Extending Context-Sensitivity in Term Rewriting

Bernhard Gramlich and Felix Schernhammer

1 Introduction

Standard term rewriting is 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.).

In the last decade, context-sensitive rewriting ([11, 13]) emerged as a suitable framework to impose such kind of restrictions. Its basic underlying idea is to divide arguments of functions in those that may be evaluated eagerly and those that may not be evaluated. To some extent, this reflects the theoretical concept of strict and non-strict function arguments, thereby being a reasonable compromise between simplicity and flexibility of context restrictions.

However, it turns out that for some examples the desired operational properties of the restricted rewrite relation, i.e., being terminating (or at least normalizing) while still being powerful enough to compute normal forms w.r.t. the relation without restrictions, cannot be achieved.

Example 1 *Consider the following, cf. e.g. [12]:*

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

Here, allowing the eager reduction of the second argument of : *leads to non-termination, while disallowing its reduction leads to incompleteness in the sense that the term* $\mathsf{2nd}(\mathsf{inf}(x))$ *cannot be normalized within the context-sensitive rewrite relation, despite having a normal form.*

To solve this problem several alternative forms of restricted rewriting were suggested, each restricting the reduction of arguments of functions. Ideas range from explicitly modeling lazy evaluation (cf. e.g. [7, 14, 15]), to imposing constraints on the order of argument evaluation of functions (cf. e.g. [8, 5]), and to combinations of these concepts, also with standard context-sensitive rewriting (cf. e.g. [12, 2]). These generalized versions of context-sensitive rewriting are quite expressive and powerful, but on the other hand tend to be hard to analyze and understand and are also error-prone, due the subtlety of the strategic information specified.

Recently it turned out that, apart from using context-sensitivity as computation model for standard term rewriting (cf. e.g. [13, 11]), context-sensitive rewrite systems naturally appear as intermediate representations in many areas relying on transformations, such as program transformation and termination analysis of rewrite systems with conditions [4, 16] / under strategies [6].

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 a la [11, 13] only stability under contexts is lost.

In this work we will sketch and discuss a generalized approach to context-sensitivity (in the sense of [11, 13]) 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 rewrite 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 is forbidden or not (if so stability under substitutions might be lost)?

2 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.

We assume familiarity with the basic notions and notations in term rewriting, cf. e.g. [3]. $O(s)$ ($O_\Sigma(s)$) 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 $\mathcal{R} = (\Sigma, R)$ we divide symbols in Σ in defined symbols (denoted by D), which are those that occur as root symbols of the left-hand side of some rule in R and constructor symbols (denoted by C), which are given by $\Sigma \setminus D$.

Definition 1 (Forbidden Pattern) *A* forbidden pattern *(w.r.t. to a signature Σ) is a triple (t, p, λ), where $t \in \mathcal{T}(\Sigma, V)$ is term, p is position from $O(t)$ and $\lambda \in \{h, i, o\}$.*

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

We denote a *finite* set of forbidden patterns for a signature Σ by δ_Σ or just δ if Σ is clear from the context or irrelevant.

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 $(t, \_, \_)$.

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 (cf. Definition 2).

Definition 2 (Forbidden Pattern Rewrite Relation) *Let $\mathcal{R} = (\Sigma, R)$ be a TRS with forbidden patterns δ_Σ. We define the forbidden pattern rewrite relation $\rightarrow_{\mathcal{R},\delta_\Sigma}$, or $\rightarrow_\delta$ for short, as the smallest relation satisfying $s \rightarrow_{\mathcal{R},\delta_\Sigma} t$ if $s \xrightarrow{p}_{\mathcal{R}} t$ for some $p \in O_\Sigma(s)$ and there is no pattern $(u, q, \lambda) \in \delta_\Sigma$ such that*

- *$s = C[u\sigma]_{q'}$ and $p = q'.q$ if $\lambda = h$,*
- *$s = C[u\sigma]_{q'}$ and $p > q'.q$ for some q', if $\lambda = i$, and*
- *$s = C[u\sigma]_{q'}$ and $p < q'.q$ for some q', if $\lambda = o$.*

Note that for a finite rewrite system $\mathcal{R}$ (with finite signature Σ) and a finite set of forbidden patterns δ_Σ it is decidable whether $s \rightarrow_{\mathcal{R},\delta_\Sigma} t$ for terms s and t.

Example 2 *Consider the TRS from Example 1. If* $\delta = \{(x : (y : \mathsf{inf}(z)), 2.2, h)\}$*, then* $\rightarrow_\delta$ *can easily be shown to be terminating. Moreover,* $\rightarrow_\delta$ *is powerful enough to compute original head-normal forms if they exist (cf. Examples 3 and 4 below).*

Traditional context-sensitive rewriting (with a replacement map μ) occurs as special case of rewriting with forbidden patterns by defining δ to contain for each function symbol f and each number $j \in \{1, \ldots .ar(f)\} \setminus \mu(f)$ the forbidden patterns $f(x_1, \ldots, x_{ar(f)}, j, h)$ and $f(x_1, \ldots, x_{ar(f)}, j, i)$.

