Title	Termination Analysis for Innermost Rewriting via Transformations
Author(s)	Netrakom, Park
Citation	
Issue Date	2017-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/14798
Rights	
Description	Supervisor:廣川　直，情報科学研究科，修士

Termination Analysis for Innermost Rewriting via Transformations

By Park Netrakom

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Nao Hirokawa

September, 2017

Termination Analysis for Innermost Rewriting via Transformations

By Park Netrakom (1510213)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Nao Hirokawa

and approved by
Associate Professor Nao Hirokawa
Professor Mizuhito Ogawa
Professor Tachio Terauchi

August, 2017 (Submitted)

Contents

Chapter 1

Introduction

Term rewriting is one of simple and powerful Turing-complete computational models, which underlies automated theorem proving (e.g. Vampire, Agda, Coq) and declarative programming languages (e.g. CafeOBJ, OCaml, Haskell). *Termination* is a fundamental property that guarantees the finiteness of computation steps, which most software should satisfy. Termination of entire software guarantees that the result can be computed in a finite time; without termination, the software may loop and become unresponsive. While in theorem proving termination of functions must be ensured.

In this chapter, we intend to provide a brief and informal introduction to term rewriting, termination, and transformation along with motivation on this thesis, while the detailed surveys of term rewriting are referred to [3, 21, 18]. Then, we mention our approach, overview, and contributions of this thesis.

1.1 Term Rewriting and Termination

Term rewriting is a computational model regarding directed equations as computation step. A *term rewrite system (TRS)* is defined as a set of directed equations.

Example 1. The term rewrite system $\mathcal{R}$ consist of the following rules:

$$\begin{array}{rr} 1: & \mathsf{s}(x) + y \to \mathsf{s}(x+y) \\ 2: & \mathsf{0} + x \to x \end{array}$$

The term rewrite system $\mathcal{R}$ represents an additive program where a number is represented by successor of zero (e.g. $\mathsf{1}$ is $\mathsf{s}(\mathsf{0})$ and $\mathsf{3}$ is $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))$). Meanwhile, the execution of the program is simulated by rewriting steps.

Regarding to the TRS $\mathcal{R}$, we illustrate the execution step of $\mathsf{2} + \mathsf{3}$ by using the term rewrite system as following rewrite step:

$$\mathsf{s}(\mathsf{s}(\mathsf{0})) + \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{0}) + \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{0} + \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))))) \to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))))$$

The term $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))))$ cannot match with any left-hand side of rules in $\mathcal{R}$. Therefore, the execution ends.

The computational result, $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))))$ is called *a normal form*. However, in general, there is no guarantee that a system always admits a normal form, and also reaches a normal form even if it exists.

Example 2. We consider the term rewrite system $\mathcal{R}$ consist of the following rules:

$$\begin{aligned} 1: &\quad \mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{a}) \\ 2: &\quad \mathsf{a} \to \mathsf{b} \end{aligned}$$

The TRS $\mathcal{R}$ is non-terminating due to the following rewrite sequence:

$$\mathsf{f}(\mathsf{a}) \to_{\mathcal{R}} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R}} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R}} \cdots$$

The TRS $\mathcal{R}$ in Example 2 is non-terminating. However, if we restrict ourselves to the innermost rewriting strategy then the system is innermost terminating.

It is wildly known that halting problem is undecidable. Since this model is one of the Turing-complete computational models, there exists a mapping from a Turing-machine to the corresponding rewrite system. Therefore, the termination property of a rewrite system is also undecidable. Moreover, the system from the result of the mapping algorithm [4] yields the particular property which is called uncurrying system. In 1996, Gramlich discover that termination and innermost termination of an uncurrying system are equivalent [12]. Therefore, the undecidability holds for innermost termination.

1.2 Transformation

Because of the importance of the termination property, there are many researchers attempt to extend decidable subclasses of terminating TRSs [2, 6, 8]. In contrast, there are fewer techniques for innermost termination analysis and even lesser for outermost termination analysis. The recent trend is using transformation techniques and attempting to solve the transformed system instead.

In this thesis, our interest is in Thiemann's transformation [23] which transforms an outermost termination problem into an innermost termination problem.

Example 3. Consider the term rewrite system $\mathcal{R}$ consisting of the following rules:

$$\begin{aligned} 1: &\quad \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x \\ 2: &\quad \mathsf{g}(\mathsf{b}) \to \mathsf{f}(\mathsf{g}(\mathsf{b})) \end{aligned}$$

In fact, $\mathcal{R}$ has the outermost termination property. We prove it by using Thiemann's transformation. The transformation yields the following rewrite system $\mathcal{R}^T$ over the signature $\{\mathsf{b}^{(0)}, \mathsf{f}^{(1)}, \mathsf{g}^{(1)}, \mathsf{f}_1^{(1)}, \mathsf{g}_1^{(1)}, \underline{\mathsf{f}}^{(1)}, \underline{\mathsf{g}}^{(1)}, \triangledown^{(1)}, \vartriangle^{(1)}, \blacktriangledown_{\mathsf{f}}^{(1)}, \blacktriangledown_{\mathsf{g}}^{(1)}, \blacktriangle^{(1)}, \mathsf{top}^{(1)}\}$:

$$\begin{array}{llll} 1: & \triangledown(\mathsf{f}(x)) \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\ 3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) \to \mathsf{f}_1(\triangledown(x)) & 4: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \to \mathsf{g}_1(\triangledown(x)) \\ 5: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & 6: & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\ 7: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) \to \vartriangle(x) & 8: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) \to \vartriangle(x) \\ 9: & \mathsf{f}_1(\vartriangle(x)) \to \vartriangle(\mathsf{f}(x)) & 10: & \mathsf{g}_1(\vartriangle(x)) \to \vartriangle(\mathsf{g}(x)) \\ 11: & \mathsf{top}(\vartriangle(x)) \to \mathsf{top}(\triangledown(x)) & & \end{array}$$

For example, the outermost rewrite step of $\mathcal{R}$

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \to_{\mathcal{R}} \mathsf{b}$$

corresponds to the four innermost rewrite steps of $\mathcal{R}^T$:

$$\begin{aligned} \mathsf{top}(\triangledown(\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\blacktriangle(\mathsf{b}))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\vartriangle(\mathsf{b})) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\triangledown(\mathsf{b})) \end{aligned}$$

Since this correspondence generally holds, we can show outermost termination of the original system $\mathcal{R}$ by proving innermost termination of the transformed system $\mathcal{R}^T$. It is also known that the transformation is *complete*, meaning that $\mathcal{R}^T$ is innermost terminating if $\mathcal{R}$ is outermost terminating.

Now the remaining question is whether one can show innermost termination of such a transformed rewrite system. The above example reveals a major problem of the approach: This kind of transformations significantly increases the complexity of term structure in rewrite rules. Unfortunately, even state-of-the-art termination provers tend to fail since it cannot analyze the complex terms and rewriting structures. For example, AProVE and TTT2, the 1st and 2nd places on the termination competition in 2015, fail to prove innermost termination of the above system $\mathcal{R}^T$.

1.3 Approach

This research aims to establish techniques for showing innermost termination of systems resulting from Thiemann's transformation. There are various transformation techniques for termination with specific strategies [10, 5]. Most of them result in rewrite systems similar to those of Thiemann's transformation.

There are two major problems of transformed systems. The first problem is that one rewrite step becomes many rewrite steps as seen in Example 3. Termination proofs are usually established by detecting decreasing parameters. However, the intermediate steps obfuscate the decreasingness. The second problem originates from the nature of innermost rewriting. Majority of existing termination techniques directly or indirectly employ the notion of reduction order, which does not fit for innermost termination proofs when the system is non-terminating.

In order to address these problems, we develop new *transformation* techniques. Exploiting *type information*, we resolve the first problem about the complexity of term structure. There is a technique to introduce many-sorts to untyped rewrite systems. As proved in the main part of the thesis, all rewrite systems induced by Thiemann's transformation admit (proper) many-sorted signatures. Based on the sort information, we can perform *type-based reachability analysis* which can be integrated with various termination techniques, such as dependency graphs [15], usable rules [13, 24], and simple freezing [14].

For handling the second problem, we introduce a transformation technique dubbed *pattern separation*. This transformation fills in the gap between ordinary rewrite step and innermost rewrite step, in the latter of which lacks the closure under substitutions. By using the aforementioned type introduction technique, pattern separation can be further improved.

Here we illustrate these techniques, contributions of this thesis. We start with *type-based reachability analysis*.

