Loops under Strategies ... Continued

René Thiemann* Christian Sternagel*

University of Innsbruck
Austria

{rene.thiemann, christian.sternagel}@uibk.ac.at

Jürgen Giesl†

RWTH Aachen University
Germany

giesl@informatik.rwth-aachen.de

Peter Schneider-Kamp

University of Southern Denmark
Denmark

petersk@imada.sdu.dk

While there are many approaches for automatically *proving* termination of term rewrite systems, up to now there exist only few techniques to *disprove* their termination automatically. Almost all of these techniques try to find *loops*, where the existence of a loop implies non-termination of the rewrite system. However, most programming languages use specific *evaluation strategies*, whereas loop detection techniques usually do not take strategies into account. So even if a rewrite system has a loop, it may still be terminating under certain strategies.

Therefore, our goal is to develop decision procedures which can determine whether a given loop is also a loop under the respective evaluation strategy. In earlier work, such procedures were presented for the strategies of innermost, outermost, and context-sensitive evaluation. In the current paper, we build upon this work and develop such decision procedures for important strategies like leftmost-innermost, leftmost-outermost, (max-)parallel-innermost, (max-)parallel-outermost, and forbidden patterns (which generalize innermost, outermost, and context-sensitive strategies). In this way, we obtain the first approach to disprove termination under these strategies automatically.

1 Introduction

Termination is an important property of term rewrite systems (TRSs). Therefore, much effort has been spent on developing and automating techniques for showing termination of TRSs. However, in order to detect bugs, it is at least as important to prove *non-termination*. Note that for rewriting under a strategy, the strategy has to be taken into account when checking for non-termination. The reason is that a TRS which is non-terminating when ignoring the strategy may still be terminating when considering the strategy. Thus, it is important to develop automated techniques to disprove termination of TRSs under strategies.

Most of the techniques for showing non-termination detect *loops* (for example, [4, 7, 8, 9, 13, 20, 21]). For a TRS $\mathcal{R}$, a loop is a derivation of the form $t \to_{\mathcal{R}}^{+} C[t\mu]$ for some context C and some substitution μ. To prove non-termination under a strategy $\mathcal{S}$, we may use a complete transformation $T_{\mathcal{S}}$ (e.g., [2, 14, 18]) where a TRS $\mathcal{R}$ terminates under the strategy $\mathcal{S}$ iff the TRS $T_{\mathcal{S}}(\mathcal{R})$ terminates when ignoring the strategy. After applying such a transformation, we may try to find a loop in the transformed system $T_{\mathcal{S}}(\mathcal{R})$. However, there are some drawbacks: The first problem is an increased search space, as loops of $\mathcal{R}$ are often transformed into much longer loops in $T_{\mathcal{S}}(\mathcal{R})$. Moreover, the complete transformations from [2, 14, 18] translate a loop $t \to_{\mathcal{R}}^{+} C[t\mu]$ into a non-looping infinite derivation in $T_{\mathcal{S}}(\mathcal{R})$, whenever

*These authors are supported by the FWF (Austrian Science Fund) project P22767-N13.

†This author is supported by the DFG (German Research Foundation) project GI 274/5-3.

$C \neq \Box$. These two problems were solved in [17, 19] by decision procedures which, given a loop in the original system $\mathcal{R}$, directly decide whether the loop is also a loop under the respective strategy. Here, [17] treats the innermost strategy whereas [19] deals with the context-sensitive [10] and the outermost strategy. Another problem is the availability of complete transformations. For the leftmost-innermost, parallel-innermost, and max-parallel-innermost strategy we know by [15] that a TRS is terminating under one of these strategies iff it is innermost terminating. Thus, we can use the decision procedure for innermost loops [17] to disprove termination under these strategies.[1] However, we are not aware of any complete transformation for the strategies leftmost-outermost, parallel-outermost, and max-parallel-outermost. Therefore, in this paper we build upon the direct methods of [17, 19] and give decision procedures for all these strategies (i.e., these procedures again decide whether a loop is also a loop under the strategy). Note that our decision procedures can also be extended to the context-sensitive case, e.g., to the leftmost-innermost context-sensitive strategy.

Finally, recently a generalization of innermost / outermost / context-sensitive rewriting has been introduced: *rewriting with forbidden patterns* [6]. In this paper we also develop a decision procedure for loops under forbidden patterns.

Before giving an overview on the contents of this paper, we present a motivating example.

Example 1. *Consider the following TRS (computing the factorial) which is a variant of [17, Ex. 1].*

$$\begin{aligned}
\mathsf{factorial}(y) &\to \mathsf{fact}(\mathsf{0},y) && (1) & \mathsf{0}\cdot y &\to \mathsf{0} && (7)\\
\mathsf{fact}(x,y) &\to \mathsf{if}(x == y, \mathsf{s}(\mathsf{0}), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x)) && (2) & \mathsf{s}(x)\cdot y &\to y + (x\cdot y) && (8)\\
\mathsf{if}(\mathsf{true},x,y) &\to x && (3) & x == y &\to \mathsf{eq}(\mathsf{chk}(x),\mathsf{chk}(y)) && (9)\\
\mathsf{if}(\mathsf{false},x,y) &\to y && (4) & \mathsf{eq}(x,x) &\to \mathsf{true} && (10)\\
\mathsf{0}+y &\to y && (5) & \mathsf{chk}(x) &\to \mathsf{false} && (11)\\
\mathsf{s}(x)+y &\to \mathsf{s}(x+y) && (6) & \mathsf{eq}(\mathsf{false},y) &\to \mathsf{false} && (12)
\end{aligned}$$

Here, $\mathsf{fact}(x,y)$ *computes* $\prod_{x\le z<y}(z+1) = (x+1)\cdot(x+2)\cdot\ldots\cdot y$. *The intended strategy is leftmost-outermost. Otherwise, rule (2) would directly cause non-termination. Moreover, this strategy is needed for the equality-test encoded by rules (9)–(12) (which takes at most three reductions). Nevertheless, we obtain the following looping leftmost-outermost reduction (the respective redexes are underlined):*

$$\begin{aligned}
t &= \underline{\mathsf{fact}(x,y)}\\
&\to \mathsf{if}(\underline{x == y}, \mathsf{s}(\mathsf{0}), \mathsf{fact}(\mathsf{s}(x),y)\cdot\mathsf{s}(x))\\
&\to \mathsf{if}(\mathsf{eq}(\underline{\mathsf{chk}(x)}, \mathsf{chk}(y)), \mathsf{s}(\mathsf{0}), \mathsf{fact}(\mathsf{s}(x),y)\cdot\mathsf{s}(x))\\
&\to \mathsf{if}(\underline{\mathsf{eq}(\mathsf{false}, \mathsf{chk}(y))}, \mathsf{s}(\mathsf{0}), \mathsf{fact}(\mathsf{s}(x),y)\cdot\mathsf{s}(x))\\
&\to \underline{\mathsf{if}(\mathsf{false}, \mathsf{s}(\mathsf{0}), \mathsf{fact}(\mathsf{s}(x),y)\cdot\mathsf{s}(x))}\\
&\to \mathsf{fact}(\mathsf{s}(x),y)\cdot\mathsf{s}(x)\\
&= C[t\mu]
\end{aligned}$$

[1] By performing all steps in a parallel-innermost step one after another, one can easily show that innermost loopingness is equivalent to parallel-innermost loopingness. Moreover, by [15] an innermost loop implies leftmost-innermost and max-parallel innermost non-termination. Yet, this does neither imply leftmost-innermost nor max-parallel-innermost loopingness. As an example, consider $\mathcal{R}' = \{\mathsf{a} \to \mathsf{f}(\mathsf{nloop},\mathsf{a})\} \cup \mathcal{R}$, where nloop is a non-terminating, but non-looping term w.r.t. $\mathcal{R}$. Then $\mathcal{R}'$ is innermost looping but neither leftmost-innermost looping nor max-parallel-innermost looping. This might be a motivation to also develop decision procedures for the various innermost strategies. Since the decision procedures for leftmost-, parallel-, and max-parallel-*outermost* loops directly also give us decision procedures for the corresponding *innermost* strategies, we will mention these results in the paper as well.

where $\mu = \{x/\mathsf{s}(x)\}$ and $C = \Box \cdot \mathsf{s}(x)$. Applying our new decision procedure developed in this paper will show that the above loop indeed is a leftmost-outermost loop, and hence, $\mathcal{R}$ does not terminate under the leftmost-outermost strategy.

The rest of the paper is structured as follows: In Section 2 we give the necessary preliminaries. Afterwards, in Section 3, we treat the special case of leftmost loops. Next, in Section 4, we consider parallel as well as max-parallel loops. Subsequently, we handle the more complicated case of loops under forbidden patterns in Section 5. Finally, in Section 6, we conclude.

2 Preliminaries

We only regard finite signatures and TRSs and refer to [1] for the basics of rewriting. We use ℓ, r, s, t, u for terms, f, g for function symbols, x, y for variables, μ, σ for substitutions, i, j, k, n, m for natural numbers, o, p, q for positions, and C, D for contexts. Here, contexts are terms which contain exactly one hole $\Box$. A position p is left of q iff $p = o\,i\,p'$, $q = o\,j\,q'$, and $i < j$. The set of variables is denoted by $\mathcal{V}$.

