Electronic Communications of the EASST
Volume 4 (2006)

Proceedings of the
Second International Workshop on
Graph and Model Transformation
(GraMoT 2006)

Termination of Algebraic Rewriting with Inhibitors

Paolo Bottoni, Kathrin Hoffmann, Francesco Parisi-Presicce

13 pages

Guest Editors: Gabor Karsai, Gabriele Taentzer
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Termination of Algebraic Rewriting with Inhibitors

Paolo Bottoni[1], **Kathrin Hoffmann**[2], **Francesco Parisi-Presicce**[1]

[1] (bottoni,parisi)@di.uniroma1.it Dipartimento di Informatica
Università di Roma "La Sapienza", Italy
[2] hoffmann@cs.tu-berlin.de Department of Software Engineering and Theoretical Computer Science
Technical University of Berlin, Germany

Abstract: We proceed with the study of termination properties in the double pushout approach to algebraic rewriting, and show a concrete termination criterion for rewriting systems with inhibitors. Inhibitors prevent elements in an algebra to participate in rule matches, so that termination depends only on whether new possibilities for matches are created. The notion of inhibitor can be extended to considering different levels of inhibition, by which the ability of an element to participate in a match is progressively reduced. We illustrate the approach by considering some application contexts in model transformation.

Keywords: termination, double pushout approach, inhibitors

1 Introduction

When defining model transformation through rewriting systems, it is often the case that rules must be applied as long as possible, so that it becomes important to reason on the termination of such processes. While termination of graph transformations is undecidable in general [Plu98], the termination of specific systems can be decided with different techniques. In a previous paper [BHPT05], we have identified an abstract notion of termination criterion for high-level replacement (HLR) systems, i.e. algebraic rewriting systems operating on objects and morphisms in adhesive HLR categories [EHPP04], in which rewriting is steered by control expressions.

Extending such a notion to rules with negative application conditions (NACs) meets with several difficulties. However, excluding matches on some parts of the host objects can also be realized through other techniques than NACs. For example, in semi-Thue systems, an inhibitor is an element which cannot be involved in a rewriting [McN97], forcing matches to occur to either its left or its right, but not across it.

We place this notion in an algebraic setting, by seeing an inhibitor as an element of a particular sort with which elements in other sorts can be tagged so that they cannot be used in any match for any rule. We show that by enriching an algebra with inhibitors, the counting of not inhibited elements acts, in the obtained algebra, as a concrete termination criterion. The main result of this paper is that termination can be proven for algebras with inhibitors. Moreover, we show how any totally ordered finite set can support a notion of partial inhibition, in which a tag for an object indicates a "level of inhibition" and the top element of the set rules definitively out the possibility for the tagged object to participate in a match. Also in this case termination can be proven. This

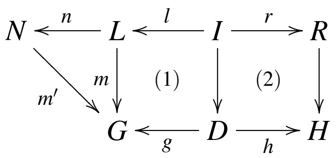

Figure 1: A DPO rule with negative application condition

can be further extended to the case of attributed rewriting systems, where attribute domains are totally ordered, and attribute value transformations have to comply with the ordering.

Paper organization. We revise some formal notions in Section 2 and investigate literature on termination and inhibitors in Section 3. Section 4 compares provability of termination for rules with NACs in graph and multiset rewriting. Section 5 shows how to introduce inhibitors in an algebraic setting. In Section 6 we state a concrete termination criterion for algebras with inhibitors and Section 7 extends the result to the case of inhibitory structures with more than two inhibition levels. Finally, Section 8 shows some applications of the concept, while Section 9 draws some conclusion and points to future work.

2 Formal background

Informally, a pushout in a category **CAT** is a gluing construction of two objects over a specific interface, while a pullback is dual to a pushout in the sense that a pullback construction extracts the common part of two objects. Let now **CAT** be a category with a distinguished morphism class $\mathscr{M}$, such that **CAT** has pushouts and pullbacks along $\mathscr{M}$-morphisms, i.e. if a morphism is in $\mathscr{M}$, then its opposite is in $\mathscr{M}$, too, and $\mathscr{M}$-morphisms are closed under pushouts and pullbacks (the notation $(\mathbf{CAT}, \mathscr{M})$ will also be used).

A rule $p : L \xleftarrow{l} I \xrightarrow{r} R$ in **CAT** is given by two morphisms l and r of $\mathscr{M}$. $L(p)$ is the left-hand side and $R(p)$ the right-hand side of p. Given an object G and a rule $p : L \xleftarrow{l} I \xrightarrow{r} R$, a match of p to G is a morphism $m : L \to G$. A direct derivation d from G to H by p and match m, $d : G \Rightarrow_{p,m} H$, is given by a double pushout (DPO) (1) and (2) in Figure 1. In a rule with NACs, as shown in Figure 1, the morphism n is an injective total morphism, so that the rule is applicable only if match m cannot be extended to a match m' such that $m' \circ n = m$ [HHT96]. Several objects N_i, and associated morphisms n_i, can be coupled with one L, indicating that no extension of m should exist for any i.