Example 4 (continued from Example 3)**.** The next sort information can be attached to the transformed system $\mathcal{R}^T$.

$$\left\{ \begin{array}{rrrr} \triangledown : \alpha \to \beta & \triangle : \alpha \to \beta & \underline{\mathsf{f}} : \alpha \to \delta & \underline{\mathsf{g}} : \alpha \to \delta \\ \blacktriangledown_{\mathsf{f}} : \delta \to \beta & \blacktriangledown_{\mathsf{g}} : \delta \to \beta & \blacktriangle : \alpha \to \delta & \mathsf{b} : \alpha \quad\quad \\ \mathsf{f} : \alpha \to \alpha & \mathsf{g} : \alpha \to \alpha & \mathsf{f}_1 : \beta \to \beta & \mathsf{g}_1 : \beta \to \beta \\ \mathsf{top} : \beta \to \gamma & & & \end{array} \right\}$$

Here we suppose that with other termination methods we succeeded in eliminate rules 5 and 6 from $\mathcal{R}^T$. Terms of form $\mathsf{top}(\triangledown(s))$ no longer reaches $\mathsf{top}(\triangle(t))$ for any terms s and t. Our type-based reachability analysis can detect this unreachability in the following way: We interpret each term to a set of function symbols that may appear in reachable terms. This can be computed by using the rewrite system $\|\mathcal{R}^T\|$ on sets:

$$\begin{array}{lrl} 1: & \{\triangledown\} \rightsquigarrow \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} & 2: \quad \{\triangledown\} \rightsquigarrow \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} \\ 3: & \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} \rightsquigarrow \{\mathsf{f}_1, \triangledown\} & 4: \quad \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} \rightsquigarrow \{\mathsf{g}_1, \triangledown\} \\ \\ 7: & \{\blacktriangledown_{\mathsf{f}}, \blacktriangle\} \rightsquigarrow \{\triangle\} & 8: \quad \{\blacktriangledown_{\mathsf{g}}, \blacktriangle\} \rightsquigarrow \{\triangle\} \\ 9: & \{\mathsf{f}_1, \triangle\} \rightsquigarrow \{\triangle\} & 10: \quad \{\mathsf{g}_1, \triangle\} \rightsquigarrow \{\triangle\} \\ 11: & \{\mathsf{top}, \triangle\} \rightsquigarrow \{\mathsf{top}, \triangledown\} & \end{array}$$

Because our terms are sorted, the interpretation of $\mathsf{top}(\triangledown(s))$, say A, does not contain the symbol $\triangle$, and moreover the set A cannot reach a set containing $\triangle$ by using $\|\mathcal{R}^T\|$. This is sufficient to conclude the announced unreachability. The information of unreachability is used for the computation of the *dependency graph* [15]. Although we omit its explanation here, this technique now shows innermost termination of $\mathcal{R}^T$. We provide details of type-based reachability analysis in Chapter 3.

Our running example can also be handled by pattern separation, which is another our contribution.

Example 5 (continued from Example 3)**.** Applying the optimization version of pattern separation on the TRS $\mathcal{R}^T$, The following TRS $\mathcal{S}_{op}(\mathcal{R}^T)$ is obtained:

$$\begin{array}{lrcllrcl}
1: & \triangledown(\mathsf{f}(x)) & \to & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) & \to & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3.1: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}(x))) & \to & \mathsf{f}_1(\triangledown(\mathsf{g}(x))) & 3.2: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{b})) & \to & \mathsf{f}_1(\triangledown(\mathsf{b})) \\
3.3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{h}(x))) & \to & \mathsf{f}_1(\triangledown(\mathsf{h}(x))) & 3.4: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{c})) & \to & \mathsf{f}_1(\triangledown(\mathsf{c})) \\
3.5: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}(x)))) & \to & \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}(x)))) & 3.6: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{b}))) & \to & \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{b}))) \\
3.7: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{h}(x)))) & \to & \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{h}(x)))) & 3.8: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{c}))) & \to & \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{c}))) \\
4.1: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}(x))) & \to & \mathsf{g}_1(\triangledown(\mathsf{f}(x))) & 4.2: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}(x))) & \to & \mathsf{g}_1(\triangledown(\mathsf{g}(x))) \\
4.3: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{h}(x))) & \to & \mathsf{g}_1(\triangledown(\mathsf{h}(x))) & 4.4: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{c})) & \to & \mathsf{g}_1(\triangledown(\mathsf{c})) \\
5: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) & \to & \blacktriangle(x) & 6: & \underline{\mathsf{g}}(\mathsf{b}) & \to & \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
7: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) & \to & \triangle(x) & 8: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) & \to & \triangle(x) \\
9: & \mathsf{f}_1(\triangle(x)) & \to & \triangle(\mathsf{f}(x)) & 10: & \mathsf{g}_1(\triangle(x)) & \to & \triangle(\mathsf{g}(x)) \\
11: & \mathsf{top}(\triangle(x)) & \to & \mathsf{top}(\triangledown(x)) & & & &
\end{array}$$

Pattern separation replaces rule 3 of $\mathcal{R}^T$ by its instantiated versions rules 3.1 – 3.8. The same idea applies to rules 4.

Innermost termination of the resulting system can be shown by existing termination provers (such as AProVE and TTT2). We want to stress that the character of innermost rewrite step has been changed by the separation: As shown in the thesis, the final system even has the *termination* property.

As shown by the experiments, in Chapter 6 our *pattern separation technique* is helpful for proving innermost termination of transformed systems. However, the condition to obtain a transformed system as an overlay system remains unclear, and this will be a future work.

1.4 Overview and Contributions

The thesis is organized as follows: In Chapter 2 we introduce background knowledge of term rewrite system in a formal way which includes definitions, notions, and notations. We propose the discovered property of a transformed system resulting from Thiemann's algorithm along with *type-based reachability analysis* in Chapter 3. Type-based reachability analysis is a reachability analysis based on type information of the system. In Chapter 4 we introduce *pattern separation* technique in two version. First with the basic version and then following with the optimized version which is exploited type information. The techniques have been implemented and tested. Chapter 5 reports experimental results where we compare transformation results of pattern separation using testing result on existing termination tools. Finally, we conclude the thesis and discuss related work and future work in Chapter 6.

Here is the list of our contributions:

- analysis of types for Thiemann's transformation,
- type-based reachability analysis,

- a pattern separation technique, and
- the optimized version of pattern separation.

Chapter 2

Term Rewriting

This chapter aims at describing background knowledge for this thesis. Term rewriting is a computational model which is based on directed equations on terms. Recalling basic notions for terms and relations, we define term rewrite systems and related computational properties. For detailed surveys of term rewriting, we refer to [3, 21, 18].

2.1 Term Rewrite Systems

We start with the definition of terms, which are built from function symbols and variables. Every function symbol f is associated with a natural number n, called *arity*, and we may write $f^{(n)}$ to indicate the arity of f. The arity stands for the number of arguments of the function.

Definition 1. Let $\mathcal{F}$ be a set of function symbols and let $\mathcal{V}$ be a countably infinite set of variables. The set $\mathcal{T}(\mathcal{F},\mathcal{V})$ of *terms* is inductively defined as follows:

- If $x \in \mathcal{V}$ then $x \in \mathcal{T}(\mathcal{F},\mathcal{V})$.
- If $f^{(n)} \in \mathcal{F}$ and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F},\mathcal{V})$ then $f(t_1, \ldots, t_n) \in \mathcal{T}(\mathcal{F},\mathcal{V})$.

The set $\mathcal{V}\mathsf{ar}(t)$ of variables in a term t is defined as follows:

$$\mathcal{V}\mathsf{ar}(t) = \begin{cases} \{t\} & \text{if } t \in \mathcal{V} \\ \displaystyle\bigcup_{i=1}^{n} \mathcal{V}\mathsf{ar}(t_i) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

A term t is called *ground* if $\mathcal{V}\mathsf{ar}(t) = \varnothing$. The set of all ground terms is denoted by $\mathcal{T}(\mathcal{F})$.

Definition 2. A term is called *linear* if no variable occurs more than once in the term.

Example 6. Let $\mathcal{F} = \{+^{(2)}, \mathsf{s}^{(1)}, \mathsf{0}^{(0)}\}$ and $x, y \in \mathcal{V}$. While the term $\mathsf{s}(0) + 0$ is ground, the term $t = \mathsf{s}((x + 0) + \mathsf{s}(\mathsf{s}(0)))$ is not as $\mathcal{V}\mathsf{ar}(t) = \{x\}$.

Definition 3. The root symbol of a term t is defined as follows:

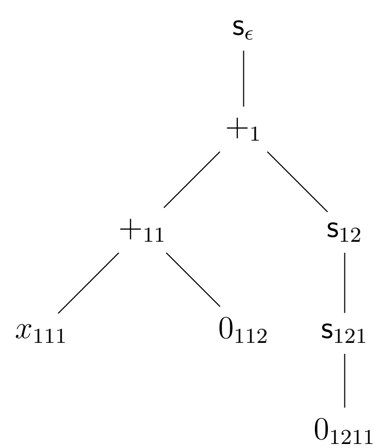
$$\mathsf{root}(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ f & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Definition 4. A *position* is a finite sequence of positive integers. The set of positions in a term t inductively defined as follows:

$$\mathcal{P}\mathsf{os}(t) = \begin{cases} \{\epsilon\} & \text{if } t \in \mathcal{V} \\ \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \text{ and } p \in \mathcal{P}\mathsf{os}(t_i)\} & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

The empty sequence, denoted by ϵ, represents the root position. We define useful notions for positions: Let p and q be positions. If $pr = q$ for some position r then we say that p is *above* q, or q is *below* p and write $p \leq q$. If $p \leq q$ and $p \neq q$ then we write $p < q$.

Example 7. The term $t = \mathsf{s}((x + 0) + \mathsf{s}(\mathsf{s}(0)))$ can be represented as a tree structure. In the figure, each node is a function symbol, and subscriptions indicate the position.



Definition 5. A *subterm* of a term t at position $p \in \mathcal{P}\mathsf{os}(t)$ is denoted by $t|_p$, which is formally defined as follows:

$$s|_p = \begin{cases} s & \text{if } p = \epsilon \\ t_i|_q & \text{if } p = iq \text{ and } s = f(t_1, \ldots, t_i, \ldots, t_n) \end{cases}$$

We say that t is a subterm of s if there exists $p \in \mathcal{P}\mathsf{os}(s)$ such that $s|_p = t$. We denote this relation of t and s by $t \trianglelefteq s$. Moreover, we call t a proper subterm ($t \triangleleft s$) if $p \neq \epsilon$.