Throughout this paper we assume a fixed TRS $\mathcal{R}$ and we write $t \to_p s$ if one can reduce t to s at position p with $\mathcal{R}$, i.e., $t = C[\ell\sigma]$ and $s = C[r\sigma]$ for some rule $\ell \to r \in \mathcal{R}$, substitution σ, and context C with $C|_p = \Box$. In this case, the term $\ell\sigma$ is called a redex at position p. The reduction is leftmost / innermost / outermost, written $t \xrightarrow{\mathsf{l}}_p / \xrightarrow{\mathsf{i}}_p / \xrightarrow{\mathsf{o}}_p s$, iff p is a leftmost / innermost / outermost position of t where $t|_p$ is a redex. The leftmost-innermost reduction is defined as $\xrightarrow{\mathsf{li}}_p = \xrightarrow{\mathsf{l}}_p \cap \xrightarrow{\mathsf{i}}_p$. Similarly, the leftmost-outermost reduction is $\xrightarrow{\mathsf{lo}}_p = \xrightarrow{\mathsf{l}}_p \cap \xrightarrow{\mathsf{o}}_p$. If the position is irrelevant we just write $\to$, $\xrightarrow{\mathsf{l}}$, $\xrightarrow{\mathsf{i}}$, $\xrightarrow{\mathsf{o}}$, $\xrightarrow{\mathsf{li}}$, and $\xrightarrow{\mathsf{lo}}$, respectively.

We also consider parallel reductions. Here, $t \xrightarrow{\mathsf{p}}_{q_1,\ldots,q_k} s$ is a parallel reduction iff $k > 0$, the q_i's are pairwise parallel positions, and $t \to_{q_1} \ldots \to_{q_k} s$. The max-parallel reduction relation is defined by $t \xrightarrow{\mathsf{m}}_{q_1,\ldots,q_k} s$ iff $t \xrightarrow{\mathsf{p}}_{q_1,\ldots,q_k} s$ and t has no further redex at a position that is parallel to all positions $q_1,\ldots,q_k$. The (max-)parallel-innermost reduction is defined by $t \xrightarrow{\mathsf{mi}} / \xrightarrow{\mathsf{pi}}_{q_1,\ldots,q_k} s$ iff $t \xrightarrow{\mathsf{m}} / \xrightarrow{\mathsf{p}}_{q_1,\ldots,q_k} s$ and all redexes $t|_{q_i}$ are innermost redexes. The (max-)parallel-outermost reductions $\xrightarrow{\mathsf{mo}}$ and $\xrightarrow{\mathsf{po}}$ are defined analogously.

To shortly illustrate the difference between the strategies, observe that for the TRS $\mathcal{R}$ of Example 1, $x == y \xrightarrow{\mathsf{i}}{}^* / \xrightarrow{\mathsf{li}}{}^* / \xrightarrow{\mathsf{mi}}{}^* / \xrightarrow{\mathsf{o}}{}^* / \xrightarrow{\mathsf{mo}}{}^* \mathsf{true}$ whereas $x == y \not\xrightarrow{\mathsf{lo}}{}^* \mathsf{true}$. Moreover, $\mathsf{0} == \mathsf{0} \xrightarrow{\mathsf{i}}{}^* / \xrightarrow{\mathsf{li}}{}^* / \xrightarrow{\mathsf{mi}}{}^* \mathsf{false}$ but $\mathsf{0} == \mathsf{0} \xrightarrow{\mathsf{o}}{}^* / \xrightarrow{\mathsf{lo}}{}^* / \xrightarrow{\mathsf{mo}}{}^* \mathsf{false}$ is not possible.

Next, we consider rewriting under *forbidden patterns*.

Definition 2 (Rewriting under forbidden patterns [6])**.** *A* forbidden pattern *is a triple (ℓ, o, λ) for a term ℓ, position $o \in \mathcal{P}\mathsf{os}(\ell)$, and $\lambda \in \{h, a, b\}$. For a set Π of forbidden patterns the induced rewrite relation $\xrightarrow{\Pi}$ is defined by $t \xrightarrow{\Pi}_p s$ iff $t \to_p s$ and there is no pattern $(\ell, o, \lambda) \in \Pi$ such that there exist a position $o' \in \mathcal{P}\mathsf{os}(t)$, a substitution σ with $t|_{o'} = \ell\sigma$, and*

- *$p = o'o$, if $\lambda = h$,*
- *$p < o'o$, if $\lambda = a$, and*
- *$p > o'o$, if $\lambda = b$.*

So a forbidden pattern (ℓ, o, h) means that the redex may not be at position o in a subterm of the form $\ell\sigma$. Similarly, (ℓ, o, a) and (ℓ, o, b) mean that the redex may not be strictly above and not strictly below position o in a subterm of the form $\ell\sigma$, respectively.

Several strategies are expressible using $\xrightarrow{\Pi}$ [6]: Innermost rewriting is obtained by setting $\Pi = \{(\ell, \varepsilon, a) \mid \ell \to r \in \mathcal{R}\}$, outermost rewriting by using $\Pi = \{(\ell, \varepsilon, b) \mid \ell \to r \in \mathcal{R}\}$, $\mathcal{Q}$-restricted-rewriting [3] by $\Pi = \{(\ell, \varepsilon, a) \mid \ell \to r \in \mathcal{Q}\}$, and context-sensitive-rewriting [10] w.r.t. the replacement map μ can be expressed by $\Pi = \{(f(x_1,\ldots,x_n),i,\lambda) \mid f \in \Sigma, i \notin \mu(f), \lambda \in \{h,b\}\}$, where Σ is the set of all function symbols of the signature.

However, even more sophisticated examples can be treated by forbidden patterns.

Example 3. *Consider the following TRS from [6, 11].*

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

This TRS is not weakly normalizing, but still some terms like $\mathsf{2nd}(\mathsf{inf}(0))$ *have a normal form. One purpose of forbidden patterns is to restrict the rewrite relation in such a way that the restriction is terminating, but that all normal forms are still being reached. Here, context-sensitive rewriting is too restrictive, since forbidding rewriting in the second argument of ":" would not allow the reduction* $\mathsf{2nd}(\mathsf{inf}(0)) \to \mathsf{2nd}(0 : \mathsf{inf}(\mathsf{s}(0))) \to \mathsf{2nd}(0 : (\mathsf{s}(0) : \mathsf{inf}(\mathsf{s}(\mathsf{s}(0))))) \to \mathsf{s}(0)$. *However, we can use rewriting with forbidden patterns where* Π *only contains the pattern* $(x : (y : \mathsf{inf}(z)), 2.2, h)$. *Note that* $(x : (y : \mathsf{inf}(z)))|_{2.2} = \mathsf{inf}(z)$. *Then,* $\xrightarrow{\Pi}$ *is terminating, but the above reduction is still allowed.*

A TRS $\mathcal{R}$ is non-terminating iff there is an infinite derivation $t_1 \to t_2 \to \cdots$. It is leftmost-innermost / leftmost-outermost / parallel-innermost / parallel-outermost / max-parallel-innermost / max-parallel-outermost / forbidden pattern non-terminating iff there is such an infinite derivation using $\xrightarrow{\mathsf{li}}$ / $\xrightarrow{\mathsf{lo}}$ / $\xrightarrow{\mathsf{pi}}$ / $\xrightarrow{\mathsf{po}}$ / $\xrightarrow{\mathsf{mi}}$ / $\xrightarrow{\mathsf{mo}}$ / $\xrightarrow{\Pi}$ instead of $\to$. To describe the infinite derivation that is induced by a loop, we use context-substitutions.

Definition 4 (Context-substitutions [19])**.** *A* context-substitution *is a pair* (C,μ) *consisting of a context* C *and a substitution* μ. *The n-fold application of* (C,μ) *to a term t, written* $t(C,\mu)^n$, *is defined as follows.*

$$t(C,\mu)^0 = t \qquad\qquad t(C,\mu)^{n+1} = C[t(C,\mu)^n\mu]$$

For example, $t(C,\mu) = C[t\mu]$, $t(C,\mu)^2 = C[C[t\mu]\mu] = C[C\mu[t\mu^2]]$, etc. So in general, in $t(C,\mu)^n$, the context C is added n-times above t and t is instantiated by μ^n. Note that also the added contexts are instantiated by μ. For the term $t(C,\mu)^3$ this is illustrated in Figure 1. Context-substitutions have similar properties to contexts and substitutions.

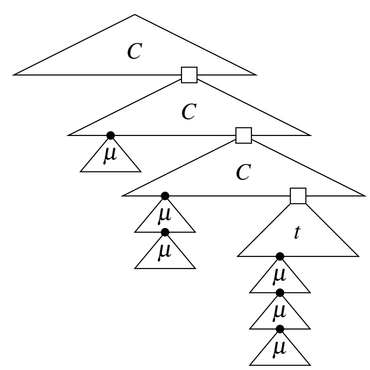

Figure 1: The term $t(C,\mu)^3$

Lemma 5 (Properties of context-substitutions [19])**.**

(i) $t(C,\mu)^n\mu = t\mu(C\mu,\mu)^n$.

(ii) $t(C,\mu)^m(C,\mu)^n = t(C,\mu)^{m+n}$.

(iii) If $C|_p = \Box$ *then* $t(C,\mu)^n|_{p^n} = t\mu^n$.

