

Extending Context-Sensitivity in Term Rewriting

Bernhard Gramlich and Felix Schernhammer*

Institute of Computer Languages, Theory and Logic Group
Vienna University of Technology
{gramlich,felix}@logic.at

We propose a generalized version of context-sensitivity in term rewriting based on the notion of “forbidden patterns”. The basic idea is that a rewrite step should be forbidden if the redex to be contracted has a certain shape and appears in a certain context. This shape and context is expressed through forbidden patterns. In particular we analyze the relationships among this novel approach and the commonly used notion of context-sensitivity in term rewriting, as well as the feasibility of rewriting with forbidden patterns from a computational point of view. The latter feasibility is characterized by demanding that restricting a rewrite relation yields an improved termination behaviour while still being powerful enough to compute meaningful results. Sufficient criteria for both kinds of properties in certain classes of rewrite systems with forbidden patterns are presented.

1 Introduction and Overview

Standard term rewriting systems (TRSs) are well-known to enjoy nice logical and closure properties. Yet, from an operational and computational point of view, i.e., when using term rewriting as computational model, it is also well-known that for non-terminating systems restricted versions of rewriting obtained by imposing context-sensitivity and/or strategy requirements may lead to better results (e.g., in terms of computing normal forms, head-normal forms, etc.).

One major goal when using reduction strategies and context restrictions is to avoid non-terminating reductions. On the other hand the restrictions should not be too strong either, so that the ability to compute useful results in the restricted rewrite systems is not lost. We introduce a novel approach to context restrictions relying on the notion of “forbidden patterns”, which generalizes existing approaches and succeeds in handling examples in the mentioned way (i.e., producing a terminating reduction relation which is powerful enough to compute useful results) where others fail.

The following example motivates the use of reduction strategies and/or context restrictions.

Example 1. Consider the following rewrite system, cf. e.g. [15]:

$$\begin{aligned}\text{inf}(x) &\rightarrow x : \text{inf}(s(x)) \\ \text{2nd}(x : (y : zs)) &\rightarrow y\end{aligned}$$

This TRS is non-terminating and not even weakly normalizing. Still some terms like $\text{2nd}(\text{inf}(x))$ are reducible to a normal form while also admitting infinite reduction sequences. One goal of context restrictions and reduction strategies is to restrict derivations in a way such that normal forms can be computed whenever they exist, while infinite reductions are avoided.

One way to address the problem of avoiding non-normalizing reductions in Example 1 is the use of reduction strategies. For instance for the class of (almost) orthogonal rewrite systems (the TRS of

*This author has been supported by the Austrian Academy of Sciences under grant 22.361.

Example 1 is orthogonal), always contracting all outermost redexes in parallel yields a normalizing strategy (i.e. whenever a term can be reduced to a normal form it is reduced to a normal form under this strategy) [18]. Indeed, one can define a sequential reduction strategy having the same property for an even wider class of TRSs [3]. One major drawback (or asset depending on one's point of view) of using reduction strategies, however, is that their use does not introduce new normal forms. This means that the set of normal forms w.r.t. to some reduction relation is the same as the set of normal forms w.r.t. to the reduction relation under some strategy. Hence, strategies can in general not be used to detect non-normalizing terms or to impose termination on not weakly normalizing TRSs (with some exceptions cf. e.g. [3, Theorem 7.4]). Moreover, the process of selecting a suitable redex w.r.t. to a reduction strategy is often complex and may thus be inefficient.

These shortcomings of reduction strategies led to the advent of proper restrictions of rewriting that usually introduce new normal forms and select respectively forbid certain reductions according to the syntactic structure of a redex and/or its surrounding context.

The most well-known approach to context restrictions is context-sensitive rewriting. There, a *replacement map* μ specifies the arguments $\mu(f) \subseteq \{1, \dots, ar(f)\}$ which can be reduced for each function f . However, regarding Example 1, context-sensitive rewriting does not improve the situation, since allowing the reduction of the second argument of \cdot leads to non-termination, while disallowing its reduction leads to incompleteness in the sense that for instance a term like $2nd(\inf(x))$ cannot be normalized via the corresponding context-sensitive reduction relation, despite having a normal form in the unrestricted system.

Other ideas of context restrictions range from explicitly modeling lazy evaluation (cf. e.g. [9, 17, 19]), to imposing constraints on the order of argument evaluation of functions (cf. e.g. [10, 7]), and to combinations of these concepts, also with standard context-sensitive rewriting (cf. e.g. [15, 2]). The latter generalized versions of context-sensitive rewriting are quite expressive and powerful (indeed some of them can be used to restrict the reduction relation of the TRS in Example 1 in a way, so that the restricted relation is terminating and still powerful enough to compute (head-)normal forms), but on the other hand tend to be hard to analyze and understand, due the subtlety of the strategic information specified.

The approach we present in this paper is simpler in that its definition only relies on matching and simple comparison of positions rather than on laziness or prioritizing the evaluation of certain arguments of functions over others. In order to reach the goal of restricting the reduction relation in such a way that it is terminating while still being powerful enough to compute useful results, we provide a method to verify termination of a reduction relation restricted by our approach (Section 5) as well as a criterion which guarantees that normal forms computed by the restricted system are head-normal forms of the unrestricted system (Section 4).

Recently it turned out that, apart from using context-sensitivity as computation model for standard term rewriting (cf. e.g. [16, 14]), context-sensitive rewrite systems naturally also appear as intermediate representations in many areas relying on transformations, such as program transformation and termination analysis of rewrite systems with conditions [6, 20] / under strategies [8].

This suggests that apart from using restrictions as guidance and thus as operational model for rewrite derivations, a general, flexible and well-understood framework of restricted term rewriting going beyond context-sensitive rewriting may be useful as a valuable tool in many other areas, too.

The major problem in building such a framework is that imposing context restrictions on term rewriting in general invalidates the closure properties of term rewriting relations, i.e., stability under contexts and substitutions. Note that in the case of context-sensitive rewriting à la [14, 16] only stability under contexts is lost.

In this work we will sketch and discuss a generalized approach to context-sensitivity (in the sense of [14, 16]) relying on *forbidden patterns* rather than on forbidden arguments of functions. From a systematic point of view we see the following design decisions to be made.

- What part of the context of a (sub)term is relevant to decide whether the (sub)term may be reduced or not?
- In order to specify the restricted reduction relation, is it better/advantageous to explicitly define the allowed or the forbidden part of the context-free reduction relation?
- What are the forbidden/allowed entities, for instance whole subterms, contexts, positions, etc.?
- Does it depend on the shape of the considered subterm itself (in addition to its outside context) whether it should be forbidden or not (if so, stability under substitutions may be lost)?
- Which restrictions on forbidden patterns seem appropriate (also w.r.t. practical feasibility) in order to guarantee certain desired closure and preservation properties.