Example 8. The proper subterms of term $s = \mathsf{s}((x + 0) + \mathsf{s}(\mathsf{s}(0)))$ are $s|_{111} = x$, $s|_{112} = s|_{1211} = 0$, $s|_{11} = x + 0$, $s|_{12} = \mathsf{s}(\mathsf{s}(0))$, $s|_{121} = \mathsf{s}(0)$, and $s|_1 = (x + 0) + \mathsf{s}(\mathsf{s}(0))$. Note that subterms of s include s itself.

Definition 6. Let s, t be terms and $p \in \mathcal{P}\mathsf{os}(s)$. The term which is obtained from s by replacing the subterm of s at position p with term t is denoted by $s[t]_p$ and formally defined as follows:

$$s[t]_p = \begin{cases} t & \text{if } p = \epsilon \\ f(s_1, \ldots, s_i[t]_q, \ldots, s_n) & \text{if } s = f(s_1, \ldots, s_i, \ldots, s_n) \text{ and } p = iq \end{cases}$$

For convenience, contexts of a term are defined in the same way as terms containing a special fresh constant symbol $\Box$, named *hole*.

Definition 7. A *context* is a term in $\mathcal{T}(\mathcal{F} \uplus \{\Box\}, \mathcal{V})$ such that the hole symbol occur in the term exactly once. The term which is obtained from replacing the hole symbol in a context C with a term t is denoted by $C[t]$ and formally defined as follows:

$$C[t] = \begin{cases} t & \text{if } C = \Box \\ f(t_1, \ldots, C'[t], \ldots, t_n) & \text{if } C = f(t_1, \ldots, C', \ldots, t_n) \end{cases}$$

where C' is a context.

The difference between a nullary function symbol (i.e. constant) and a variable is that the variable can be substituted with a term.

Definition 8. Let $\mathcal{V}$ be a countably infinite set. A substitution is a mapping from $\mathcal{V}$ to $\mathcal{T}(\mathcal{F}, \mathcal{V})$, denoted as a set of variable bindings $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. A term t and a substitution σ, the term $t\sigma$ is an instance of t, denoted by $t \geq t\sigma$. The term $t\sigma$ is defined as follows:

$$t\sigma = \begin{cases} \sigma(t) & \text{if } t \in \mathcal{V} \\ f(t_1\sigma, \ldots, t_n\sigma) & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

A substitution σ is *more general* than a substitution σ' whenever $\sigma' = \delta\sigma$ for some substitution δ. The relation is denoted by $\sigma \leq \sigma'$.

Equation solving in the area of term algebra is centered around *unification*. In term algebra, satisfiability is called unifiability and solutions are called unifiers.

Definition 9. Let s and t be terms. We say s and t are *unifiable* if there exists a substitution σ such that $s\sigma = t\sigma$. This substitution is called *unifier* of s and t. The *most general unifier (mgu)* is a unifier σ such that $\sigma \leq \sigma'$ for all unifiers σ' of them.

Example 9. The terms $(x + 0) + \mathsf{s}(\mathsf{s}(y))$ and $(\mathsf{s}(0) + 0) + \mathsf{s}(z)$ are unifiable. Substitution $\{x \mapsto \mathsf{s}(0), z \mapsto \mathsf{s}(0), y \mapsto 0\}$ is one of their unifiers, while the most general unifier is $\{x \mapsto \mathsf{s}(0), z \mapsto \mathsf{s}(y)\}$.

Now we are ready to define term rewrite systems and rewrite steps.

Definition 10. A pair (ℓ, r) of terms with $\ell \notin \mathcal{V}$ and $\mathcal{V}\mathsf{ar}(r) \subseteq \mathcal{V}\mathsf{ar}(\ell)$ is called a *rewrite rule*. Such a pair is often denoted by $\ell \to r$. A *term rewrite system (TRS)* is a set of rewrite rules.

Example 10. The following set is an instance of TRSs.

$$\mathcal{R} = \left\{ \begin{array}{c} \mathsf{s}(x) + y \to \mathsf{s}(x+y) \\ \mathsf{0} + x \to x \end{array} \right\}$$

Definition 11. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. A function symbol $f \in F$ is called a defined symbol if there exists a rewrite rule $\ell \to r \in \mathcal{R}$ such that $f = \mathsf{root}(\ell)$. The subset of $\mathcal{F}$ consisting of all defined symbols is denoted by $\mathcal{F}_{\mathcal{D}}$. The subset of $\mathcal{F}$ consisting of all non-defined symbols is denoted by $\mathcal{F}_{\mathcal{C}}$.

Definition 12. Let $\mathcal{R}$ be a TRS. We define function $lhs(\mathcal{R})$ and $rhs(\mathcal{R})$ as follow:

$$lhs(\mathcal{R}) = \{\ell \mid \ell \to r \in \mathcal{R}\} \qquad rhs(\mathcal{R}) = \{r \mid \ell \to r \in \mathcal{R}\}$$

We call a TRS $\mathcal{R}$ is left-linear if every term in $lhs(\mathcal{R})$ are linear. A right-linear TRS is defined in the same way.

Definition 13. Let $\mathcal{F}$ be a signature and let $\mathcal{V}$ be a set of variables. The rewrite relation $\to_{\mathcal{R}}$ on terms is defined as the smallest set such that if $\ell \to r \in \mathcal{R}$ then $C[\ell\sigma] \to_{\mathcal{R}} C[r\sigma]$ for all contexts C and substitutions σ.

Example 11. Consider the TRS $\mathcal{R}$ in Example 10. The term $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))+\mathsf{s}(x))$ is rewritten in the following way:

$$\begin{aligned} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})) + \mathsf{s}(x)) &\to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}) + \mathsf{s}(x))) \\ &\to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0} + \mathsf{s}(x)))) \\ &\to_{\mathcal{R}} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x)))) \end{aligned}$$

Next, we will introduce some basic symbols which are often combined with rewrite relations. Here we omit $\mathcal{R}$ and use only $\to$ for rewrite relation.

Definition 14. Given two relations A and B, their composition $A \circ B$ is the set of pairs (x, z) which satisfied $(x, y) \in A$ and $(y, z) \in B$ for some y. The following notions are defined on composition of rewrite relation with itself.

$$\begin{array}{lcll} \to^0 & = & \{(x,x) \mid x \in \mathcal{T}(\mathcal{F},\mathcal{V})\} & \text{identity relation} \\ \to^{i+1} & = & \to^i \circ \to & \text{i+1 step relation} \\ \to^+ & = & \bigcup_{i>0} \to^i & \text{transitive closure} \\ \to^* & = & \to^+ \cup \to^0 & \text{reflexive transitive closure} \\ \leftarrow & = & \{(y,x) \mid x \to y\} & \text{inverse relation} \end{array}$$

The following terminologies are used for describing relations between terms.

- x is *reducible or a redex* if there exists y such that $x \to y$.
- x is *in a normal form* if x is not reducible.
- y is *a normal form* of x if $x \to^* y$ and y is in normal form.

Example 12. Consider TRS $\mathcal{R}$ in the Example 10.

- $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})) + \mathsf{s}(x))$ is reducible
- $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x))))$ is in normal form
- $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(x))))$ is a normal form of $\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})) + \mathsf{s}(x))$

Definition 15. Given a TRS $\mathcal{R}$ and two terms s and t, we say that s reaches t if there are substitutions σ and τ such that $s\sigma \to^*_{\mathcal{R}} t\tau$. We denote it by $s \hookrightarrow_{\mathcal{R}} t$.

Definition 16. A term s *overlaps* with a term t if s is unifiable with a non-variable subterm of t. A TRS is called an *overlay system* if overlaps between rules occur only at the root position.

2.2 Many-sorted Term Rewrite Systems

In this section, we consider terms structure equipped with sort (i.e. type) information. We restrict sort information to every function symbol in the following way, each function has a sort and expects sort of each argument of the function. From here on, we denote $\mathcal{S}$ is a set of sort symbols.

Definition 17. An $\mathcal{S}$-sorted signature $\mathcal{F}$ is consisted of the set of sort symbols $\mathcal{S}$, the signature $\mathcal{F}$ and the sort of function symbol and expected sorts for its arguments written as a set of:

$$f : \alpha_1 \times \cdots \times \alpha_n \to \beta$$

where $f \in \mathcal{F}$, $\alpha_1, \ldots, \alpha_n, \beta \in \mathcal{S}$, and n is a number of the arguments in f. We say that f has sort β

Example 13. Let $\mathcal{S} = \{\alpha, \beta, \gamma\}$. The next set forms a sorted signature:

$$\mathcal{F} = \left\{ \begin{array}{ll} \mathsf{0} : \alpha & \mathsf{1} : \alpha \\ \mathsf{g} : \alpha \to \beta & \mathsf{h} : \beta \to \beta \\ \mathsf{f} : \beta \to \gamma & \end{array} \right\}$$

Definition 18. We define function $\mathsf{st} : \mathcal{F} \to \mathcal{S}$ and $\mathsf{ar} : \mathcal{F} \times \mathbb{N} \to \mathcal{S}$ as follow:

$$\mathsf{st}(f) = \beta \qquad \mathsf{ar}(f, i) = \alpha_i$$

where $f : \alpha_1 \times \cdots \times \alpha_n \to \beta$ and $i \in \{1, \ldots, n\}$.

Definition 19. The sort of a term is defined as the sort of root symbol of the term, so we have $\mathsf{sort}(t) = \mathsf{st}(\mathsf{root}(t))$, and we may write t^α to indicate that the sort of t is α. We inductively define the set of *well-sorted* terms $\mathcal{WT}(\mathcal{F}, \mathcal{V})$ as follows:

- If $x \in \mathcal{V}$ then $x \in \mathcal{WT}(\mathcal{F}, \mathcal{V})$.