(iv) Whenever $t \to_q s$ *and* $C|_p = \Box$ *then* $t(C,\mu)^n \to_{p^n q} s(C,\mu)^n$.

Here, property (i) is similar to the fact that $C[t]\mu = C\mu[t\mu]$, and (ii) shows that context-substitutions can be combined just like substitutions where $\mu^m\mu^n = \mu^{m+n}$. Property (iii) shows that the n-fold application of (C,μ) to t yields a term containing the n-fold application of μ to t. Finally, stability and monotonicity of rewriting are used to show in (iv) that rewriting is closed under context-substitutions. Using context-substitutions we can now concisely present the infinite derivation resulting from a loop $t \to^+ C[t\mu] = t(C,\mu)$.

$$t(C,\mu)^0 \to^+ t(C,\mu)^0(C,\mu) = t(C,\mu)^1 \to^+ \cdots \to^+ t(C,\mu)^n \to^+ \cdots$$

So for every n, the positions of the reductions in the loop are prefixed by an additional p^n where p is the position of the hole in C, cf. Lemma 5 (iv).

Definition 6 ($\mathcal{S}$-loops [19])**.** *Let $\mathcal{S}$ be a strategy.*[2] *A loop $t_1 \to_{q_1} t_2 \to_{q_2} \cdots \to_{q_m} t_{m+1} = t_1(C,\mu)$ with $C|_p = \Box$ is an $\mathcal{S}$-*loop *iff the reduction $t_i(C,\mu)^n \to_{p^n q_i} t_{i+1}(C,\mu)^n$ respects the strategy $\mathcal{S}$ for all $i \leq m$ and all $n \in \mathbb{N}$.*

As a direct consequence of Definition 6, we can conclude that every $\mathcal{S}$-loop of a rewrite system $\mathcal{R}$ proves non-termination of $\mathcal{R}$ under the strategy $\mathcal{S}$. Moreover, Definition 6 also shows that being a loop is a modular property in the following sense.

Corollary 7 (Loops of intersection strategies)**.** *Let $\mathcal{S}$, $\mathcal{S}_1$, and $\mathcal{S}_2$ be strategies such that $\xrightarrow{\mathcal{S}}_p = \xrightarrow{\mathcal{S}_1}_p \cap \xrightarrow{\mathcal{S}_2}_p$ for all positions p. Then a loop is an $\mathcal{S}$-loop iff it is both an $\mathcal{S}_1$-loop and an $\mathcal{S}_2$-loop.*

Hence, to decide whether a loop is leftmost-innermost / leftmost-outermost, we just require a decision procedure for leftmost loops and a decision procedure for innermost / outermost loops. As decision procedures for innermost loops and outermost loops have already been developed [17, 19], it remains to construct a decision procedure for leftmost loops (see Section 3).

For rewriting with forbidden patterns, we observe that $\xrightarrow{\Pi}_p = \bigcap_{(\ell,o,\lambda)\in\Pi} \xrightarrow{\{(\ell,o,\lambda)\}}_p$, and hence, by Corollary 7 it suffices to consider loops w.r.t. single forbidden patterns which is the content of Section 5.

3 Leftmost Loops

Recall the definition of $\xrightarrow{\mathrm{l}}$. A leftmost reduction of all terms $t(C,\mu)^n$ at positions $p^n q$ requires that for no n there is a redex at a position left of $p^n q$. This is illustrated in Figure 2: The reduction of the subterm at the black position $p^n q$ respects the leftmost strategy iff $p^n q$ is leftmost. This is the case whenever there are no redexes at positions $\odot$.

We want to be able to decide whether all $p^n q$ point to leftmost redexes in the term $t(C,\mu)^n$. There are four possibilities why $p^n q$ might not point to a leftmost redex in that term. These cases are marked with (i)-(iv) in Figure 2.

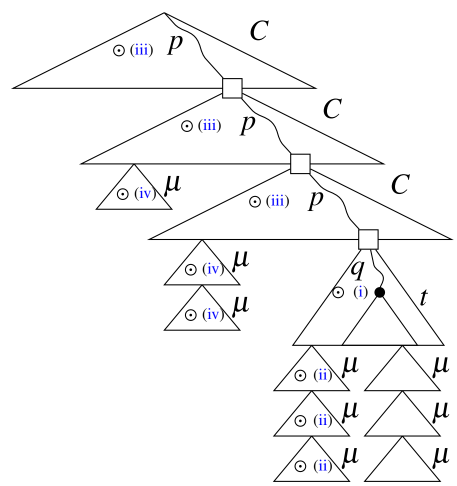

Figure 2: Leftmost redexes

(i) There might be a redex within $t\mu^n$ at a position $q' \in \mathcal{P}\mathsf{os}(t)$ which is left of q. Hence, we have to consider all finitely many subterms $u = t|_{q'}$ where q' is left of q and guarantee that $u\mu^n$ is no redex.

(ii) There might be a redex within $t\mu^n$ at a position $q' \in \mathcal{P}\mathsf{os}(t\mu^n) \setminus \mathcal{P}\mathsf{os}(t)$ which is left of q. Hence, this redex is of the form $u\mu^k$ for some $k \leq n$ and some subterm $u \trianglelefteq x\mu$ where x is a variable that occurs within some of v, $v\mu$, $v\mu^2$, $\ldots$ for some subterm $v = t|_{q'}$ where q' is left of q.[3] Note that there are only finitely many such variables x and hence, again we obtain a finite set of terms where for each of these terms u and each n we have to guarantee that $u\mu^n$ is not a redex.

[2] In this paper we use a rather liberal definition of a strategy: a strategy is just a restriction of the rewrite relation.

[3] It does not suffice to only consider the variables x that occur in v and $v\mu$. This can be seen for $v = y$ and $\mu = \{y/y_1, y_1/y_2, y_2/y_3, \ldots y_{n-1}/x, x/f(\ldots)\}$. Here, x does neither occur in v nor in $v\mu$, but in $v\mu^n$. Hence, the potential redex $f(\ldots)$ is detected only after n iterations.

(iii) There might be a redex where the root is within C and left of the path p. Here, we have to consider all finitely many subterms $u = C|_{p'}$ where p' is left of p and guarantee that $u\mu^n$ is not a redex.

(iv) In analogy to (ii) we also have to consider redexes within μ where now the variables x are taken from the subterms $u = C|_{p'}$ where p' is left of p.

To summarize, we generate a finite set U of terms u such that (a) and (b) are equivalent:

(a) For every n, the reduction $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ is leftmost.

(b) There is no $u \in U$ and no number n such that $u\mu^n$ is a redex.

Note that the question whether $u\mu^n$ is a redex for some n can be formulated as the kind of matching problem that was encountered for deciding innermost loops.

Definition 8 (Matching problems [17])**.** *A* matching problem *is a pair* $(u > \ell, \mu)$. *It is* solvable *iff there are n and σ such that $u\mu^n = \ell\sigma$.*

Thus, following the possibilities (i) - (iv) above, we can formally define a set of matching problems to analyze leftmost reductions.

Definition 9 (Leftmost matching problems)**.** *The set of* leftmost matching problems *for a reduction $t \to_q t'$ and a context-substitution (C,μ) with $C|_p = \Box$ is defined as the set consisting of:*

$(u > \ell, \mu)$ *for each* $\ell \to r \in \mathcal{R}$ *and* $q' \in \mathcal{P}\mathsf{os}(t)$ *where* q' *is left of* q*, and* $u = t|_{q'}$

$(u > \ell, \mu)$ *for each* $\ell \to r \in \mathcal{R}$ *and* $q' \in \mathcal{P}\mathsf{os}(t)$ *where* q' *is left of* q*,* $x \in \bigcup_{i \in \mathbb{N}} \mathcal{V}(t|_{q'}\mu^i)$*, and* $u \trianglelefteq x\mu$

$(u > \ell, \mu)$ *for each* $\ell \to r \in \mathcal{R}$ *and* $p' \in \mathcal{P}\mathsf{os}(C)$ *where* p' *is left of* p*, and* $u = C|_{p'}$

$(u > \ell, \mu)$ *for each* $\ell \to r \in \mathcal{R}$ *and* $p' \in \mathcal{P}\mathsf{os}(C)$ *where* p' *is left of* p*,* $x \in \bigcup_{i \in \mathbb{N}} \mathcal{V}(C|_{p'}\mu^i)$*, and* $u \trianglelefteq x\mu$

Note that the sets of variables in the second and fourth case are finite and can easily be computed. The above considerations prove the following theorem.

Theorem 10 (Soundness of leftmost matching problems)**.** *Let $t \to_q t'$ and let (C,μ) be a context-substitution such that $C|_p = \Box$. All reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are leftmost iff none of the leftmost matching problems for $t \to_q t'$ and (C,μ) is solvable.*

Using Theorem 10 in combination with the decision procedures for matching problems yields the following corollary.

Corollary 11 (Leftmost loops are decidable)**.** *Let there be a loop $t_1 \to_{q_1} t_2 \to_{q_2} \cdots \to_{q_m} t_{m+1} = t_1(C,\mu)$ with $C|_p = \Box$. Then it is decidable whether the loop is a leftmost loop.*

Combining Corollary 11 and Corollary 7 with the decision procedures for innermost and outermost loops of [17,19] yields decision procedures which determine whether a given loop is a leftmost-innermost loop or a leftmost-outermost loop: for each loop construct the leftmost matching problems, ensure that all these matching problems are not satisfiable (then leftmost reductions are guaranteed), and moreover use the decision procedures of [17, 19] to further ensure that the loop is an innermost or outermost loop.

