Non-Termination of Term Rewrite Systems Using Pattern Unfolding

Etienne Payet
LIM - Université de la Réunion, France

Abstract

We present a revisit, based on a new unfolding technique, of an approach introduced in term rewriting for the automatic detection of infinite non-looping derivations from patterns of rules.

2012 ACM Subject Classification Theory of computation → Rewrite systems; Theory of computation → Program analysis

Keywords and phrases Non-Termination, Non-Looping, Unfolding, Term Rewriting

1 Introduction

A derivation w.r.t. a term rewrite system (TRS) is called *non-looping* if it does not contain any loop, *i.e.*, any finite rewrite sequence where an instance of the starting term re-occurs as a subterm of the last term. In this paper, we present a work in progress on the automatic detection of infinite non-looping derivations w.r.t. a given TRS. We describe a reformulation of the approach of [2]: our contribution is the replacement of the nine inference rules of [2] for producing *pattern rules*, together with the strategy for their automated application, by a new unfolding technique. All we can say at the moment is that it provides a more compact presentation than that of [2], but we still need to compare the two approaches more precisely.

2 Preliminaries

$\mathbb{N}$ denotes the set of natural numbers. For all binary relations $\Rightarrow$ on a set A, a $\Rightarrow$*-chain* is a (possibly infinite) sequence $a_0 \Rightarrow a_1 \Rightarrow \cdots$ and $\Rightarrow^+$ (resp. $\Rightarrow^*$) denotes the transitive (resp. reflexive and transitive) closure of $\Rightarrow$.

2.1 Terms

We use the same definitions as [1] for terms. From now on, we fix a signature Σ and a set $H = \{\square_n \mid n \in \mathbb{N} \setminus \{0\}\}$ of constant symbols (called *holes*) disjoint from Σ. We also fix an infinite countable set X of variables disjoint from $\Sigma \cup H$. We let $S(\Sigma, X)$ denote the set of all substitutions from X to $T(\Sigma, X)$. We let $mgu(s,t)$ denote the set of most general unifiers of the terms (or sequences of terms) s and t. We denote function symbols by words in the *sans serif* font, *e.g.*, f, $\mathsf{0}$, while... We use the superscript notation to denote several successive applications of a unary function symbol, *e.g.*, $\mathsf{s}^2(\mathsf{0})$ stands for $\mathsf{s}(\mathsf{s}(\mathsf{0}))$ and $\mathsf{s}^0(\mathsf{0}) = \mathsf{0}$.

Let n be a positive integer. An n*-context* is an element of $T(\Sigma \cup H, X)$ that contains occurrences of $\square_1, \ldots, \square_n$ but no occurrence of another hole. For all n-contexts c and all $s_1, \ldots, s_n \in T(\Sigma \cup H, X)$, we let $c(s_1, \cdots, s_n)$ denote the element of $T(\Sigma \cup H, X)$ obtained from c by replacing all the occurrences of $\square_i$ by s_i, for all $1 \leq i \leq n$. We use the superscript notation for denoting several successive embeddings of a 1-context c into itself: $c^0 = \square_1$ and, for all $n \in \mathbb{N}$, $c^{n+1} = c(c^n)$. For all $n \in \mathbb{N}$, we denote by $\chi^{(n)}$ the set of n-contexts that contain exactly one occurrence of $\square_1, \ldots, \square_n$.

2.2 Term Rewriting

A *rule* is an element of $T(\Sigma, X)^2$ and a *term rewrite system (TRS)* is a set of rules. Given a rule (u, v), we let $[(u, v)] = \{(u\gamma, v\gamma) \mid \gamma$ is a renaming$\}$ denote its *equivalence class modulo renaming*. Moreover, for all TRSs $\mathcal{R}$, we let $[\mathcal{R}] = \bigcup_{r\in\mathcal{R}}[r]$. The rules of a TRS allow one to rewrite terms. This is formalised by the following binary relation $\rightarrow_\mathcal{R}$.