The remainder of the paper is structured as follows. In Section 2 we briefly recall some basic notions and notations. Rewriting with forbidden patterns is defined, discussed and exemplified in Section 3. In the main Sections 4 and 5 we develop some theory about the expressive power of rewriting with forbidden patterns (regarding the ability to compute original (head-)normal forms), and about how to prove ground termination for such systems via a constructive transformational approach. Crucial aspects are illustrated with the two running Examples 1 and 3. Finally, in Section 6 we summarize our approach and its application in the examples, discuss its relationship to previous approaches and briefly touch the important perspective and open problem of (at least partially) automating the generation of suitable forbidden patterns in practice.¹

2 Preliminaries

We assume familiarity with the basic notions and notations in term rewriting, cf. e.g. [4], [5].

Since we develop our approach in a many-sorted setting, we recall a few basics on many-sorted equational reasoning (cf. e.g. [5]). A many-sorted signature \mathcal{F} is a pair (S, Ω) where S is a set of sorts and Ω is a family of (mutually disjoint) sets of typed function symbols: $\Omega = (\Omega_{\omega, s} \mid \omega \in S^*, s \in S)$. We also say, f is of type $\omega \rightarrow s$ (or just s if $\omega = \emptyset$) if $f \in \Omega_{\omega, s}$. $V = (V_s \mid s \in S)$ is a family of (mutually disjoint) countably infinite sets of typed variables (with $V \cap \Omega = \emptyset$). The set $\mathcal{T}(\mathcal{F}, V)_s$ of (well-formed) terms of sort s is the least set containing V_s , and whenever $f \in \Omega_{(s_1, \dots, s_n), s}$ and $t_i \in \mathcal{T}(\mathcal{F}, V)_{s_i}$ for all $1 \leq i \leq n$, then $f(t_1, \dots, t_n) \in \mathcal{T}(\mathcal{F}, V)_s$. The sort of a term t is denoted by $\text{sort}(t)$. Rewrite rules are pairs of terms $l \rightarrow r$ where $\text{sort}(l) = \text{sort}(r)$. Subsequently, we make the types of terms and rewrite rules explicit only if they are relevant. Throughout the paper x, y, z represent (sorted) variables.

Positions are possibly empty sequences of natural numbers (the empty sequence is denoted by ε). We use the standard partial order \leq on positions given by $p \leq q$ if there is some position p' , such that $p.p' = q$ (i.e., p is a prefix of q). $\text{Pos}(s)$ ($\text{Pos}_{\mathcal{F}}(s)$) denotes the set of (non-variable) positions of a term s . By $s \xrightarrow{p} t$ we mean rewriting at position p . Given a TRS $\mathcal{R} = (\mathcal{F}, R)$ we partition \mathcal{F} into the set D of defined function symbols, which are those that occur as root symbols of left-hand sides of rules in R , and the set C of constructors (given by $\mathcal{F} \setminus D$). For TRSs $\mathcal{R} = (\mathcal{F}, R)$ we sometimes confuse \mathcal{R} and R , e.g., by omitting the signature.

¹Due to lack of space the obtained results are presented without proofs. The latter can be found in the full technical report version of the paper, cf. <http://www.logic.at/staff/{gramlich,schernhammer}/>.

3 Rewriting with Forbidden Patterns

In this section we define a generalized approach to rewriting with context restrictions relying on term patterns to specify forbidden subterms/superterms/positions rather than on a replacement map as in context-sensitive rewriting.

Definition 1 (forbidden pattern). *A forbidden pattern (w.r.t. to a signature \mathcal{F}) is a triple $\langle t, p, \lambda \rangle$, where $t \in \mathcal{T}(\mathcal{F}, V)$ is a term, p a position from $\text{Pos}(t)$ and $\lambda \in \{h, b, a\}$.*

The intended meaning of the last component λ is to indicate whether the pattern forbids reductions

- exactly at position p , but not outside (i.e., strictly above or parallel to p) or strictly below – (h for here), or
- strictly below p , but not at or outside p – (b for below), or
- strictly above position p , but not at, below or parallel to p – (a for above).

Abusing notation we sometimes say a forbidden pattern is linear, unifies with some term etc. when we actually mean that the term in the first component of a forbidden pattern has this property.

We denote a *finite* set of forbidden patterns for a signature \mathcal{F} by $\Pi_{\mathcal{F}}$ or just Π if \mathcal{F} is clear from the context or irrelevant. For brevity, patterns of the shape $\langle -, -, h/b/a \rangle$ are also called $h/b/a$ -patterns, or *here/below/above*-patterns.²

Note that if for a given term t we want to specify more than just one restriction by a forbidden pattern, this can easily be achieved by having several triples of the shape $\langle t, -, - \rangle$.

In contrast to context-sensitive rewriting, where a replacement map defines the allowed part of the reduction, the patterns are supposed to explicitly define its *forbidden* parts, thus implicitly yielding allowed reduction steps as those that are not forbidden.

Definition 2 (forbidden pattern reduction relation). *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS with forbidden patterns $\Pi_{\mathcal{F}}$. The forbidden pattern reduction relation $\rightarrow_{\mathcal{R}, \Pi_{\mathcal{F}}}$, or \rightarrow_{Π} for short, induced by some set of forbidden patterns Π and \mathcal{R} , is given by $s \rightarrow_{\mathcal{R}, \Pi_{\mathcal{F}}} t$ if $s \xrightarrow{p} t$ for some $p \in \text{Pos}_{\mathcal{F}}(s)$ such that there is no pattern $\langle u, q, \lambda \rangle \in \Pi_{\mathcal{F}}$, no context C and no position q' with*

- $s = C[u\sigma]_{q'}$ and $p = q'.q$, if $\lambda = h$,
- $s = C[u\sigma]_{q'}$ and $p > q'.q$, if $\lambda = b$, and
- $s = C[u\sigma]_{q'}$ and $p < q'.q$, if $\lambda = a$.

Note that for a finite rewrite system \mathcal{R} (with finite signature \mathcal{F}) and a finite set of forbidden patterns $\Pi_{\mathcal{F}}$ it is decidable whether $s \rightarrow_{\mathcal{R}, \Pi_{\mathcal{F}}} t$ for terms s and t . We write (\mathcal{R}, Π) for rewrite systems with associated forbidden patterns. Such a rewrite system (\mathcal{R}, Π) is said to be Π -terminating (or just terminating if no confusion arises) if $\rightarrow_{\mathcal{R}, \Pi}$ is well-founded. We also speak of Π -normal forms instead of $\rightarrow_{\mathcal{R}, \Pi}$ -normal forms.