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

The definition of forbidden patterns and rewriting with forbidden patterns is very 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 tradeoff between power and simplicity. An example of such a restriction is the following adaption of canonicity to the forbidden pattern approach.

In context-sensitive rewriting the concept of canonicity is central, because it ensures completeness in the simulation of the unrestricted rewrite relation, i.e., it guarantees that the restricted relation is powerful enough to compute original head-normal forms. Obviously, the "outer" patterns, i.e., those which forbid reductions outside the identified position, and patterns of the form $(\_, \epsilon, h)$ are incompatible with the desired root-normalizing behaviour, thus we exclude them in the following definition.

In order to mimic the concept of canonicity in the realm of rewriting with forbidden patterns, we need to restrict the syntactical structure of applicable forbidden patterns. The goal of this restriction is that reducibility of subterms should only depend on the *outside context* and there only on the structure above but not parallel to the position in question. This idea is covered by the notion of *simple* forbidden patterns.

We say that a forbidden pattern $(t, p, \_)$ is *simple* if t is linear and p is a variable position or a maximal non-variable position in t and all positions in t parallel to p are variable positions as well, i.e., for each $q \in O(t)$ where $q > p$ or $p||q$, $t|_q$ is a variable. δ_Σ is simple if all its elements are simple.

Definition 3 (Canonical rewriting with forbidden patterns) *Let* $\mathcal{R} = (\Sigma, R)$ *where* $\Sigma = (\mathcal{C} \uplus \mathcal{D})$ *be a TRS with simple forbidden patterns* δ_Σ *(such w.l.o.g.* R *and* δ_Σ *have no variables in common). Then,* δ_Σ *is* $\mathcal{R}$-canonical *(or just* canonical*) if it does not contain forbidden patterns of the form* $(\_, \_, o)$ *or* $(\_, \epsilon, h)$ *and for all rules* $l \rightarrow r \in R$

- *there is no pattern* (t, p, λ) *such that* $root(t) \in \mathcal{D}$*,* $t|_q$ *and* l *unify for some* $q \in O_\Sigma(t)$*,* $q > \epsilon$*, and such that there exists a position* $q' \in O_\Sigma(l)$ *with* $q.q' = p$ *for* $\lambda = h$ *respectively* $q.q' > p$ *for* $\lambda = i$*; and*
- *there is no pattern* (t, p, λ) *such that* t *and* $l|_q$ *unify for some* $q \in O_\Sigma(l)$ *and such that there exists a position* q' *with* $q.q' \in O_\Sigma(l)$ *and* $q' = p$ *for* $\lambda = h$ *respectively* $q' > p$ *for* $\lambda = i$*,*

where $t' = t[x]_p$ *and* x *is a fresh variable.*

Proposition 1 *Let* $\mathcal{R} = (\Sigma, R)$ *be a* left-linear *TRS with* canonical *and* simple *forbidden patterns* δ_Σ*. Then* $\rightarrow_{\mathcal{R},\delta_\Sigma}$*-normal forms are* $\rightarrow_\mathcal{R}$*-head-normal forms.*

Example 3 *As the forbidden pattern defined in Example 2 is simple and canonical, Proposition 1 yields completeness w.r.t. head-normal forms of the rewrite relation with forbidden patterns.*

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 "here" patterns, i.e., having h as the last component, 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

in a standard (i.e., context-free) one by explicitly putting all rewrite rules (that may eventually be forbidden) in all their allowed contexts (including a designated top symbol representing the *empty* context).

In particular, for a given rewrite system with forbidden patterns $(\mathcal{R}, \delta)$, we identify for each rule $l \to r$ the potentially relevant patterns for this rule.

$$\delta_{l\to r} = \{(t, p, h) \in \delta \mid l \text{ and } t|_p \text{ unify}\}.$$

Moreover, we define the set $R^{ac}_{l\to r}$ of rules which represents the rule $l \to r$ embedded in all its *allowed contexts* w.r.t. δ:

$$\begin{array}{lll} R^{ac}_{l\to r} = \{C[l\sigma]_q \to C[r\sigma]_q & \mid & C[x] \in \mathcal{T}(\Sigma \cup \{\mathsf{top}\}, V), \mathsf{top} \text{ may only occur at} \\ & & \text{the root of C, and there is no position } o \text{ with} \\ & & q = o.p \text{ such that } C[l\sigma]_q|_o \text{ and } t \text{ unify for some} \\ & & (t, p, h) \in \delta_{l\to r} \text{ and } C[l\sigma]_q \text{ is minimal,} \\ & & \text{i.e., no proper subterm (above } q\text{) and/or more general term (w.r.t.} \\ & & \text{the instantiation ordering) has the same property}\} \end{array}$$

The precondition that δ only consists of linear "here" patterns ensures that $R^{ac}_{l\to r}$ is finite for each rule $l \to r$ (provided the signature of the initial rewrite system is finite). The transformed system $\mathcal{R}^{ac}$ of $(\mathcal{R}, \delta)$ ("ac" for allowed contexts) consists of the rules $\bigcup_{l\to r\in R} R^{ac}_{l\to r}$.

