Proving Non-Looping Non-Termination Automatically*

Fabian Emmes, Tim Enger, and Jürgen Giesl

LuFG Informatik 2, RWTH Aachen University, Germany

Abstract. We introduce a technique to prove non-termination of term rewrite systems automatically. Our technique improves over previous approaches substantially, as it can also detect non-looping non-termination.

1 Introduction

Approaches to prove termination of term rewrite systems (TRSs) have been studied for decades and there exist several techniques to prove termination of programs via a translation to TRSs. In contrast, techniques to *disprove* termination of TRSs have received much less attention, although this is highly relevant to detect bugs during program development. To prove non-termination of a TRS, one has to provide a finite description of an infinite rewrite sequence.

The most common way for this is to find a *loop*, i.e., a finite rewrite sequence $s \to_{\mathcal{R}}^{+} C[s\mu]$ for some term s, context C, and substitution μ. Indeed, any loop gives rise to an infinite rewrite sequence $s \to_{\mathcal{R}}^{n} C[s\mu] \to_{\mathcal{R}}^{n} C[C\mu[s\mu^2]] \to_{\mathcal{R}}^{n} \dots$ for some $n > 0$. While this is a very intuitive way to prove non-termination, it cannot capture non-periodic infinite rewrite sequences.

```
while (gt(x,y)){
 x = dbl(x);
 y = y + 1;     }
```

For instance, consider the imperative program fragment on the side which does not terminate if `x` > `y` and `x` > `0`. However, if `gt` (greater than) and `dbl` (double) are user-defined, then the number of evaluation steps needed for `gt` and `dbl` increases in each loop iteration. Hence, this is a non-periodic form of non-termination.

The following TRS $\mathcal{R}$ corresponds to the imperative program fragment above.

$$\begin{aligned}
\mathsf{f}(\mathsf{tt}, x, y) &\to \mathsf{f}(\mathsf{gt}(x, y), \mathsf{dbl}(x), \mathsf{s}(y)) & \mathsf{dbl}(x) &\to \mathsf{times}(\mathsf{s}(\mathsf{s}(0)), x)\\
\mathsf{gt}(\mathsf{s}(x), 0) &\to \mathsf{tt} & \mathsf{times}(x, 0) &\to 0\\
\mathsf{gt}(0, y) &\to \mathsf{ff} & \mathsf{times}(x, \mathsf{s}(y)) &\to \mathsf{plus}(\mathsf{times}(x, y), x)\\
\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)) &\to \mathsf{gt}(x, y) & \mathsf{plus}(x, 0) &\to x\\
& & \mathsf{plus}(x, \mathsf{s}(y)) &\to \mathsf{plus}(\mathsf{s}(x), y)
\end{aligned}$$

This TRS is non-terminating, but not looping. For $n > m$, we have

$$\begin{aligned}
&\mathsf{f}(\mathsf{tt}, \mathsf{s}^n(0), \mathsf{s}^m(0)) && \to_{\mathcal{R}} \mathsf{f}(\mathsf{gt}(\mathsf{s}^n(0), \mathsf{s}^m(0)), \mathsf{dbl}(\mathsf{s}^n(0)), \mathsf{s}^{m+1}(0)) && \to_{\mathcal{R}}^{m+1}\\
&\mathsf{f}(\mathsf{tt}, \mathsf{dbl}(\mathsf{s}^n(0)), \mathsf{s}^{m+1}(0)) && \to_{\mathcal{R}} \mathsf{f}(\mathsf{tt}, \mathsf{times}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}^n(0)), \mathsf{s}^{m+1}(0)) && \to_{\mathcal{R}}^{4 \cdot n}\\
&\mathsf{f}(\mathsf{tt}, \mathsf{s}^{2 \cdot n}(\mathsf{times}(\mathsf{s}(\mathsf{s}(0)), 0)), \mathsf{s}^{m+1}(0)) && \to_{\mathcal{R}} \mathsf{f}(\mathsf{tt}, \mathsf{s}^{2 \cdot n}(0), \mathsf{s}^{m+1}(0)) && \to_{\mathcal{R}} \dots
\end{aligned}$$

Since the number of steps required to evaluate `gt` and `dbl` increases in every

* Supported by the DFG grant GI 274/5-3

iteration, this is a non-periodic sequence that cannot be represented as a loop.

While interesting classes of non-looping TRSs were identified in earlier papers (e.g., [3, 14]), up to now virtually all methods to prove non-termination of TRSs automatically were restricted to loops (e.g., [4, 5, 11, 13, 15, 16]).[1] A notable exception is a technique and tool for non-termination of non-looping *string* rewrite systems (SRSs) in [10]. To represent rewrite sequences, this approach uses rules between string patterns of the form $u\,v^n\,w$. Here, u, v, w are strings and n can be instantiated by any natural number. We will extend this idea in order to prove non-termination of (possibly non-looping) *term* rewrite systems automatically.

To detect loops, one can start with a rule and repeatedly *narrow* it using other rules, until it has the form of a loop. To handle non-looping TRSs as well, we generate *pattern rules* which represent a whole set of rewrite sequences and also allow narrowing with pattern rules. In this way, one can create more and more pattern rules until one obtains a pattern rule that is obviously non-terminating. In Sect. 2, we define pattern rules formally and introduce a set of inference rules to derive pattern rules from a TRS automatically. Sect. 2 also contains a criterion to detect pattern rules that are obviously non-terminating. In Sect. 3 we present a strategy for the application of our inference rules. We implemented our contributions in the automated termination tool AProVE [6] and in Sect. 4, we present an experimental evaluation of our technique.

2 Pattern Rules

To represent rewrite sequences, we extend the idea of [10] from SRSs to TRSs and define *pattern terms* and *pattern rules* which are parameterized over $\mathbb{N}$.

A *pattern term* describes a set of terms.[2] Formally, a pattern term is a mapping from natural numbers to terms which are constructed from a *base term*, a *pumping substitution* that is applied multiple times to the base term, and a *closing substitution* that is applied once to "close" the term. For example, to represent $\mathsf{gt}(\mathsf{s}^2(x), \mathsf{s}(\mathsf{0}))$, $\mathsf{gt}(\mathsf{s}^3(x), \mathsf{s}^2(\mathsf{0}))$, $\mathsf{gt}(\mathsf{s}^4(x), \mathsf{s}^3(\mathsf{0}))$, ..., we use the pattern term $n \mapsto \mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ [x/\mathsf{s}(x), y/\mathsf{s}(y)]^n\ [x/\mathsf{s}(x), y/\mathsf{0}]$, where $\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))$ is the base term, $[x/\mathsf{s}(x), y/\mathsf{s}(y)]$ is the pumping substitution, and $[x/\mathsf{s}(x), y/\mathsf{0}]$ is the closing substitution. For $n = 0$ this pattern term evaluates to $\mathsf{gt}(\mathsf{s}^2(x), \mathsf{s}(\mathsf{0}))$, for $n = 1$ to $\mathsf{gt}(\mathsf{s}^3(x), \mathsf{s}^2(\mathsf{0}))$, etc. In the following, $\mathcal{T}(\Sigma, \mathcal{V})$ denotes the set of terms over the underlying signature Σ and the infinite set of variables $\mathcal{V}$.

Definition 1 (Pattern Terms and Rules). *A function $\mathbb{N} \to \mathcal{T}(\Sigma, \mathcal{V})$ is a* pattern term *if it is a mapping $n \mapsto t\sigma^n\mu$ where $t \in \mathcal{T}(\Sigma, \mathcal{V})$ and σ, μ are*

[1] Similarly, most existing automated approaches for non-termination of *programs* also just detect loops. For Java Bytecode, we recently presented an approach that can also prove non-periodic non-termination, provided that there are no sub-loops and that non-termination is due to operations on integers [2]. However, this approach is not suitable for TRSs where one treats terms instead of integers and where sub-loops (i.e., recursively defined auxiliary functions like gt and times) are common.

[2] In contrast to *tree automata*, pattern terms can also describe non-regular sets.

substitutions. For readability, we omit "$n \mapsto$" if it is clear that we refer to a pattern term. For a pattern term $p = t\sigma^n\mu$, its base term *is* $\mathrm{base}(p) = t$*, its* pumping substitution *is* σ*, and its* closing substitution *is* μ*. We also say that* σ, μ *are its* pattern substitutions*. Its* domain variables *are* $\mathrm{dv}(p) = \mathrm{dom}(\sigma) \cup \mathrm{dom}(\mu)$*.*

If p, q *are pattern terms, then* $p \hookrightarrow q$ *is a* pattern rule*. A pattern rule* $p \hookrightarrow q$ *is* correct w.r.t. a TRS $\mathcal{R}$ *if* $p(n) \to_{\mathcal{R}}^{+} q(n)$ *holds for all* $n \in \mathbb{N}$*.*

As an example, consider the pattern rule

$$\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ [x/\mathsf{s}(x), y/\mathsf{s}(y)]^n\ [x/\mathsf{s}(x), y/\mathsf{0}] \quad \hookrightarrow \quad \mathsf{tt}\ \varnothing^n\ \varnothing, \tag{1}$$

where $\varnothing$ denotes the empty (identical) substitution. This pattern rule is correct w.r.t. the TRS $\mathcal{R}$ in Sect. 1, since $\mathsf{gt}(\mathsf{s}^{n+2}(x), \mathsf{s}^{n+1}(\mathsf{0})) \to_{\mathcal{R}}^{+} \mathsf{tt}$ holds for all $n \in \mathbb{N}$. Thus, a pattern rule describes a set of rewrite sequences of arbitrary length.