The DPO approach has been largely used for graph transformation systems (GTS) allowing the study of notions like confluence, sequential or parallel independence, etc. and supporting several extensions, e.g. to typed and attributed graphs. High-level replacement (HLR) systems extended the approach to other algebraic structures, by considering only their categorical properties [EHKP91]. A HLR system in a category **CAT** consists of a set of rules $\mathscr{P}$ and may be extended by a set of finite input objects. Moreover, some form of control on rule applications can be exploited like layering and rule expressions [BHPT05].

Let $\mathscr{G}$ be the class of all objects in a category $(\mathbf{CAT}, \mathscr{M})$ and p be a rule with morphisms in $\mathscr{M}$. We say that p terminates on $G \in \mathscr{G}$ if $\forall H \in \mathscr{G}$, $(G \Longrightarrow_p^* H) \Rightarrow (\exists K$ such that: $H \Longrightarrow_p^* K$

$\wedge \nexists Z \neq K$ for which $K \Longrightarrow_p Z$). Rule p is said to terminate if $\forall G \in \mathscr{G}$, p terminates on G. A function $F : \mathscr{G} \rightarrow \mathscr{N}$ from objects to natural numbers is a termination criterion for $(\mathbf{CAT}, \mathscr{M})$ if for any two morphisms $a : C \rightarrow A$ and $b : C \rightarrow B$ in $\mathscr{M}$, the value $F(A +_C B)$ of the pushout object $A +_C B$ of a and b is given by $F(A +_C B) = F(A) + F(B) - F(C)$. Given a rule p, if there exists a function F which is a termination criterion for **CAT**, such that $F(L(p)) > F(R(p))$, then p terminates [BHPT05].

Adhesive HLR categories [EHPP04] are based on adhesive categories [LS04] and HLR systems. In contrast to HLR systems where properties are given in a more ad-hoc manner, adhesive HLR categories introduce a more simplified version of these properties; especially the so-called van Kampen-square property for pushouts along M-morphisms is sufficient to prove fundamental results such as Local Church-Rosser etc. In general, a van Kampen-square is a pushout which is stable under pullbacks in a commutative cube. Typical examples of adhesive HLR categories are the categories of sets and graphs.

3 Related work

The possibility of forbidding rule application, based on the presence or absence of some properties in the object to be rewritten has been treated in several formalisms. Graph transformations with negative application conditions have been long used [HHT96], but their termination properties started to be investigated only recently. In particular, Ehrig *et al.* proved termination of layered GTSs [EEL$^+$05] with NACs. Results for the general case are still missing[1]. The expressibility of the Post Correspondence Problem in the form of GTSs has been exploited by Plump to prove undecidability of termination and confluence for GTSs without NACs [Plu98, Plu05]. The decidability of termination for some specific GTSs has been used in an experimental setting to prove safety properties [KMP02].

The presence of a forbidding context for rule application has been exploited in different forms. In particular, context-free grammars have been equipped with several types of control. For example, conditional grammars allow the use of a rule only if the host string belongs to a certain regular language [Nav70]. In semi-conditional grammars, for each rule p, a regular language R defines the language of subwords that have to appear in the string for p to be applicable, while a different regular language Q defines the set of subwords that must not appear [Pau79, Pau85]. In a random context grammar, these sets are reduced to subsets of nonterminals [vdW71]. For grammars, termination is related to the problems of non-emptiness and finiteness of the generated language. The first three types of grammars produce type-0 languages if erasing productions are allowed, and context-sensitive languages otherwise. Membership, finiteness and emptiness of a language are all decidable problems for random context grammars, which instead have an expressive power not related to grammars in the Chomsky hierarchy.

Moving from grammars to general string rewriting systems, semi-Thue systems have rules of type $x \rightarrow y$, $x, y \in \Sigma^*$ being strings on an alphabet Σ. For these systems, termination is undecidable in general. McNaughton has proposed semi-Thue systems of one rule with inhibitors, where an inhibitor is a special symbol $\iota \notin \Sigma$, which cannot appear in x, but has to appear at least once

[1] The problem is only significant if a non-trivial NAC exists for each rule, which we assume throughout the rest of the paper.

in y [McN97]. In this case termination becomes decidable, in that inhibitors divide a string in fragments such that each rule match is constrained to occur on a single fragment.

In Petri nets, inhibitors are special arcs connecting places to transitions, such that the place must not contain the number of tokens indicated by the arc. The presence of inhibitors strictly increases the expressive power of nets, making problems such as reachability or liveness become undecidable [Hac76].

In multisets, differently from sets, several copies of the same element can appear. Rewriting simply replaces the elements in the antecedent with those in the consequent. In this framework, inhibitors have been introduced for P-Systems, a generalization of multiset rewriting where rewriting is restricted to occur within an isolated region. An inhibitor for a rule is a multiset of symbols, such that the rule cannot be applied in a region which contains this multiset. This allows the characterization of recursively enumerable languages of Parikh vectors, in a significantly more compact way than by traditional methods in the field of membrane computing, e.g. catalysts, priorities, target indications [BMPR02].