▶ **Definition 1.** *Let $\mathcal{R}$ be a TRS. We let $\rightarrow_\mathcal{R} = \bigcup\{\rightarrow_r \mid r \in \mathcal{R}\}$ where, for all $r = (u, v) \in \mathcal{R}$, $\rightarrow_r = \{(c(u\theta), c(v\theta)) \in T(\Sigma, X)^2 \mid c \in \chi^{(1)},\ \theta \in S(\Sigma, X)\}$. A rule (u, v) is* correct *w.r.t. $\mathcal{R}$ if $u \rightarrow^+_\mathcal{R} v$ holds. A TRS is correct w.r.t. $\mathcal{R}$ if all its rules are. We say that $\mathcal{R}$ is* non-terminating *(or* does not terminate*) if there exists an infinite $\rightarrow_\mathcal{R}$-chain.*

A *loop* in a TRS $\mathcal{R}$ is a finite $\rightarrow_\mathcal{R}$-chain of the form $s \rightarrow_{r_1} \cdots \rightarrow_{r_n} c(s\theta)$ where $s \in T(\Sigma, X)$, $r_1, \dots, r_n \in \mathcal{R}$, $c \in \chi^{(1)}$ and $\theta \in S(\Sigma, X)$. It gives rise to an infinite $\rightarrow_\mathcal{R}$-chain $s \rightarrow_{r_1} \cdots \rightarrow_{r_n} c(s\theta) \rightarrow_{r_1} \cdots \rightarrow_{r_n} c(c\theta(s\theta^2)) \rightarrow_{r_1} \cdots$. We say that a $\rightarrow_\mathcal{R}$-chain is *non-looping* if it does not contain any loop.

We unfold TRSs as follows. For the sake of readability, we write $[(u, v) \mid \dots]$ instead of $[\{(u, v) \mid \dots\}]$. Moreover, $(r_1, \dots, r_n) \ll_r [R]$ means that $(r_1, \dots, r_n)$ is a sequence of elements of $[R]$ variable disjoint from r and from each other.

▶ **Definition 2** (Unfolding). *For all TRSs $\mathcal{R}$ and R, we let*

$$U_\mathcal{R}(R) = \left[(u\theta, c(v_1, \dots, v_n)\theta) \;\middle|\; \begin{array}{l} r = (u, c(s_1, \dots, s_n)) \in \mathcal{R},\ c \in \chi^{(n)} \\ ((u_1, v_1), \dots, (u_n, v_n)) \ll_r [R] \\ \theta = mgu((s_1, \dots, s_n), (u_1, \dots, u_n)) \end{array} \right]$$

The unfolding of $\mathcal{R}$ from R *is the set $unf(\mathcal{R}, R) = (U_\mathcal{R})^*(R)$.*

▶ **Proposition 3.** *Let $\mathcal{R}$ and R be TRSs such that R is correct w.r.t. $\mathcal{R}$. Then, for all $n \in \mathbb{N}$, $U^n_\mathcal{R}(R)$ is correct w.r.t. $\mathcal{R}$, which implies that $unf(\mathcal{R}, R)$ also is.*

▶ **Example 4.** Let $\mathcal{R}$ be the TRS which consists of the rules

$$\begin{array}{ll} r_1 = (\mathsf{while}(\mathsf{true}, x, y), \mathsf{while}(\mathsf{gt}(x, y), \mathsf{add}(x, y), \mathsf{s}(y))) & r_4 = (\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)), \mathsf{gt}(x, y)) \\ r_2 = (\mathsf{gt}(\mathsf{s}(x), \mathsf{0}), \mathsf{true}) & r_5 = (\mathsf{add}(x, \mathsf{0}), x) \\ r_3 = (\mathsf{gt}(\mathsf{0}, y), \mathsf{false}) & r_6 = (\mathsf{add}(x, \mathsf{s}(y)), \mathsf{s}(\mathsf{add}(x, y))) \end{array}$$

and which corresponds to the imperative program fragment

```
while (x > y) { x = x + y; y = y + 1; }
```

Note that this fragment does not terminate if it is run from integers x, y such that $x > y > 0$. Moreover, for all $n > m > 0$, we have the infinite $\rightarrow_\mathcal{R}$-chain

$$\mathsf{while}(\mathsf{true}, \mathsf{s}^n(\mathsf{0}), \mathsf{s}^m(\mathsf{0})) \left(\underset{r_1}{\rightarrow} \circ \overset{m}{\underset{r_4}{\rightarrow}} \circ \underset{r_2}{\rightarrow} \circ \overset{m}{\underset{r_6}{\rightarrow}} \circ \underset{r_5}{\rightarrow} \right) \mathsf{while}(\mathsf{true}, \mathsf{s}^{n+m}(\mathsf{0}), \mathsf{s}^{m+1}(\mathsf{0}))$$
$$\left(\underset{r_1}{\rightarrow} \circ \overset{m+1}{\underset{r_4}{\rightarrow}} \circ \underset{r_2}{\rightarrow} \circ \overset{m+1}{\underset{r_6}{\rightarrow}} \circ \underset{r_5}{\rightarrow} \right) \cdots$$

It is non-looping because the number of applications of r_4 and r_6 gradually increases. Let us compute some elements of $unf(\mathcal{R}, \mathcal{R})$ by applying Def. 2.