In the following, we present 9 inference rules to derive correct pattern rules from a TRS automatically. As soon as one finds a correct pattern rule that is obviously non-terminating, one has proved non-termination of the original TRS.

The inference rules have the form $\frac{p_1 \hookrightarrow q_1 \quad \dots \quad p_k \hookrightarrow q_k}{p \hookrightarrow q}$. In Thm. 7 we will prove their soundness, i.e., if all the pattern rules $p_1 \hookrightarrow q_1, \dots, p_k \hookrightarrow q_k$ are correct w.r.t. a TRS $\mathcal{R}$, then the pattern rule $p \hookrightarrow q$ is also correct w.r.t. $\mathcal{R}$.

The inference rules in Sect. 2.1 create initial pattern rules from a TRS. Sect. 2.2 shows how to modify the pattern terms in a pattern rule without changing the represented set of terms. Sect. 2.3 introduces inference rules in order to instantiate pattern rules and to combine them by narrowing. Finally, Sect. 2.4 shows how to detect whether a pattern rule directly leads to non-termination.

2.1 Creating Pattern Rules

The first inference rule converts rules from the TRS to equivalent pattern rules by simply using the identity $\varnothing$ as pattern substitution. Since a pattern term $\ell\,\varnothing^n\,\varnothing$ just represents the (ordinary) term ℓ, this inference rule is clearly sound. So by applying **(I)** to the recursive gt-rule from Sect. 1, we obtain the pattern rule

(I) Pattern Rule from TRS

$$\frac{}{\ell\,\varnothing^n\,\varnothing \ \hookrightarrow\ r\,\varnothing^n\,\varnothing} \quad \text{if } \ell \to r \in \mathcal{R}$$

$$\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ \varnothing^n\ \varnothing \quad \hookrightarrow \quad \mathsf{gt}(x, y)\ \varnothing^n\ \varnothing. \tag{2}$$

The next inference rule generates pattern rules that represent the repeated application of a rewrite sequence at the same position. Here, we say that two substitutions θ and σ *commute* iff $x\theta\sigma = x\sigma\theta$ holds for all variables $x \in \mathcal{V}$. When applying **(II)** to Rule (2), we have $s = \mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))$ and $t = \mathsf{gt}(x, y)$. By choosing $\theta = \varnothing$ and $\sigma = [x/\mathsf{s}(x), y/\mathsf{s}(y)]$, we obtain $s\theta = t\sigma$. Moreover since θ is the identical substitution, θ and σ obviously commute. Hence, by **(II)** we obtain the following new pattern rule which describes how (2) can be applied repeatedly on terms of the form $\mathsf{gt}(\mathsf{s}^n(x), \mathsf{s}^n(y))$.

(II) Pattern Creation 1

$$\frac{s\,\varnothing^n\,\varnothing \ \hookrightarrow\ t\,\varnothing^n\,\varnothing}{s\,\sigma^n\,\varnothing \ \hookrightarrow\ t\,\theta^n\,\varnothing} \quad \begin{array}{l}\text{if } s\theta = t\sigma, \text{ and} \\ \theta \text{ commutes with } \sigma\end{array}$$

$$\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ [x/\mathsf{s}(x), y/\mathsf{s}(y)]^n\ \varnothing \quad \hookrightarrow \quad \mathsf{gt}(x, y)\ \varnothing^n\ \varnothing \tag{3}$$

To see why commutation of θ and σ is needed for the soundness of Rule **(II)**, consider $s = \mathsf{f}(x, \mathsf{a})$ and $t = \mathsf{f}(\mathsf{b}, x)$ for a TRS $\mathcal{R}' = \{s \to t\}$. Then for $\theta = [x/\mathsf{b}]$ and $\sigma = [x/\mathsf{a}]$ we have $s\theta = t\sigma$. But θ and σ do not commute and $s\sigma = \mathsf{f}(\mathsf{a}, \mathsf{a}) \not\to^+_{\mathcal{R}'} \mathsf{f}(\mathsf{b}, \mathsf{b}) = t\theta$. Thus, $s\,\sigma^n\,\varnothing \;\hookrightarrow\; t\,\theta^n\,\varnothing$ is not correct w.r.t. $\mathcal{R}'$.

To automate the application of inference rule **(II)**, one has to find substitutions θ and σ that satisfy the conditions for its applicability. In our implementation, we use a sufficient criterion which proved useful in our experiments: We first apply unification to find the most general substitutions θ and σ such that $s\theta = t\sigma$. Then we check whether θ and σ commute. More precisely, to find θ and σ with $s\theta = t\sigma$, we use a variable renaming ρ which renames all variables in $\mathcal{V}(s)$ to fresh ones. If there exists $\tau = \mathrm{mgu}(s\rho, t)$, then we set $\theta = (\rho\,\tau\,\rho^{-1})|_{\mathcal{V}(s)}$ and $\sigma = (\tau\rho^{-1})|_{\mathcal{V}(t)}$. Now we have $s\theta = s\rho\,\tau\,\rho^{-1} = t\,\tau\,\rho^{-1} = t\sigma$ and thus, it remains to check whether θ commutes with σ. So in our example, we use a renaming ρ with $x\rho = x'$ and $y\rho = y'$. The mgu of $s\rho = \mathsf{gt}(\mathsf{s}(x'), \mathsf{s}(y'))$ and $t = \mathsf{gt}(x, y)$ is $\tau = [x/\mathsf{s}(x'), y/\mathsf{s}(y')]$. Hence, we obtain $x\theta = x\rho\,\tau\,\rho^{-1} = x$, $y\theta = y$, $x\sigma = x\,\tau\,\rho^{-1} = \mathsf{s}(x)$, and $y\sigma = y\,\tau\,\rho^{-1} = \mathsf{s}(y)$. Here, θ and σ obviously commute.

The next inference rule generates pattern rules to represent rewrite sequences where the context around the redex increases in each iteration. For instance, the times-rule of Sect. 1 can be applied repeatedly to rewrite terms of the form $\mathsf{times}(x, \mathsf{s}^n(y))$ to $\mathsf{plus}(\mathsf{plus}(\ldots\mathsf{plus}(\mathsf{times}(x, y), x), \ldots, x), x)$. But since these rewrite steps (except for the first) occur below the root, instead of **(II)** we need Rule **(III)**. As usual, $t[z]_\pi$ results from replacing the subterm at position π by z. Moreover, $\sigma \cup [z/t[z]_\pi]$ is the extension of the substitution σ which maps the fresh variable z to $t[z]_\pi$.

(III) Pattern Creation 2

$$\frac{s\,\varnothing^n\,\varnothing \;\hookrightarrow\; t\,\varnothing^n\,\varnothing}{s\,\sigma^n\,\varnothing \;\hookrightarrow\; t[z]_\pi\;(\sigma \cup [z/t[z]_\pi])^n\;[z/t|_\pi]} \quad \begin{array}{l}\text{if } \pi \in \mathcal{P}\mathrm{os}(t),\\ s = t|_\pi\,\sigma,\\ \text{and } z \in \mathcal{V} \text{ is fresh}\end{array}$$

Rule **(III)** can easily be automated, since one only has to check whether some subterm[3] of t matches s. For example, regard the pattern rule $\mathsf{times}(x, \mathsf{s}(y))\,\varnothing^n\,\varnothing \hookrightarrow \mathsf{plus}(\mathsf{times}(x, y), x)\,\varnothing^n\,\varnothing$ resulting from the times-rule. Here, $s = \mathsf{times}(x, \mathsf{s}(y))$ and $t = \mathsf{plus}(\mathsf{times}(x, y), x)$. For the subterm $t|_\pi = \mathsf{times}(x, y)$ at position $\pi = 1$ we have $s = t|_\pi\,\sigma$ with $\sigma = [y/\mathsf{s}(y)]$. Hence, by **(III)** we obtain the pattern rule

$$\mathsf{times}(x, \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,\varnothing \;\hookrightarrow\; \mathsf{plus}(z, x)\,[y/\mathsf{s}(y), z/\mathsf{plus}(z, x)]^n\,[z/\mathsf{times}(x, y)]. \quad (4)$$

Note that if π is the root position, then inference rule **(III)** is the special case of inference rule **(II)** where θ is the identity. In this case, both inference rules create a pattern rule equivalent to $s\,\sigma^n\,\varnothing \;\hookrightarrow\; t\,\varnothing^n\,\varnothing$.

2.2 Using Equivalence of Pattern Terms

As mentioned in the introduction, a common technique to prove that a TRS is looping is to construct loops via repeated narrowing operations. Narrowing is similar to rewriting, but uses unification instead of matching.