- If $f^{(n)} \in \mathcal{F}$, $t_1, \ldots, t_n \in \mathcal{WT}(\mathcal{F}, \mathcal{V})$ and $\mathsf{ar}(f, i) = \mathsf{sort}(t_i)$ for all $1 \leq i \leq n$ such that then $f(t_1, \ldots, t_n) \in \mathcal{WT}(\mathcal{F}, \mathcal{V})$.

Definition 20. We say that $\mathcal{S}$-sorted signature $\mathcal{F}$ is *compatible* with TRS $\mathcal{R}$ if the following restrictions hold for all $\ell \to r \in \mathcal{R}$

- $\ell \in \mathcal{WT}(\mathcal{F}, \mathcal{V})$ and $r \in \mathcal{WT}(\mathcal{F}, \mathcal{V})$
- $\mathsf{sort}(\ell) = \mathsf{sort}(r)$

Example 14. Consider the TRS $\mathcal{R}$ consisting of the following rule:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ \mathsf{g}(\mathsf{0}) \to \mathsf{h}(\mathsf{g}(\mathsf{1})) \end{array} \right\}$$

The following $\mathcal{S}$-sorted signature are compatible with TRS $\mathcal{R}$

$$\left\{ \begin{array}{ll} \mathsf{0} : \alpha & \mathsf{1} : \alpha \\ \mathsf{g} : \alpha \to \beta & \mathsf{h} : \beta \to \beta \\ \mathsf{f} : \beta \to \gamma & \end{array} \right\} \qquad \left\{ \begin{array}{ll} \mathsf{0} : \alpha & \mathsf{1} : \alpha \\ \mathsf{g} : \alpha \to \alpha & \mathsf{h} : \alpha \to \alpha \\ \mathsf{f} : \alpha \to \alpha & \end{array} \right\}$$

Unsorted TRSs are regarded as one-sorted TRSs. The technique to attach sort information to a TRS is called *type introduction* [27], resulting in the left set. This process can be done by initializing all distinct sorts for every function symbols and find a compatible many-sorted signature while forcing minimal sorts to be equal. However, TRS without sort information can be considered as a single sorted-signature, which is shown in the right set.

2.3 Termination

Termination is an important property of term rewrite systems. Unfortunately, it is known to be undecidable. However, there are many researches on automated termination analysis such as dependency pairs [2], matrix interpretations [6] and polynomial interpretations [8].

In this section, we explain the reduction order, which is an important tool for proving termination of rewrite systems. To prove termination of a rewrite system, we find an appropriate reduction order for the system. Thus, various reduction orders have been invented and widely used for proving termination. First, we start with the definition of termination of rewrite systems.

Definition 21. A relation $\to$ is terminating if there is no infinite sequence.

$$a_0 \to a_1 \to a_2 \to \cdots$$

Definition 22. A TRS $\mathcal{R}$ is terminating if $\to_{\mathcal{R}}$ is terminating.

It is easy to imagine how to prove that a TRS is non-terminating. Simply, we try to find a witness of infinite rewrite sequence.

Example 15. Consider the TRS $\mathcal{R}$:

$$\left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ \mathsf{g}(\mathsf{0}) \to \mathsf{h}(\mathsf{g}(\mathsf{1})) \end{array} \right\}$$

There exists the following infinite rewrite sequence.

$$\mathsf{f}(\mathsf{g}(x)) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\mathsf{0})) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\mathsf{0})) \cdots$$

Definition 23. An order $>$ is said to be well-founded order over a set A if and only if for all nonempty subsets S of A there exists $m \in S$ such that $m \not> s$ for all $s \in S$.

The basic idea to prove termination of TRS $\mathcal{R}$ is to find a well-founded order $>$ on terms that is compatible with the rewrite relation $\to_{\mathcal{R}}$ (i.e. if $s \to_{\mathcal{R}} t$ then $s > t$ for all terms s, t). The motivation of reduction order definition is to find an order that capable of proving termination while checking only $\ell > r$ for finitely many rules $\ell \to r \in \mathcal{R}$ rather than $s > t$ for infinitely many pairs s, t with $s \to_{\mathcal{R}} t$. However, the order must satisfy additional properties.

Definition 24. A reduction order is a well-founded order $>$ that

- is *closed under contexts*:

  for every term s, t and context C, if $s > t$ then $C[s] > C[t]$, and

- is *closed under substitutions*:

  for every term s, t and substitution σ, if $s > t$ then $s\sigma > t\sigma$.

Theorem 1. *A TRS $\mathcal{R}$ is terminating if and only if there exists a reduction order $>$ such that $\ell > r$ for all $\ell \to r \in \mathcal{R}$.*

In the next example, we introduce a simple reduction order based on size of terms and a number of variables occurrences.

Example 16. The strict order $>$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$ defined

$$s > t \text{ if and only if } |s| > |t| \text{ and, } |s|_x \geq |t|_x \text{ for all variables x}$$

forms a reduction order. The following TRS $\mathcal{R}$ can be shown terminating by this reduction order

$$\mathcal{R} = \{\mathsf{0} + x \to x\}$$

Since $|\mathsf{0} + x| = 3 > 1 = |x|$ and $|\mathsf{0} + x|_x = 1 \geq 1 = |x|_x$. Therefore, Theorem 1 applies.

Here, we introduce dependency pairs [2] and dependency graphs [15].

Definition 25. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$ and $\mathcal{F}^\sharp = \mathcal{F} \uplus \{f^{\sharp(n)} | f^{(n)} \in \mathcal{F}_\mathcal{D}\}$. The set of dependency pairs of $\mathcal{R}$ is denoted by $\mathsf{DP}(\mathcal{R})$ and formally defined as follows:

$$\mathsf{DP}(\mathcal{R}) = \{\ell^\sharp \to u^\sharp \mid u \trianglelefteq r \text{ and } \mathsf{root}(u) \in \mathcal{F}_\mathcal{D} \text{ and } u \not\triangleleft \ell \text{ and } \ell \to r \in \mathcal{R}\}$$

where, if $u = f(u_1, \ldots, u_n)$ then $u^\sharp = f^\sharp(u_1, \ldots, u_n)$.

Definition 26. Let $\mathcal{R}$ be a TRS. A dependency graphs $\mathsf{DG}(\mathcal{R})$ is the directed graph whose vertexes set is $\mathsf{DP}(\mathcal{R})$ and edges set E is given by the following condition:

$$(s \to t, u \to v) \in E \text{ if and only if } t \hookrightarrow_\mathcal{R} u$$

Theorem 2. *A TRS $\mathcal{R}$ is terminating if $\mathsf{DG}(\mathcal{R})$ contains no cycle*[1].

2.4 Innermost and Outermost Strategies

In Example 15, we saw that the following TRS is non-terminating.

$$\left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{g}(0) \to \mathsf{h}(\mathsf{g}(1)) \end{array} \right\}$$

However, this TRS is innermost terminating. In most of the cases, a system requires only termination under some strategies. We do not need the full termination property for the system.

The innermost strategy is one of the various rewriting strategies. The idea of innermost rewriting strategy is at the rewritable position there is no other rewritable position below. Termination of rewrite systems may depend on a rewrite strategy. Unfortunately, the undecidability still remains, even termination under rewriting strategies.

Example 17. Consider the TRS consisting the following rules:

$$\begin{array}{c} \mathsf{f}(\mathsf{a}) \to \mathsf{f}(\mathsf{a}) \\ \mathsf{a} \to \mathsf{b} \end{array}$$

This is a simple system that innermost terminating but not terminating in general.

Definition 27. Let $\mathcal{R}$ be a TRS. The *innermost step* $\xrightarrow{i}_\mathcal{R}$ is defined as follows: $s \xrightarrow{i}_\mathcal{R} t$ if there is a rule $\ell \to r \in \mathcal{R}$, a context C, and a substitution σ such that $s = C[\ell\sigma]$, $t = C[r\sigma]$ and all proper subterms of $\ell\sigma$ are in normal form.

Definition 28. A TRS $\mathcal{R}$ is *innermost terminating* if $\xrightarrow{i}_\mathcal{R}$ is terminating.

There is an important theorem on a relation between terminating and innermost terminating that inspire our work. The theorem was discovered by Masahiko Sakai [20].

[1] A non-empty set $(v_1, \ldots, v_n)$ of vertexes is a *cycle* if $v_1 E v_2 E \cdot E v_n E v_1$

Theorem 3. *A Right-Linear Overlay system is terminating if and only if it is innermost terminating.*

The idea of outermost rewriting strategy is at the rewritable position there is no other rewritable position above.

Example 18. Consider the TRS consisting the following rules:

$$\begin{aligned} \mathsf{a} &\to \mathsf{f}(\mathsf{a}) \\ \mathsf{f}(\mathsf{a}) &\to \mathsf{b} \end{aligned}$$

This is a simple system that outermost terminating but not terminating in general.

Definition 29. Let $\mathcal{R}$ be a TRS. The *outermost step* $\xrightarrow{o}_{\mathcal{R}}$ is defined as follows: $s \xrightarrow{o}_{\mathcal{R}} t$ if there are a rule $\ell \to r \in \mathcal{R}$, a context C, a position p, and a substitution σ such that $s = C[\ell\sigma]$, $t = C[r\sigma]$, $s|_p = \ell\sigma$ and for all $q \in \mathcal{P}\mathsf{os}(s)$ such that if $q < p$ then $s|_q$ is not a redex.

Definition 30. A TRS $\mathcal{R}$ is *outermost terminating* if $\xrightarrow{o}_{\mathcal{R}}$ is terminating.

However, only a few techniques are available to analyze outermost termination directly. One way to prove outermost termination is to transform a system to prove it as innermost termination. The technique was proposed by Thiemann in 2009 [23].