One can observe, that, with the exception of McNaughton's work, the use of forbidding contexts increases the expressive power of the underlying formalism (context-free grammars, Petri nets, P-Systems), thus making several problems undecidable, which were not so in the basic version of the formalism. The basic difference is that in the semi-Thue case inhibitors identify portions of the string to be rewritten which are definitely excluded from the possibility of a match, while in other cases they prevent the application of a specific rule, which can later become applicable again if the context changes. We leverage this concept to propose our notion of inhibitors as a way to prevent sorted elements to participate in a match, independently of the concrete structure to which they belong, but in a purely algebraic setting.

4 Examples: Rules with Negative Application Conditions

We discuss difficulties in the identification of termination criteria for DPO rules with NACs of a general form. We consider the case of nondeleting rules, i.e. in which an inclusion morphism $L \hookrightarrow R$ can be constructed coherently with the two inclusions $I \overset{l}{\hookrightarrow} L$, $I \overset{r}{\hookrightarrow} R$. For rules which cause some deletion, we can simply consider counting functions as termination criteria, disregarding the presence of NACs. In particular, even restricting N to be included in R, or viceversa, does not solve the problem for GTSs. We also discuss rewriting of multisets, i.e. empty graphs, and show how in this case rules are either trivially terminating, or their termination depends on the source multiset to which they are applied.

4.1 Graphs

Let $F : \mathscr{G} \to \mathscr{N}$ be a termination criterion. Inclusion of N into R does not provide an immediate criterion for termination, at least under the assumption that the morphisms $I \overset{l}{\hookrightarrow} L$, $L \overset{n}{\hookrightarrow} N$, $I \overset{r}{\hookrightarrow} R$ exist. Indeed, the left hand side of Figure 2 shows a terminating rule, while the right hand side of Figure 2 a case of nontermination. In both cases, due to the relations between L, N and R, we must have, for any choice of F, $F(L) \leq F(N) \leq F(R)$.

The case in which the inclusion runs from R to N should produce terminating rule applications

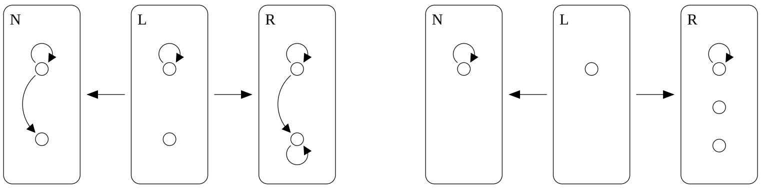

Figure 2: Rules with $N \subseteq R$.

when the repeated creation of matches for R comes to saturate the possibility of matches for L not forbidden by N. However, while this is true for the rule in the left hand side of Figure 3, no match is ruled out by its application for the right hand side of Figure 3. In this case, we have $F(L) \leq F(R) \leq F(N)$.

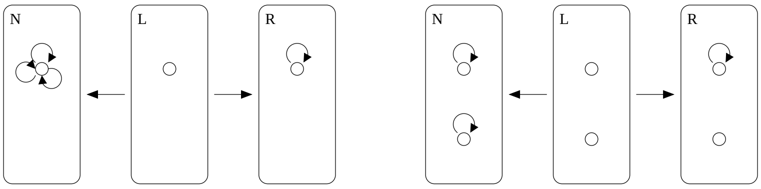

Figure 3: Rules with $R \subseteq N$.

Finally, we consider the case where no immediate relation can be identified between R and N (still maintaining $L \hookrightarrow N$ and $L \hookrightarrow R$). Again, the left hand side of Figure 4 shows a terminating rule (the rule can be repeated at most until all nodes in the graph have two loops attached). The right hand side of Figure 4 shows a rule which may terminate or not, depending on the presence in the original host graph of at least two nodes. In both cases two new edges and zero new nodes are created.

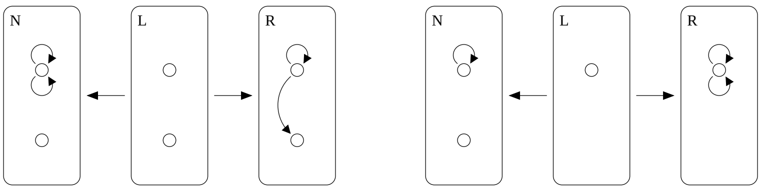

Figure 4: Rules with $N \not\subseteq R$ and $R \not\subseteq N$.