Special degenerate cases of (\mathcal{R}, Π) include e.g. $\Pi = \emptyset$ where $\rightarrow_{\mathcal{R}, \Pi} = \rightarrow_{\mathcal{R}}$, and $\Pi = \{\langle l, \varepsilon, h \rangle \mid l \rightarrow r \in R\}$ where $\rightarrow_{\mathcal{R}, \Pi} = \emptyset$.

In the sequel we use the notions of *allowed* and *forbidden* (by Π) redexes. A redex $s|_p$ of a term s is allowed if $s \xrightarrow{p}_{\Pi} t$ for some term t , and forbidden otherwise.

² Here and subsequently we use a wildcard notation for forbidden patterns. For instance, $\langle -, -, i \rangle$ stands for $\langle t, p, i \rangle$ where t is some term and p some position in t of no further relevance.

Example 2. Consider the TRS from Example 1. If $\Pi = \{(x : (y : \text{inf}(z)), 2, 2, h)\}$, then \rightarrow_Π can automatically be shown to be terminating. Moreover, \rightarrow_Π is powerful enough to compute original head-normal forms if they exist (cf. Examples 6 and 11 below).

Example 3. Consider the non-terminating TRS \mathcal{R} given by

$$\begin{array}{ll} \text{take}(0, y : ys) \rightarrow y & \text{app}(\text{nil}, ys) \rightarrow ys \\ \text{take}(s(x), y : ys) \rightarrow \text{take}(x, ys) & \text{app}(x : xs, ys) \rightarrow x : \text{app}(xs, ys) \\ \text{take}(x, \text{nil}) \rightarrow 0 & \text{inf}(x) \rightarrow \text{inf}(s(x)) \end{array}$$

with two sorts $S = \{\text{Nat}, \text{NatList}\}$, where the types of function symbols are as follows: $\text{nil} : \text{NatList}$, $0 : \text{Nat}$, $s : \text{Nat} \rightarrow \text{Nat}$, $:$ is of type $\text{Nat}, \text{NatList} \rightarrow \text{NatList}$, $\text{inf} : \text{Nat} \rightarrow \text{NatList}$, $\text{app} : \text{NatList}, \text{NatList} \rightarrow \text{NatList}$ and $\text{take} : \text{Nat}, \text{NatList} \rightarrow \text{Nat}$. If one restricts rewriting in \mathcal{R} via Π given by

$$\langle x : \text{inf}(y), 2, h \rangle \quad \langle x : \text{app}(\text{inf}(y), zs), 2, 1, h \rangle \quad \langle x : \text{app}(y : \text{app}(z, zs), us), 2, h \rangle,$$

then \rightarrow_Π is terminating and still every well-formed ground term can be normalized with the restricted relation \rightarrow_Π (provided the term is normalizing). See Examples 7 and 12 below for justifications of these claims.

Several well-known approaches to restricted term rewriting as well as to rewriting guided by reduction strategies occur as special cases of rewriting with forbidden patterns. In the following we provide some examples. Context-sensitive rewriting, where a replacement map μ specifies the arguments $\mu(f) \subseteq \{1, \dots, \text{ar}(f)\}$ which can be reduced for each function f , arises as special case of rewriting with forbidden patterns by defining Π to contain for each function symbol f and each $j \in \{1, \dots, \text{ar}(f)\} \setminus \mu(f)$ the forbidden patterns $(f(x_1, \dots, x_{\text{ar}(f)}), j, h)$ and $(f(x_1, \dots, x_{\text{ar}(f)}), j, b)$.

Moreover, with forbidden patterns it is also possible to simulate position-based reduction strategies such as innermost and outermost rewriting. The innermost reduction relation of a TRS \mathcal{R} coincides with the forbidden pattern reduction relation if one uses the forbidden patterns $\langle l, \varepsilon, a \rangle$ for the left-hand sides l of each rule of \mathcal{R} . Dually, if patterns $\langle l, \varepsilon, b \rangle$ are used, the forbidden pattern reduction relation coincides with the *outermost* reduction relation w.r.t. \mathcal{R} .

However, note that more complex layered combinations of the aforementioned approaches, such as innermost context-sensitive rewriting cannot be modeled by forbidden patterns as proposed in this paper.

Still, the definition of forbidden patterns and rewriting with forbidden patterns is rather general and leaves many parameters open. In order to make this approach feasible in practice, it is necessary to identify interesting classes of forbidden patterns that yield a reasonable trade-off between power and simplicity. For these interesting classes of forbidden patterns we need methods which guarantee that the results (e.g. normal forms) computed by rewriting with forbidden patterns are meaningful, in the sense that they have some natural correlation with the actual results obtained by unrestricted rewriting. For instance, it is desirable that normal forms w.r.t. the restricted rewrite system are original head-normal forms. In this case one can use the restricted reduction relation to compute original normal forms (by an iterated process) whenever they exist (provided that the TRS in question is left-linear, confluent and the restricted reduction relation is terminating) (cf. Section 4 below for details). We define a criterion ensuring that normal forms w.r.t. the restricted system are original head-normal forms in the following section.

4 Computing Meaningful Results

We are going to use canonical context-sensitive rewriting as defined in [14, 16] as an inspiration for our approach. There, for a given (left-linear) rewriting system \mathcal{R} certain restrictions on the associated

replacement map μ guarantee that \rightarrow_μ -normal forms are $\rightarrow_{\mathcal{R}}$ -head-normal-forms. Hence, results computed by \rightarrow_μ and $\rightarrow_{\mathcal{R}}$ share the same root symbol.

The basic idea is that reductions that are essential to create a more outer redex should not be forbidden. In the case of context-sensitive rewriting this is guaranteed by demanding that whenever an f -rooted term t occurs (as subterm) in the left-hand side of a rewrite rule and has a non-variable direct subterm $t|_i$, then $i \in \mu(f)$.

It turns out that for rewriting with forbidden patterns severe restrictions on the shape of the patterns are necessary in order to obtain results similar to the ones for canonical context-sensitive rewriting in [14]. First, no forbidden patterns of the shape $\langle -, \varepsilon, h \rangle$ or $\langle -, -, a \rangle$ may be used as they are in general not compatible with the desired root-normalizing behaviour of our forbidden pattern rewrite system.

Moreover, for each pattern $\langle t, p, - \rangle$ we demand that

- t is linear,
- p is a variable or maximal (w.r.t. to the prefix ordering \leq on positions) non-variable position in t , and
- for each position $q \in \text{Pos}(t)$ with $q||p$ we have $t|_q \in V$.