2.5 Thiemann's Transformation

This work is concerned with Thiemann's transformation because of two reasons. First, transformed systems are highly different from hand-crafted systems. Systems resulting from transformations are consist of heavily mutual recursion and deeply nested functions. Second, transformed systems have a nice property when applying type information. Here, we start with Thiemann's transformation.

Definition 31. Let $\mathcal{R}$ be a TRS. The transformed TRS $\mathcal{R}^{\mathsf{T}}$ consists of following rules:

$$\begin{array}{lcl} \triangledown(d(x_1,\ldots,x_n)) & \to & \blacktriangledown_d(\underline{d}(x_1,\ldots,x_n)) \\ \blacktriangledown_f(\underline{f}(x_1,\ldots,x_m)) & \to & f_i(x_1,\ldots,\triangledown(x_i),\ldots,x_m) \\ \underline{g}(\ell_1,\ldots,\ell_n) & \to & \blacktriangle(r) \\ \blacktriangledown_d(\blacktriangle(x)) & \to & \vartriangle(x) \\ h_i(x_1,\ldots,\vartriangle(x_i)\ldots,x_m) & \to & \vartriangle(h(x_1,\ldots,x_m)) \\ \mathsf{top}(\vartriangle(x)) & \to & \mathsf{top}(\triangledown(x)) \\ \triangledown(c(x_1,\ldots,x_m)) & \to & c_i(x_1,\ldots,\triangledown(x_i),\ldots,x_m) \\ \triangledown(a) & \to & \vartriangle(r) \end{array}$$

where $d^{(n)}, f^{(m)} \in \mathcal{F}_{\mathcal{D}},\ c^{(m)} \in \mathcal{F}_{\mathcal{C}},\ h^{(m)} \in \mathcal{F},\ a^{(0)} \to r,\ g(\ell_1,\ldots,\ell_n) \to r \in \mathcal{R}$, $n > 0$ and $i \in \{1\ldots m\}$.

Theorem 4. *TRS $\mathcal{R}$ is outermost terminating if and only if $\mathcal{R}^T$ is innermost terminating.*

We illustrate the transformation with an example.

Example 19. In the running example, we consider the TRS $\mathcal{R}$ consisting of the following rule:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x \\ \mathsf{g}(\mathsf{b}) \to \mathsf{f}(\mathsf{g}(\mathsf{b})) \end{array} \right\}$$

The transformed TRS $\mathcal{R}^T$ results as:

$$\mathcal{R}^T = \left\{ \begin{array}{rr} \triangledown(\mathsf{f}(x)) \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\ \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) \to \mathsf{f}_1(\triangledown(x)) & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \to \mathsf{g}_1(\triangledown(x)) \\ \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\ \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) \to \triangle(x) & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) \to \triangle(x) \\ \mathsf{f}_1(\triangle(x)) \to \triangle(\mathsf{f}(x)) & \mathsf{g}_1(\triangle(x)) \to \triangle(\mathsf{g}(x)) \\ \mathsf{top}(\triangle(x)) \to \mathsf{top}(\triangledown(x)) & \end{array} \right\}$$

Next, we provide illustration of rewrite sequences of $\mathcal{R}$ and $\mathcal{R}^T$. The outermost rewrite step of $\mathcal{R}$

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \to_{\mathcal{R}} \mathsf{b}$$

corresponds to the four innermost rewrite steps of $\mathcal{R}^T$:

$$\begin{aligned} \mathsf{top}(\triangledown(\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\blacktriangle(\mathsf{b}))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\triangle(\mathsf{b})) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\triangledown(\mathsf{b})) \end{aligned}$$

Since this correspondence generally holds, we can show outermost termination of the original system $\mathcal{R}$ by proving innermost termination of the transformed system $\mathcal{R}^T$. It is also known that the transformation is *complete*, meaning that if $\mathcal{R}^T$ is innermost terminating then $\mathcal{R}$ is outermost terminating.

It is obvious that the TRS $\mathcal{R}$ in Example 19 is outermost terminating. However, innermost termination of the transformed system cannot be proved with AProVE [9] or TTT2 [16]. In this thesis, we will propose pattern separation technique which further transforms the system and makes the innermost termination provable.

Chapter 3

Type-based Reachability Analysis

In this chapter, we introduce *type-based reachability analysis* which exploits sort information to analyze systems. The idea of this technique is to map a term into a set and check subset relation for analyzing reachability. First of all, we will explain the motivation and follow with related definitions.

3.1 Typing Transformed Systems

Considering Thiemanns transformation [23], if one applies type introduction [27] to a transformed system, original function symbols in the transformed system can be distinguished which inspire the technique we propose in this thesis. Given a unsorted TRS, type introduction tries to find a typing in such a way that all rules are well-typed. This can be done systematically in a way similar to unification; in this way, a sorted TRS is obtained.

Theorem 5. *Suppose that $\mathcal{R}$ is a TRS over a signature $\mathcal{F}$ and the result of type introduction on $\mathcal{R}^T$ is $\mathcal{F}'$. Then the following equality holds.*

$$\{\mathsf{st}(f) \mid f \in \mathcal{F}\} \cap \{\mathsf{st}(g) \mid g \in \mathcal{F}' \setminus \mathcal{F}\} = \varnothing$$

Proof. Since type induction yields the most general result, if there exists a sorted signature which satisfies Theorem 5 then the sorted signature induce by type introduction also satisfies the equality. Let $\mathcal{R}$ be an arbitrary TRS. We construct a sorted signature which is compatible with the transformed system and satisfied the claim by attaching sorts into the definition of transformation.

$$\begin{array}{lcl}
\triangledown(d(x_1^\alpha, \ldots, x_n^\alpha)^\alpha)^\beta & \to & \blacktriangledown_d(\underline{d}(x_1^\alpha, \ldots, x_n^\alpha)^\delta)^\beta \\
\blacktriangledown_f(\underline{f}(x_1^\alpha, \ldots, x_m^\alpha)^\delta)^\beta & \to & f_i(x_1^\alpha, \ldots, \triangledown(x_i^\alpha)^\beta, \ldots, x_m^\alpha)^\beta \\
\underline{g}(\ell_1^\alpha, \ldots, \ell_n^\alpha)^\delta & \to & \blacktriangle(r^\alpha)^\delta \\
\blacktriangledown_d(\blacktriangle(x^\alpha)^\delta)^\beta & \to & \vartriangle(x^\alpha)^\beta \\
h_i(x_1^\alpha, \ldots, \vartriangle(x_i^\alpha)^\beta \ldots, x_m^\alpha)^\beta & \to & \vartriangle(h(x_1^\alpha, \ldots, x_m^\alpha)^\alpha)^\beta \\
\mathsf{top}(\vartriangle(x^\alpha)^\beta)^\gamma & \to & \mathsf{top}(\triangledown(x^\alpha)^\beta)^\gamma \\
\triangledown(c(x_1^\alpha, \ldots, x_m^\alpha)^\alpha)^\beta & \to & c_i(x_1^\alpha, \ldots, \triangledown(x_i^\alpha)^\beta, \ldots, x_m^\alpha)^\beta \\
\triangledown(a^\alpha)^\beta & \to & \vartriangle(r^\alpha)^\beta
\end{array}$$

where $d^{(n)}, f^{(m)} \in \mathcal{F}_\mathcal{D},\ c^{(m)} \in \mathcal{F}_\mathcal{C},\ h^{(m)} \in \mathcal{F},\ a^{(0)} \to r,\ g(\ell_1, \ldots, \ell_n) \to r \in \mathcal{R},\ n > 0$ and $i \in \{1 \ldots m\}$. We can guarantee that $\ell_1, \ldots, \ell_n$ and r are well-sorted terms because all functions from the original system are of type α. From the construction above, the set of all sorts of new symbols is $\{\beta, \gamma, \delta\}$ and the set of sorts of original symbols is $\{\alpha\}$ are disjoint. $\square$

Example 20. Let signature $\mathcal{F} = \{\mathsf{f}, \mathsf{g}, \mathsf{b}\}$, and let TRS $\mathcal{R}$ be the following system.

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x \\ \mathsf{g}(\mathsf{b}) \to \mathsf{f}(\mathsf{g}(\mathsf{b})) \end{array} \right\}$$

The transformed system $\mathcal{R}^T$ results as

$$\left\{ \begin{array}{rr} \triangledown(\mathsf{f}(x)) \to \blacktriangledown_\mathsf{f}(\underline{\mathsf{f}}(x)) & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_\mathsf{g}(\underline{\mathsf{g}}(x)) \\ \blacktriangledown_\mathsf{f}(\underline{\mathsf{f}}(x)) \to \mathsf{f}_1(\triangledown(x)) & \blacktriangledown_\mathsf{g}(\underline{\mathsf{g}}(x)) \to \mathsf{g}_1(\triangledown(x)) \\ \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\ \blacktriangledown_\mathsf{f}(\blacktriangle(x)) \to \triangle(x) & \blacktriangledown_\mathsf{g}(\blacktriangle(x)) \to \triangle(x) \\ \mathsf{f}_1(\triangle(x)) \to \triangle(\mathsf{f}(x)) & \mathsf{g}_1(\triangle(x)) \to \triangle(\mathsf{g}(x)) \\ \mathsf{top}(\triangle(x)) \to \mathsf{top}(\triangledown(x)) & \end{array} \right\}$$

The signature of $\mathcal{R}^T$ is $\mathcal{F}' = \{\triangledown, \blacktriangledown_\mathsf{f}, \blacktriangledown_\mathsf{g}, \blacktriangledown_\mathsf{f}, \blacktriangledown_\mathsf{g}, \mathsf{f}_1, \mathsf{g}_1, \mathsf{top}, \mathsf{f}, \mathsf{g}, \mathsf{b}\}$. The following sort information is obtained from applying type introduction on $\mathcal{R}^T$

$$\left\{ \begin{array}{rl} \triangledown : \alpha \to \beta & \triangle : \alpha \to \beta \\ \underline{\mathsf{f}} : \alpha \to \delta & \underline{\mathsf{g}} : \alpha \to \delta \\ \blacktriangledown_\mathsf{f} : \delta \to \beta & \blacktriangledown_\mathsf{g} : \delta \to \beta \\ \blacktriangle : \alpha \to \delta & \mathsf{b} : \alpha \\ \mathsf{f} : \alpha \to \alpha & \mathsf{g} : \alpha \to \alpha \\ \mathsf{f}_1 : \beta \to \beta & \mathsf{g}_1 : \beta \to \beta \\ \mathsf{top} : \beta \to \gamma & \end{array} \right\}$$

So we have $\{\mathsf{st}(g) \mid g \in \mathcal{F}' \setminus \mathcal{F}\} = \{\beta, \gamma, \delta\}$ and $\{\mathsf{st}(f) \mid f \in \mathcal{F}\} = \{\alpha\}$. Therefore, the set of original symbols sorts $\{\mathsf{st}(f) \mid f \in \mathcal{F}\}$ and the set of new symbols sorts$\{\mathsf{st}(g) \mid g \in \mathcal{F}' \setminus \mathcal{F}\}$ are disjoint.