4.2 Multisets

Empty graphs, i.e. those in which each node is isolated, can be modelled as multisets. A multiset M on an alphabet Σ is defined by a membership function $f_M : \Sigma \rightarrow \mathcal{N}$ such that $f_M(x)$ indicates the number of times a copy of x appears in M, $\forall x \in \Sigma$. Let A and B two multisets on the same alphabet Σ defined by two functions f_A and f_B, respectively. The usual notions of union, difference and intersection of sets are redefined by suitable functions $f_{AopB} : \Sigma \rightarrow \mathcal{N}$. In particular,

$\forall x \in \Sigma$: 1) $f_{A\cup B}(x) = f_A(x) + f_B(x)$. 2) $f_{A\setminus B}(x) = f_A(x) \ominus f_B(x)$, where $m \ominus n = m - n$ if $m > n$, 0 otherwise. 3) $f_{A\cap B}(x) = min(f_A(x), f_B(x))$, while for inclusion $A \subseteq B \Rightarrow f_A(x) \leq f_B(x)$.

Multiset rewriting can be immediately set in the DPO framework by considering the morphisms, in particular the match morphism, induced by the subset relation, so that $I = L \cap R$. Each $F_{M|x}$ counting the number of occurrences of $x \in \Sigma$ in M is a termination criterion.

The notion of negative application conditions for multisets amounts to inhibiting the application of a rule if some subset is present in the host multiset. Again, we can restrict ourselves to the case of nondeleting rules, such that I, L and R are multisets, with $L = I \subset R$ and $L \subset N$.

Theorem 1 (Decidability for multiset rewriting) *Termination is decidable for single rule multiset rewriting with DPO rules with NACs.*

Proof. For $p : L \overset{l}{\hookleftarrow} I \overset{r}{\hookrightarrow} R$ a DPO rule on multisets and M a host multiset s.t. $L \subset M$, three cases are given.

$N \subseteq R$. Then either $N \subset M$, so that p terminates trivially, or $N \not\subset M$. Now, applying p produces M', with $R \subset M'$ and, via $N \subseteq R$, $N \subset M'$, so that p can no longer be applied.

$R \subset N$. Let $Y = N \setminus L$ and $Z = R \setminus L$. Let $Z_i = Z_{i-1} \cup Z$, for $i > 0$, with $Z_0 = \emptyset$ and $M_i = M \cup Z_i$. Z_i is the set of elements added to M after i applications of p. Let $y = \mid Y \mid$ and $z = \mid Z \mid$. $R \subset N \Rightarrow y \geq z$. Then, $\exists! j \geq 0$, depending on R, N, and L, such that $z \cdot j \leq y < z \cdot (j+1)$. Hence, after at most j applications of p, either $N \subset M_j$, or $N \not\subset M_h$ for any $h \geq j$.

$N \not\subseteq R \wedge R \not\subseteq N$. Following the argument above, termination or nontermination is decidable for any pair $(rule, host)$ through static analysis, considering the two different cases $z \geq y$ and $z < y$.

□

By reconsidering the examples shown above, one can notice that for the graph rewriting case, the essential problem lies in the possibility of repeating or not the same match on a subsequent application of the same rule, if no new match is created. For the multiset case, all matches are equivalent. Hence, we look for some formal device that allows us to consider the occurrence of a match on some element of the host structure in a uniform way. We thus turn our attention to a general notion of (match) inhibitor.

5 Many sorted algebras with inhibitors

Inhibitors are here introduced by a suitable extension of many-sorted algebras [Wec92]. Given a set of sorts S, a signature SIG defines a set OP of operations and an arity for each operation as a function $ar : OP \to S \times S$, i.e. we only consider unary operations. A many-sorted SIG-algebra is defined by providing a carrier set A, such that a typing function $tp : A \to S$, assigns a sort to each value in A. We also use the notion A_s to summarize the values in A of type $s \in S$. The realization of an operation must then be consistent with its declared arity. Moreover, **SIG-Alg** is the category with the class of algebras over a suitable signature SIG as objects and a set of homomorphisms as morphisms.

Definition 1 ((Un-)Inhibitor) Let $(Flag, \leq)$ be a totally ordered set, with $Flag = \{\surd, \bullet\}$ and $\leq = \{(\surd, \surd), (\surd, \bullet), (\bullet, \bullet)\}$. We call $\surd$ the uninhibitor and $\bullet$ the inhibitor.

Definition 2 (Inhibitable version of SIG) Given a signature SIG, then an inhibitable version of SIG, $SIG' = (S', OP')$, is such that $S' = S \cup \{flag\}$, where $flag$ is the new sort of (un-)inhibitors. Given a set of inhibitable sorts $X \subseteq S$, OP' is extended by adding for each $x \in X$ an operation in_x of arity $(x, flag)$.

Due to the construction in Definition 2, SIG is a subsignature of its inhibitable version, i.e. the signature morphism $I : SIG \rightarrow SIG'$ is an inclusion.

Definition 3 (Inhibitable version of A) Let A be a SIG-algebra. Then a SIG'-algebra A' is an inhibitable version of A if $A'_{flag} = Flag$ and $A'_{|SIG} = A$, where $A'_{|SIG}$ is the SIG-reduct of A'.