- We have $r_4 = (\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y)), c(\mathsf{gt}(x, y))) \in \mathcal{R}$ where $c = \square_1 \in \chi^{(1)}$. Moreover, we have $(\mathsf{gt}(\mathsf{s}(x_1), \mathsf{0}), \mathsf{true}) \ll_{r_4} [\mathcal{R}]$ and $\theta = \{x \mapsto \mathsf{s}(x_1), y \mapsto \mathsf{0}\} = mgu(\mathsf{gt}(x, y), \mathsf{gt}(\mathsf{s}(x_1), \mathsf{0}))$. So, $r^{\mathsf{gt}}_1 = (\mathsf{gt}(\mathsf{s}(x), \mathsf{s}(y))\theta, c(\mathsf{true})\theta) \in U_\mathcal{R}(\mathcal{R})$ with $r^{\mathsf{gt}}_1 = (\mathsf{gt}(\mathsf{s}^2(x_1), \mathsf{s}(\mathsf{0})), \mathsf{true})$.

- More generally, for all $n \in \mathbb{N}$, $[r_n^{\mathsf{gt}}] \subseteq U_{\mathcal{R}}^n(\mathcal{R})$ with $r_n^{\mathsf{gt}} = \big(\mathsf{gt}(\mathsf{s}^{n+1}(x), \mathsf{s}^n(0)), \mathsf{true}\big)$.
- Identically, from r_6 and r_5 one gets: for all $n \in \mathbb{N}$, $[r_n^{\mathsf{add}}] \subseteq U_{\mathcal{R}}^n(\mathcal{R})$ with $r_n^{\mathsf{add}} = \big(\mathsf{add}(x, \mathsf{s}^n(0)), \mathsf{s}^n(x)\big)$.
- Let $n \in \mathbb{N}$. From r_1, r_n^{gt} and r_n^{add} one gets: $[r_n^{\mathsf{while}}] \subseteq U_{\mathcal{R}}^{n+1}(\mathcal{R})$ where

$$r_n^{\mathsf{while}} = (\mathsf{while}(\mathsf{true}, \mathsf{s}^{n+1}(x), \mathsf{s}^n(0)), \mathsf{while}(\mathsf{true}, \mathsf{s}^{2n+1}(x), \mathsf{s}^{n+1}(0)))$$

  As $\mathcal{R}$ is correct w.r.t. $\mathcal{R}$, $U_{\mathcal{R}}^{n+1}(\mathcal{R})$ also is (Prop. 3), *i.e.*, r_n^{while} also is. So, we have

$$\mathsf{while}(\mathsf{true}, \mathsf{s}^{n+1}(x), \mathsf{s}^n(0)) \xrightarrow[\mathcal{R}]{+} \mathsf{while}(\mathsf{true}, \mathsf{s}^{2n+1}(x), \mathsf{s}^{n+1}(0))$$

For all $n > m > 0$, we also have the infinite $\rightarrow_{\mathit{unf}(\mathcal{R},\mathcal{R})}$-chain

$$\mathsf{while}(\mathsf{true}, \mathsf{s}^n(0), \mathsf{s}^m(0)) \underset{r_m^{\mathsf{while}}}{\rightarrow} \mathsf{while}(\mathsf{true}, \mathsf{s}^{n+m}(x_1), \mathsf{s}^{m+1}(0)) \underset{r_{m+1}^{\mathsf{while}}}{\rightarrow} \cdots$$

It is non-looping because a new rule (not occurring before) is used at each step.

3 Pattern Unfolding

Now, we describe a reformulation, based on unfolding, of the pattern approach of [2].