3.2 Type-based Reachability

It is well known that reachability is undecidable in general [21]. Therefore, reachability analysis mostly based on over-approximation technique. The next definition shows how one can approximate a term over a sorted signature to a set. Note that, sorted signature can be given to a standard TRS by type inference.

Example 21. Consider the running example TRS $\mathcal{R}^T$ which consisting of the following rules:

$$\begin{array}{rrclrrcl}
1: & \triangledown(\mathsf{f}(x)) & \to & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) & \to & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & \to & \mathsf{f}_1(\triangledown(x)) & 4: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) & \to & \mathsf{g}_1(\triangledown(x)) \\
5: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) & \to & \blacktriangle(x) & 6: & \underline{\mathsf{g}}(\mathsf{b}) & \to & \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
7: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) & \to & \triangle(x) & 8: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) & \to & \triangle(x) \\
9: & \mathsf{f}_1(\triangle(x)) & \to & \triangle(\mathsf{f}(x)) & 10: & \mathsf{g}_1(\triangle(x)) & \to & \triangle(\mathsf{g}(x)) \\
11: & \mathsf{top}(\triangle(x)) & \to & \mathsf{top}(\triangledown(x)) & & & &
\end{array}$$

We use the following substitutions

- $\sigma = \{x \mapsto \mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b})))\}$
- $\tau = \{y \mapsto \mathsf{b}\}$

and following rewrite step as a witness to show term $\mathsf{top}(\triangledown(x))$ reach term $\mathsf{top}(\triangle(y))$.

$$\begin{aligned}
\mathsf{top}(\triangledown(\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) \\
&\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\blacktriangle(\mathsf{b}))) \\
&\to_{\mathcal{R}^T} \mathsf{top}(\triangle(\mathsf{b})) \\
&\to_{\mathcal{R}^T} \mathsf{top}(\triangledown(\mathsf{b}))
\end{aligned}$$

Definition 32. Let $\mathcal{S}'$ be a subset of $\mathcal{S}$ on $\mathcal{S}$-sort signature $\mathcal{F}$ and a term t^α. We define the mapping $\|t^\alpha\|$ as follows:

$$\|t^\alpha\| = \begin{cases} \{t\} & \text{if } t \in \mathcal{V} \text{ and } \alpha \in \mathcal{S}' \\ \{f\} \cup \|t_1\| \cup \ldots \cup \|t_n\| & \text{if } t = f(t_1, \ldots, t_n) \text{ and } \alpha \in \mathcal{S}' \\ \varnothing & \text{otherwise} \end{cases}$$

Moreover, let $\mathcal{R}$ be a TRS. We define the *set extending system* $\|\mathcal{R}\|$ as the set of pairs of symbols sets:

$$\|\mathcal{R}\| = \{\|\ell\| \rightsquigarrow \|r\| \mid \ell \to r \in \mathcal{R}\}$$

Now, we select a special set $\mathcal{S}'$ which makes reachability approximation possible in the following way.

Definition 33. The set $\mathcal{S}'$ is the maximal subset of $\mathcal{S}$, satisfying

$$(\|\ell\| \cup \|r\|) \cap \mathcal{V} = \varnothing \text{ for every } \ell \to r \in \mathcal{R}$$

An intention behind this definition is to avoid accessible of variables while mapping a term to a set.

Lemma 1. *Suppose t be a well-sorted term of many-sorted TRS $\mathcal{R}$. Then the following equality holds:*

$$\|t\sigma\| = \|t\| \textit{ for all substitutions } \sigma$$

Example 22 (continued from Example 21)**.** The next sort information can be attached to the transformed system $\mathcal{R}^T$.

$$\left\{ \begin{array}{rlll} \triangledown : \alpha \to \beta & \vartriangle : \alpha \to \beta & \underline{\mathsf{f}} : \alpha \to \delta & \underline{\mathsf{g}} : \alpha \to \delta \\ \blacktriangledown_{\mathsf{f}} : \delta \to \beta & \blacktriangledown_{\mathsf{g}} : \delta \to \beta & \blacktriangle : \alpha \to \delta & \mathsf{b} : \alpha \\ \mathsf{f} : \alpha \to \alpha & \mathsf{g} : \alpha \to \alpha & \mathsf{f}_1 : \beta \to \beta & \mathsf{g}_1 : \beta \to \beta \\ \mathsf{top} : \beta \to \gamma & & & \end{array} \right\}$$

The set $\mathcal{S}'$ for this system is $\{\beta, \gamma, \delta\}$. Here, we obtain the set extending system $\|\mathcal{R}\|$ as follow:

$$\begin{array}{lrcllrcl} 1: & \{\triangledown\} & \rightsquigarrow & \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} & 2: & \{\triangledown\} & \rightsquigarrow & \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} \\ 3: & \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} & \rightsquigarrow & \{\mathsf{f}_1, \triangledown\} & 4: & \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} & \rightsquigarrow & \{\mathsf{g}_1, \triangledown\} \\ 5: & \{\underline{\mathsf{f}}\} & \rightsquigarrow & \{\blacktriangle\} & 6: & \{\underline{\mathsf{g}}\} & \rightsquigarrow & \{\blacktriangle\} \\ 7: & \{\blacktriangledown_{\mathsf{f}}, \blacktriangle\} & \rightsquigarrow & \{\vartriangle\} & 8: & \{\blacktriangledown_{\mathsf{g}}, \blacktriangle\} & \rightsquigarrow & \{\vartriangle\} \\ 9: & \{\mathsf{f}_1, \vartriangle\} & \rightsquigarrow & \{\vartriangle\} & 10: & \{\mathsf{g}_1, \vartriangle\} & \rightsquigarrow & \{\vartriangle\} \\ 11: & \{\mathsf{top}, \vartriangle\} & \rightsquigarrow & \{\mathsf{top}, \triangledown\} & & & & \end{array}$$

From the above definition, a term can be seen as a set, and a rewrite system becomes a set extending system. In this definition, we simulate the rewrite relation of a TRS with a *extend relation.*

Definition 34. A set A is extended to the set B denoted by $A \rightsquigarrow_R B$ if there exists $\ell \rightsquigarrow r \in R$ such that $\ell \subseteq A$, $B = A \cup r$ and $|A| < |B|$. For for convenience usage we define the following notions:

- A set A is *extendable* if and only if there exists B such that $A \rightsquigarrow_R B$.
- A set A is in *normal form* if and only if it is not extendable.
- The set B is the normal form of A if $A \rightsquigarrow_R^* B$ and B is in normal form. We denoted by $A_{\wr R}$.

Example 23. Consider the set extending system R on Example 22.

- $\{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\}$ is extendable.
- $\{\mathsf{b}^{(0)}, \mathsf{f}^{(1)}, \mathsf{g}^{(1)}, \mathsf{f}_1^{(1)}, \mathsf{g}_1^{(1)}, \underline{\mathsf{f}}^{(1)}, \underline{\mathsf{g}}^{(1)}, \triangledown^{(1)}, \vartriangle^{(1)}, \blacktriangledown_{\mathsf{f}}^{(1)}, \blacktriangledown_{\mathsf{g}}^{(1)}, \blacktriangle^{(1)}\}$ is the normal form of $\{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\}$.

Next, we illustrate the following rewrite step

$$\begin{aligned} \mathsf{top}(\triangledown(\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) &\xrightarrow{i}_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(\mathsf{b}))))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\blacktriangledown_{\mathsf{f}}(\blacktriangle(\mathsf{b}))) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\vartriangle(\mathsf{b})) \\ &\to_{\mathcal{R}^T} \mathsf{top}(\triangledown(\mathsf{b})) \end{aligned}$$

with this extend step

$$\begin{aligned} \{\mathsf{top}, \triangledown\} &\rightsquigarrow_R \{\mathsf{top}, \triangledown\} \cup \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} \\ &\rightsquigarrow_R \{\mathsf{top}, \triangledown, \blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} \cup \{\blacktriangle\} \\ &\rightsquigarrow_R \{\mathsf{top}, \triangledown, \blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}, \blacktriangle\} \cup \{\vartriangle\} \\ &\rightsquigarrow_R \{\mathsf{top}, \triangledown, \blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}, \blacktriangle, \vartriangle\} \cup \{\triangledown\} \end{aligned}$$

Before the proof, we show the next lemma.