Thus, for each $x \in X$ the operations in_x of arity $(x, flag)$ map each x-sorted element of A' into $\surd$ or $\bullet$. Moreover, the notion of restriction can be generalized to the forgetful functor U_I : **SIG'-Alg** $\rightarrow$ **SIG-Alg** induced by the signature morphism I, such that for each homomorphism in **SIG'-Alg** there is a corresponding homomorphism in **SIG-Alg**.

Lemma 1 ((**SIG'-Alg** , $\mathcal{M}$) is adhesive) *(**SIG'-Alg** , $\mathcal{M}$) with $\mathcal{M}$ the class of injective homomorphisms is an adhesive HLR category.*

Proof. Since SIG' contains only unary operation symbols, we exploit results in [EEPT05]: $\mathcal{M}$ is closed under composition and decomposition, (**SIG'-Alg**, $\mathcal{M}$) has pushouts and pullbacks along $\mathcal{M}$-morphisms and $\mathcal{M}$-morphisms are closed under pushouts and pullbacks, and pushouts along $\mathcal{M}$-morphisms in **SIG'-Alg** are van Kampen-squares. Thus, (**SIG'-Alg**, $\mathcal{M}$) is an adhesive HLR category. □

Example 1 (Graphs) *Let $GRAPH$ be the signature of graphs and let $nodes$ be an inhibitable sort. Then we get the following inhibitable version of $GRAPH$, $GRAPH'$.*

$$\begin{array}{rll} GRAPH & = & \\ sort: & nodes, edges & \\ opns: & source : edges \rightarrow nodes & \\ & target : edges \rightarrow nodes & \end{array} \qquad \begin{array}{rll} GRAPH' & = & \\ sort: & nodes, edges, flag & \\ opns: & source : edges \rightarrow nodes & \\ & target : edges \rightarrow nodes & \\ & in_{nodes} : nodes \rightarrow flag & \end{array}$$

Let A be a $GRAPH$-algebra with carriers $A_{nodes} = \{n_1, n_2\}$ and $A_{edges} = \{e_1, e_2\}$ and realizations $source_A(e_1) = n_1$ and $source_A(e_2) = target_A(e_1) = target_A(e_2) = n_2$. The algebras A'_1 and A'_2 with carriers $A'_{1,nodes} = A'_{2,nodes} = A_{nodes}$, $A'_{1,edges} = A'_{2,edges} = A_{edges}$, and $A'_{1,flag} = A'_{2,flag} = \{\surd, \bullet\}$ and additional realizations $in_{A'_1,nodes}(n_1) = \surd$ and $in_{A'_1,nodes}(n_2) = \surd$ resp. $in_{A'_2,nodes}(n_1) = \bullet$ and $in_{A'_2,nodes}(n_2) = \bullet$ are inhibitable versions of A.

Let B be a $GRAPH$-algebra with carriers and realizations as follows: $B_{nodes} = \{n\}, B_{edges} = \{e\}$, and $source_B(e) = target_B(e) = n$. There are two different inhibitable versions of B, B'_i, $i \in \{1, 2\}$, defined by the carriers as above and $B'_{i,flag} = \{\surd, \bullet\}$. The realizations are given as before, but for in_{nodes} we have $in_{B'_1,nodes}(n) = \surd$ and $in_{B'_2,nodes}(n) = \bullet$ respectively .

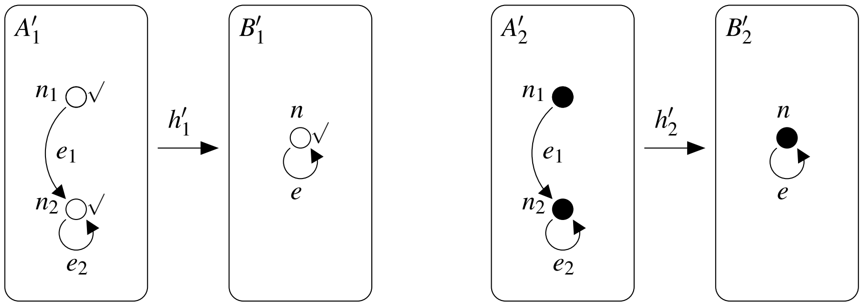

Figure 5: Homomorphisms $h'_1 : A'_1 \to B'_1$ and $h'_2 : A'_2 \to B'_2$.

Example 2 (Graph morphisms) *Let the inhibitable versions of A and B as defined in Example 1 and $h'_1 : A'_1 \to B'_1$ and $h'_2 : A'_2 \to B'_2$ (see Figure 5) be two different homomorphisms in* **SIG'-Alg**. *Due to the forgetful functor U_I there is a corresponding homomorphism $h : A \to B$ in* **SIG-Alg** *defined by $h_{nodes}(n_1) = h_{nodes}(n_2) = n$ and $h_{edges}(e_1) = h_{edges}(e_2) = e$.*

6 Concrete termination criteria for algebras with inhibitors

In this section we define a termination criterion for algebras with inhibitors.