▶ **Definition 5.** *A* pattern substitution *is a pair $\theta = (\sigma, \mu)$ of elements of $S(\Sigma, X)$. We rather denote it as $\sigma \star \mu$. For all $n \in \mathbb{N}$, we let $\theta(n) = \sigma^n \mu$. A* pattern term *is a pair $p = (s, \theta)$ where $s \in T(\Sigma, X)$ and θ is a pattern substitution. We denote it as $s \star \sigma \star \mu$ if $\theta = \sigma \star \mu$. For all $n \in \mathbb{N}$, we let $p(n) = s\theta(n)$. For all $s \in T(\Sigma, X)$, we let $s^\star = s \star \emptyset \star \emptyset$.*

For instance, if $\sigma = \{x \mapsto \mathsf{s}(x), y \mapsto \mathsf{s}(y)\}$ and $\mu = \{x \mapsto \mathsf{s}(x), y \mapsto 0\}$ then $\theta = \sigma \star \mu$ is a pattern substitution and $p = \mathsf{gt}(x, y) \star \sigma \star \mu$ is a pattern term. For all $n \in \mathbb{N}$, we have $\theta(n) = \sigma^n \mu = \{x \mapsto \mathsf{s}^{n+1}(x), y \mapsto \mathsf{s}^n(0)\}$ and $p(n) = \mathsf{gt}(x, y)\sigma^n \mu = \mathsf{gt}(\mathsf{s}^{n+1}(x), \mathsf{s}^n(0))$.

From pattern terms one can define pattern rules.

▶ **Definition 6.** *A* pattern rule *is a pair $r = (p, q)$ of pattern terms. It describes the set $\mathit{rules}(r) = \{(p(n), q(n)) \mid n \in \mathbb{N}\} \subseteq T(\Sigma, X)^2$.*

▶ **Example 7** (Ex. 4 continued)**.** Let $u = \mathsf{while}(\mathsf{true}, x, y)$ be the left-hand side of r_1, $\sigma = \{x \mapsto \mathsf{s}(x), y \mapsto \mathsf{s}(y)\}$, $\sigma' = \{x \mapsto \mathsf{s}(x)\}$ and $\mu = \{x \mapsto \mathsf{s}(x), y \mapsto 0\}$. The pattern terms

$$p = u\sigma \star \sigma \star \mu = \mathsf{while}(\mathsf{true}, \mathsf{s}(x), \mathsf{s}(y)) \star \sigma \star \mu$$
$$q = u\sigma^2 \star \sigma\sigma' \star \mu = \mathsf{while}(\mathsf{true}, \mathsf{s}^2(x), \mathsf{s}^2(y)) \star \{x \mapsto \mathsf{s}^2(x), y \mapsto \mathsf{s}(y)\} \star \mu$$

respectively describe the sets of terms $\big\{p(n) = \mathsf{while}\big(\mathsf{true}, \mathsf{s}^{n+2}(x), \mathsf{s}^{n+1}(0)\big) \bigm| n \in \mathbb{N}\big\}$ and $\big\{q(n) = \mathsf{while}\big(\mathsf{true}, \mathsf{s}^{2n+3}(x), \mathsf{s}^{n+2}(0)\big) \bigm| n \in \mathbb{N}\big\}$. Moreover,

$$\begin{aligned}\mathit{rules}((p, q)) &= \big\{\big(\mathsf{while}(\mathsf{true}, \mathsf{s}^{n+2}(x), \mathsf{s}^{n+1}(0)), \mathsf{while}(\mathsf{true}, \mathsf{s}^{2n+3}(x), \mathsf{s}^{n+2}(0))\big) \bigm| n \in \mathbb{N}\big\} \\ &= \big\{r_n^{\mathsf{while}} \mid n > 0\big\} \subseteq \big\{r_n^{\mathsf{while}} \mid n \in \mathbb{N}\big\} \subseteq \mathit{unf}(\mathcal{R}, \mathcal{R}) \text{ (see Ex. 4)}\end{aligned}$$

We adapt the notion of correctness (Def. 1) to pattern rules (and we get the notion defined in [2]).