We call the class of patterns obtained by the above restrictions *simple patterns*.

Definition 3 (simple patterns). *A set Π of forbidden patterns is called simple if it does not contain patterns of the shape $\langle -, \varepsilon, h \rangle$ or $\langle -, -, a \rangle$ and for every pattern $\langle t, p, - \rangle \in \Pi$ it holds that t is linear, $t|_p \in V$ or $t|_p = f(x_1, \dots, x_{ar(f)})$ for some function symbol f , and for each position $q \in \text{Pos}(t)$ with $q||p$ we have that $t|_q$ is a variable.*

Basically these syntactical properties of forbidden patterns are necessary to ensure that reductions which are essential to enable other, more outer reductions are not forbidden. Moreover, these properties, contrasting those defined in Definition 4 below, are independent of any concrete rewrite system.

The forbidden patterns of the TRS (\mathcal{R}, Π) in Example 4 below are not simple, since the patterns contain terms with parallel non-variable positions. This is the reason why it is not possible to head-normalize terms (w.r.t \mathcal{R}) with \rightarrow_Π :

Example 4. *Consider the TRS \mathcal{R} given by*

$$f(b, b) \rightarrow g(f(a, a)) \quad a \rightarrow b$$

and forbidden patterns $\langle f(a, a), 1, h \rangle$ and $\langle f(a, a), 2, h \rangle$. $f(a, a)$ is linear and 1 and 2 are maximal positions (w.r.t. \leq) within this term. However, positions 1 and 2 are both non-variable and thus e.g. for $\langle f(a, a), 1, h \rangle$ there exists a position $2||1$ such that $f(a, a)|_2 = a \notin V$. Hence, Π is too restrictive to compute all \mathcal{R} -head-normal forms in this example. Indeed, $f(a, a) \rightarrow_{\mathcal{R}}^ f(b, b) \rightarrow_{\mathcal{R}} g(f(a, a))$ where the latter term is a \mathcal{R} -head-normal form.*

The term $f(a, a)$ is a Π -normal form, although it is not a head-normal form (w.r.t. \mathcal{R}). Note also that the (first components of) forbidden patterns are not unifiable with the left-hand side of the rule that is responsible for the (later) possible root-step when reducing $f(a, a)$, not even if the forbidden subterms in the patterns are replaced by fresh variables.

Now we are ready to define canonical rewriting with forbidden patterns within the class of simple forbidden patterns. To this end, we demand that patterns do not overlap with left-hand sides of rewrite rules in a way such that reductions necessary to create a redex might be forbidden.

Definition 4 (canonical forbidden patterns). *Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS with simple forbidden patterns $\Pi_{\mathcal{F}}$ (w.l.o.g. we assume that R and $\Pi_{\mathcal{F}}$ have no variables in common). Then, $\Pi_{\mathcal{F}}$ is \mathcal{R} -canonical (or just canonical) if the following holds for all rules $l \rightarrow r \in R$:*

1. There is no pattern (t, p, λ) such that
 - $t'|_q$ and l unify for some $q \in \text{Pos}_{\mathcal{F}}(t)$ where $t' = t[x]_p$ and $q > \varepsilon$, and
 - there exists a position $q' \in \text{Pos}_{\mathcal{F}}(l)$ with $q.q' = p$ for $\lambda = h$ respectively $q.q' > p$ for $\lambda = b$.
2. There is no pattern (t, p, λ) such that
 - t' and $l|_q$ unify for some $q \in \text{Pos}_{\mathcal{F}}(l)$ where $t' = t[x]_p$, and
 - there exists a position q' with $q.q' \in \text{Pos}_{\mathcal{F}}(l)$ and $q' = p$ for $\lambda = h$ respectively $q' > p$ for $\lambda = b$.

Here, x denotes a fresh variable.

Example 5. Consider the TRS \mathcal{R} given by the single rule

$$l = f(g(h(x))) \rightarrow x = r.$$

Then, $\Pi = \{(t, p, h)\}$ with $t = g(f(a))$, $p = 1.1$ is not canonical since $t[x]_p|_q = g(f(y))|_1 = f(y)$ and l unify where $q = q' = 1$ and thus $q.q' = p$ (hence $\text{root}(l|_{q'}) = g$). Moreover, also $\Pi = \{(t, p, h)\}$ with $t = g(i(x))$, $p = 1$ is not canonical, since $l|_q = g(h(x))$ and $t[x]_p = f(y)$ unify for $q = 1$ and $q.p = 1.1$ is a non-variable position in l .

On the other hand, $\Pi = \{(g(g(x)), 1.1, h)\}$ is canonical. Note that all of the above patterns are simple.

In order to prove that normal forms obtained by rewriting with simple and canonical forbidden patterns are actually head-normal forms w.r.t. unrestricted rewriting, and also to provide more intuition on canonical rewriting with forbidden patterns, we define the notion of a *partial redex* (w.r.t. to a rewrite system \mathcal{R}) as a term that is matched by a non-variable term l' which in turn matches the left-hand side of some rule of \mathcal{R} . We call l' a *witness* for the partial match.

Definition 5 (Partial redex). Given a rewrite system $\mathcal{R} = (\mathcal{F}, R)$, a partial redex is a term s that is matched by a non-variable term l' which in turn matches the left-hand side of some rule in R . The (non-unique) term l' is called witness for a partial redex s .

Thus, a partial redex can be viewed as a candidate for a future reduction step, which can only be performed if the redex has actually been created through more inner reduction steps. Hence, the idea of canonical rewriting with forbidden patterns could be reformulated as guaranteeing that the reduction of subterms of partial redexes is allowed whenever these reductions are necessary to create an actual redex.

Lemma 1. Let $\mathcal{R} = (\mathcal{F}, R)$ be a left-linear TRS with canonical (hence, in particular simple) forbidden patterns $\Pi_{\mathcal{F}}$. Moreover, let s be a partial redex w.r.t. to the left-hand side of some rule l with witness l' such that $l|_p \notin V$ but $l'|_p \in V$. Then in the term $C[s]_q$ the position $q.p$ is allowed by $\Pi_{\mathcal{F}}$ for reduction provided that q is allowed for reduction.

Theorem 1. Let $\mathcal{R} = (\mathcal{F}, R)$ be a left-linear TRS with canonical (hence in particular simple) forbidden patterns $\Pi_{\mathcal{F}}$. Then $\rightarrow_{\mathcal{R}, \Pi_{\mathcal{F}}}$ -normal forms are $\rightarrow_{\mathcal{R}}$ -head-normal forms.