Definition 4 (Rule with inhibitors) A rule with inhibitors $p : L \xleftarrow{l} I \xrightarrow{r} R$ is given by three SIG'-algebras L, I and R with finite carrier sets and two injective homomorphisms l and r of $\mathscr{M}$. In $L(p)$ the realizations of the operation in_x for each $x \in X$ are defined by $in_{L,x}(o) = \surd$ where o is a value from the carriers of the sorts in X.

Next, we define a suitable termination criterion for $(\textbf{SIG'-Alg}, \mathscr{M})$.

Definition 5 (Termination criterion) Let $\mathscr{G}$ be the class of all objects in **SIG'-Alg** and $F : \mathscr{G} \to \mathscr{N}$ from objects $A \in \mathscr{G}$ to natural numbers be the function which counts the number of uninhibited elements, i.e., the number of values o from the carriers of the sorts in X with $in_{A,x}(o) = \surd$.

$$F(A) = |Un(A)| = |\{o | x \in X, in_{A,x}(o) = \surd\}|$$

Lemma 2 (Termination criterion) *The function F of Definition 5 is a termination criterion for* $(\textbf{SIG'-Alg}, \mathscr{M})$.

Proof. We have to show that for any two arbitrary morphisms $a : C \to A$ and $b : C \to B$ in $\mathscr{M}$, the value $F(A +_C B)$ of the pushout object $A +_C B$ of a and b is given by $F(A +_C B) = F(A) + F(B) - F(C)$. Let $\uplus$ denote disjoint union. Because a is an injective homomorphism, we have

$$\begin{array}{ll} & \forall s \in S : (A_s +_{C_s} B_s) = B_s \uplus (A_s - a_s(C_s)) \\ \Rightarrow & \forall x \in X : Un(A_x +_{C_x} B_x) = Un(B)_x \uplus (Un(A)_x - a_x(Un(C)_x)) \\ \Rightarrow & |Un(A +_C B)| = |Un(B)| + |Un(A)| - |Un(C)| \\ \Rightarrow & F(A +_C B) = F(A) + F(B) - F(C) \end{array}$$

□

Theorem 2 (Termination of rules with inhibitors) *Let p be a rule with inhibitors as in Definition 4. If $F(L(p)) > F(R(p))$, the application of p, using injective $\mathscr{M}$-morphisms and inclusions as match morphism, terminates.*

Proof. From Lemma 2, F is a termination criterion for (**SIG'-Alg**, $\mathscr{M}$), thus it is a termination criterion for p. Hence, if $F(L(p)) > F(R(p))$, the iteration of rule application terminates [BHPT05]. □

The result of Theorem 2 can be extended to a set of rules $\mathscr{P}$ (see Theorem 1 in [BHPT05]), i.e. a replacement system terminates if F is a termination criterion which holds for all rules in $\mathscr{P}$. Additionally, [BHPT05] shows how this termination criterion can be used to prove termination of replacement units with control structures like sequential composition, choice, and as long as possible operators.

It is worth noting that for single-sorted algebras, termination for single rule rewriting with inhibitors becomes decidable, as witnessed by Theorem 3.

Theorem 3 (Decidability for single-sorted algebras) *Let SIG' be the inhibitable version of a signature SIG with a single sort $s \in X$. Let p be a rule with inhibitors. Then p terminates in* **SIG'-Alg** *if and only if there exists a termination criterion for it.*

Proof. The *if* direction is a consequence of Theorem 2. For the *only if* direction, as *SIG* contains only one sort s, for p to terminate there must exist some element x of its carrier which either appears in $L(p)$, but not in $R(p)$, or which appears in $L(p)$ tagged with $\surd$ and in $R(p)$ tagged with $\bullet$. In the first case, a function counting the number of occurrences of x is a termination criterion, in the second case, the function F of Definition 5 is a termination criterion. □

Note that if *SIG* is a signature with a single sort, each object in **SIG'-Alg** is isomorphic to a multiset.

7 Multiple levels of inhibition

The carrier $Flag$ can be extended in such a way that its elements constitute a finite set W, equipped with a partial order $\leq$ and a distinguished element k, called *top inhibitor level*, such that $\nexists x \in W, x \neq k$ with $k \leq x$. Each element $i \in W$ is called a *level*.

Definition 6 (Multi level inhibitors) Given $(W, \leq)$ as above, the multi level inhibitable version of *SIG* and objects A of **SIG-Alg** are given as in Definitions 2 and 3.

Here the realization of the operations in_x of arity $(x, flag)$ for each $x \in X$ is a mapping of each x-sorted element of A into a level $i \in W$.

Definition 7 (Rule with multi level inhibitors) A rule with multi level inhibitors $p : L \stackrel{l}{\leftarrow} I \stackrel{r}{\rightarrow} R$ is given by three SIG'-algebras L, I and R with finite carrier sets and two injective homomorphisms l and r of $\mathscr{M}$. Moreover, $\forall x \in X\ \forall o \in L_x : in_{L,x}(o) \neq k$, where o is a value from the carriers of the sorts in X and k is the top inhibitor level.

Next, we define a suitable termination criterion for **SIG'-Alg**.