Indeed, whenever $s \to_{\mathcal{R},\delta} t$, then also $\mathsf{top}(s) \to_{\mathcal{R}^{ac}} \mathsf{top}(t)$, thus termination of the rewrite systems with forbidden patterns is implied by termination of the transformed system.

Example 4 *Transforming the TRS of Example 2 in the described way yields*

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{2nd}(\mathsf{inf}(x)) & \to & \mathsf{2nd}(x : \mathsf{inf}(\mathsf{s}(x))) & \mathsf{2nd}(x : (y : zs)) & \to & y \\ \mathsf{s}(\mathsf{inf}(x)) & \to & \mathsf{s}(x : \mathsf{inf}(\mathsf{s}(x))) & \mathsf{inf}(\mathsf{inf}(x)) & \to & \mathsf{inf}(x : \mathsf{inf}(\mathsf{s}(x))) \\ \mathsf{inf}(x) : y & \to & (x : \mathsf{inf}(\mathsf{s}(x))) : y & (x' : \mathsf{inf}(x)) : y & \to & (x' : (x : \mathsf{inf}(\mathsf{s}(x)))) : y \\ \mathsf{top}(\mathsf{inf}(x)) & \to & \mathsf{top}(x : \mathsf{inf}(\mathsf{s}(x))) & \mathsf{2nd}(x' : \mathsf{inf}(x)) & \to & \mathsf{2nd}(x' : x : \mathsf{inf}(\mathsf{s}(x))) \\ \mathsf{s}(x' : \mathsf{inf}(x)) & \to & \mathsf{s}(x' : (x : \mathsf{inf}(\mathsf{s}(x)))) & \mathsf{inf}(x' : \mathsf{inf}(x)) & \to & \mathsf{inf}(x' : (x : \mathsf{inf}(\mathsf{s}(x)))) \\ \mathsf{top}(x' : \mathsf{inf}(x)) & \to & \mathsf{top}(x' : (x : \mathsf{inf}(\mathsf{s}(x)))) & & & \end{array}$$

This system is terminating (proved with VMTL [17]), hence also the TRS with forbidden patterns from Example 2.

3 Conclusion and Future 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. Through 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.

Apart from these known special cases it is crucial to identify reasonable classes of patterns that provide tradeoffs between practical feasibility, simplicity and power, favoring either component to a certain degree. We have sketched and illustrated two approaches to deal with completeness (w.r.t. head-normal forms) and termination, respectively, of the obtained systems that are specified via such restricted classes of forbidden patterns.

In particular "here" patterns (i.e., patterns with h in their last component) 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. [9, Definition 23] [1, Definition 7] and [10, Definitions 1-3]).

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. Improving on-demand strategy annotations. In M. Baaz and A. Voronkov, eds., *Proc. LPAR'02, Tbilisi, Georgia*, LNCS 2514, pp. 1–18, 2002. Springer.

[3] F. Baader and T. Nipkow. *Term rewriting and All That.* Cambridge University Press, 1998.

[4] 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.

[5] S. Eker. Term rewriting with operator evaluation strategies. *Electr. Notes Theor. Comput. Sci.*, 15:311–330, 1998. Proc. WRLA'98, Abbaye des Pr'emontr'es at Pont-à-Mousson, France, September 1998.

[6] 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, Springer, June 2009. to appear.

[7] 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.

[8] 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.

[9] J. Giesl and A. Middeldorp. Transformation techniques for context-censitive rewrite systems. *Journal of Functional Programming*, 14(4):379–427, Jul. 2004.

[10] 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.

[11] S. Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1), Jan. 1998.

[12] S. Lucas. Termination of on-demand rewriting and termination of OBJ programs. In *Proc. PPDP'01, September 5-7, 2001, Florence, Italy*, pp. 82–93. ACM, 2001.

[13] S. Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):294–343, 2002.

[14] S. Lucas. Lazy rewriting and context-sensitive rewriting. *Electr. Notes Theor. Comput. Sci.*, 64:234–254, 2002. Proc. WFLP'01, Kiel, Germany, September 13-15, 2001, Selected Papers.

[15] F. Schernhammer and B. Gramlich. Termination of lazy rewriting revisited. *Electronic Notes in Theoretical Computer Science*, 204:35–51, Apr. 2008. Final Proc. WRS'07, Jürgen Giesl, ed., 2008

[16] F. Schernhammer and B. Gramlich. Characterizing and proving operational termination of deterministic conditional term rewriting systems. Technical Report E1852-2009-01, Theory and Logic Group, Institute of Computer Languages, TU Wien, Austria, Mar. 2009.

[17] F. Schernhammer and B. Gramlich. VMTL – a modular termination laboratory. In *Proc. RTA'09, Brasilia, Brazil, June 29 - July 1, 2009*, LNCS, Springer, June 2009. to appear.