[3] In the automation, we restrict Rule **(III)** to non-variable subterms $t|_\pi$ in order to obtain pattern rules with "small" terms in the ranges of the pumping substitutions.

For instance, to narrow the right-hand side of the recursive rule $\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)) \to \mathsf{gt}(x, y)$ with the rule $\mathsf{gt}(\mathsf{s}(x), \mathsf{0}) \to \mathsf{tt}$, one could first instantiate the recursive rule using the substitution $[x/\mathsf{s}(x), y/\mathsf{0}]$, which yields $\mathsf{gt}(\mathsf{s}(\mathsf{s}(x)), \mathsf{s}(\mathsf{0})) \to \mathsf{gt}(\mathsf{s}(x), \mathsf{0})$. Now its right-hand side can be rewritten by the non-recursive gt-rule, which results in the new rule $\mathsf{gt}(\mathsf{s}(\mathsf{s}(x)), \mathsf{s}(\mathsf{0})) \to \mathsf{tt}$.

Our goal is to extend this concept to pattern rules. However, the problem is that the pattern terms in the rules may have different pattern substitutions. Thus, to narrow the right-hand side of a pattern rule $p \hookrightarrow q$ with another pattern rule $p' \hookrightarrow q'$, we first transform the rules such that the pattern substitutions in all four terms p, q, p', q' are the same. Then $p \hookrightarrow q$ and $p' \hookrightarrow q'$ have the form $s\,\sigma^n\,\mu \;\hookrightarrow\; t\,\sigma^n\,\mu$ and $u\,\sigma^n\,\mu \;\hookrightarrow\; v\,\sigma^n\,\mu$, respectively (i.e., the *same* pattern substitutions σ and μ are used on both sides of both pattern rules). To achieve that, it is often useful to modify the pattern terms in the rules appropriately without changing the set of terms represented by the pattern terms.

Definition 2 (Equivalent Pattern Terms). *We say that two pattern terms p and p' are* equivalent *iff $p(n) = p'(n)$ holds for all $n \in \mathbb{N}$.*

Based on Def. 2, we immediately obtain inference rule **(IV)** that allows us to replace pattern terms by equivalent other pattern terms. To apply rule **(IV)** automatically, in Lemmas 4, 6, and 9 we will present three criteria for equivalence of pattern terms.

(IV) Equivalence

$$\frac{p \hookrightarrow q}{p' \hookrightarrow q'}$$

if p is equivalent to p' and q is equivalent to q'

The first criterion allows us to rename the *domain variables* in the pattern substitutions. For example, in the pattern term $\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ [x/\mathsf{s}(x), y/\mathsf{s}(y)]^n\ \varnothing$ one can rename its domain variables x and y to x' and y'. This results in the pattern term $\mathsf{gt}(\mathsf{s}(x'), \mathsf{s}(y'))\ [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/x, y'/y]$ which is equivalent, since for every n, both pattern terms represent $\mathsf{gt}(\mathsf{s}^n(x), \mathsf{s}^n(y))$.

Definition 3 (Domain Renamings). *For any substitution σ, let* $\mathrm{range}(\sigma) = \{x\sigma \mid x \in \mathrm{dom}(\sigma)\}$ *and* $\mathcal{V}(\sigma) = \mathrm{dom}(\sigma) \cup \mathcal{V}(\mathrm{range}(\sigma))$. *Let ρ be a variable renaming on* $\mathrm{dom}(\rho)$, *i.e.,* $\mathrm{range}(\rho) \subseteq \mathcal{V}$ *and ρ is injective on* $\mathrm{dom}(\rho)$. *This allows us to define ρ^{-1} as $\rho^{-1}(y) = x$ if there is some $x \in \mathrm{dom}(\rho)$ with $x\rho = y$ and as $\rho^{-1}(y) = y$, otherwise. Note that $x\rho\rho^{-1} = x$ holds for all $x \in \mathrm{dom}(\rho)$ and also for all $x \notin \mathrm{range}(\rho)$. For any pattern term $p = t\sigma^n\mu$, we define its variables as $\mathcal{V}(p) = \mathcal{V}(t) \cup \mathcal{V}(\sigma) \cup \mathcal{V}(\mu)$. We say that a variable renaming ρ is a* domain renaming *for a pattern term p if* $\mathrm{dom}(\rho) \subseteq \mathrm{dv}(p)$ *and* $\mathrm{range}(\rho) \cap \mathcal{V}(p) = \varnothing$. *For a pattern term $p = t\sigma^n\mu$, we define the result of renaming p by ρ as $p^\rho = t'\,\sigma'^n\,\mu'$ where $t' = t\rho$, $\sigma' = [\,x\rho/s\rho \mid x/s \in \sigma\,]$, and $\mu' = [\,x\rho/s \mid x/s \in \mu\,]\,\rho^{-1}$.*

To illustrate Def. 3, consider $\rho = [x/x', y/y']$. This is indeed a variable renaming on $\mathrm{dom}(\rho) = \{x, y\}$ and we have $\rho^{-1} = [x'/x, y'/y]$. Moreover, we regard the pattern term $p = \mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\ [x/\mathsf{s}(x), y/\mathsf{s}(y)]^n\ \varnothing$. Thus, its base term is $t = \mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))$, and it has the pattern substitutions $\sigma = [x/\mathsf{s}(x), y/\mathsf{s}(y)]$ and $\mu = \varnothing$. Hence, ρ is a domain renaming for p since $\mathrm{dom}(\rho) \subseteq \mathrm{dv}(p) = \{x, y\}$ and since $\mathrm{range}(\rho) = \{x', y'\}$ is disjoint from $\mathcal{V}(p) = \mathcal{V}(t) \cup \mathcal{V}(\sigma) \cup \mathcal{V}(\mu) = \{x, y\}$. Thus, the result of renaming p by ρ is $p^\rho = \mathsf{gt}(\mathsf{s}(x'), \mathsf{s}(y'))\ [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/x, y'/y]$.

Lemma 4 gives the first criterion for obtaining equivalent pattern terms (in order to apply inference rule **(IV)** automatically).

Lemma 4 (Equivalence by Domain Renaming). *Let p be a pattern term and let ρ be a domain renaming for p. Then p is equivalent to p^ρ.*

Proof. Let $p = t\,\sigma^n\,\mu$, $\sigma' = [\,x\rho/s\rho \mid x/s \in \sigma\,]$, and $\mu' = [\,x\rho/s \mid x/s \in \mu\,]\,\rho^{-1}$. We first show the following conjecture:

$$x\,\sigma\rho = x\,\rho\,\sigma' \text{ for all } x \in \mathcal{V}(p) \tag{5}$$

For (5), let $x \in \mathcal{V}(p)$. If $x \in \mathrm{dom}(\sigma)$, then $x\,\rho\,\sigma' = x\,\sigma\,\rho$ by the definition of σ'. If $x \notin \mathrm{dom}(\sigma)$, then $x\rho \notin \mathrm{dom}(\sigma')$. Thus, $x\,\rho\,\sigma' = x\rho = x\,\sigma\,\rho$, which proves (5).

Moreover, we show the following conjecture:

$$x\,\mu = x\,\rho\,\mu' \text{ for all } x \in \mathcal{V}(p) \tag{6}$$

For (6), let $x \in \mathcal{V}(p)$. If $x \in \mathrm{dom}(\mu)$, then $x\,\rho\,\mu' = x\,\mu\,\rho^{-1}$ by the definition of μ'. Since $\mathcal{V}(x\mu) \subseteq \mathcal{V}(p)$, we have $\mathrm{range}(\rho) \cap \mathcal{V}(x\mu) = \varnothing$. Thus, $x\,\rho\,\mu' = x\,\mu\,\rho^{-1} = x\mu$.

Otherwise, if $x \notin \mathrm{dom}(\mu)$, then $x\mu = x$ and $x\,\rho\,\mu' = x\,\rho\,\rho^{-1} = x$. This concludes the proof of Conjecture (6).

Now we show the lemma. We have $p(n) = t\,\sigma^n\,\mu$. By (6), this is equal to $t\,\sigma^n\,\rho\,\mu'$. Using Conjecture (5) n times, we get $t\,\sigma^n\,\rho\,\mu' = t\,\rho\,\sigma'^n\,\mu' = p^\rho(n)$. □

Thus, we can apply inference rule **(IV)** (using Lemma 4 with the domain renaming $\rho = [x/x', y/y']$) to obtain the following pattern rule from Rule (3).

$$\mathsf{gt}(\mathsf{s}(x'), \mathsf{s}(y'))\ [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/x, y'/y] \quad \hookrightarrow \quad \mathsf{gt}(x, y)\ \varnothing^n\ \varnothing \tag{7}$$

Recall that to perform narrowing of pattern rules, we would like to have the same pattern substitutions on both sides of the rule. So the above domain renaming has the advantage that the variables x', y' used for "pumping" are now different from the variables x, y occurring in the final term. This allows us to add the pattern substitutions also on the right-hand side of the rule, since they only concern variables x', y' that are not *relevant* in the right-hand side up to now.