▶ **Definition 8.** *Let $\mathcal{R}$ be a TRS. A pattern rule r is* correct *w.r.t. $\mathcal{R}$ if $\mathit{rules}(r)$ is. A set of pattern rules is correct w.r.t. $\mathcal{R}$ if all its elements are.*

So, if a pattern rule (p, q) is correct w.r.t. $\mathcal{R}$ then $p(n) \to^+_\mathcal{R} q(n)$ holds for all $n \in \mathbb{N}$. In Ex. 7, we have $rules((p, q)) \subseteq unf(\mathcal{R}, \mathcal{R})$. As $\mathcal{R}$ is correct w.r.t. $\mathcal{R}$, $unf(\mathcal{R}, \mathcal{R})$ also is (Prop. 3), *i.e.*, (p, q) also is. So, $\mathsf{while}(\mathsf{s}^{n+2}(x), \mathsf{s}^{n+1}(\mathsf{0})) \to^+_\mathcal{R} \mathsf{while}(\mathsf{s}^{2n+3}(x), \mathsf{s}^{n+2}(\mathsf{0}))$ holds for all $n \in \mathbb{N}$.

The next result allows one to infer correct pattern rules from a TRS. It considers pairs of rules that have the same form as (r_4, r_2), (r_4, r_3) and (r_6, r_5) in Ex. 4.

▶ **Proposition 9.** *Suppose that a TRS $\mathcal{R}$ contains two rules $r = (u, v)$, $r' = (u', v')$ s.t.*

- $u = c(c_1(x_1), \dots, c_m(x_m))$, $v = c'(c(x_1, \dots, x_m))$, $u' = c(t_1, \dots, t_m)$,
- $c_1, \dots, c_m, c'$ *are* 1*-contexts and* c *is an* m*-context with* $Var(c_1, \dots, c_m, c', c) = \emptyset$,
- $x_1, \dots, x_m \in X$ *are distinct and* $t_1, \dots, t_m \in T(\Sigma, X)$.

Let $\sigma = \{x_k \mapsto c_k(x_k) \mid 1 \le k \le m\}$ and $\mu = \{x_k \mapsto t_k \mid 1 \le k \le m\}$. Then, the pattern rule $\big(c(x_1, \dots, x_m) \star \sigma \star \mu, x_1 \star \{x_1 \mapsto c'(x_1)\} \star \{x_1 \mapsto v'\}\big)$ is correct w.r.t. $\mathcal{R}$

▶ **Example 10.** In Ex. 4, we have $r_4 = \big(c(c_1(x), c_2(y)), c'(c(x, y))\big)$ and $r_2 = (c(t_1, t_2), \mathsf{true})$ for $c = \mathsf{gt}(\square_1, \square_2)$, $c_1 = c_2 = \mathsf{s}(\square_1)$, $c' = \square_1$, $t_1 = \mathsf{s}(x)$ and $t_2 = \mathsf{0}$. So, by Prop. 9, (p_1, q_1) is correct w.r.t. $\mathcal{R}$ where $p_1 = \mathsf{gt}(x, y) \star \{x \mapsto \mathsf{s}(x), y \mapsto \mathsf{s}(y)\} \star \{x \mapsto \mathsf{s}(x), y \mapsto \mathsf{0}\}$ and $q_1 = x \star \emptyset \star \{x \mapsto \mathsf{true}\}$. Identically, from r_6 and r_5 one gets: (p_2, q_2) is correct w.r.t. $\mathcal{R}$ where $p_2 = \mathsf{add}(x, y) \star \{y \mapsto \mathsf{s}(y)\} \star \{y \mapsto \mathsf{0}\}$ and $q_2 = x \star \{x \mapsto \mathsf{s}(x)\} \star \emptyset$.

Unification for pattern terms is not considered in [2]. As we need it in our development (see Def. 13 below), we define it here.

▶ **Definition 11.** *Let p and q be pattern terms and θ be a pattern substitution. Then, θ is a* unifier *of p and q if $p(n)\theta(n) = q(n)\theta(n)$ for all $n \in \mathbb{N}$. Moreover, θ is a* most general unifier *(mgu) of p and q if $\theta(n) \in mgu(p(n), q(n))$ for all $n \in \mathbb{N}$. We let $mgu(p, q)$ be the set of all mgu's of p and q. All this is naturally extended to finite sequences of pattern terms.*

In Sect. 2.2, we have defined the equivalence class of a rule modulo renaming. We also need to adapt this concept to pattern rules.