Given a left-linear and confluent rewrite system \mathcal{R} and a set of canonical forbidden patterns Π such that \rightarrow_{Π} is well-founded, one can thus normalize a term s (provided that s is normalizing) by computing the \rightarrow_{Π} -normal form t of s which is \mathcal{R} -root-stable according to Theorem 1, and then do the same recursively for the immediate subterms of t . Confluence of \mathcal{R} assures that the unique normal form of s will indeed be computed this way.

Example 6. As the forbidden pattern defined in Example 2 is (simple and) canonical, Theorem 1 yields that $\rightarrow_{\mathcal{R}, \delta}$ -normal forms are $\rightarrow_{\mathcal{R}}$ -head-normal forms. For instance we get $2nd(\text{inf}(0)) \rightarrow_{\Pi}^* s(0)$.

Example 7. Consider the TRS with \mathcal{R} and forbidden patterns Π from Example 3. We will prove below that \mathcal{R} is Π -terminating (cf. Example 12).

Furthermore we are able to show that every well-formed ground term that is reducible to a normal form in \mathcal{R} is reducible to the same normal form with $\rightarrow_{\mathcal{R}, \Pi}$ and that every $\rightarrow_{\mathcal{R}}$ -normal form is root-stable w.r.t. $\rightarrow_{\mathcal{R}}$.

5 Proving Termination

We provide another example of a result on a restricted class of forbidden patterns, this time concerning termination. We exploit the fact that, given a finite signature and linear h -patterns, a set of allowed contexts complementing each forbidden one can be constructed. Thus, we can transform a rewrite system with this kind of forbidden patterns into a standard (i.e., context-free) one by explicitly instantiating and embedding all rewrite rules (in a minimal way) in contexts (including a designated top-symbol representing the *empty* context) such that rewrite steps in these contexts are allowed.

To this end we propose a transformation that proceeds by iteratively instantiating and embedding rules in a minimal way. This is to say that the used substitutions map variables only to terms of the form $f(x_1, \dots, x_{ar(f)})$ and the contexts used for the embeddings have the form $g(x_1, \dots, x_{i-1}, \square, x_{i+1}, x_{ar(g)})$ for some function symbols $f \in \mathcal{F}$, $g \in \mathcal{F} \uplus \{\text{top}\}$ and some argument position i of f (resp. g). It is important to keep track of the position of the initial rule inside the embeddings. Thus we associate to each rule introduced by the transformation a position pointing to the embedded original rule. To all initial rules of \mathcal{R} we thus associate ε .

Note that it is essential to consider a new unary function symbol top_s for every sort $s \in S$ (of type $s \rightarrow s$) representing the empty context. This is illustrated by the following example.

Example 8. Consider the TRS given by

$$a \rightarrow f(a) \quad f(x) \rightarrow x$$

with $\mathcal{F} = \{a, f\}$ and the set of forbidden patterns $\Pi = \{\langle f(x), 1, h \rangle\}$. This system is not Π -terminating as we have

$$a \rightarrow_{\Pi} f(a) \rightarrow_{\Pi} a \rightarrow_{\Pi} \dots$$

Whether a subterm $s|_p = a$ is allowed for reduction by Π depends on its context. Thus, according to the idea of our transformation we try to identify all contexts $C[a]_p$ such that the reduction of a at position p is allowed by Π . However, there is no such (non-empty) context, although a may be reduced if C is the empty context. Moreover, there cannot be a rule $l \rightarrow r$ in the transformed system where $l = a$, since that would allow the reduction of terms that might be forbidden by Π . Our solution to this problem is to introduce a new function symbol top explicitly representing the empty context. Thus, in the example the transformed system will contain a rule $\text{top}(a) \rightarrow \text{top}(f(a))$.

Abusing notation we subsequently use only one top -symbol, while we actually mean the top_s -symbol of the appropriate sort. Moreover, in the following by rewrite rules we always mean rewrite rules with an associated (embedding) position, unless stated otherwise. All forbidden patterns used in this section (particularly in the lemmata) are linear here-patterns. We will make this general assumption explicit only in the more important results.

Definition 6 (instantiation and embedding). Let $\mathcal{F} = (S, \Omega)$ be a signature, let $\langle l \rightarrow r, p \rangle$ be a rewrite rule of sort s over \mathcal{F} and let Π be a set of forbidden patterns (linear, h). The set of minimal instantiated

and embedded rewrite rules $T_\Pi(\langle l \rightarrow r, p \rangle)$ (or just $T(\langle l \rightarrow r, p \rangle)$) is $T_\Pi^i(\langle l \rightarrow r, p \rangle) \uplus T_\Pi^e(\langle l \rightarrow r, p \rangle)$ where

$$\begin{aligned} T^e(\langle l \rightarrow r, p \rangle) &= \{ \langle C[l] \rightarrow C[r], i.p \rangle \mid C = f(x_1, \dots, x_{i-1}, \square, x_{i+1}, \dots, x_{ar(f)}), \\ &\quad f \in \Omega_{(s_1, \dots, s_{i-1}, s, s_{i+1}, \dots, s_{ar(f)})}, s', f \in \mathcal{F} \uplus \{\text{top}_s \mid s \in S\}, i \in \{1, \dots, ar(f)\}, \\ &\quad \exists \langle u, o, h \rangle \in \Pi. u|_q \theta = l\theta \wedge q \neq \varepsilon \wedge o = q.p \} \\ T_\Pi^i(\langle l \rightarrow r, p \rangle) &= \{ \langle l\sigma \rightarrow r\sigma, p \rangle \mid x\sigma = f(x_1, \dots, x_{ar(f)}), \text{sort}(x) = \text{sort}(f(x_1, \dots, x_{ar(f)})), \\ &\quad f \in \mathcal{F}, y \neq x \Rightarrow y\sigma = y, x \in RV_\Pi(l, p) \} \end{aligned}$$

and $RV_\Pi(l, p) = \{x \in \text{Var}(l) \mid \exists \langle u, o, h \rangle \in \Pi. \theta = \text{mgu}(u, l|_q) \wedge q.o = p \wedge x\theta \notin V\}$.

We also call the elements of $T(\langle l \rightarrow r, p \rangle)$ the one-step T -successors of $\langle l \rightarrow r, p \rangle$. The reflexive-transitive closure of the one-step T -successor relation is the many-step T -successor relation or just T -successor relation. We denote the set of all many-step T -successors of a rule $\langle l \rightarrow r, p \rangle$ by $T^*(\langle l \rightarrow r, p \rangle)$.

The set $RV_\Pi(l, p)$ of “relevant variables” is relevant in the sense that their instantiation might contribute to a matching by some (part of a) forbidden pattern term.