Definition 5 (Relevant Variables). *For a pattern term $p = t\sigma^n\mu$, we define its* relevant variables *as $\mathrm{rv}(p) = \mathcal{V}(\{t, t\sigma, t\sigma^2, \ldots\})$, i.e., $\mathrm{rv}(p)$ is the smallest set such that $\mathcal{V}(t) \subseteq \mathrm{rv}(p)$ and such that $\mathcal{V}(x\sigma) \subseteq \mathrm{rv}(p)$ holds for all $x \in \mathrm{rv}(p)$.*

So the relevant variables of the pattern term $\mathsf{gt}(x, y)\ \varnothing^n\ \varnothing$ are x and y. In contrast, a pattern term $\mathsf{gt}(x, y)\ [x/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ \varnothing$ would have the relevant variables x, x', and y. Lemma 6 states that one can modify pattern substitutions as long as this only concerns variables that are not relevant in the pattern term.

Lemma 6 (Equivalence by Irrelevant Pattern Substitutions). *Let $p = t\,\sigma^n\,\mu$ be a pattern term and let σ' and μ' be substitutions such that $x\sigma = x\sigma'$ and $x\mu = x\mu'$ holds for all $x \in \mathrm{rv}(p)$. Then p is equivalent to $t\,\sigma'^n\,\mu'$.*

Proof. We prove $t\sigma^n = t\sigma'^n$ by induction on n. For $n = 0$ this is trivial. For $n > 0$, the induction hypothesis implies $t\sigma^{n-1} = t\sigma'^{n-1}$, and since $\mathcal{V}(t\sigma^{n-1}) \subseteq \mathrm{rv}(p)$, we also obtain $t\sigma^n = t\sigma'^n$. Finally, $\mathcal{V}(t\sigma^n) \subseteq \mathrm{rv}(p)$ implies $t\sigma^n\mu = t\sigma'^n\mu'$. □

Hence, since x', y' are not relevant in the pattern term $\mathsf{gt}(x,y)\ \varnothing^n\ \varnothing$, we can add the pattern substitutions from the left-hand side of Rule (7) also on its right-hand side. Thus, by applying **(IV)** (using Lemma 6) to (7), we obtain

$$\begin{array}{rll} & \mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y'))\ [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/x,y'/y] & \\ \hookrightarrow & \mathsf{gt}(x,y) \qquad [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/x,y'/y]. & \end{array} \tag{8}$$

Recall that our goal was to narrow the recursive gt-rule (resp. (8)) with the non-recursive gt-rule $\mathsf{gt}(\mathsf{s}(x),\mathsf{0}) \to \mathsf{tt}$. As a first step towards this goal, we now made the pattern substitutions on both sides of (8) equal.

2.3 Modifying Pattern Rules by Instantiation and Narrowing

For the desired narrowing, we have to instantiate the recursive pattern rule (8) such that the base term of its right-hand side contains the left-hand side of the rule $\mathsf{gt}(\mathsf{s}(x),\mathsf{0}) \to \mathsf{tt}$. To this end, we use inference rule **(V)**. For any two substitutions σ and ρ, let σ_ρ result from the composition of σ and ρ, but restricted to the domain of σ. Thus, $\sigma_\rho = [x/s\rho \mid x/s \in \sigma]$.

(V) Instantiation

$$\frac{s\,\sigma_s^n\,\mu_s\ \hookrightarrow\ t\,\sigma_t^n\,\mu_t}{(s\rho)\ (\sigma_s)_\rho^n\ (\mu_s)_\rho\ \hookrightarrow\ (t\rho)\ (\sigma_t)_\rho^n\ (\mu_t)_\rho} \text{ if } \mathcal{V}(\rho) \cap (\mathrm{dom}(\sigma_s) \cup \mathrm{dom}(\mu_s) \cup \mathrm{dom}(\sigma_t) \cup \mathrm{dom}(\mu_t)) = \varnothing$$

Hence, we now apply inference rule **(V)** on the pattern rule (8) using $\rho = [x/\mathsf{s}(x), y/\mathsf{0}]$. The domain variables of (8) are x' and y'. Thus, due to the domain renaming in Sect. 2.2 they are disjoint from $\mathcal{V}(\rho) = \{x, y\}$. In the resulting pattern rule, the base terms are instantiated with ρ and the new pattern substitutions result from composing the previous pattern substitutions with ρ (restricted to the domains of the previous substitutions). So for $\sigma = [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]$ we have $\sigma_\rho = \sigma$ and for $\mu = [x'/x, y'/y]$, we obtain $\mu_\rho = [x'/\mathsf{s}(x), y'/\mathsf{0}]$. This yields

$$\begin{array}{rll} & \mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y'))\ [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & \\ \hookrightarrow & \mathsf{gt}(\mathsf{s}(x),\mathsf{0}) \qquad [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & \end{array} \tag{9}$$

For the narrowing, the original rule $\mathsf{gt}(\mathsf{s}(x),\mathsf{0}) \to \mathsf{tt}$ of the TRS can be transformed to a pattern rule $\mathsf{gt}(\mathsf{s}(x),\mathsf{0})\ \varnothing^n\ \varnothing\ \hookrightarrow \mathsf{tt}\ \varnothing^n\ \varnothing$ by **(I)**. Afterwards, one can add the pattern substitutions of (9) by Rule **(IV)** using Lemma 6, since x', y' are not relevant in the pattern rule:

$$\begin{array}{rll} & \mathsf{gt}(\mathsf{s}(x),\mathsf{0})\ [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & \\ \hookrightarrow & \mathsf{tt} \qquad [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & \end{array} \tag{10}$$

Now all pattern terms in (9) and (10) have the same pattern substitutions.

Hence, we can apply the narrowing rule **(VI)** which rewrites the right-hand side of one pattern rule with another pattern rule, if the pattern substitutions of all pattern terms coincide.

(VI) Narrowing

$$\frac{s\,\sigma^n\,\mu\ \hookrightarrow\ t\,\sigma^n\,\mu \qquad u\,\sigma^n\,\mu\ \hookrightarrow\ v\,\sigma^n\,\mu}{s\,\sigma^n\,\mu\ \hookrightarrow\ t[v]_\pi\,\sigma^n\,\mu} \text{ if } t|_\pi = u$$

In our example, $s\,\sigma^n\,\mu\ \hookrightarrow\ t\,\sigma^n\,\mu$ is the pattern rule (9) and $u\,\sigma^n\,\mu\ \hookrightarrow\ v\,\sigma^n\,\mu$ is the pattern rule (10). Thus, we have $t = \mathsf{gt}(\mathsf{s}(x),\mathsf{0}) = u$ and we obtain the following new pattern rule (which corresponds to Rule (1) in the introduction).

$$\begin{array}{rll} & \mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y'))\ [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & (11)\\ \hookrightarrow & \mathsf{tt}\qquad\qquad\quad [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & \end{array}$$

In general, to make the narrowing rule **(VI)** applicable for two rules $s\,\sigma_s^n\,\mu_s \hookrightarrow t\,\sigma_t^n\,\mu_t$ and $u\,\sigma_u^n\,\mu_u \hookrightarrow v\,\sigma_v^n\,\mu_v$, one should first instantiate the base terms t, u such that t contains u. Then one should try to make the substitutions $\sigma_s, \sigma_t, \sigma_u, \sigma_v$ equal and finally, one should try to make $\mu_s, \mu_t, \mu_u, \mu_v$ identical.

To illustrate that, let us try to narrow the pattern rule $\mathsf{f}(\mathsf{tt},x,y)\,\varnothing^n\,\varnothing \hookrightarrow \mathsf{f}(\mathsf{gt}(x,y),\mathsf{dbl}(x),\mathsf{s}(y))\,\varnothing^n\,\varnothing$ resulting from the f-rule with the above pattern rule (11) for gt. To let the base term $\mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y'))$ of (11)'s left-hand side occur in the right-hand side of f's pattern rule, we instantiate the latter with the substitution $[x/\mathsf{s}(x'),y/\mathsf{s}(y')]$. Thus, inference rule **(V)** yields

$$\mathsf{f}(\mathsf{tt},\mathsf{s}(x'),\mathsf{s}(y'))\,\varnothing^n\,\varnothing \hookrightarrow \mathsf{f}(\mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y')),\mathsf{dbl}(\mathsf{s}(x')),\mathsf{s}^2(y'))\,\varnothing^n\,\varnothing. \qquad (12)$$

Now we try to replace the current pumping substitution σ of Rule (12) by the one of (11). To this end, we use inference rule **(VII)** which allows us to instantiate pumping substitutions.

(VII) Instantiating σ

$$\frac{s\,\sigma_s^n\,\mu_s \hookrightarrow t\,\sigma_t^n\,\mu_t}{s\,(\sigma_s\rho)^n\,\mu_s \hookrightarrow t\,(\sigma_t\rho)^n\,\mu_t}$$

if ρ commutes with σ_s, μ_s, σ_t, and μ_t

So in our example, we apply inference rule **(VII)** to the pattern rule (12) using the substitution $\rho = [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]$. Since the pattern substitutions of (12) are just $\varnothing$, ρ trivially commutes with them. Hence, we obtain

$$\begin{array}{rll} & \mathsf{f}(\mathsf{tt},\mathsf{s}(x'),\mathsf{s}(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ \varnothing & (13)\\ \hookrightarrow & \mathsf{f}(\mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y')),\mathsf{dbl}(\mathsf{s}(x')),\mathsf{s}^2(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ \varnothing. & \end{array}$$

Note that **(VII)** differs from the previous instantiation rule **(V)** which does not add new variables to the domains of the pattern substitutions (i.e., with **(V)** we would not have been able to modify the pattern substitutions of (12)).