▶ **Definition 12.** *For all pattern rules r, we let $[r]$ be the set of all pattern rules r' such that $rules(r') \subseteq [rules(r)]$. Moreover, for all sets of pattern rules R, we let $[R] = \bigcup_{r \in R} [r]$.*

Now, using the above concepts, we provide a counterpart of Def. 2 for pattern rules.

▶ **Definition 13.** *For all TRSs $\mathcal{R}$ and all sets of pattern rules R, we let*

$$U^\pi_\mathcal{R}(R) = \left[(p, q) \;\middle|\; \begin{array}{l} r = (u, c(s_1, \dots, s_n)) \in \mathcal{R},\ c \in \chi^{(n)} \\ ((p_1, v_1 \star \sigma_1 \star \mu_1), \dots, (p_n, v_n \star \sigma_n \star \mu_n)) \ll_r [R] \\ \sigma \star \mu \in mgu\left((s_1{}^\star, \dots, s_n{}^\star), (p_1, \dots, p_n)\right) \\ \sigma \text{ commutes with the } \sigma_i\text{s and } \mu_i\text{s} \\ p = u \star \sigma \star \mu \text{ and } q = c(v_1, \dots, v_n) \star \sigma_1 \dots \sigma_n \sigma \star \mu_1 \dots \mu_n \mu \end{array} \right]$$

The pattern unfolding *of $\mathcal{R}$ from R is the set $patunf(\mathcal{R}, R) = \big(U^\pi_\mathcal{R}\big)^*(R)$.*

▶ **Example 14** (Ex. 4 continued)**.** Let $R = \{(p_1, q_1), (p_2, q_2)\}$ (see Ex. 10) and

$$p'_1 = \mathsf{gt}(x_1, y_1) \star \{x_1 \mapsto \mathsf{s}(x_1), y_1 \mapsto \mathsf{s}(y_1)\} \star \{x_1 \mapsto \mathsf{s}(x_1), y_1 \mapsto \mathsf{0}\}$$
$$q'_1 = x_1 \star \emptyset \star \{x_1 \mapsto \mathsf{true}\}$$
$$p'_2 = \mathsf{add}(x_2, y_2) \star \{y_2 \mapsto \mathsf{s}(y_2)\} \star \{y_2 \mapsto \mathsf{0}\} \qquad q'_2 = x_2 \star \{x_2 \mapsto \mathsf{s}(x_2)\} \star \emptyset$$

Then, we have $((p'_1, q'_1), (p'_2, q'_2)) \ll_{r_1} [R]$ where $r_1 = (\mathsf{while}(\mathsf{true}, x, y), c(\mathsf{gt}(x, y), \mathsf{add}(x, y)))$ for $c = \mathsf{while}(\square_1, \square_2, \mathsf{s}(y)) \in \chi^{(2)}$. Moreover, $\rho \star \nu \in mgu\left(\left(\mathsf{gt}(x, y)^\star, \mathsf{add}(x, y)^\star\right), (p'_1, p'_2)\right)$

where $\rho = \{x \mapsto \mathsf{s}(x),\ y \mapsto \mathsf{s}(y),\ x_2 \mapsto \mathsf{s}(x_2)\}$ and $\nu = \{x \mapsto \mathsf{s}(x_1),\ y \mapsto \mathsf{0},\ x_2 \mapsto \mathsf{s}(x_1)\}$. We note that ρ commutes with the substitutions of q'_1 and q'_2. So, $r^{\mathsf{while}} = (\mathsf{while}(\mathsf{true}, x, y) \star \rho \star \nu, c(x_1, x_2) \star \rho' \star \nu') \in U^{\pi}_{\mathcal{R}}(R)$ where $\rho' = \emptyset\{x_2 \mapsto \mathsf{s}(x_2)\}\rho = \{x \mapsto \mathsf{s}(x),\ y \mapsto \mathsf{s}(y),\ x_2 \mapsto \mathsf{s}^2(x_2)\}$ and $\nu' = \{x_1 \mapsto \mathsf{true}\}\emptyset\nu = \nu \cup \{x_1 \mapsto \mathsf{true}\}$.

The following result corresponds to the Soundness Thm. 7 of [2].

▶ **Theorem 15.** *Let $\mathcal{R}$ be a TRS and R be a set of pattern rules. If R is correct w.r.t. $\mathcal{R}$ then $\mathit{patunf}(\mathcal{R}, R)$ also is.*

Non-termination can be detected from a pattern rule using the following criterion.