Lemma 2. *If* $s \to_{\mathcal{R}} t$ *then* $\|t\|_{\downarrow\|\mathcal{R}\|} \subseteq \|s\|_{\downarrow\|\mathcal{R}\|}$.

Proof. Suppose $s \rightsquigarrow_{\mathcal{R}} t$. We can write $s = C[\ell\sigma]$, $t = C[r\sigma]$ and $s|_p = \ell\sigma$. The context can be considered in two cases.

- If there exists $q \in \mathcal{P}\mathsf{os}(s)$ such that $q \geq p$ and $\|s|_q\| = \varnothing$, then there exists contexts C_1 and C_2 such that $s|_q = C_2[\ell\sigma]$ and $C = C_1[C_2]$. Therefore, $\|s\| = \|C_1\| = \|t\|$ and the claim holds.

- If for all $q \in \mathcal{P}\mathsf{os}(s)$ such that if $q \geq p$ then $\|s|_q\| \neq \varnothing$, then $\|s\| = \|C\| \cup \|\ell\|$ and $\|t\| = \|C\| \cup \|r\|$. We can extend $\|s\|$ one more step to obtain $\|C\| \cup \|\ell\| \cup \|r\|$. Since $\|t\| \subseteq \|C\| \cup \|\ell\| \cup \|r\|$, we have $\|t\|_{\downarrow\|\mathcal{R}\|} \subseteq \|s\|_{\downarrow\|\mathcal{R}\|}$.

$\square$

The approximation of reachability can be computed by computing a normal form with respect to extending relation and simply check subset relation. First, we prove computability of approximation by proving a normal form with respect to extending relation is unique and computable.

Theorem 6. *The relation* $\rightsquigarrow_{\|\mathcal{R}\|}$ *is complete.*

Proof. We show Theorem 6 by proving termination and confluence property with respect to extending relation.

- Termination proof: Since set over function symbols is a finite set, we claim the termination property by showing if $A \rightsquigarrow_{\|\mathcal{R}\|} B$ then $A \subsetneq B$. Suppose $A \rightsquigarrow_{\|\mathcal{R}\|} B$ then we can write $B = A \cup \|r\|$ for some r in right hand-side of $\mathcal{R}$ and $|A| < |B|$. Therefore $A \subsetneq B$.

- Confluence proof: Since the relation is terminating, we only show local confluence for claiming confluence. Suppose $A \rightsquigarrow_{\|\mathcal{R}\|} B$ and $A \rightsquigarrow_{\|\mathcal{R}\|} C$ then we can write $B = A \cup \|r_1\|$ and $C = A \cup \|r_2\|$ such that $\|\ell_1\|, \|\ell_2\| \subseteq A$ for some $\ell_1 \to r_1, \ell_2 \to r_2 \in \mathcal{R}$. Therefore $B \rightsquigarrow_{\|\ell_2\| \rightsquigarrow \|r_2\|} A \cup \|r_1\| \cup \|r_2\|$ and $C \rightsquigarrow_{\|\ell_1\| \rightsquigarrow \|r_1\|} A \cup \|r_1\| \cup \|r_2\|$.

$\square$

Hence, a normal form with respect to $\rightsquigarrow_{\|\mathcal{R}\|}$ is unique and computable. Next, we claim the following theorem by induction on rewrite step.

Theorem 7. *If* $s \hookrightarrow_{\mathcal{R}} t$, *then* $\|t\| \subseteq \|s\|_{\downarrow\|\mathcal{R}\|}$

Proof. Our hypothesis is if $s\sigma \to^n_{\mathcal{R}} t\tau$ for some substitutions σ and τ, then $\|t\| \subseteq \|s\|_{\downarrow\|\mathcal{R}\|}$.

- if $n = 0$, then $s\sigma = t\tau$. Since $\|s\sigma\| = \|s\|$ and $\|t\tau\| = \|t\|$, we know that $\|s\| = \|t\|$. We have $\|t\| \subseteq \|t\|_{\downarrow\|\mathcal{R}\|}$ because $A \rightsquigarrow_{\|\mathcal{R}\|} B$ implies $A \subset B$. Therefore, if $t \to^{\epsilon}_{\mathcal{R}} u$, then $\|u\| \subseteq \|t\|_{\downarrow\|\mathcal{R}\|}$.

- if $n > \epsilon$, then $s\sigma \to_{\mathcal{R}} t_1 \to_{\mathcal{R}}^{n-1} t\tau$. We have $\|t\| \subseteq \|t_1\|_{\wr_{\|\mathcal{R}\|}}$ by induction hypothesis. Since $s\sigma \to_{\mathcal{R}} t_1$, we know that $\|t_1\|_{\wr_{\|\mathcal{R}\|}} \subseteq \|s\sigma\|_{\wr_{\|\mathcal{R}\|}}$ by Lemma 2. Because $\|s\sigma\| = \|s\|$. So, $\|s\sigma\|_{\wr_{\|\mathcal{R}\|}} = \|s\|_{\wr_{\|\mathcal{R}\|}}$. Therefore, $\|t\| \subseteq \|s\|_{\wr_{\|\mathcal{R}\|}}$.

Hence, if $s \hookrightarrow_{\mathcal{R}} t$, then $\|t\| \subseteq \|s\|_{\wr_{\|\mathcal{R}\|}}$. □

We will introduce *ICAP* function [11] and reachability approximation by *ICAP*. Then we compare *type-based reachability analysis* and *ICAP* on the running example.

Definition 35. Let $\mathcal{R}$ be a TRS and let t be a term. We define *ICAP* as follows:

$$ICAP(t) = \begin{cases} t & \text{if } t \in \mathcal{V} \\ u & \text{if } t = f(t_1, \ldots, t_n) \text{ and } u \text{ does not unify with any } v \in \mathit{lhs}(\mathcal{R}) \\ \text{fresh variable} & \text{otherwise} \end{cases}$$

where, $u = f(ICAP(t_1), \ldots, ICAP(t_n))$.

Theorem 8. *If $s \hookrightarrow_{\mathcal{R}} t$, then $ICAP(s)$ and t are unifiable.*

Example 24 (continued from Example 21)**.** Here we suppose that with other termination methods we succeeded to eliminate rules 5 and 6 from $\mathcal{R}^T$. Terms of form $\mathsf{top}(\triangledown(s))$ no longer reaches $\mathsf{top}(\vartriangle(t))$ for any terms s and t. In this setting, let consider the dependency graph approximation $\mathsf{DG}(\mathcal{R}^T)$, the corresponding dependency pairs of rule 11 in TRS $\mathcal{R}^T$ will form a self-cycle if the reachability approximation of $\mathsf{top}^\sharp(\triangledown(x))$ and $\mathsf{top}^\sharp(\vartriangle(y))$ yields yes answer. Now, the accuracy of approximations become importance to remove the cycle and lead to termination.

Our type-based reachability analysis can detect this unreachability in the following way: We interpret each term to a set of function symbols that may appear in reachable terms. This can be computed by using the rewrite system $\|\mathcal{R}^T\|$ on sets:

$$\begin{array}{llll}
1: & \{\triangledown\} \rightsquigarrow \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} & 2: & \{\triangledown\} \rightsquigarrow \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} \\
3: & \{\blacktriangledown_{\mathsf{f}}, \underline{\mathsf{f}}\} \rightsquigarrow \{\mathsf{f}_1, \triangledown\} & 4: & \{\blacktriangledown_{\mathsf{g}}, \underline{\mathsf{g}}\} \rightsquigarrow \{\mathsf{g}_1, \triangledown\} \\
 & & & \\
7: & \{\blacktriangledown_{\mathsf{f}}, \blacktriangle\} \rightsquigarrow \{\vartriangle\} & 8: & \{\blacktriangledown_{\mathsf{g}}, \blacktriangle\} \rightsquigarrow \{\vartriangle\} \\
9: & \{\mathsf{f}_1, \vartriangle\} \rightsquigarrow \{\vartriangle\} & 10: & \{\mathsf{g}_1, \vartriangle\} \rightsquigarrow \{\vartriangle\} \\
11: & \{\mathsf{top}, \vartriangle\} \rightsquigarrow \{\mathsf{top}, \triangledown\} & &
\end{array}$$

Here, we compare approximation of $\mathsf{top}^\sharp(\triangledown(x)) \hookrightarrow_{\mathcal{R}^T} \mathsf{top}^\sharp(\vartriangle(y))$ with this approach and *ICAP* function.

- The approximation *ICAP* yield reachable answer. Since $ICAP(\mathsf{top}(\triangledown(x))) = \mathsf{top}(x')$, we have $\mathsf{top}(x')$ and $\mathsf{top}(\vartriangle(y))$ are unifiable.

- $$\begin{aligned} \|\mathsf{top}(\vartriangle(x))\| &= \{\mathsf{top}, \vartriangle\} \\ &\not\subseteq \{\mathsf{top}, \blacktriangledown_{\mathsf{s}}, \underline{\mathsf{s}}\} \\ &= \{\mathsf{top}, \triangledown\}_{\wr_{\|\mathcal{R}^T\|}} \end{aligned}$$

In this example, type based reachability analysis shows a performance to remove cycle in $\mathsf{DG}(\mathcal{R}^T)$ which supports termination proof in dependency graphs technique.

This analysis can be integrated into techniques such as dependency graphs [2], usable rules [13, 24], and usable replacement maps [14].

Chapter 4

Pattern Separation Transformation

This chapter describes a transformation technique for innermost termination analysis, dubbed *pattern separation*. This technique focuses on systems resulting from Thiemann's transformation stated in Chapter 2. The key point is to reduce the gap between innermost rewriting and reduction order based termination techniques. The analysis provides more possibility to prove the system innermost terminating.