To make also the closing substitutions of the f-rule (13) and the gt-rule (11) identical, we use inference rule **(VIII)** which allows arbitrary instantiations of pattern rules (i.e., in contrast to **(V)** and **(VII)**, here we impose no conditions on ρ).

(VIII) Instantiating μ

$$\frac{s\,\sigma_s^n\,\mu_s \hookrightarrow t\,\sigma_t^n\,\mu_t}{s\,\sigma_s^n\,(\mu_s\rho) \hookrightarrow t\,\sigma_t^n\,(\mu_t\rho)}$$

Applying inference rule **(VIII)** to Rule (13) with $\rho = [x'/\mathsf{s}(x),y'/\mathsf{0}]$ yields

$$\begin{array}{rll} & \mathsf{f}(\mathsf{tt},\mathsf{s}(x'),\mathsf{s}(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & (14)\\ \hookrightarrow & \mathsf{f}(\mathsf{gt}(\mathsf{s}(x'),\mathsf{s}(y')),\mathsf{dbl}(\mathsf{s}(x')),\mathsf{s}^2(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}]. & \end{array}$$

By **(VI)**, now one can narrow (14) with the gt-rule (11) which yields

$$\begin{array}{rll} & \mathsf{f}(\mathsf{tt},\mathsf{s}(x'),\mathsf{s}(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}] & (15)\\ \hookrightarrow & \mathsf{f}(\mathsf{tt},\mathsf{dbl}(\mathsf{s}(x')),\mathsf{s}^2(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x),y'/\mathsf{0}]. & \end{array}$$

So to narrow a pattern rule with another one, we require identical pattern substitutions. Moreover, we only allow narrowing of the *base term* (i.e., the

narrowing rule **(VI)** does not modify terms in the ranges of the pattern substitutions). In contrast, rewriting with ordinary rules is also allowed in the pattern substitutions and moreover, here the two pattern terms in the pattern rule may also have different pattern substitutions.

(IX) Rewriting

$$\frac{p \;\hookrightarrow\; t\,\sigma^n\,\mu}{p \;\hookrightarrow\; t'\,\sigma'^n\,\mu'} \quad \begin{array}{l}\text{if } t \to_{\mathcal{R}}^* t',\ \forall x \in \mathcal{V} : x\sigma \to_{\mathcal{R}}^* x\sigma', \\ \text{and } \forall x \in \mathcal{V} : x\mu \to_{\mathcal{R}}^* x\mu'\end{array}$$

While no rewriting is possible for the terms in the ranges of the pattern substitutions of (15), one can rewrite the base term using the dbl-rule:

$$\begin{array}{lll} & \mathsf{f}(\mathsf{tt}, \mathsf{s}(x'), \mathsf{s}(y')) & [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x), y'/\mathsf{0}] \\ \hookrightarrow & \mathsf{f}(\mathsf{tt}, \mathsf{times}(\mathsf{s}^2(\mathsf{0}), \mathsf{s}(x')), \mathsf{s}^2(y')) & [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x), y'/\mathsf{0}] \end{array} \tag{16}$$

To continue our example further, we now want to narrow the above f-rule (16) with the pattern rule (4) for times. To make the narrowing rule **(VI)** applicable, the base term of (4)'s left-hand side must occur in (16) and all four pattern terms in the rules must have the same pattern substitutions. Thus, one first has to transform the pattern rules by the equivalence rule **(IV)** (using Lemmas 4 and 6) and instantiations (using **(V)**, **(VII)**, and **(VIII)**). After the narrowing, one can simplify the resulting pattern rule by rewriting (Rule **(IX)**) and by removing irrelevant parts of substitutions (Rule **(IV)** using Lemma 6), which yields

$$\begin{array}{lll} & \mathsf{f}(\mathsf{tt}, \mathsf{s}(x'), \mathsf{s}(y')) & [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(x), y'/\mathsf{0}] \\ \hookrightarrow & \mathsf{f}(\mathsf{tt}, \mathsf{s}^2(z), \mathsf{s}^2(y')) & [y'/\mathsf{s}(y'), z/\mathsf{s}^2(z)]^n\ [y'/\mathsf{0}, z/\mathsf{times}(\mathsf{s}^2(\mathsf{0}), \mathsf{s}(x))]. \end{array} \tag{17}$$

The following theorem shows that all our inference rules are sound.

Theorem 7 (Soundness of Inference Rules). *For all inference rules* **(I)** - **(IX)** *of the form* $\frac{p_1 \hookrightarrow q_1 \quad \dots \quad p_k \hookrightarrow q_k}{p \hookrightarrow q}$*, if all pattern rules* $p_1 \hookrightarrow q_1, \dots, p_k \hookrightarrow q_k$ *are correct w.r.t. a TRS* $\mathcal{R}$*, then the pattern rule* $p \hookrightarrow q$ *is also correct w.r.t.* $\mathcal{R}$*.*

Proof. Soundness of Rule **(I)** is trivial. Soundness of Rule **(II)** is proved by induction on n. For $n = 0$, we have $s\,\sigma^0 = s \to_{\mathcal{R}}^+ t = t\,\theta^0$, since $s\,\varnothing^n\,\varnothing \;\hookrightarrow\; t\,\varnothing^n\,\varnothing$ is correct w.r.t. $\mathcal{R}$. For $n > 0$, we obtain $s\,\sigma^n \to_{\mathcal{R}}^+ t\,\theta^{n-1}\,\sigma$ by the induction hypothesis. Since θ and σ commute, we have $t\,\theta^{n-1}\,\sigma = t\sigma\,\theta^{n-1} = s\,\theta^n \to_{\mathcal{R}}^+ t\,\theta^n$.

Soundness of Rule **(III)** is also proved by induction on n. For $n = 0$, we have $s\,\sigma^0 = s \to_{\mathcal{R}}^+ t = t[z]_\pi\,[z/t|_\pi] = t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^0\,[z/t|_\pi]$. For $n > 0$, we obtain

$$\begin{array}{lll} s\,\sigma^n & = & s\,\sigma^{n-1}\,\sigma \\ & \to_{\mathcal{R}}^+ & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n-1}\,[z/t|_\pi]\,\sigma & \text{by induction hypothesis} \\ & = & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n-1}\,(\sigma \cup [z/t|_\pi\sigma]) & \text{since } z \notin \mathrm{dom}(\sigma) \\ & = & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n-1}\,(\sigma \cup [z/s]) & \\ & \to_{\mathcal{R}}^+ & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n-1}\,(\sigma \cup [z/t]) & \\ & = & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n-1}\,(\sigma \cup [z/t[z]_\pi])\,[z/t|_\pi] & \text{since } z \notin \mathrm{range}(\sigma) \\ & = & t[z]_\pi\,(\sigma \cup [z/t[z]_\pi])^{n}\,[z/t|_\pi] & \end{array}$$

Rule **(IV)** is trivially sound. For Rule **(V)**, note that correctness of $s\,\sigma_s^n\,\mu_s \;\hookrightarrow\; t\,\sigma_t^n\,\mu_t$ also implies correctness of $s\,\sigma_s^n\,(\mu_s\rho) \;\hookrightarrow\; t\,\sigma_t^n\,(\mu_t\rho)$. But we have

$$\begin{array}{lll} s\;\sigma_s^n\;(\mu_s\,\rho) = s\;\sigma_s^n\;\rho\;\mu_{s_\rho} & & \text{since } \mathcal{V}(\rho) \cap \mathrm{dom}(\mu_s) = \varnothing \\ \qquad = (s\rho)\;(\sigma_s)_\rho^n\;(\mu_s)_\rho & & \text{since } \mathcal{V}(\rho) \cap \mathrm{dom}(\sigma_s) = \varnothing. \end{array}$$

Similarly, $t\,\sigma_t^n\,(\mu_t\,\rho) = (t\rho)\,(\sigma_t)_\rho^n\,(\mu_t)_\rho$, which implies soundness of Rule **(V)**.

Soundness of Rule **(VI)** is trivial. For soundness of Rule **(VII)**, correctness of $s\,\sigma_s^n\,\mu_s \;\hookrightarrow\; t\,\sigma_t^n\,\mu_t$ also implies correctness of $s\,\sigma_s^n\,(\mu_s\,\rho^n) \;\hookrightarrow\; t\,\sigma_t^n\,(\mu_t\,\rho^n)$. As ρ commutes with σ_s, μ_s, σ_t, μ_t, this is equivalent to $s\,(\sigma_s\,\rho)^n\,\mu_s \;\hookrightarrow\; t\,(\sigma_t\,\rho)^n\,\mu_t$. Soundness of Rules **(VIII)** and **(IX)** is again straightforward. □

2.4 Detecting Non-Termination

Thm. 8 introduces a criterion to detect pattern rules that directly lead to non-termination. Hence, whenever we have inferred a new pattern rule that satisfies this criterion, we can conclude non-termination of our TRS.

