongs to the copy of $\tilde{X}$, or if not, it has a pre-image in Y under ψ_1. Similarly for an edge e in X, either e is in the copy of $\tilde{X}$, or it has at least one endpoint with a pre-image in x.

This means that X must be of the following form: it must contain an exact copy of $\tilde{X}$, and images of all elements in Y (since ψ_1 must be total). An element in the copy of $\tilde{X}$ cannot have a preimage in Y under ψ_1 since otherwise ψ_2 would not be total. But the elements of Y may be be merged with each other in any fashion. It cannot contain any other nodes, but it may contain any number of additional edges, so long as at least one endpoint of these edges lies outside the copy of $\tilde{X}$.

[8] Here "appropriate" means that whenever two edges are in the equivalence relation, all their adjacent nodes must be pairwise equivalent.

For any such X, we can delete these extra edges, and we will obtain a minor X' of X, which is still a pushout complement. Since this minor contains only a copy of $\tilde{X}$ and extra elements from Y, it can be obtained by the above construction. Thus the above construction allows us to compute all the minimal pushout complements.

Finally, whenever ψ_1 is conflict-free wrt. r, we can find a matching X' for which $\psi_1'\colon L \to X'$ is also conflict-free, since the minor morphism $X \mapsto X'$ only deletes edges, but never contracts them. So ψ_1' must be conflict-free whenever ψ_1 is. □

Proposition 20. *Let $r\colon L \rightharpoonup R$ be a fixed rule. Furthermore let L, M and S be graphs, with a partial morphism $\varphi_1 : L \rightharpoonup M$ and a total morphism $\psi_2 : M \to S$. Then, if we apply the following procedure we only construct pushout complements X' of φ_1, ψ_2 and any other pushout complement X (with $\psi_1\colon L \to X$ where ψ_1 is total and conflict-free wrt. r) has one of them as a minor.*

1. *Split φ_1 into two morphisms as follows: let $\varphi_1' : L \rightharpoonup dom(\varphi_1)$ be an inverse injection and let $\varphi_1'' : dom(\varphi_1) \to M$ be total.*
2. *Now consider the total morphisms $\varphi_1'' : dom(\varphi_1) \to M$, and $\psi_2 : M \to S$. Construct all their pushout complements as usual for total morphisms.*[9]
3. *Let $\tilde{X}$ be any such pushout complement with $\eta\colon dom(\varphi_1) \to \tilde{X}$.*
4. *For φ_1', η use the construction of Lemma 19 in order to obtain the minimal pushout complements X' (with total and conflict-free ψ_1').*
5. *Finally, from all such pushout complements X' take the minimal ones.*

The situation is depicted in the diagram below.

$$\begin{array}{ccccc}
L & \xrightharpoonup{\varphi_1'} & dom(\varphi_1) & \xrightarrow{\varphi_1''} & M \\
\downarrow{\scriptstyle \psi_1'} & & \downarrow{\scriptstyle \eta} & & \downarrow{\scriptstyle \psi_2} \\
X' & \rightharpoonup & \tilde{X} & \longrightarrow & S
\end{array}$$

Proof. Since both squares in the diagram above are pushouts (the left square by Lemma 19, the right square by construction), the outer square must be a pushout and hence X' is a pushout complement of φ_1, ψ_2.

Now take any pushout complement X such that $\psi_1\colon L \to X$ is total and conflict-free wrt. r. The situation is as depicted below on the left:

$$\begin{array}{ccccc}
L & \xrightharpoonup{\varphi_1'} & dom(\varphi_1) & \xrightarrow{\varphi_1''} & M \\
\downarrow{\scriptstyle \psi_1} & & & & \downarrow{\scriptstyle \psi_2} \\
X & & \rightharpoonup & & S
\end{array}
\qquad
\begin{array}{ccccc}
L & \xrightharpoonup{\varphi_1'} & dom(\varphi_1) & \xrightarrow{\varphi_1''} & M \\
\downarrow{\scriptstyle \psi_1} & & \downarrow{\scriptstyle \eta} & & \downarrow{\scriptstyle \psi_2} \\
X & \rightharpoonup & \tilde{X} & \longrightarrow & S
\end{array}$$

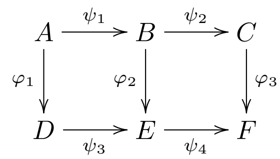
As shown above on the right we can now split the pushout leading to two pushout diagrams, where $\tilde{X}$ is one of the pushout complements of φ_1'', ψ_2 computed above. Hence by Lemma 19 the graph X must have one of the constructed graphs X' as a minor. □

B Pushout Properties

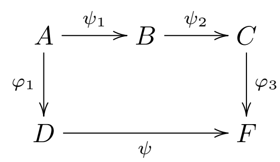
We shortly summarize several properties of pushouts which hold in any category and which are used in the proofs.

[9] We do not describe this construction here, but it is well-known that there are only finitely many such pushout complements and that they can be constructed effectively.

If in the diagram below both squares (consisting of morphisms $\varphi_1, \psi_1, \psi_3, \varphi_2$ and $\varphi_2, \psi_2, \psi_4, \varphi_3$) are pushouts, then the outer rectangle (consisting of $\varphi_1, \psi_2 \circ \psi_1, \varphi_3, \psi_4 \circ \psi_3, \varphi_3$) is also a pushout.



If the rectangle below (consisting of morphisms $\varphi_1, \psi_2 \circ \psi_1, \psi, \varphi_3$) is a pushout, then there is a unique way to factorize ψ into $\psi = \psi_3 \circ \psi_4$ such that the rectangle splits into two pushout squares as shown above.



Finally, an important tool in proofs is the so-called "mediating morphism" which uniquely connects a pushout with another commuting square. In Definition 8 this morphism is denoted by η.