Corollary 12 (Leftmost-innermost and leftmost-outermost loops are decidable)**.** *Let there be a loop $t_1 \to_{q_1} t_2 \to_{q_2} \cdots \to_{q_m} t_{m+1} = t_1(C,\mu)$ with $C|_p = \Box$. Then the following two questions are decidable.*

- *Is the loop a leftmost-innermost loop?*
- *Is the loop a leftmost-outermost loop?*

Example 13. *Using Corollary 12, we can decide that the loop given in Example 1 is a leftmost loop, since for this loop, the set of leftmost matching problems is empty (as there is never a position left of the used redex). Moreover, by the results of [17, 19] we can decide that the loop is an outermost loop, but not an innermost loop. Hence, the loop is a leftmost-outermost loop, but not a leftmost-innermost loop.*

Example 14. *We consider the following loop for the TRS of Example 1*

$$\begin{aligned}
t &= \underline{\mathsf{fact}(x,y)}\\
&\to \mathsf{if}(\underline{x == y}, \mathsf{s}(0), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x))\\
&\to \mathsf{if}(\mathsf{eq}(\underline{\mathsf{chk}(x)}, \mathsf{chk}(y)), \mathsf{s}(0), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x))\\
&\to \mathsf{if}(\mathsf{eq}(\mathsf{false}, \underline{\mathsf{chk}(y)}), \mathsf{s}(0), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x))\\
&\to \mathsf{if}(\underline{\mathsf{eq}(\mathsf{false},\mathsf{false})}, \mathsf{s}(0), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x))\\
&\to \mathsf{if}(\mathsf{false}, \mathsf{s}(0), \mathsf{fact}(\mathsf{s}(x),y)\cdot \mathsf{s}(x))\\
&= C[t\mu]
\end{aligned}$$

where $C = \mathsf{if}(\mathsf{false}, \mathsf{s}(0), \Box\cdot \mathsf{s}(x))$ *and* $\mu = \{x/\mathsf{s}(x)\}$. *We decide that this loop is a leftmost loop by constructing the leftmost matching problems*

- $(\mathsf{false} \gtrdot \ell, \mu)$ *for all left-hand sides* ℓ *(due to the reduction* $\mathsf{if}(\mathsf{eq}(\mathsf{false}, \underline{\mathsf{chk}(y)}),\dots) \to \dots$*)*
- $(\mathsf{false} \gtrdot \ell, \mu)$, $(0 \gtrdot \ell, \mu)$, *and* $(\mathsf{s}(0) \gtrdot \ell, \mu)$ *for all left-hand sides* ℓ *(since* $C = \mathsf{if}(\mathsf{false}, \mathsf{s}(0), \Box\cdot\dots)$*)*

and observing that none of them is solvable. This loop is also an innermost loop, but not an outermost loop and hence, it is a leftmost-innermost loop, but not a leftmost-outermost loop.

Whereas in the previous two examples it is rather easy to see that the loops are leftmost, since the leftmost matching problems are trivially not solvable, we now present two more examples where the resulting matching problems are more involved.

Example 15. *Consider the TRS*

$$\begin{aligned}
\mathsf{f}(x,y,z) &\to \mathsf{h}(\mathsf{g}(x,y), \mathsf{f}(y,z,z))\\
\mathsf{g}(x,x) &\to x
\end{aligned}$$

and the loop $t = \mathsf{f}(x,y,z) \to \mathsf{h}(\mathsf{g}(x,y), \mathsf{f}(y,z,z)) = C[t\mu]$ *for* $C = \mathsf{h}(\mathsf{g}(x,y), \Box)$ *and* $\mu = \{x/y, y/z\}$. *Here, we construct the non-solvable leftmost matching problems* $(u \gtrdot \ell, \mu)$ *for all left-hand sides* ℓ *and* $u \in \{x,y,z\}$. *But additionally we construct the leftmost matching problem* $(\mathsf{g}(x,y) \gtrdot \mathsf{g}(x,x), \mu)$ *which is solvable, since* $\mathsf{g}(x,y)\mu^2 = \mathsf{g}(y,z)\mu = \mathsf{g}(z,z) = \mathsf{g}(x,x)\sigma$ *for* $\sigma = \{x/z\}$. *Hence, the loop is not a leftmost loop.*

Example 16. *Consider the TRS*

$$\begin{aligned}
\mathsf{f}(x,y,z) &\to \mathsf{h}(\mathsf{g}(x), \mathsf{f}(y,z,\mathsf{s}(x)))\\
\mathsf{g}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x)))) &\to x
\end{aligned}$$

and the loop $t = \mathsf{f}(x,y,z) \to \mathsf{h}(\mathsf{g}(x), \mathsf{f}(y,z,\mathsf{s}(x))) = C[t\mu]$ *for* $C = \mathsf{h}(\mathsf{g}(x), \Box)$ *and* $\mu = \{x/y, y/z, z/\mathsf{s}(x)\}$. *Here, we construct the non-solvable leftmost matching problems* $(u \gtrdot \ell, \mu)$ *for all left-hand sides* ℓ *and* $u \in \{x,y,z,\mathsf{s}(x)\}$. *But additionally we construct the leftmost matching problem* $(\mathsf{g}(x) \gtrdot \mathsf{g}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x)))), \mu)$ *which is solvable, since* $\mathsf{g}(x)\mu^9 = \mathsf{g}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x))))$. *Hence, the loop is not a leftmost loop.*

4 Parallel and Max-Parallel Loops

For the parallel innermost / outermost strategies it suffices to use the decision procedures for innermost- and outermost loops. The reason is that $t(C,\mu)^n \xrightarrow{\mathsf{p}}_{p^n q_1,\dots,p^n q_k} t'(C,\mu)^n$ is a $\xrightarrow{\mathsf{pi}}$ / $\xrightarrow{\mathsf{po}}$-reduction iff for every $1 \le i \le k$ there is some s_i such that $t(C,\mu)^n \to_{p^n q_i} s_i$ is an innermost / outermost reduction.

Hence, for the rest of the section we consider the max-parallel strategies $\xrightarrow{\mathsf{mi}}$ and $\xrightarrow{\mathsf{mo}}$. Again, the innermost or outermost aspect can be decided by the respective decision procedures using a variant of Corollary 7 where one allows parallel rewrite steps. It remains to consider the max-parallel aspect, i.e., we have to decide whether $t(C,\mu)^n \xrightarrow{\mathsf{m}}_{p^n q_1,\dots,p^n q_k} t'(C,\mu)^n$ for all n.

Here, we essentially proceed as in the leftmost case, where we replace the condition that some position is left of p or q by the condition that it is parallel to p or to each q_i.

Definition 17 (Max-parallel matching problems)**.** *The set of* max-parallel matching problems *for a reduction $t \xrightarrow{\mathsf{p}}_{q_1,\dots,q_k} t'$ and a context-substitution (C,μ) with $C|_p = \Box$ is defined as the set consisting of:*

$(u \gtrdot \ell, \mu)$ *for each $\ell \to r \in \mathcal{R}$ and $q' \in \mathcal{P}\mathsf{os}(t)$ where q' is parallel to all positions q_i, and $u = t|_{q'}$*

$(u \gtrdot \ell, \mu)$ *for each $\ell \to r \in \mathcal{R}$ and $q' \in \mathcal{P}\mathsf{os}(t)$ where q' is parallel to all q_i, $x \in \bigcup_{i\in\mathbb{N}} \mathcal{V}(t|_{q'}\mu^i)$, and $u \trianglelefteq x\mu$*

$(u \gtrdot \ell, \mu)$ *for each $\ell \to r \in \mathcal{R}$ and $p' \in \mathcal{P}\mathsf{os}(C)$ where p' is parallel to p, and $u = C|_{p'}$*

$(u \gtrdot \ell, \mu)$ *for each $\ell \to r \in \mathcal{R}$ and $p' \in \mathcal{P}\mathsf{os}(C)$ where p' is parallel to p, $x \in \bigcup_{i\in\mathbb{N}} \mathcal{V}(C|_{p'}\mu^i)$, and $u \trianglelefteq x\mu$*

Using this finite set of matching problems we again obtain a decision procedure.

Theorem 18 (Soundness of max-parallel matching problems)**.** *Let $t \xrightarrow{\mathsf{p}}_{q_1,\dots,q_k} t'$ and let (C,μ) be a context-substitution such that $C|_p = \Box$. All reductions $t(C,\mu)^n \xrightarrow{\mathsf{p}}_{p^n q_1,\dots,p^n q_k} t'(C,\mu)^n$ are max-parallel iff none of the max-parallel matching problems for $t \xrightarrow{\mathsf{p}}_{q_1,\dots,q_k} t'$ and (C,μ) is solvable.*

Corollary 19 (Max-parallel loops are decidable)**.** *Let $t_1 \xrightarrow{\mathsf{p}}_{q^1_1,\dots,q^1_{k_1}} t_2 \xrightarrow{\mathsf{p}}_{q^2_1,\dots,q^2_{k_2}} \cdots \xrightarrow{\mathsf{p}}_{q^m_1 \dots q^m_{k_m}} t_{m+1}$ be a loop with $t_{m+1} = t_1(C,\mu)$ and $C|_p = \Box$. Then the following questions are decidable.*

- *Is the loop a max-parallel loop?*
- *Is the loop a parallel-innermost loop? Is it a max-parallel-innermost loop?*
- *Is the loop a parallel-outermost loop? Is it a max-parallel-outermost loop?*

Note that in the corollary we did not list the question "Is the loop a parallel loop?" since every loop is trivially also a parallel loop.