For a pattern rule $s\,\sigma^n\,\mu \;\hookrightarrow\; t\,\sigma_t^n\,\mu_t$, we check whether the pattern substitutions of the right-hand side are specializations of the pattern substitutions of the left-hand side. More precisely, there must be an $m \in \mathbb{N}$ such that $\sigma_t = \sigma^m\,\sigma'$ and $\mu_t = \mu\,\mu'$ for some σ' and μ', where σ' commutes with σ and μ. Then one only has to check whether there is a $b \in \mathbb{N}$ such that $s\,\sigma^b$ is equal to some subterm of t.

Theorem 8 (Detecting Non-Termination). *Let $s\,\sigma^n\,\mu \;\hookrightarrow\; t\,\sigma_t^n\,\mu_t$ be correct w.r.t. a TRS $\mathcal{R}$ and let there be an $m \in \mathbb{N}$ such that $\sigma_t = \sigma^m\,\sigma'$ and $\mu_t = \mu\,\mu'$ for some substitutions σ' and μ', where σ' commutes with both σ and μ. If there is a $\pi \in \mathcal{P}\mathrm{os}(t)$ and some $b \in \mathbb{N}$ such that $s\,\sigma^b = t|_\pi$, then $\mathcal{R}$ is non-terminating.*

Proof. We show that for all $n \in \mathbb{N}$, the term $s\,\sigma^n\,\mu$ rewrites to a term containing an instance of $s\,\sigma^{m\cdot n+b}\mu$. By repeating these rewrite steps on this subterm, we obtain an infinite rewrite sequence. Here, $\trianglerighteq$ denotes the superterm relation.

$$\begin{array}{lll}
s\,\sigma^n\,\mu \to_{\mathcal{R}}^+ t\,\sigma_t^n\,\mu_t & & \text{since } s\,\sigma^n\,\mu \;\hookrightarrow\; t\,\sigma_t^n\,\mu_t \text{ is correct} \\
\quad \trianglerighteq\; t|_\pi\,\sigma_t^n\,\mu_t & & \\
\quad =\; s\,\sigma^b\,\sigma_t^n\,\mu_t & & \\
\quad =\; s\,\sigma^b\,(\sigma^m\,\sigma')^n\,(\mu\,\mu') & & \\
\quad =\; s\,\sigma^{m\cdot n+b}\,\mu\,\sigma'^n\,\mu' & & \text{since } \sigma' \text{ commutes with both } \sigma \text{ and } \mu
\end{array}$$ □

To apply Thm. 8 to the pattern rule (17) obtained in our example, we have to transform the rule such that the pattern substitutions on the right-hand side become specializations of the pattern substitutions on the left-hand side. Thus, we use a domain renaming for the right-hand side to rename the variable z to x' (using Rule **(IV)** with Lemma 4). Moreover, we would like to get rid of the closing substitution $[x'/\mathsf{s}(x)]$ on the left-hand side. To this end, we first apply $[x/x']$ to the whole pattern rule (using inference rule **(VIII)**) and remove irrelevant parts of the pattern substitutions (Rule **(IV)** with Lemma 6), which yields

$$\begin{array}{lll}
 & \mathsf{f}(\mathsf{tt},\mathsf{s}(x'),\mathsf{s}(y')) & [x'/\mathsf{s}(x'),y'/\mathsf{s}(y')]^n \;\; [x'/\mathsf{s}(x'),y'/\mathsf{0}] \\
\hookrightarrow\; & \mathsf{f}(\mathsf{tt},\mathsf{s}^2(x'),\mathsf{s}^2(y')) & [x'/\mathsf{s}^2(x'),y'/\mathsf{s}(y')]^n\;[x'/\mathsf{times}(\mathsf{s}^2(\mathsf{0}),\mathsf{s}(x')),y'/\mathsf{0}].
\end{array} \tag{18}$$

Now the closing substitution $[x'/\mathsf{s}(x')]$ on the left-hand side of the rule can be moved from the closing substitution to the base term. This is stated by the following lemma, which can be used in addition to Lemmas 4 and 6 in order to transform pattern terms to equivalent other pattern terms in inference rule **(IV)**.

Lemma 9 (Equivalence by Simplifying μ). *Let $p = t\,\sigma^n\,\mu$ be a pattern term and let $\mu = \mu_1\,\mu_2$ where μ_1 commutes with σ. Then p is equivalent to $(t\,\mu_1)\,\sigma^n\,\mu_2$.*

Proof. For any n, $t\,\sigma^n\,\mu = t\,\sigma^n\,\mu_1\mu_2 = t\mu_1\,\sigma^n\,\mu_2$, as μ_1 commutes with σ. □

The closing substitution μ of (18)'s left-hand side has the form $\mu = \mu_1\,\mu_2$ for $\mu_1 = [x'/\mathsf{s}(x')]$ and $\mu_2 = [y'/\mathsf{0}]$. Since μ_1 commutes with $\sigma = [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]$, by inference rule **(IV)** and Lemma 9, we can replace the left-hand side of (18) by the equivalent pattern term $\mathsf{f}(\mathsf{tt}, \mathsf{s}^2(x'), \mathsf{s}(y'))\ [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ [y'/\mathsf{0}]$.

Moreover, by rewriting $\mathsf{times}(\mathsf{s}^2(\mathsf{0}), \mathsf{s}(x'))$ on the right-hand side using Rule **(IX)**, the right-hand side is transformed to $\mathsf{f}(\mathsf{tt}, \mathsf{s}^2(x'), \mathsf{s}^2(y'))\ [x'/\mathsf{s}^2(x'), y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}^2(\mathsf{times}(\mathsf{s}^2(\mathsf{0}), x')), y'/\mathsf{0}]$. So now its closing substitution μ' has the form $\mu' = \mu'_1\,\mu'_2$ for $\mu'_1 = [x'/\mathsf{s}(x')]$ and $\mu'_2 = [x'/\mathsf{s}(\mathsf{times}(\mathsf{s}^2(\mathsf{0}), x')), y'/\mathsf{0}]$. Since μ'_1 commutes with the pumping substitution $\sigma' = [x'/\mathsf{s}^2(x'), y'/\mathsf{s}(y')]$, by applying inference rule **(IV)** and Lemma 9 also on the right-hand side, we get

$$\begin{aligned} & \mathsf{f}(\mathsf{tt}, \mathsf{s}^2(x'), \mathsf{s}(y'))\ \ [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]^n\ \ [y'/\mathsf{0}] \qquad (19)\\ \hookrightarrow\ & \mathsf{f}(\mathsf{tt}, \mathsf{s}^3(x'), \mathsf{s}^2(y'))\ [x'/\mathsf{s}^2(x'), y'/\mathsf{s}(y')]^n\ [x'/\mathsf{s}(\mathsf{times}(\mathsf{s}^2(\mathsf{0}), x')), y'/\mathsf{0}]. \end{aligned}$$

The resulting rule (19) satisfies the conditions of Thm. 8, i.e., one can directly detect its non-termination. It has the form $s\,\sigma^n\,\mu \hookrightarrow t\,\sigma_t^n\,\mu_t$ with $\sigma = [x'/\mathsf{s}(x'), y'/\mathsf{s}(y')]$ and $\mu = [y'/\mathsf{0}]$, where $\sigma_t = \sigma\,\sigma'$ for $\sigma' = [x'/\mathsf{s}(x')]$ and $\mu_t = \mu\,\mu'$ for $\mu' = [x'/\mathsf{s}(\mathsf{times}(\mathsf{s}^2(\mathsf{0}), x'))]$. Clearly σ' commutes with σ and μ and moreover, $s\sigma = t$. Thus, non-termination of the TRS in Sect. 1 is proved.

Note that with our inference rules and the criterion of Thm. 8, one can also prove non-termination of any looping TRS $\mathcal{R}$. The reason is that then there is also a loop $s \to_{\mathcal{R}}^+ C[s\mu]$ where the first rewrite step is on the root position. By translating the rules of the TRS to pattern rules (Rule **(I)**) and by performing instantiation (Rule **(V)**) followed by narrowing or rewriting (Rule **(VI)** or **(IX)**) repeatedly, we can also obtain a corresponding pattern rule $s\,\varnothing^n\,\varnothing \hookrightarrow C[s\mu]\,\varnothing^n\,\varnothing$. To detect its non-termination by Thm. 8, we replace the closing substitution $\varnothing$ by μ (using Rule **(VIII)**) which yields $s\,\varnothing^n\,\mu \hookrightarrow C[s\mu]\,\varnothing^n\,\mu$. Simplifying the closing substitution on the left-hand side (Rule **(IV)** with Lemma 9) yields $(s\mu)\,\varnothing^n\,\varnothing \hookrightarrow C[s\mu]\,\varnothing^n\,\mu$. Since the closing substitution μ on the right-hand side is a specialization of the closing substitution $\varnothing$ on the left-hand side and since $s\mu$ is equal to a subterm of $C[s\mu]$, Thm. 8 now detects non-termination.