Definition 8 (Termination criterion) Let $\overline{F} = (F_i)_{i \in W}$ a family of functions $F_i : \mathscr{G} \rightarrow \mathscr{N}$ from objects to natural numbers, each counting the number of uninhibited elements of a specific level $i \in W$.

$$F_i(A) = |Un_i(A)| = |\{o | x \in X, i \in W, in_{A,x}(o) = i\}|$$

Lemma 3 (Termination criterion) *Each function F_i as defined in Definition 8 is a termination criterion.*

Proof. The proof follows from the fact that for any two arbitrary morphisms $a : C \rightarrow A$ and $b : C \rightarrow B$ in $\mathscr{M}$, the value $F_i(A +_C B)$ of the pushout object $A +_C B$ of a and b is given by

$$\begin{aligned} F_i(A +_C B) &= |Un_i(A +_C B)| \\ &= |Un_i(A)| + |Un_i(B)| - |Un_i(C)| \\ &= F_i(A) + F_i(B) - F_i(C) \end{aligned}$$

□

Theorem 4 (Termination of rules with multi level inhibitors) *Let p be a rule with multi level inhibitors as given in Definition 4. If $\exists i \in W : F_i(L(p)) > F_i(R(p))$, application of p using inclusion as match morphism terminates.*

Proof. From Lemma 3, if $\exists i \in W : F_i(L(p)) > F_i(R(p))$, then the iteration of rule application terminates. □

The results in this Section can be extended to the case where W contains a subset K such that $k \in K \Leftrightarrow \nexists x \in W$ with $k \leq x$. Each $k \in K$ can then act as top level inhibitor.

8 Applications

Providing full support of NACs in the DPO approach is costly and in many situations the notion of inhibitor provides a simpler alternative. As an example, the definition of singleton classes requires that only one instance of that class can be created. This can be achieved by creating the object representing a class in an uninhibited version, and inhibiting it after the first object creation, as shown in Figure 6.

In general, a model transformation process exploiting inhibitors would develop by first immersing the original model in its inhibitable version with all elements uninhibited, then per-

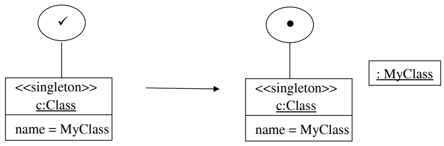

Figure 6: The rule for guaranteeing creation of singletons.

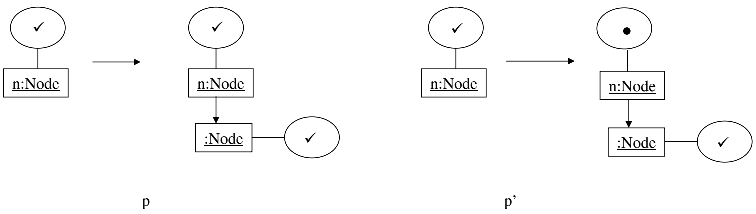

Figure 7: A pair of rules to create ramifications (p) up to the point when p’ is applied.

forming the transformations using rules with inhibitors, and finally reprojecting on the original signature. The process can be iterated several times, possibly removing from the set of rules those which have been exploited to inhibit some element.

The approach exploited for singletons could be extended to any situation in which we want to limit or discontinue the possibility of generating elements in relation with another specific element. For example, in the creation of trees, one could prevent further ramification of a node. A pair of rules such as those depicted in Figure 7 would then allow an arbitrary level of ramification, but also prevent, for instance in an interactive or collaborative setting, a node to be further expanded. Note that, while the rule of Figure 6 terminates, as the number of uninhibited elements decreases with every application, the composed rule obtained from the rules of Figure 7 does not terminate, nor does any individual rule.

Expansion limitation can occur at specific configurations instead that at arbitrary derivation steps. For example, in modeling visual languages, plex languages allow only a well-defined number of connections to depart or enter into nodes of a given type. Hence, by adopting multi-level inhibitors, the creation of connections could increase the level of saturation up to the point when all attachment points are busy, without the need to define rules for each possible configuration of already connected points.

With this solution the inhibited elements would not be available for any other rule. This can be mitigated by relaxing the notion of inhibitor, so that instead of having a “general-purpose” inhibition, one could define “rule-specific” inhibitor sets. In this case, carriers for the $flag$ sort would be partitioned in pairs $\{\surd_p, \bullet_p\}$, where p identifies uniquely a rule with respect to which an inhibitor is defined. Instead of having a total order, one would have a partial order $\leq$ s.t. $\forall p \in P$, $\{(\surd_p, \surd_p), (\surd_p, \bullet_p), (\bullet_p, \bullet_p)\} \subset \leq$ and $\forall p' \neq p$, $x_p \in \{\surd_p, \bullet_p\}$, $x_{p'} \in \{\surd_{p'}, \bullet_{p'}\} \Rightarrow x_p \not\leq x_{p'} \wedge x_{p'} \not\leq x_p$.