Example 20. *It is easy to see that neither the loop of Example 1 nor the loop of Example 14 is a max-parallel loop. The reason is that both loops violate the max-parallel strategy already in the second reduction step. However, the TRS of Example 1 is both max-parallel-outermost and -innermost looping which is proved by the following two loops which could be obtained automatically using a loop detection technique and our decision procedure of Theorem 18.*

The max-parallel-outermost loop needs two parallel reductions:

$$
\begin{aligned}
t &= \mathsf{if}(\underline{\mathsf{eq}(\mathsf{false},\mathsf{false})},1,\mathsf{if}(\mathsf{eq}(\underline{\mathsf{chk}(\mathsf{s}(x))},\underline{\mathsf{chk}(y)}),1,\mathsf{if}(\underline{\mathsf{s}^2(x) == y},1,\underline{\mathsf{fact}(\mathsf{s}^3(x),y)} \cdot \mathsf{s}^3(x)) \cdot \mathsf{s}^2(x)) \cdot \mathsf{s}(x))\\
&\xrightarrow{\mathsf{mo}} \underline{\mathsf{if}(\mathsf{false},1,\mathsf{if}(\mathsf{eq}(\mathsf{false},\mathsf{false}),1,\mathsf{if}(\mathsf{eq}(\mathsf{chk}(\mathsf{s}^2(x)),\mathsf{chk}(y)),1,\mathsf{if}(\mathsf{s}^3(x) == y,1,\mathsf{fact}(\mathsf{s}^4(x),y) \cdot \mathsf{s}^4(x)) \cdot \mathsf{s}^3(x)) \cdot \mathsf{s}^2(x)) \cdot \mathsf{s}(x))}\\
&\xrightarrow{\mathsf{mo}} \mathsf{if}(\mathsf{eq}(\mathsf{false},\mathsf{false}),1,\mathsf{if}(\mathsf{eq}(\mathsf{chk}(\mathsf{s}^2(x)),\mathsf{chk}(y)),1,\mathsf{if}(\mathsf{s}^3(x) == y,1,\mathsf{fact}(\mathsf{s}^4(x),y) \cdot \mathsf{s}^4(x)) \cdot \mathsf{s}^3(x)) \cdot \mathsf{s}^2(x)) \cdot \mathsf{s}(x)\\
&= C[t\mu]
\end{aligned}
$$

where $C = \Box \cdot \mathsf{s}(x)$, $\mu = \{x/\mathsf{s}(x)\}$, *and where* 1 *abbreviates* $\mathsf{s}(0)$. *For the max-parallel-innermost loop one parallel reduction suffices:*

$$\begin{aligned} t &= \mathsf{if}(\underline{\mathsf{eq}(\mathsf{false},\mathsf{false})}, 1, \mathsf{if}(\mathsf{eq}(\underline{\mathsf{chk}(\mathsf{s}(x))}, \underline{\mathsf{chk}(y)}), 1, \mathsf{if}(\underline{\mathsf{s}^2(x) == y}, 1, \underline{\mathsf{fact}(\mathsf{s}^3(x), y)} \cdot \mathsf{s}^3(x)) \cdot \mathsf{s}^2(x)) \cdot \mathsf{s}(x)) \\ &\xrightarrow{\mathsf{mi}} \mathsf{if}(\mathsf{false}, 1, \mathsf{if}(\mathsf{eq}(\mathsf{false},\mathsf{false}), 1, \mathsf{if}(\mathsf{eq}(\mathsf{chk}(\mathsf{s}^2(x)), \mathsf{chk}(y)), 1, \mathsf{if}(\mathsf{s}^3(x) == y, 1, \mathsf{fact}(\mathsf{s}^4(x), y) \cdot \mathsf{s}^4(x)) \cdot \mathsf{s}^3(x)) \cdot \mathsf{s}^2(x)) \cdot \mathsf{s}(x)) \\ &= C[t\mu] \end{aligned}$$

where $C = \mathsf{if}(\mathsf{false}, 1, \Box \cdot \mathsf{s}(x))$ *and* $\mu = \{x/\mathsf{s}(x)\}$.

5 Loops for Rewriting with Forbidden Patterns

For rewriting with forbidden patterns we have to investigate for given t, t', C, μ with $C|_p = \Box$ and $t \to_q t'$, whether all reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are allowed w.r.t. some fixed forbidden pattern (ℓ, o, λ). In other words, we have to check whether

$$\text{there are } n,\ o', \text{ and } \sigma \text{ with } t(C,\mu)^n|_{o'} = \ell\sigma \text{ and } \begin{cases} p^n q = o'o, & \text{if } \lambda = h, \\ p^n q < o'o, & \text{if } \lambda = a, \text{ and} \\ p^n q > o'o, & \text{if } \lambda = b. \end{cases} \tag{13}$$

In the subsections 5.1-5.3, we investigate the three cases of λ. We show that for all of them, (13) is decidable. To this end, we reuse algorithms that have been developed to decide innermost and outermost loops.

5.1 Deciding Loops for Forbidden Patterns of Type $(\cdot, \cdot, h)$

We start with the easiest case where $\lambda = h$. Given p, q, and o, here we first want to figure out whether there are n and o' such that the condition $p^n q = o'o$ of (13) is satisfied. Obviously, once n has been fixed, then o' is uniquely determined. Therefore, we first compute n_0 as the minimal value of n such that $p^n q = o'o$ is satisfied for some o' and then uniquely determine o'_0 such that $p^{n_0} q = o'_0 o$.

This can be done as follows. If $p = \varepsilon$, then one can set $n_0 = 0$ and just has to determine whether q has o as a suffix. Otherwise, one has to ensure that $p^n q$ is at least as long as o. This is done by choosing $n_0 = \lceil \frac{|o| \dot{-} |q|}{|p|} \rceil$. If there is an n where $\exists o'. p^n q = o'o$ can be satisfied, then n_0 is the minimal such number. Here, "$\dot{-}$" is the subtraction on natural numbers where $x \dot{-} y = \max(x - y, 0)$. Afterwards one just checks whether $p^{n_0} q$ contains o as suffix. If this holds, then there is obviously a unique o'_0 such that $p^{n_0} q = o'_0 o$. Otherwise, there cannot be any n and o' which satisfy $p^n q = o'o$. The reason is that for any solution $p^n q = o'o$ we know that $n \geq n_0$ and hence, $p^{n-n_0} p^{n_0} q = p^n q = o'o$ shows that o is a suffix of $p^{n_0} q$ as $|p^{n_0} q| \geq |o|$.

In this way we can compute the minimal number n_0 and the corresponding o'_0 such that $p^{n_0} q = o'_0 o$, or we detect that $p^n q = o'o$ is unsatisfiable. In the latter case we are finished since we know that the forbidden pattern will not restrict any of the desired reductions. In the former case we can represent the set of solutions of $p^n q = o'o$ conveniently:

$$\{(n, o') \mid p^n q = o'o\} = \{(k + n_0, p^k o'_0) \mid k \in \mathbb{N}\}$$

Hence, it remains to check whether there are $k \in \mathbb{N}$ and σ with $t(C,\mu)^{k+n_0}|_{p^k o'_0} = \ell\sigma$. Note that this problem can be simplified using Lemma 5:

$$t(C,\mu)^{k+n_0}|_{p^k o'_0} = t(C,\mu)^{n_0}(C,\mu)^k|_{p^k}|_{o'_0} = t(C,\mu)^{n_0}\mu^k|_{o'_0} = (t(C,\mu)^{n_0}|_{o'_0})\mu^k$$

Thus, for the concrete terms $u = t(C,\mu)^{n_0}|_{o'_0}$ and ℓ, we have to decide whether there are k and σ such that $u\mu^k = \ell\sigma$.

Definition 21 ((ℓ,o,h) matching problems)**.** *The set of* (ℓ,o,h) matching problems *for a term t, a position $q \in \mathcal{P}\mathsf{os}(t)$, and a context-substitution (C,μ) with $C|_p = \Box$ is defined as*

- *the empty set, if there are no n and o' such that $p^n q = o'o$*
- $\{(t(C,\mu)^{n_0}|_{o'_0} \geqslant \ell,\mu)\}$*, otherwise, where n_0 and o'_0 form the unique minimal solution to the equation $p^n q = o'o$*

By the discussion above, we have proved the following theorem.

Theorem 22 (Soundness of (ℓ,o,h) problems)**.** *Let $t \to_q t'$ and let (C,μ) be a context-substitution such that $C|_p = \Box$. All reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are allowed w.r.t. the pattern (ℓ,o,h) iff none of the (ℓ,o,h) matching problems for t, q, and (C,μ) is solvable.*

Using Theorem 22 in combination with the decision procedure of [17] for solvability of matching problems, one can decide whether all reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are allowed w.r.t. the pattern (ℓ,o,h).