3 A Strategy to Prove Non-Termination Automatically

The inference rules in Sect. 2 constitute a powerful calculus to prove non-termination. We now present a strategy for their automated application which turned out to be successful in our implementation in the tool AProVE, cf. Sect. 4.

The strategy first transforms all rules of the TRS[4] into pattern rules using Rule **(I)** and if possible, one uses Rules **(II)** and **(III)** afterwards to obtain

[4] It is preferable to check non-termination within the *dependency pair framework* [5, 7, 8]. In this way, one can automatically decompose the TRS into parts where termination can easily be proved and into parts which can potentially cause non-termination.

pattern rules with non-empty pattern substitutions. Then for every pattern rule $p \hookrightarrow q$, one repeatedly tries to rewrite its right-hand side (Rule **(IX)**) or to narrow it with every pattern rule $p' \hookrightarrow q'$ (see below). Whenever a new pattern rule is obtained, one checks whether it satisfies the non-termination criterion of Thm. 8.[5] In this case, the procedure stops and non-termination has been proved.

Before trying to narrow $p \hookrightarrow q$ with $p' \hookrightarrow q'$ at some $\pi \in \mathcal{P}\mathrm{os}(\mathrm{base}(q))$, to avoid conflicting instantiations of variables, one uses domain renamings to ensure that $\mathrm{dv}(p)$, $\mathrm{dv}(q)$, $\mathrm{dv}(p')$, and $\mathrm{dv}(q')$ are pairwise disjoint (Rule **(IV)** with Lemma 4). Moreover, pattern rules are made variable-disjoint (using Rule **(V)**). Then the strategy proceeds by the following steps to make the narrowing rule **(VI)** applicable. After presenting the strategy, we illustrate it by an example.

1. Make $\mathrm{base}(q)|_\pi$ equal to $\mathrm{base}(p')$: If $\mathrm{base}(q)|_\pi$ and $\mathrm{base}(p')$ do not unify, then abort with failure. If $\mathrm{base}(q)|_\pi = \mathrm{base}(p')$, then go to Step 2. Otherwise, let $\theta = \mathrm{mgu}(\mathrm{base}(q)|_\pi, \mathrm{base}(p'))$, let $x \in \mathrm{dom}(\theta)$, and let $s = \theta(x)$. W.l.o.g. we assume $x \in \mathcal{V}(p')$ (the case where $x \in \mathcal{V}(q)$ works analogously).
   (a) If $x \notin \mathrm{dv}(p')$ and $s \notin \mathrm{dv}(p')$, then let s' result from s by renaming all variables from $\mathrm{dv}(p')$ occurring in s by pairwise different fresh variables. Instantiate $p' \hookrightarrow q'$ with $\rho = [x/s']$ (Rule **(V)**) and go back to Step 1.
   (b) If $x \notin \mathrm{dv}(p')$ and $s \in \mathrm{dv}(p')$, then use Rule **(VII)** to add x to the domain of p''s pumping substitution, such that it operates on x as it operates on s. To make Rule **(VII)** applicable, some pre-processing with Rules **(VIII)** and **(IV)** may be required. Then go back to Step 1 (resp. to case (c)). The case where $x \in \mathrm{dv}(p')$ and $s \in \mathcal{V}(p') \setminus \mathrm{dv}(p')$ is analogous.
   (c) If both $x, s \in \mathrm{dv}(p')$ and $[x/s]$ commutes with p''s pumping substitution, then apply **(VIII)** on $p' \hookrightarrow q'$ such that p''s closing substitution gets the form $[x/s]\,\mu$ for some μ. Then, move $[x/s]$ from p''s closing substitution to p''s base term with Rule **(IV)** (using Lemma 9) and go to Step 1.
   (d) If $x \in \mathrm{dv}(p')$ and $s \in \mathcal{V} \setminus \mathcal{V}(p')$, then apply Rule **(IV)** (using Lemma 4) with the domain renaming $[x/s]$ on $p' \hookrightarrow q'$ and go back to Step 1.
   (e) Otherwise, abort with failure.
2. Make the pumping substitutions of p, q, p', and q' equal (without changing $\mathrm{base}(q), \mathrm{base}(p')$): resolve all conflicts using Rules **(VII)** and **(IV)**.
3. Make the closing substitutions of p, q, p', q' equal (without changing pumping substitutions or $\mathrm{base}(q), \mathrm{base}(p')$): resolve conflicts by **(VIII)** and **(IV)**.
4. Apply narrowing according to Rule **(VI)**.

To illustrate the strategy, consider the TRS with the plus-rules of Sect. 1 and

$$\mathsf{f}(\mathsf{tt}, x) \to \mathsf{f}(\mathsf{isNat}(x), \mathsf{plus}(x, x)), \qquad \mathsf{isNat}(\mathsf{0}) \to \mathsf{tt}, \qquad \mathsf{isNat}(\mathsf{s}(y)) \to \mathsf{isNat}(y).$$

[5] To this end, one tries to transform the pattern rule using Rules **(IV)** and **(VIII)** such that the pattern substitutions on the right-hand sides become specializations of the corresponding pattern substitutions on the left-hand sides.

After creating pattern rules for f, isNat, and plus, we narrow the recursive isNat- and plus-rules with the non-recursive ones. For plus, this results in

$$\mathsf{plus}(x, \mathsf{s}(y'))\,[y'/\mathsf{s}(y')]^n\,[y'/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}(x')\,[x'/\mathsf{s}(x')]^n\,[x'/x]. \tag{20}$$

Moreover, we use the resulting isNat-rule to narrow the f-rule, which yields

$$\mathsf{f}(\mathsf{tt}, \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{f}(\mathsf{tt}, \mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y)))\,[y/\mathsf{s}(y)]^n\,[y/\mathsf{0}]. \tag{21}$$

Now our goal is to narrow the f-rule (21) with the plus-rule (20). We begin with Step 1 in the strategy. The mgu of $\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y))$ (in (21)'s right-hand side q) and $\mathsf{plus}(x, \mathsf{s}(y'))$ (in (20)'s left-hand side p') is $\theta = [y'/y, x/\mathsf{s}(y)]$. Let us first regard the variable y'. Since $y' \in \mathrm{dv}(p')$ and $y \in \mathcal{V} \setminus \mathcal{V}(p')$, we are in Case (d). Thus, we apply the domain renaming $[y'/y]$ to (20) (with Rule **(IV)**) and obtain

$$\mathsf{plus}(x, \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}(x')\,[x'/\mathsf{s}(x')]^n\,[x'/x]. \tag{22}$$

Now $\theta = \mathrm{mgu}(\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y)), \mathsf{plus}(x, \mathsf{s}(y))) = [x/\mathsf{s}(y)]$. Since x is no domain variable of (22)'s left-hand side and $\mathsf{s}(y) \notin \mathcal{V}$, we are in Case (a). Thus, we apply **(V)** with $\rho = [x/\mathsf{s}(z)]$ for a fresh $z \in \mathcal{V}$. After simplification with **(IV)**, we get

$$\mathsf{plus}(\mathsf{s}(z), \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}(x')\,[x'/\mathsf{s}(x')]^n\,[x'/\mathsf{s}(z)]. \tag{23}$$

Now $\theta = \mathrm{mgu}(\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y)), \mathsf{plus}(\mathsf{s}(z), \mathsf{s}(y))) = [z/y]$. Since z is no domain variable of (23)'s left-hand side, but y is, we are in Case (b). Hence, our goal is to extend the pumping substitution $[y/\mathsf{s}(y)]$ to operate on z as on y (i.e., we want to add $[z/\mathsf{s}(z)]$). To make Rule **(VII)** applicable, we have to remove the closing substitution $[x'/\mathsf{s}(z)]$ on (23)'s right-hand side which does not commute with $[z/\mathsf{s}(z)]$. To this end, we instantiate (23)'s closing substitutions with $[z/x']$ (Rule **(VIII)**) and simplify both sides of (23) using Rule **(IV)** with Lemmas 9 and 6.

$$\mathsf{plus}(\mathsf{s}(x'), \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}^2(x')\,[x'/\mathsf{s}(x')]^n\,\varnothing \tag{24}$$

Now $\theta = \mathrm{mgu}(\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y)), \mathsf{plus}(\mathsf{s}(x'), \mathsf{s}(y))) = [x'/y]$ for the non-domain variable x' and the domain variable y. Thus, we can proceed according to Case (b) and add $[x'/\mathsf{s}(x')]$ to the pumping substitutions of (24) using Rule **(VII)**.

$$\mathsf{plus}(\mathsf{s}(x'), \mathsf{s}(y))\,[x'/\mathsf{s}(x'), y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}^2(x')\,[x'/\mathsf{s}^2(x')]^n\,\varnothing \tag{25}$$