▶ **Theorem 16** (Thm. 8 of [2])**.** *Let $(s \star \sigma_s \star \mu_s, t \star \sigma_t \star \mu_t)$ be correct w.r.t. a TRS $\mathcal{R}$ and let there be an $m \in \mathbb{N}$ such that $\sigma_t = \sigma_s^m \sigma'$ and $\mu_t = \mu_s \mu'$ for some $\sigma', \mu' \in S(\Sigma, X)$, where σ' commutes with σ_s and μ_s. If there is a $\pi \in \mathit{Pos}(t)$ and some $b \in \mathbb{N}$ such that $s\sigma_s^b = t|_\pi$, then $s\sigma_s^n \mu_s$ starts an infinite (possibly non-looping) $\to_{\mathcal{R}}$-chain for all $n \in \mathbb{N}$.*

We note that the infinite chain of Thm. 16 may contain a loop (*e.g.*, if $m = 1$ and $b = 0$). But, as the following example illustrates, this is not always the case.

▶ **Example 17** (Ex. 7 and Ex. 14 continued)**.** Let us regard the pattern rule (p, q) of Ex. 7. As $\mathit{rules}((p, q)) \subseteq \{r_n^{\mathsf{while}} \mid n \in \mathbb{N}\} \subseteq [r_n^{\mathsf{while}} \mid n \in \mathbb{N}] = [\mathit{rules}(r^{\mathsf{while}})]$ (see Ex. 14), we have $(p, q) \in [r^{\mathsf{while}}]$ by Def. 12. So, as $[r^{\mathsf{while}}] \subseteq \mathit{patunf}(\mathcal{R}, R)$, we have $(p, q) \in \mathit{patunf}(\mathcal{R}, R)$. Moreover, as R is correct w.r.t. $\mathcal{R}$ (by Prop. 9), $\mathit{patunf}(\mathcal{R}, R)$ is correct w.r.t. $\mathcal{R}$ (by Thm. 15). So, (p, q) is correct w.r.t. $\mathcal{R}$. On the other hand, $(p, q) = (u\sigma \star \sigma \star \mu, u\sigma^2 \star \sigma\sigma' \star \mu)$ (see Ex. 7) and σ' commutes with σ and μ. Hence, by Thm. 16, for all $n \in \mathbb{N}$ the term $p(n) = u\sigma\sigma^n\mu = u\sigma^{n+1}\mu = \mathsf{while}(\mathsf{true}, \mathsf{s}^{n+2}(x), \mathsf{s}^{n+1}(\mathsf{0}))$ starts an infinite $\to_{\mathcal{R}}$-chain. This implies that for all $n > m > 0$, $\mathsf{while}(\mathsf{true}, \mathsf{s}^n(\mathsf{0}), \mathsf{s}^m(\mathsf{0}))$ starts an infinite $\to_{\mathcal{R}}$-chain (Ex. 4).

4 Conclusion

We have presented a work in progress on the detection of infinite non-looping chains in TRSs. There are still many tasks to be completed. We have to implement our approach and to compare it with that of [2]: for the moment, we simply observed that it is a reformulation of [2], but we have to investigate further. Note that we already implemented a similar approach in logic programming [6] and that we tested it on logic programs obtained by translating TRSs introduced by the authors of [2] to evaluate their work. Our experiments suggest that our approach and that of [2] are not orthogonal and do not completely overlap. We also have to compare our work with other techniques for detecting non-looping chains [3, 4, 5].

References

1 F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

2 F. Emmes, T. Enger, and J. Giesl. Proving non-looping non-termination automatically. In *Proc. IJCAR'12*, volume 7364 of *LNCS*, pages 225–240. Springer, 2012.

3 J. Endrullis and H. Zantema. Proving non-termination by finite automata. In *Proc. RTA'15*, volume 36 of *LIPIcs*, pages 160–176. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

4 M. Oppelt. Automatische Erkennung von Ableitungsmustern in nichtterminierenden Wortersetzungssystemen. Diploma Thesis, HTWK Leipzig, Germany, 2008.

5 É. Payet. Non-termination in term rewriting and logic programming. *Journal of Automated Reasoning*, 68(4):24 pages, 2024.

6 É. Payet. Non-termination of logic programs using patterns. To appear in Theory and Practice of Logic Programming, 2025.