Example 23. *We consider the TRS of Example 3 and $\Pi = \{(x : (y : \mathsf{inf}(z)), 2.2, h)\}$. Here, we have the looping reduction $t = \mathsf{inf}(x) \to x : \mathsf{inf}(\mathsf{s}(x)) = C[t\mu]$ for $C = x : \Box$ and $\mu = \{x/\mathsf{s}(x)\}$. Hence, to investigate whether this loop is a Π-loop, we have $p = 2$ as the position of $\Box$ in C, $q = \varepsilon$ since the reduction is on the root position of t, and $o = 2.2$. Then we compute $n_0 = \lceil \frac{|o| \dot{-} |q|}{|p|} \rceil = \lceil \frac{2 \dot{-} 0}{1} \rceil = 2$ and observe that $p^{n_0} q = 2.2$ has $o = 2.2$ as a suffix, and set $o'_0 = \varepsilon$. Hence, we construct the matching problem $(t(C,\mu)^{n_0}|_{o'_0} \geqslant \ell,\mu) = (\mathsf{inf}(x)(C,\mu)^2 \geqslant \ell,\mu) = (x : (\mathsf{s}(x) : \mathsf{inf}(\mathsf{s}(\mathsf{s}(x)))) \geqslant x : (y : \mathsf{inf}(z)),\mu)$ which is solvable because $(x : (\mathsf{s}(x) : \mathsf{inf}(\mathsf{s}(\mathsf{s}(x)))))\mu^n = (x : (y : \mathsf{inf}(z)))\sigma$ by choosing $n = 0$ and $\sigma = \{y/\mathsf{s}(x), z/\mathsf{s}(\mathsf{s}(x))\}$. Thus, by Theorem 22 we know that this loop is not a Π-loop.*

5.2 Deciding Loops for Forbidden Patterns of Type $(\cdot,\cdot,a)$

Also for patterns of type $(\cdot,\cdot,a)$ we want to generate a finite set of matching problems such that the loop respects a pattern (ℓ,o,a) iff none of these matching problems is solvable. Essentially, we replace the condition $p^n q = o'o$ of the previous subsection by $p^n q < o'o$, i.e., $o'o$ must now be strictly below the redex.

The plan is to systematically represent all terms $t(C,\mu)^n|_{o'}$ for all numbers n and all positions o' where $p^n q < o'o$. We consider two alternatives: either the term starts within $C^n[t]$ and not in the substitutions below t, or the term starts within the substitutions that are below t. To distinguish these possibilities, we define the finite set of positions $\mathcal{P} = \{q' \mid qq' \in \mathcal{P}\mathsf{os}(t)\}$. Then the first alternative corresponds to the constraint $o' \leq p^n qq'$ for some $q' \in \mathcal{P}$, and the second alternative corresponds to the constraint $o' > p^n qq'$ for some maximal position $q' \in \mathcal{P}$.

For the first alternative, we start to fix the unknown n by choosing $n_0 = 0$ if $p = \varepsilon$, and $n_0 = \lceil \frac{|o| \dot{-} |q|}{|p|} \rceil$ otherwise. We will show later that if $\exists o'.p^n q < o'o$ can be satisfied by some n, then it can also be satisfied using some $n \geq n_0$. For $n \geq n_0$, we will see that $t(C,\mu)^n|_{o'}$ must be of the form $t(C,\mu)^{n_0}|_{o''}\mu^k$ for some o'' and k. Hence, we build the finite set of matching problems

$$\mathcal{M}_1 = \{(t(C,\mu)^{n_0}|_{o''} \geqslant \ell,\mu) \mid o'' \leq p^{n_0} qq' \wedge q' \in \mathcal{P} \wedge p^{n_0} q < o''o\}.$$

For the second alternative where $o' > p^n qq'$ for some maximal $q' \in \mathcal{P}$, we first define the set $\mathcal{W} = \bigcup_{k\in\mathbb{N}} \mathcal{V}(t|_q \mu^k)$ of variables that can occur below $t|_q$ when applying μ an arbitrary number of times. Note that for substitutions with finite domains, $\mathcal{W}$ is finite and can easily be computed by iteratively applying μ on $t|_q$ until no new variables appear. We define the second set of matching problems as

$$\mathcal{M}_2 = \{(u \gtrdot \ell, \mu) \mid u \trianglelefteq x\mu \wedge x \in \mathcal{W}\}.$$

We will show soundness of these matching problems by the following key lemma which handles both alternatives.

Lemma 24 (Connection of (13) and $\mathcal{M}_1 \cup \mathcal{M}_2$)**.** *Let t be a term, $q \in \mathcal{P}\mathsf{os}(t)$, and let (C, μ) be a context-substitution such that $C|_p = \Box$ and such that $t|_q$ is not a variable.*

(i) If (13) is satisfied with $o' \leq p^n q q'$ for some $q' \in \mathcal{P}$, then a problem in $\mathcal{M}_1$ is solvable.

(ii) If (13) is satisfied with $o' > p^n q q'$ for some maximal $q' \in \mathcal{P}$, then a problem in $\mathcal{M}_2$ is solvable.

(iii) If a problem in $\mathcal{M}_1 \cup \mathcal{M}_2$ is solvable then (13) is satisfied.

Proof. (i) Assume (13) holds and let n, o', $q' \in \mathcal{P}$, and σ be such that $t(C,\mu)^n|_{o'} = \ell\sigma$, $o' \leq p^n q q'$, and $p^n q < o' o$. If $p = \varepsilon$ then $n_0 = 0$, and we define $o'' = o'$ and $k = n$. Hence, using Lemma 5

$$t(C,\mu)^{n_0}|_{o''}\mu^k = t|_{o''}\mu^k = t|_{o'}\mu^n = t\mu^n|_{o'} = t(C,\mu)^n|_{p^n}|_{o'} = t(C,\mu)^n|_{\varepsilon^n}|_{o'} = t(C,\mu)^n|_{o'} = \ell\sigma$$

shows that the matching problem $(t(C,\mu)^{n_0}|_{o''} \gtrdot \ell, \mu)$ is solvable, and since $o'' = o' \leq p^n q q' = p^{n_0} q q'$ and $p^{n_0} q = \varepsilon^{n_0} q = \varepsilon^n q = p^n q < o' o = o'' o$ we also know that this matching problem is contained in $\mathcal{M}_1$. Otherwise, $p \neq \varepsilon$ and $n_0 = \lceil \frac{|o| \dot{-} |q|}{|p|} \rceil$. W.l.o.g. one can assume that $n \geq n_0$.[4] Hence, the position p^{n-n_0} is well formed. Next, we prove that $o' \geq p^{n-n_0}$. Note that o' cannot be parallel to p^{n-n_0} as $o' \leq p^n q q'$. If we had $o' < p^{n-n_0}$, then $|p^{n-n_0}| + |p^{n_0} q| = |p^n q| < |o' o| = |o'| + |o| < |p^{n-n_0}| + |o|$ shows that $n_0 \cdot |p| + |q| < |o|$, and hence yields the contradiction $n_0 \cdot |p| = \lceil \frac{|o| \dot{-} |q|}{|p|} \rceil \cdot |p| < |o| \dot{-} |q|$. So there is some o'' such that $o' = p^{n-n_0} o''$ and since $o' \leq p^n q q' = p^{n-n_0} p^{n_0} q q'$ we know that $o'' \leq p^{n_0} q q'$. Moreover, as $p^{n-n_0} p^{n_0} q = p^n q < o' o = p^{n-n_0} o'' o$ we also know that $p^{n_0} q < o'' o$. Thus, $o'' \leq p^{n_0} q q'$ and $p^{n_0} q < o'' o$ and hence, $(t(C,\mu)^{n_0}|_{o''} \gtrdot \ell, \mu) \in \mathcal{M}_1$. It remains to show that this matching problem is solvable which is established using Lemma 5:

$$t(C,\mu)^{n_0}|_{o''}\mu^{n-n_0} = t(C,\mu)^{n_0}\mu^{n-n_0}|_{o''} = t(C,\mu)^{n_0}(C,\mu)^{n-n_0}|_{p^{n-n_0}}|_{o''} = t(C,\mu)^n|_{o'} = \ell\sigma.$$

(ii) We now assume that (13) is satisfiable where $o' > p^n q q'$ for some maximal position $q' \in \mathcal{P}$, and show that there is also some matching problem in $\mathcal{M}_2$ that is solvable. So, let n, o', q', and σ be such that $t(C,\mu)^n|_{o'} = \ell\sigma$, $o' > p^n q q'$, $p^n q < o' o$, and q' is a maximal position in $\mathcal{P}$. Hence, $o' = p^n q q' o''$ for some $o'' \neq \varepsilon$ and thus by Lemma 5,

$$t(C,\mu)^n|_{o'} = t(C,\mu)^n|_{p^n}|_{qq'o''} = t\mu^n|_{qq'o''} = t|_{qq'}\mu^n|_{o''}.$$

Since q' was maximal and $o'' \neq \varepsilon$ we know that $t|_{qq'}$ must be a variable. Then one can show as in the proof of [17, Thm. 10] that $t|_{qq'}\mu^n|_{o''} = u\mu^k$ for some $u \trianglelefteq x\mu$, $x \in \mathcal{W}$, and k. Hence, $(u \gtrdot \ell, \mu)$ is a matching problem of $\mathcal{M}_2$ and it is solvable since

$$\ell\sigma = t(C,\mu)^n|_{o'} = t|_{qq'}\mu^n|_{o''} = u\mu^k.$$

[4] If $n < n_0$ then one can replace n, o', and σ by $n + n_0$, $p^{n_0} o'$, and $\sigma\mu^{n_0}$. These new values also satisfy (13).