We still have $\theta = \mathrm{mgu}(\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y)), \mathsf{plus}(\mathsf{s}(x'), \mathsf{s}(y))) = [x'/y]$. But now both x', y are domain variables of (25)'s left-hand side, i.e., we are in Case (c). Indeed, now $[x'/y]$ commutes with the pumping substitution $[x'/\mathsf{s}(x'), y/\mathsf{s}(y)]$. So we instantiate the closing substitutions of (25) with $\rho = [x'/\mathsf{0}]$ (Rule **(VIII)**). Then the closing substitution $[y/\mathsf{0}, x'/\mathsf{0}]$ of (25)'s left-hand side has the form $[x'/y][y/\mathsf{0}]$ and hence, Rule **(IV)** with Lemma 9 yields

$$\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y))\,[x'/\mathsf{s}(x'), y/\mathsf{s}(y)]^n\,[y/\mathsf{0}] \;\hookrightarrow\; \mathsf{s}^2(x')\,[x'/\mathsf{s}^2(x')]^n\,[x'/\mathsf{0}]. \tag{26}$$

Thus, now the term $\mathsf{plus}(\mathsf{s}(y), \mathsf{s}(y))$ from the right-hand side of (21) also occurs on the left-hand side of (26), i.e., Step 1 is finished. In Step 2 of the strategy, we have to make the pumping substitutions of (21) and (26) equal. By Rule **(IV)** with Lemma 6 we first remove the irrelevant substitution $[x'/\mathsf{s}(x')]$ from the left-hand side of (26) and then extend the pumping substitutions by new irrelevant parts such that they all become $[x'/\mathsf{s}^2(x'), y/\mathsf{s}(y)]$. Similarly, in Step 3 of the strategy,

all closing substitutions are extended to $[x'/0, y/0]$ by Rule **(IV)** with Lemma 6. Now narrowing the f- with the plus-rule (by Rule **(VI)**) and subsequent removal of irrelevant substitutions (by Rule **(IV)** with Lemma 6) yields

$$\mathsf{f}(\mathsf{tt}, \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/0] \;\hookrightarrow\; \mathsf{f}(\mathsf{tt}, \mathsf{s}^2(x'))\,[x'/\mathsf{s}^2(x')]^n\,[x'/0]. \tag{27}$$

Hence, we now have to check whether (27) leads to non-termination due to Thm. 8. As in Footnote 5, to this end we apply a domain renaming $[x'/y]$ to (27)'s right-hand side in order to turn the pattern substitutions on the right-hand side into a specialization of the pattern substitutions on the left-hand side.

$$\mathsf{f}(\mathsf{tt}, \mathsf{s}(y))\,[y/\mathsf{s}(y)]^n\,[y/0] \;\hookrightarrow\; \mathsf{f}(\mathsf{tt}, \mathsf{s}^2(y))\,[y/\mathsf{s}^2(y)]^n\,[y/0]. \tag{28}$$

Rule (28) satisfies the criterion of Thm. 8. If σ is the pumping substitution $[y/\mathsf{s}(y)]$ of (28)'s left-hand side, then (28)'s right-hand side has the pumping substitution $\sigma\,\sigma$. Moreover, if s resp. t are the base terms of the two sides, then $s\sigma = t$. Thus, non-termination of the original (non-looping) TRS is proved.

4 Evaluation and Conclusion

We introduced a new technique to prove non-termination of possibly non-looping TRSs automatically. To this end, we adapted an idea of [10] from string to term rewriting and introduced *pattern rules* which represent a whole class of rewrite sequences. Afterwards, we presented 9 inference rules to deduce new pattern rules, a strategy for the application of these rules, and a criterion to detect non-terminating pattern rules. In this way, one can now repeatedly generate pattern rules until one obtains a rule which is detected to be non-terminating.

We implemented our contributions in the tool AProVE [6] and compared the new version AProVE-NL (for non-loop) with the previous version AProVE '11 and 3 other powerful tools for non-termination of TRSs (NTI [11], T$_{\text{T}}$T$_2$ [9], VMTL [12]). We ran the tools on the 1438 TRSs of the *Termination Problem Data Base* (*TPDB*) used in the annual *International Termination Competition*.[6] In the table, we consider those 241 TRSs of the TPDB where at least one tool proved non-termination. Moreover, we also tested the tools on 58 typical non-looping non-terminating TRSs obtained from actual programs and other sources (*"nl"*). We used a time-out of 1 minute for each example. "**N**" indicates how often **N**on-termination was proved and "**R**" gives the average **R**untime in seconds for each example. Thus, AProVE-NL could solve 75.9 % of the non-looping examples without compromising its power on looping examples, whereas the other tools cannot handle non-looping non-termination. To access our implementation via a web interface and for further details on our experiments, we refer to [1].

	TPDB		*nl*	
	N	**R**	**N**	**R**
AProVE-NL	232	6.6	44	5.2
AProVE '11	228	6.6	0	60.0
NTI	214	7.3	0	60.0
T$_{\text{T}}$T$_2$	194	2.5	0	10.4
VMTL	95	16.5	0	42.8

Future work will be concerned with (i) improving our strategy for applying inference rules and with (ii) extending the notion of pattern rules. To motivate (i),

[6] See `http://termination-portal.org/wiki/Termination_Competition`

we compared AProVE-NL with the tools Knocked for Loops (KFL) [15], Matchbox [13], and nonloop [10] for non-termination of *string* rewriting on the 1316 SRSs of the TPDB. The table regards those 156 SRSs where at least one tool proved non-termination. Only AProVE-NL and nonloop handle non-looping non-terminating SRSs, and AProVE-NL succeeds whenever nonloop succeeds. However, some looping SRSs are found by other tools, but not by our current strategy which mainly focuses on *term* rewriting.

	N	**R**
KFL	147	6.2
AProVE-NL	120	19.0
Matchbox	111	22.0
AProVE '11	97	31.1
nonloop	95	26.3
NTI	67	37.1
T$_{\mathsf{T}}$T$_2$	24	51.4
VMTL	0	56.8

For (ii), while our approach is "complete" for looping TRSs, there are TRSs whose non-termination cannot be proved with our inference rules. An example is the TRS with rules for isNat, double, and $\mathsf{f}(\mathsf{tt},\mathsf{tt},x,\mathsf{s}(y)) \to \mathsf{f}(\mathsf{isNat}(x),\mathsf{isNat}(y),\mathsf{s}(x),\mathsf{double}(\mathsf{s}(y)))$. Here, one needs the rule $\mathsf{f}(\mathsf{tt},\mathsf{tt},x,\mathsf{s}(y))\,[x/\mathsf{s}(x)]^n\,[y/\mathsf{s}(y)]^m\,[x/\mathsf{0},y/\mathsf{0}] \hookrightarrow \mathsf{f}(\mathsf{tt},\mathsf{tt},\mathsf{s}(x),\mathsf{s}(\mathsf{s}(y)))\,[x/\mathsf{s}(x)]^n\,[y/\mathsf{s}(\mathsf{s}(y))]^m\,[x/\mathsf{0},y/\mathsf{0}]$ with *two parameters* n and m, which goes beyond our current notion of pattern rules.

References

1. `http://aprove.informatik.rwth-aachen.de/eval/NonLooping/`.
2. M. Brockschmidt, T. Ströder, C. Otto, and J. Giesl. Automated detection of non-termination and `NullPointerExceptions` for Java Bytecode. In *Proc. FoVeOOS '11*, LNCS. To appear. Available from [1].
3. A. Geser and H. Zantema. Non-looping string rewriting. *Informatique Théorique et Applications*, 33(3):279–302, 1999.
4. A. Geser, D. Hofbauer, and J. Waldmann. Termination proofs for string rewriting systems via inverse match-bounds. *J. Automated Reasoning*, 34(4):365–385, 2005.
5. J. Giesl, R. Thiemann, P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In *Proc. FroCoS '05*, LNAI 3717, pages 216–231, 2005.
6. J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic termination proofs in the dependency pair framework. In *Proc. IJCAR '06*, LNAI 4130, pages 281–286, 2006.
7. J. Giesl, R. Thiemann, P. Schneider-Kamp, and S. Falke. Mechanizing and improving dependency pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.
8. N. Hirokawa and A. Middeldorp. Automating the dependency pair method. *Information and Computation*, 199(1-2):172–199, 2005.
9. M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean Termination Tool 2. In *Proc. RTA '09*, LNCS 5595, pages 295–304, 2009.
10. M. Oppelt. Automatische Erkennung von Ableitungsmustern in nichtterminierenden Wortersetzungssystemen, 2008. Diploma Thesis, HTWK Leipzig, Germany.
11. É. Payet. Loop detection in term rewriting using the eliminating unfoldings. *Theoretical Computer Science*, 403:307–327, 2008.
12. F. Schernhammer and B. Gramlich. VMTL - A modular termination laboratory. In *Proc. RTA '09*, LNCS 5595, pages 285–294, 2009.
13. J. Waldmann. Matchbox: A tool for match-bounded string rewriting. In *RTA '04*, LNCS 3091, pages 85–94, 2004.
14. Y. Wang and M. Sakai. On non-looping term rewriting. *WST '06*, p. 17-21, 2006.
15. H. Zankl, C. Sternagel, D. Hofbauer, and A. Middeldorp. Finding and certifying loops. In *Proc. SOFSEM '10*, LNCS 5901, pages 755–766, 2010.
16. H. Zantema. Termination of string rewriting proved automatically. *Journal of Automated Reasoning*, 34:105–139, 2005.