Note that in the generated rules $\langle l' \rightarrow r', p' \rangle$ in $T_\Pi(\langle l \rightarrow r, p \rangle)$, a fresh top_s -symbol can only occur at the root of both l' and r' or not at all, according to the construction in Definition 6.

Example 9. Consider the TRS (\mathcal{R}, Π) where $\mathcal{R} = (\{a, f, g\}, \{f(x) \rightarrow g(x)\})$ and the forbidden patterns Π are given by $\{\langle g(g(f(a))), 1.1, h \rangle\}$. $T(\langle f(x) \rightarrow g(x), \varepsilon \rangle)$ consists of the following rewrite rules.

$$\langle f(f(x)) \rightarrow g(f(x)), \varepsilon \rangle \quad (1)$$

$$\langle f(g(x)) \rightarrow g(g(x)), \varepsilon \rangle \quad (2)$$

$$\langle f(a) \rightarrow g(a), \varepsilon \rangle \quad (3)$$

$$\langle f(f(x)) \rightarrow f(g(x)), 1 \rangle \quad (4)$$

$$\langle g(f(x)) \rightarrow g(g(x)), 1 \rangle \quad (5)$$

Note that $RV_\Pi(f(x), \varepsilon) = \{x\}$ because $g(g(f(a)))_{1.1} = f(a)$ unifies with $f(x)$ and $\text{mgu } \theta$ where $x\theta = a \notin V$. On the other hand $RV_\Pi(f(f(x)), 1) = \emptyset$.

Lemma 2 (finiteness of instantiation and embedding). *Let $\langle l \rightarrow r, p \rangle$ be a rewrite rule and let Π be a set of forbidden patterns. The set of (many-step) instantiations and embeddings of $\langle l \rightarrow r, p \rangle$ (i.e. $T^*(\langle l \rightarrow r, p \rangle)$) is finite.*

The transformation we are proposing proceeds by iteratedly instantiating and embedding rewrite rules. The following definitions identify the rules for which no further instantiation and embedding is needed.

Definition 7 (Π -stable). *Let $\langle l \rightarrow r, p \rangle$ be a rewrite rule and let Π be a set of forbidden patterns. $\langle l \rightarrow r, p \rangle$ is Π -stable ($\text{stb}_\Pi(\langle l \rightarrow r, p \rangle)$ for short) if there is no context C and no substitution σ such that $C[l\sigma]_q|_{q'} = u\theta$ and $q.p = q'.o$ for any forbidden pattern $\langle u, o, h \rangle \in \Pi$ and any θ .*

Note that Π -stability is effectively decidable (for finite signatures and finite Π), since only contexts and substitutions involving terms not exceeding a certain depth depending on Π need to be considered.

Definition 8 (Π -obsolete). *Let $\langle l \rightarrow r, p \rangle$ be a rewrite rule and let Π be a set of forbidden patterns. $\langle l \rightarrow r, p \rangle$ is Π -obsolete ($\text{obs}_\Pi(\langle l \rightarrow r, p \rangle)$ for short) if there is a forbidden pattern $\Pi = \langle u, o, h \rangle$ such that $l|_q = u\theta$ and $p = q.o$.*

In Example 9, the rules (1), (2) and (4) are Π -stable, while rules (3) and (5) would be processed further. After two more steps e.g. a rule $\langle g(g(f(a))) \rightarrow g(g(g(a))), 1.1 \rangle$ is produced that is Π -obsolete.

The following lemmata state some properties of Π -stable rules.

Lemma 3. Let Π be a set of forbidden patterns and let $\langle l' = C[l\sigma]_p \rightarrow C[r\sigma]_p = r', p \rangle$ be a Π -stable rewrite rule corresponding to $l \rightarrow r$. If $s \rightarrow t$ with $l' \rightarrow r'$, then $s \rightarrow_{\Pi} t$ with $l \rightarrow r$.

Lemma 4. Let $\langle l \rightarrow r, p \rangle$ be a rule and Π be a set of forbidden patterns. If $T(\langle l \rightarrow r, p \rangle) = \emptyset$, then $\langle l \rightarrow r, p \rangle$ is either Π -stable or Π -obsolete.

Definition 9. Let $\mathcal{R} = (\mathcal{F}, R)$ be a TRS with an associated set of forbidden patterns Π where $\mathcal{F} = (S, \Omega)$. The transformation T maps TRSs with forbidden patterns to standard TRSs $T(\mathcal{R}, \Pi)$. It proceeds in 5 steps.

1. $R^{tmp} = \{ \langle l \rightarrow r, \varepsilon \rangle \mid l \rightarrow r \in R \}$
 $R^{acc} = \emptyset$
2. $R^{acc} = \{ \langle l \rightarrow r, p \rangle \in R^{tmp} \mid stb_{\Pi}(\langle l \rightarrow r, p \rangle) \}$
 $R^{tmp} = \{ \langle l \rightarrow r, p \rangle \in R^{tmp} \mid \neg stb_{\Pi}(\langle l \rightarrow r, p \rangle) \wedge \neg obs_{\Pi}(\langle l \rightarrow r, p \rangle) \}$
3. $R^{tmp} = \bigcup_{\langle l \rightarrow r, p \rangle \in R^{tmp}} T(\langle l \rightarrow r, p \rangle)$
4. If $R^{tmp} \neq \emptyset$ go to 2
5. $T(\mathcal{R}, \Pi) = (\mathcal{F} \uplus \{top_s \mid s \in S\}, \{l \rightarrow r \mid \langle l \rightarrow r, p \rangle \in R^{acc}\})$

In the transformation rewrite rules are iteratively created and collected in R^{tmp} (temporary rules). Those rules that are Π -stable and will thus be present in the final transformed system are collected in R^{acc} (accepted rules).

Lemma 5. Let \mathcal{R} be a rewrite system and Π be a set of forbidden (linear h -)patterns. If $s \rightarrow_{\mathcal{R}, \Pi} t$ for ground terms s and t , then $top(s) \rightarrow top(t)$ in $T(\mathcal{R}, \Pi)$.

Theorem 2. Let \mathcal{R} be a TRS and Π be a set of linear here-patterns. We have $s \rightarrow_{\Pi}^+ t$ for ground terms s and t if and only if $top(s) \rightarrow_{T(\mathcal{R}, \Pi)}^+ top(t)$.

Proof. The result is a direct consequence of Lemmata 3 and 5. □

Corollary 1. Let \mathcal{R} be a TRS and Π be a set of linear h -patterns. \mathcal{R} is ground terminating under Π if and only if $T(\mathcal{R}, \Pi)$ is ground terminating.

Note that the restriction to ground terms is crucial in Corollary 1. Moreover, ground termination and general termination do not coincide in general for rewrite systems with forbidden patterns (observe that the same is true for other important rewrite restrictions and strategies such as the outermost strategy).