(iii) Assume that a problem in $\mathcal{M}_1$ is solvable. Hence, there exist k, σ, o'', and $q' \in \mathcal{P}$ such that $t(C,\mu)^{n_0}|_{o''}\mu^k = \ell\sigma$, $o'' \leq p^{n_0}qq'$, and $p^{n_0}q < o''o$. Then we define $n = n_0 + k$ and $o' = p^k o''$ and achieve

$$t(C,\mu)^n|_{o'} = t(C,\mu)^{n_0}(C,\mu)^k|_{p^k}|_{o''} = t(C,\mu)^{n_0}\mu^k|_{o''} = t(C,\mu)^{n_0}|_{o''}\mu^k = \ell\sigma$$

and moreover $p^n q = p^k p^{n_0} q < p^k o'' o = o'o$. Hence, if one of the matching problems in $\mathcal{M}_1$ is solvable, then also (13) holds.

We now assume that a matching problems in $\mathcal{M}_2$ is solvable and show that then (13) is satisfied. Here, we need the additional assumption that $t|_q$ is not a variable. This assumption is not severe as we are interested in terms t where $t \to_q t'$, which implies that $t|_q$ is not a variable for well-formed TRSs.[5] So, let u, x, k, k', and σ be given such that $x \in \mathcal{V}(t|_q\mu^{k'})$, $u \trianglelefteq x\mu$, and $u\mu^k = \ell\sigma$. Let o'' and o''' be positions such that $t|_q\mu^{k'}|_{o''} = x$ and $x\mu|_{o'''} = u$. We define $n = k + k' + 1$ and $o' = p^n q o'' o'''$ and show for these values that (13) is satisfied (again, using Lemma 5):

$$t(C,\mu)^n|_{o'} = t(C,\mu)^n|_{p^n}|_{qo''o'''} = t\mu^n|_{qo''o'''} = t|_q\mu^{k'+1+k}|_{o''o'''} = x\mu^{1+k}|_{o'''} = u\mu^k = \ell\sigma$$

and $p^n q < p^n q o'' o''' o = o'o$ since $o'' \neq \varepsilon$. That o'' is indeed non-empty follows from the fact that $t|_q$ and thus also $t\mu^{k'}|_q$ is not a variable, but $t\mu^{k'}|_{qo''} = t|_q\mu^{k'}|_{o''} = x$.

$\square$

Using Lemma 24 it is now easy to derive the following theorem.

Theorem 25 (Soundness of (ℓ,o,a) problems)**.** *Let $t \to_q t'$ and let (C,μ) be a context-substitution such that $C|_p = \square$ and such that $t|_q$ is not a variable. All reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are allowed w.r.t. the pattern (ℓ,o,a) iff none of the matching problems in $\mathcal{M}_1 \cup \mathcal{M}_2$ is solvable.*

Note that when encoding innermost rewriting by using forbidden patterns, the resulting matching problems one obtains in [17] are essentially $\mathcal{M}_1 \cup \mathcal{M}_2$.

5.3 Deciding Loops for Forbidden Patterns of Type $(\cdot,\cdot,b)$

Finally, for patterns (ℓ,o,b), we replace the condition $p^n q = o'o$ by $p^n q > o'o$, i.e., $o'o$ has to be strictly above the redex. First note that $o'o \in \mathcal{P}\mathsf{os}(C^n[t])$. Now, we consider the following two cases: either $o'o$ ends in t (i.e., $o'o \geq p^n$), or otherwise it ends in some occurrence of C (i.e., $o'o < p^n$).

In the first case there are only finitely many positions in t above q in which $o'o$ could end. Thus, we reduce this case to finitely many $(\cdot,\cdot,h)$ cases. For each $\bar{q}$ above q in t, we consider the pattern (ℓ,o,h) for a reduction at position $\bar{q}$. Hence, we define

$$\mathcal{M}_3 = \bigcup_{\bar{q}<q} \mathcal{M}_{\bar{q}}, \text{ where } \mathcal{M}_{\bar{q}} \text{ is the set of } (\ell,o,h) \text{ matching problems for } t,\ \bar{q}, \text{ and } (C,\mu).$$

In the second case $o'o$ is a non-hole position of C^n, i.e., $p^n > o'o$. Then $p \neq \varepsilon$, since otherwise we would obtain the contradiction $\varepsilon = p^n > o'o$. So there is a $k < n$ and a $p''' \leq p$ with $o' = p^k p'''$. Let p'' be the position with $p = p'''p''$. Then we have $o < p''p^{n_0}$ for some n_0. To examine all possible choices for o', we consider all prefixes p''' of p, i.e., all contexts D with $\square \triangleleft D \trianglelefteq C$ where $C|_{p'''} = D$, $D|_{p''} = \square$, and $p = p'''p''$. Let n_0 be the smallest number such that $|p''| + |p^{n_0}| > |o|$ (since $p > \varepsilon$, such

[5] It is also possible to define $\mathcal{M}_2$ in a way that $t|_q$ can be a variable. However, then the definitions would become even more technical. Essentially, one just would have to perform some additional book-keeping to check whether one is strictly below $t|_q$.

a number always exists). Then we have to check whether $o < p''p^{n_0}$. If that is not the case, then we do not result in any additional matching problems. Otherwise, we obtain an *extended matching problem* $(D \gtrdot \ell, C\mu, t(C,\mu)^{n_0}\mu, \mu)$ for each $\Box \lhd D \unlhd C$.

$$\mathcal{M}_4 = \{(D \gtrdot \ell, C\mu, t(C,\mu)^{n_0}\mu, \mu) \mid \\ \Box \lhd D \unlhd C, D|_{p''} = \Box, n_0 \text{ is least number with } |p''| + n_0|p| > |o|, p''p^{n_0} > o\}$$

These are the same kind of extended matching problem as for deciding outermost loops.

Definition 26 (Extended matching problems [19]). *We call a quadruple $(D \gtrdot \ell, C, t, \mu)$ an* extended matching problem*. It is* solvable *iff there are m, k, σ, such that $D[t(C,\mu)^m]\mu^k = \ell\sigma$.*

Lemma 27 (Connection of (13) and $\mathcal{M}_3 \cup \mathcal{M}_4$). *Let $t \to_q t'$ and let (C,μ) be a context-substitution such that $C|_p = \Box$.*

(i) (13) is satisfied with $o'o \geq p^n$ iff a problem in $\mathcal{M}_3$ is solvable.

(ii) (13) is satisfied with $o'o < p^n$ iff a problem in $\mathcal{M}_4$ is solvable.

Proof. (i) Suppose that a (ℓ, o, h) matching problem in $\mathcal{M}_3$ for $\bar{q} < q$ is solvable. By Theorem 22 we obtain $\bar{q}$, m, o', and σ with $t(C,\mu)^n|_{o'} = \ell\sigma$ and $p^n\bar{q} = o'o$. Since $\bar{q} < q$, this implies $p^n q > o'o$ and $o'o \geq p^n$. Thus we satisfy the case of (13) where $\lambda = b$ and $o'o \geq p^n$.

Conversely, assume that there are n, o', and σ such that $t(C,\mu)^n|_{o'} = \ell\sigma$, $p^n q > o'o$, and $o'o \geq p^n$. Thus, there is some $o'' \neq \varepsilon$ with $p^n q = o'oo''$. Since we are in the case where $o'o \geq p^n$, this implies that o'' is a suffix of q. Hence, there is some position $\bar{q}$ such that $q = \bar{q}o''$ and $p^n\bar{q} = o'o$. As $o'' \neq \varepsilon$ we know that $\bar{q} < q$ and hence, one of the (ℓ, o, h) matching problems in $\mathcal{M}_3$ is solvable using Theorem 22.

(ii) Suppose that an extended matching problem in $\mathcal{M}_4$ is solvable. Thus there are m, k, and σ such that $D[t(C,\mu)^{n_0}\mu(C\mu,\mu)^m]\mu^k = \ell\sigma$ and $p''p^{n_0} > o$. Let $o' = p^k p'''$ and $n = k + n_0 + m + 1$. Hence, by Lemma 5

$$t(C,\mu)^n|_{o'} = t(C,\mu)^{k+n_0+m+1}|_{p^k p'''} = t(C,\mu)^{n_0+m+1}\mu^k|_{p'''} = C[t(C,\mu)^{n_0+m}\mu]\mu^k|_{p'''} \\ = D[t(C,\mu)^{n_0+m}\mu]\mu^k = D[t(C,\mu)^{n_0}\mu(C\mu,\mu)^m]\mu^k = \ell\sigma$$

and moreover $p^n = p^k p^{n_0} p^m p \geq p^k p p^{n_0} = p^k p''' p'' p^{n_0} > p^k p''' o = o'o$ and thus, also $p^n q > o'o$.