The inhibitor structure can be also implemented through some clever usage of attributes. In some cases, the original model already provides attributes whose values can be interpreted as an indication of the availability of the model element for a rule. In other cases, one can devise suitable extensions of the model to define analogous of the inhibition mechanism. The analysis conducted in this paper can help identify the necessary adaptations of the model, according to the desired type of inhibition.

9 Conclusions

A study of decidability of termination for high level replacement rules with negative application conditions in the DPO approach shows the difficulties of extending the notion of termination criterion, presented in [BHPT05] to rules of this form. We have therefore turned to the weaker, but somehow related, notion of inhibitor, by which we mean tagging terms in a many-sorted algebra with a flag removing it from further matches. We have also shown that the termination results obtained for simple inhibitors can be extended to multiple levels of inhibition (coming to saturate the availability of an element to matches). Some uses of inhibitors in model transformation applications have been discussed.

We plan to study termination properties for rule-specific inhibitors, preventing the possibility of involving an element in a match for a specific rule, and to investigate the possibility of more general inhibitors, associated not with individual elements, but with generic subobjects in a category. Moreover, we plan to investigate confluence properties for rewriting systems with inhibitors and to set a general framework for proving properties of such systems.

Acknowledgements: Partially supported by the EC under Research and Training Network SegraVis. Kathrin Hoffmann acknowledges the financial support provided through the European Community's Human Potential Programme under contract HPRN-CT-2002-00275, SegraVis.

Bibliography

[BHPT05] P. Bottoni, K. Hoffmann, F. Parisi-Presicce, G. Taentzer. High-Level Replacement Units and their Termination Properties. *Journal of Visual Languages and Computing* 16:485–507, 2005.

[BMPR02] P. Bottoni, C. Martín-Vide, G. Paun, G. Rozenberg. Membrane systems with promoters/inhibitors. *Acta Informatica* 38(10):695–720, 2002.

[EEL$^+$05] H. Ehrig, K. Ehrig, J. de Lara, G. Taentzer, D. Varró, S. Varró-Gyapay. Termination criteria for model transformation. In *Proc. FASE 2005*. LNCS 3442, pp. 49–63. Springer, 2005.

[EEPT05] H. Ehrig, K. Ehrig, U. Prange, G. Taentzer. Fundamental Theory for Typed Attributed Graphs and Graph Transformation based on Adhesive HLR Categories. *Fundamenta Informaticae* 20:1–31, 2005.

[EHKP91] H. Ehrig, A. Habel, H.-J. Kreowski, F. Parisi-Presicce. Parallelism and Concurrency in High-Level Replacement Systems. *Math. Struct. in Comp. Science* 1:361–404, 1991.

[EHPP04] H. Ehrig, A. Habel, J. Padberg, U. Prange. Adhesive high-level replacement categories and systems. In H. Ehrig and G. Engels and F. Parisi-Presicce and G. Rozenberg (ed.), *Proc. ICGT'04*. LNCS 3256, pp. 144–160. Springer, 2004.

[Hac76] M. Hack. Decidability questions for Petri nets. Technical report 161, Laboratory for Computer Science, M.I.T., June 1976.

[HHT96] A. Habel, R. Heckel, G. Taentzer. Graph Grammars with Negative Application Conditions. *Fundamenta Informaticae* 26(3,4):287–313, 1996.

[KMP02] M. Koch, L. V. Mancini, F. Parisi-Presicce. Decidability of Safety in Graph-Based Models for Access Control. In Gollmann et al. (eds.), *Proc. ESORICS'02*. LNCS 2502, pp. 229–243. Springer, 2002.

[LS04] S. Lack, P. Sobocinski. Adhesive Categories. In Walukiewicz (ed.), *Proc. FOSSACS'04*. LNCS 2987, pp. 273–288. 2004.

[McN97] R. McNaughton. Semi-Thue Systems with an Inhibitor. *Journal of Automated Reasoning* 26:409–431, 1997.

[Nav70] E. Navratil. Context-free grammars with regular conditions. *Kybernetika* 2:118–126, 1970.

[Pau79] G. Paun. On the generative capacity of conditional grammars. *Information and Control* 43(2):178–186, 1979.

[Pau85] G. Paun. A Variant of Random Context Grammars: Semi-Conditional Grammars. *Theoretical Computer Science* 41:1–17, 1985.

[Plu98] D. Plump. Termination of graph rewriting is undecidable. *Fundamenta Informaticae* 33(2):201–209, 1998.

[Plu05] D. Plump. Confluence of Graph Transformation Revisited. In Middeldorp et al. (eds.), *Processes, Terms and Cycles: Steps on the Road to Infinity: Essays Dedicated to Jan Willem Klop on the Occasion of His 60th Birthday*. LNCS 3838, pp. 280–308. Springer, 2005.

[vdW71] A. van der Walt. Random context grammars. In *Proc. IFIP Congress 1970*. Pp. 66–68. North-Holland Publ. Co., 1971.

[Wec92] W. Wechler. *Universal Algebra for Computer Scientists*. Springer, 1992.