Example 10. Consider the TRS $\mathcal{R} = (\mathcal{F}, R)$ given by $\mathcal{F} = \{a, f\}$ (where a is a constant) and R consisting of the rule

$$f(x) \rightarrow f(x).$$

Moreover, consider the set of forbidden patterns $\Pi = \{ \langle f(a), \varepsilon, h \rangle, \langle f(f(x)), \varepsilon, h \rangle \}$. Then \mathcal{R} is not Π -terminating because we have $f(x) \rightarrow_{\Pi} f(x)$ but it is Π -terminating on all ground terms, as can be shown by Theorem 2, since $T(\mathcal{R}, \Pi) = \emptyset$.

Example 11. Consider the TRS of Example 2. We use two sorts *NatList* and *Nat*, with function symbol types $2nd : NatList \rightarrow Nat$, $inf : Nat \rightarrow NatList$, $top : NatList \rightarrow NatList$ (note that another “top” symbol of type $Nat \rightarrow Nat$ is not needed here), $s : Nat \rightarrow Nat$, $0 : Nat$, $nil : NatList$ and $:$ of type $Nat, NatList \rightarrow NatList$. According to Definition 9, the rules of $T(\mathcal{R}, \Pi)$ are:

$$\begin{array}{ll} 2nd(inf(x)) \rightarrow 2nd(x : inf(s(x))) & 2nd(x : (y : zs)) \rightarrow y \\ top(inf(x)) \rightarrow top(x : inf(s(x))) & 2nd(x' : inf(x)) \rightarrow 2nd(x' : (x : inf(s(x)))) \\ top(x' : inf(x)) \rightarrow top(x' : (x : inf(s(x)))) & \end{array}$$

This system is terminating (and termination can be verified automatically, e.g. by AProVE [12]). Hence, by Corollary 1 also the TRS with forbidden patterns from Example 2 is ground terminating.

Example 12. The TRS \mathcal{R} and forbidden patterns Π from Example 3 yield the following system $T(\mathcal{R}, \Pi)$. For the sake of saving space we abbreviate app by a , take by t and inf by i .

$$\begin{array}{ll}
\text{top}(i(x)) \rightarrow \text{top}(x : i(s(x))) & t(y, i(x)) \rightarrow t(y, x : i(s(x))) \\
a(y, i(x)) \rightarrow a(y, x : i(s(x))) & \text{top}(a(i(x), y)) \rightarrow \text{top}(a(x : i(s(x)), y)) \\
t(a(i(x), y), z) \rightarrow t(a(x : i(s(x)), y), z) & t(z, a(i(x), y)) \rightarrow t(z, a(x : i(s(x)), y)) \\
a(a(i(x), y), z) \rightarrow a(a(x : i(s(x)), y), z) & a(z, a(i(x), y)) \rightarrow a(z, a(x : i(s(x)), y)) \\
\text{top}(a(x : xs, ys)) \rightarrow \text{top}(x : a(xs, ys)) & t(z, a(x : xs, ys)) \rightarrow t(z, x : a(xs, ys)) \\
a(a(x : xs, ys), z) \rightarrow a(x : a(xs, ys), z) & a(z, a(x : xs, ys)) \rightarrow a(z, x : a(xs, ys)) \\
a(x : i(zs), ys) \rightarrow x : a(i(zs), ys) & a(x : s(zs), ys) \rightarrow x : a(s(zs), ys) \\
a(x : (y : zs), ys) \rightarrow x : a(y : zs, ys) & a(\text{nil}, x) \rightarrow x \\
t(s(x), y : ys) \rightarrow t(x, ys) & t(0, y : ys) \rightarrow y \\
t(x, \text{nil}) \rightarrow 0 &
\end{array}$$

This system is terminating (and termination can be verified automatically, e.g. by AProVE [12]). Hence, again by Corollary 1 also the TRS with forbidden patterns from Example 3 is ground terminating.

6 Conclusion and Related Work

We have presented and discussed a novel approach to rewriting with context restrictions using forbidden patterns to specify forbidden/allowed positions in a term rather than arguments of functions as it was done previously in context-sensitivity. Thanks to their flexibility and parametrizability, forbidden patterns are applicable to a wider class of TRSs than traditional methods. In particular, position-based strategies and context-sensitive rewriting occur as special cases of such patterns.

For the TRSs in Examples 1 and 3 nice operational behaviours can be achieved by using rewriting with forbidden patterns. The restricted reduction relation induced by the forbidden patterns is terminating while still being powerful enough to compute (head-) normal forms. When using simpler approaches such as position-based strategies or context-sensitive rewriting in these examples, such operational properties cannot be achieved. For instance, consider Example 1. There is an infinite reduction sequence starting from $\text{inf}(x)$ with the property that every term has exactly one redex. Thus, non-termination is preserved under any reduction strategy (as strategies do not introduce new normal forms by definition). On the other hand, in order to avoid this infinite sequence using context-sensitive rewriting, we must set $2 \notin \mu(\cdot)$ (regardless of any additional reduction strategy). But in this case \rightarrow_μ does not compute head-normal forms.

In [15] *on-demand rewriting* was introduced, which is able to properly deal with the TRS of Example 1. This means that with the on-demand rewriting the reduction relation induced by the TRS of Example 1 can be restricted in a way such that it becomes terminating while still normal forms w.r.t. the restricted relation are head-normal forms w.r.t. the unrestricted one. Indeed, Example 1 was the main motivating example for the introduction of on-demand rewriting in [15].

However, for Example 3 we get that by restricting rewriting by the proposed forbidden patterns we obtain a terminating relation that is able to compute the normal forms of all well-formed ground terms. As the system is orthogonal, any outermost-fair reduction strategy, e.g. parallel outermost, is normalizing. Yet, by using such a strategy the relation still remains non-terminating. In particular, our forbidden

patterns approach yields an effective procedure for deciding whether a ground term is normalizing or not (it is not normalizing if its \rightarrow_{Π} -normal form is not an \rightarrow -normal form) for this example.

On the other hand, by using context-sensitive rewriting, termination can only be obtained if $2 \notin \mu(\cdot)$ which in turn implies that the term $0 : \text{app}(\text{nil}, \text{nil})$ cannot be normalized despite having a normal form $0 : \text{nil}$.