In order to prove the other direction, assume that there are n, o', and σ such that $t(C,\mu)^n|_{o'} = \ell\sigma$ and $p^n > o'o$. Let $k = \lfloor \frac{|o'|}{|p|} \rfloor$. Hence, there is some $p''' < p$ such that $o' = p^k p'''$. Since $p''' < p$, there is also some p'' with $p = p'''p''$. From the fact that o' is a strict prefix of p^n, we obtain some $m \in \mathbb{N}$ such that $p^n = p^k p''' p'' p^m = o'p''p^m$. Thus, $o'p''p^m = p^n > o'o$ which implies $p''p^m > o$ and so, $|p''| + |p^m| > |o|$. Hence, m is greater than or equal to the smallest number n_0 satisfying $|p''| + |p^{n_0}| > |o|$ and thus $m = n_0 + m'$ for some $m' \in \mathbb{N}$. From $p^n = p^k p''' p'' p^m$, we also obtain $n = k + m + 1$. Let $D = C|_{p'''}$.

$$\ell\sigma = t(C,\mu)^n|_{o'} = t(C,\mu)^{k+m+1}|_{p^k p'''} = t(C,\mu)^{m+1}\mu^k|_{p'''} = C[t(C,\mu)^m\mu]\mu^k|_{p'''} = D[t(C,\mu)^m\mu]\mu^k \\ = D[t(C,\mu)^{n_0+m'}\mu]\mu^k = D[t(C,\mu)^{n_0}(C,\mu)^{m'}\mu]\mu^k = D[t(C,\mu)^{n_0}\mu(C\mu,\mu)^{m'}]\mu^k$$

By m', k, σ, we obtain a solution of the extended matching problem $(D \gtrdot \ell, C\mu, t(C,\mu)^{n_0}\mu, \mu)$. Note that $\Box \lhd D$ since otherwise $p''' = p$ which contradicts $p''' < p$. Moreover, since $p''p^m > o$ and $|p''| + |p^{n_0}| > |o|$, we have $p''p^{n_0} > o$. Hence, the matching problem $(D \gtrdot \ell, C\mu, t(C,\mu)^{n_0}\mu, \mu)$ is contained in $\mathcal{M}_4$.

$\Box$

Using Lemma 27, we have proved the following theorem.

Theorem 28 (Soundness of (ℓ, o, b) problems)**.** *Let $t \to_q t'$ and let (C, μ) be a context-substitution such that $C|_p = \Box$. All reductions $t(C,\mu)^n \to_{p^n q} t'(C,\mu)^n$ are allowed w.r.t. the pattern (ℓ, o, b) iff none of the matching problems in $\mathcal{M}_3 \cup \mathcal{M}_4$ is solvable.*

Note that as in the innermost case, when encoding outermost rewriting by using forbidden patterns, the resulting matching problems one obtains in [19] are $\mathcal{M}_3 \cup \mathcal{M}_4$. So Theorem 28 is a generalization of the result in [19].

By combining Corollary 7 with Theorem 22, Theorem 25, and Theorem 28, we finally obtain the following corollary.

Corollary 29 (Forbidden loops are decidable)**.** *Let $t_1 \to_{q_1} t_2 \to_{q_2} \cdots \to_{q_m} t_{m+1} = t_1(C,\mu)$ be a loop with $C|_p = \Box$ and let Π be a set of forbidden patterns. Then it is decidable whether the loop is a loop under the strategy Π.*

6 Conclusion

In this paper, we developed approaches to disprove termination of rewriting under strategies like leftmost-innermost, leftmost-outermost, (max-)parallel-innermost, (max-)parallel-outermost, and forbidden patterns automatically. To this end, we introduced decision procedures which check whether a given loop is also a loop under the respective strategy. By combining these procedures with techniques to detect loops automatically, one obtains methods to prove non-termination of term rewriting under these strategies.

The general idea of our decision procedures is to generate a set of *(extended) matching problems* from every loop such that one of these matching problems is solvable iff the given loop violates the strategy. We presented a decision problem for solvability of matching problems in [17] (for extended matching problems this was done in [19]).

We started with defining *leftmost* matching problems in Section 3 which shows that it is decidable whether a loop is a leftmost loop. By combining this result with the decision procedures for innermost and outermost loops from [17,19], it is also decidable whether a loop is a leftmost-innermost or leftmost-outermost loop.

In Section 4 we considered parallel- and max-parallel-rewriting, where in the latter case, *all* redexes at parallel positions must be reduced simultaneously. Similar to leftmost matching problems, here we defined *max-parallel* matching problems and showed that it is decidable whether a given loop is also a max-parallel, a (max-)parallel-innermost, or a (max-)parallel-outermost loop.

Finally, in Section 5 we extended our approach to strategies defined by *forbidden patterns* [6]. Forbidden patterns are very expressive and in particular, they can also be used to describe strategies such as innermost, outermost, or context-sensitive rewriting. There are three variants of such patterns which restrict rewriting on, above, or below certain positions of certain subterms. For each of these classes of forbidden patterns, we showed how to generate corresponding matching problems such that one of these matching problems is solvable iff the given loop violates the restriction described by the pattern. Thus, it is decidable whether a loop is also a loop under a strategy expressed by a set of forbidden patterns.

Our results constitute the first automatic approach for disproving termination under these strategies. Future work will be concerned with extending and adapting our results such that they can be integrated in rewriting-based approaches for termination analysis of programming languages (e.g., [16,5,12]).

Acknowledgments. We thank the referees for many helpful suggestions.

References

[1] F. Baader & T. Nipkow (1998): *Term Rewriting and All That*. Cambridge University Press.

[2] J. Giesl & A. Middeldorp (2004): *Transformation Techniques for Context-Sensitive Rewrite Systems*. Journal of Functional Programming 14(4), pp. 379–427.

[3] J. Giesl, R. Thiemann & P. Schneider-Kamp (2005): *The Dependency Pair Framework: Combining Techniques for Automated Termination Proofs*. In: Proc. LPAR '04, LNAI 3452, pp. 301–331.

[4] J. Giesl, R. Thiemann & P. Schneider-Kamp (2005): *Proving and Disproving Termination of Higher-Order Functions*. In: Proc. FroCoS '05, LNAI 3717, pp. 216–231.

[5] J. Giesl, M. Raffelsieper, P. Schneider-Kamp, S. Swiderski & R. Thiemann (2010): *Automated Termination Proofs for Haskell by Term Rewriting*. ACM Transactions on Programming Languages and Systems To appear. Preliminary version appeared in *Proc. RTA '06*, LNCS 4098, pp. 297–312, 2006.

[6] B. Gramlich & F. Schernhammer (2010): *Extending Context-Sensitivity in Term Rewriting*. In: Proc. WRS '09, EPTCS 15, pp. 56–68.

[7] J. Guttag, D. Kapur & D. Musser (1983): *On Proving Uniform Termination and Restricted Termination of Rewriting Systems*. SIAM Journal of Computation 12, pp. 189–214.

[8] W. Kurth (1990): *Termination und Konfluenz von Semi-Thue-Systemen mit nur einer Regel*. Ph.D. thesis, Technische Universität Clausthal, Germany.

[9] D. Lankford & D. Musser (1978): *A Finite Termination Criterion*. Unpublished Draft. USC Information Sciences Institute.

[10] S. Lucas (1998): *Context-Sensitive Computations in Functional and Functional Logic Programs*. Journal of Functional and Logic Programming 1, pp. 1–61.

[11] S. Lucas (2001): *Termination of On-Demand Rewriting and Termination of OBJ Programs*. In: Proc. PPDP '01, pp. 82–93.

[12] C. Otto, M. Brockschmidt, C. von Essen & J. Giesl (2010): *Automated Termination Analysis of Java Bytecode by Term Rewriting*. In: Proc. RTA '10, LIPIcs 6, pp. 259–276.

[13] É. Payet (2008): *Loop Detection in Term Rewriting Using the Eliminating Unfoldings*. Theoretical Computer Science 403(2-3), pp. 307–327.

[14] M. Raffelsieper & H. Zantema (2009): *A Transformational Approach to Prove Outermost Termination Automatically*. In: Proc. WRS '08, ENTCS 237, pp. 3–21.

[15] M. R. K. Krishna Rao (2000): *Some Characteristics of Strong Innermost Normalization*. Theoretical Computer Science 239, pp. 141–164.

[16] P. Schneider-Kamp, J. Giesl, A. Serebrenik & R. Thiemann (2009): *Automated Termination Proofs for Logic Programs by Term Rewriting*. ACM Transactions on Computational Logic 11(1).

[17] R. Thiemann, J. Giesl & P. Schneider-Kamp (2008): *Deciding Innermost Loops*. In: Proc. RTA '08, LNCS 5117, pp. 366–380.

[18] R. Thiemann (2009): *From Outermost Termination to Innermost Termination*. In: Proc. SOFSEM '09, LNCS 5404, pp. 533–545.

[19] R. Thiemann & C. Sternagel (2009): *Loops under Strategies*. In: Proc. RTA '09, LNCS 5595, pp. 17–31.

[20] J. Waldmann (2004): Matchbox*: A Tool for Match-Bounded String Rewriting*. In: Proc. RTA '04, LNCS 3091, pp. 85–94.

[21] H. Zantema (2005): *Termination of String Rewriting Proved Automatically*. Journal of Automated Reasoning 34, pp. 105–139.