For Examples 1 and 3 effective strategies like parallel outermost or \mathcal{S}_{ω} of [3] are normalizing (though under either strategy there are still infinite derivations). We provide another example for which these strategies fail to provide normalization while the use of appropriate forbidden patterns yields normalization (and termination)

Example 13. Consider the TRS \mathcal{R} consisting of the following rules

$$\begin{array}{lll} a \rightarrow b & b \rightarrow a & c \rightarrow c \\ g(x, x) \rightarrow d & f(b, x) \rightarrow d & \end{array}$$

Using a parallel outermost strategy the term $g(a, b)$ is not reduced to its (unique) normal form d . Using \mathcal{S}_{ω} , $f(a, c)$ is not reduced to its (unique) normal form d .

However, it is easy to see that when using a $\Pi = \{\langle c, \varepsilon, h \rangle, \langle b, \varepsilon, h \rangle\}$, \rightarrow_{Π} is terminating and all \mathcal{R} -normal forms can be computed.

Note however, that the forbidden patterns used in Example 13 are not canonical. Thus it is not clear how to come up with such patterns automatically.

We argued that for our forbidden pattern approach it is crucial to identify reasonable classes of patterns that provide trade-offs between practical feasibility, simplicity and power, favoring either component to a certain degree. We have sketched and illustrated two approaches to deal with the issues of verifying termination and guaranteeing that it is possible to compute useful results (in our case original head-normal forms) with the restricted rewrite relation. To this end we proposed a transformation from rewrite systems with forbidden patterns to ordinary rewrite systems and showed that ground termination of both induced reduction relations coincide. Moreover, we provided a criterion based on canonical rewriting with forbidden patterns to ensure that normal forms w.r.t. the restricted reduction relation are original head-normal forms.

In particular “here”-patterns seem interesting as their use avoids context restrictions to be *non-local*. That is to say that whether a position is allowed for reduction or not depends only on a restricted “area” around the position in question regardless of the actual size of the whole object term. Note that this is not true for ordinary context-sensitive rewriting and has led to various complications in the theoretical analysis (cf. e.g. [11, Definition 23] [1, Definition 7] and [13, Definitions 1-3]).

Regarding future work, among many interesting questions and problems one particularly important aspect is to identify conditions and methods for the automatic (or at least automatically supported) synthesis of appropriate forbidden pattern restrictions.

Acknowledgements: We are grateful to the anonymous referees for numerous helpful and detailed comments and criticisms.

References

- [1] B. Alarcón, F. Emmes, C. Fuhs, J. Giesl, R. Gutiérrez, S. Lucas, P. Schneider-Kamp and R. Thiemann. Improving context-sensitive dependency pairs. In I. Cervesato, H. Veith and A. Voronkov, eds., *Proc. LPAR’08, Doha, Qatar, November 22-27, 2008*, LNCS 5330, pp. 636–651. Springer, 2008.

- [2] M. Alpuente, S. Escobar, B. Gramlich and S. Lucas. On-demand strategy annotations revisited: An improved on-demand evaluation Strategy. *Theoretical Computer Science*, 411(2):504–541, 2010.
- [3] S. Antoy and A. Middeldorp. A sequential reduction strategy. *Theoretical Computer Science*, 165(1):75–95, 1996.
- [4] F. Baader and T. Nipkow. *Term rewriting and All That*. Cambridge University Press, 1998.
- [5] M. Bezem, J. Klop, and R. de Vrijer, eds. *Term Rewriting Systems*. Cambridge Tracts in Theoretical Computer Science 55. Cambridge University Press, Mar. 2003.
- [6] F. Durán, S. Lucas, C. Marché, J. Meseguer and X. Urbain. Proving operational termination of membership equational programs. *Higher-Order and Symbolic Computation*, 21(1-2):59–88, 2008.
- [7] S. Eker. Term rewriting with operator evaluation strategies. *Electr. Notes Theor. Comput. Sci.*, 15:311–330 (Proc. WRLA’98, Abbaye des Prémontrés at Pont-à-Mousson, France, September 1998, C. Kirchner and H. Kirchner, eds.), 1998.
- [8] J. Endrullis and D. Hendriks. From outermost to context-sensitive rewriting. In R. Treinen, ed., *Proc. RTA’09, Brasilia, Brazil, June 29 - July 1, 2009*, LNCS 5595, pp. 305–319, Springer, June 2009.
- [9] W. Fokkink, J. Kamperman and P. Walters. Lazy rewriting on eager machinery. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 22(1):45–86, 2000.
- [10] K. Futatsugi, J. Goguen, J.-P. Jouannaud and J. Meseguer. Principles of OBJ2. In *Conference Record of the 12th Annual ACM Symposium on Principles of Programming Languages (POPL’85)*, pp. 52–66. ACM Press, 1985.
- [11] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4):379–427, Jul. 2004.
- [12] J. Giesl, P. Schneider-Kamp and R. Thiemann AProVE 1.2: Automatic termination proofs in the dependency pair framework. In U. Furbach and N. Shankar, eds., *Proc. IJCAR’06, Seattle, Washington, USA, August 17-20, 2006*, LNCS 4130, pp. 281–286. Springer, 2006.
- [13] B. Gramlich and S. Lucas. Generalizing Newman’s Lemma for left-linear rewrite systems. In F. Pfenning, ed., *Proc. RTA’06, Seattle, Washington, USA, August 12-14, 2006*, LNCS 4098, pp. 66–80. Springer, 2006.
- [14] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1), Jan. 1998.
- [15] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. PDP’01, September 5-7, 2001, Florence, Italy*, pp. 82–93. ACM, 2001.
- [16] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):294–343, 2002.
- [17] S. Lucas. Lazy rewriting and context-sensitive rewriting. *Electr. Notes Theor. Comput. Sci.* 64:234–254 (Proc. WFLP’01, Kiel, Germany, September 13-15, 2001, Selected Papers, M. Hanus, ed.), 2002.
- [18] M.J. O’Donnell. Computing in systems described by equations. LNCS 58, Springer, 1977.
- [19] F. Schernhammer and B. Gramlich. Termination of lazy rewriting revisited. *Electronic Notes in Theoretical Computer Science*, 204:35–51 (Final Proc. WRS’07, Jürgen Giesl, ed.), Apr. 2008.
- [20] F. Schernhammer and B. Gramlich. Characterizing and proving operational termination of deterministic conditional term rewriting systems. *Journal of Logic and Algebraic Programming*, Selected revised papers of NWPT’08, to appear, 2009.
- [21] F. Schernhammer and B. Gramlich. VMTL – a modular termination laboratory. In R. Treinen, ed., *Proc. RTA’09, Brasilia, Brazil, June 29 - July 1, 2009*, LNCS 5595, pp. 285–294, Springer, June 2009.
- [22] F. Schernhammer and B. Gramlich. On some implementation aspects of VMTL. In A. Geser and J. Waldmann, eds., *Proc. WST’09, Leipzig, Germany*, pp. 72-75, June 2009.