4.1 Complementation

First of all, we recall to the complement algorithm introduced by Lazrek, Lescanne, and Thiel [17]. The complement algorithm plays a key role in pattern separation. We here reformulate their algorithm as a simple recursive function.

Definition 36. Let t be a term over signature $\mathcal{F}$. The complement of t is a set of terms which an element is not instant of t. The complement $\mathcal{C}(t)$ defined as follows:

$$\mathcal{C}(t) = \begin{cases} \varnothing & \text{if } t \in \mathcal{V} \\ A_1 \cup \ldots \cup A_n \cup B & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Here, Ai and B stand for the following sets:

$$\begin{aligned} A_i &= \{f(t_1, \ldots, t_{i-1}, c_i, x_{i+1}, \ldots, x_n) \mid c_i \in \mathcal{C}(t_i)\} \\ B &= \{g(x_1, \ldots, x_m) \mid g \in \mathcal{F} \setminus \{f\}\} \end{aligned}$$

where, $x_1, x_2, \ldots$ are fresh variables.

Example 25. Consider the signature $\mathcal{F} = \{\mathsf{f}^{(1)}, \mathsf{g}^{(1)}, \mathsf{h}^{(1)}, \mathsf{0}^{(0)}, \mathsf{1}^{(0)}\}$. We illustrate computation of $\mathcal{C}(\mathsf{f}(\mathsf{g}(\mathsf{0})))$. Below we follow notations in Definition 36 to compute the set step by step

We start with computing $\mathcal{C}(\mathsf{0})$:

$$\begin{aligned} \mathcal{C}(\mathsf{0}) &= \{g(x_1, \ldots, x_m) \mid g \in \mathcal{F} \text{ and } \mathsf{0} \neq g\} \\ &= \{\mathsf{f}(x), \mathsf{g}(x), \mathsf{h}(x), \mathsf{1}\} \end{aligned}$$

Then we compute $\mathcal{C}(\mathsf{g}(\mathsf{0}))$:

$$\begin{aligned}\mathcal{C}(\mathsf{g}(\mathsf{0})) &= \{\mathsf{g}(c) \mid c \in \mathcal{C}(\mathsf{0})\} \cup \{g(x_1, \dots, x_m) \mid g \in \mathcal{F} \text{ and } \mathsf{g} \neq g\} \\ &= \{\mathsf{g}(c) \mid c \in \mathcal{C}(\mathsf{0})\} \cup \{\mathsf{f}(x), \mathsf{h}(x), \mathsf{1}, \mathsf{0}\} \\ &= \{\mathsf{g}(\mathsf{f}(x)), \mathsf{g}(\mathsf{g}(x)), \mathsf{g}(\mathsf{h}(x)), \mathsf{g}(\mathsf{1}), \mathsf{f}(x), \mathsf{h}(x), \mathsf{1}, \mathsf{0}\}\end{aligned}$$

Finally, we can compute $\mathcal{C}(\mathsf{f}(\mathsf{g}(\mathsf{0})))$:

$$\begin{aligned}\mathcal{C}(\mathsf{f}(\mathsf{g}(\mathsf{0}))) &= \{\mathsf{f}(c) \mid c \in \mathcal{C}(\mathsf{g}(\mathsf{0}))\} \cup \{g(x_1, \dots, x_m) \mid g \in \mathcal{F} \text{ and } \mathsf{f} \neq g\} \\ &= \{\mathsf{f}(c) \mid c \in \mathcal{C}(\mathsf{g}(\mathsf{0}))\} \cup \{\mathsf{g}(x), \mathsf{h}(x), \mathsf{1}, \mathsf{0}\} \\ &= \left\{\begin{array}{cccc} \mathsf{f}(\mathsf{g}(\mathsf{f}(x))), & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))), & \mathsf{f}(\mathsf{g}(\mathsf{h}(x))), & \mathsf{f}(\mathsf{g}(\mathsf{1})), \\ \mathsf{f}(\mathsf{f}(x)), & \mathsf{f}(\mathsf{h}(x)), & \mathsf{f}(\mathsf{1}), & \mathsf{f}(\mathsf{0}), \\ \mathsf{g}(x), & \mathsf{h}(x), & \mathsf{1}, & \mathsf{0} \end{array}\right\}\end{aligned}$$

Definition 37. We define $\Sigma(t)$ as the set of all ground instantiations of a term t over a signature. Moreover, for a set of term T we defined $\Sigma(T)$ as follows:

$$\Sigma(T) = \bigcup_{t \in T} \Sigma(t)$$

Lemma 3. *The identity $\mathcal{T}(\mathcal{F}) = \Sigma(t) \uplus \Sigma(\mathcal{C}(t))$ holds for all linear terms t*

The set of all ground instantiations of t and set of all ground instantiations of set $\mathcal{C}(t)$ are disjoint and union of them result as the set of all ground terms. We divide the proof of Lemma 3 in to two part

- $\Sigma(t)$ and $\Sigma(\mathcal{C}(t))$ are disjoint for every term t
- $\mathcal{T}(\mathcal{F})$ and $\Sigma(t) \cup \Sigma(\mathcal{C}(t))$ are equivalent for all linear terms t.

Lemma 4. *The sets $\Sigma(t)$ and $\Sigma(\mathcal{C}(t))$ are disjoint.*

Proof. We show the claim by induction on term t.

- If t is a variable, then we have $\mathcal{C}(t) = \varnothing$ by Definition 36, then $\Sigma(\mathcal{C}(t)) = \varnothing$. Therefore $\Sigma(t) \cap \Sigma(\mathcal{C}(t)) = \varnothing$.
- Suppose $t = f(t_1, \dots, t_n)$. Assume to contrary $u \in \Sigma(t) \cap \Sigma(\mathcal{C}(t))$. As $u \in \Sigma(t)$, there is a substitution σ with $u = t\sigma$. As $u \in \Sigma(\mathcal{C}(t))$, there exist $v \in \mathcal{C}(t) = A_1 \cup \dots \cup A_n \cup B$ such that $u = v\tau$ for some substitution τ. We distinguish two cases.
  - If $v \in \Sigma(B)$, then we can write $v = g(x_1, \dots, x_m)$ for fresh variables $x_1, \dots, x_n$. We know that $f \neq g$ by Definition 36. Hence, $u = f(t_1\sigma, \dots, t_n\sigma) \neq g(x_1\tau, \dots, x_m\tau) = v\tau$ which contradicted.

– Suppose $v \in \Sigma(A_i)$ for some $i \in \{1, \ldots, n\}$. We can write $v = f(t_1, \ldots, t_{i-1}, c_i, x_{i+1}, \ldots, x_n)$ for an arbitrary i and fresh variables $x_{i+1}, \ldots, x_n$. We have $c_i\tau \in \Sigma(\mathcal{C}(t_i))$ then $c_i\tau \notin \Sigma(t_i)$ from induction hypothesis. Therefore $c_i\tau \neq t_i\sigma$. Hence $f(t_1\sigma, \ldots, t_n\sigma) \neq f(t_1\tau, \ldots, t_{i-1}\tau, c_i\tau, x_{i+1}\tau, \ldots, x_n\tau)$ which contradicted.

Therefore $v \notin \mathcal{C}(t)$.

Here, we can conclude that $\Sigma(t)$ and $\Sigma(\mathcal{C}(t))$ are disjoint for every term t. □

Next, we prove that $\mathcal{T}(\mathcal{F})$ and $\Sigma(t) \cup \Sigma(\mathcal{C}(t))$ are equivalent for all linear terms t. Since $\Sigma(t) \cup \Sigma(\mathcal{C}(t)) \subseteq \mathcal{T}(\mathcal{F})$ is trivial. We only show that $\mathcal{T}(\mathcal{F}) \subseteq \Sigma(t) \cup \Sigma(\mathcal{C}(t))$.

Lemma 5. *The set $\mathcal{T}(\mathcal{F})$ is a subset of $\Sigma(t) \cup \Sigma(\mathcal{C}(t))$ for all linear terms t.*

Proof. Let $u \in \mathcal{T}(\mathcal{F})$. We show the claim by induction on u. As u is ground, one can write $u = f(u_1, \ldots, u_n)$.

- If t is variable, then $u \in \Sigma(t)$.
- If $t = g(t_1, \ldots, t_m)$ with $f \neq g$, then $f(x_1, \ldots, x_n) \in \mathcal{C}(t)$ for fresh variables $x_1, \ldots, x_n$. We take σ as $\{x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\}$. Hence, $u = f(x_1, \ldots, x_n)\sigma \in \Sigma(\mathcal{C}(t))$.
- If $t = f(t_1, \ldots, t_n)$, then the induction hypothesis yields $u_i \in \Sigma(t_i) \uplus \Sigma(\mathcal{C}(t_i))$ for all i. If $u_i \in \Sigma(t_i)$ for all i, then $u \in \Sigma(t)$. Otherwise, there exists an i such that $u_j \in \Sigma(t_j)$ for all $0 < j < i$ and $u_i \in \Sigma(\mathcal{C}(t_i))$. We have $f(t_1, \ldots, t_{i-1}, c_i, x_{i+1}, \ldots, x_n) \in A_i \subseteq \mathcal{C}(t)$ for some $c_i \in \mathcal{C}(t_i)$ and fresh variables $x_{i+1}, \ldots, x_n$. We take σ_j as the most general unification of u_j and t_j. We can choose c_i such that $c_i\sigma_i = u_i$ and take σ as $\sigma_1, \ldots, \sigma_i\{x_{i+1} \mapsto u_1, \ldots, x_n \mapsto u_n\}$. We can safely substitute since t is a linear term. Hence, $u = f(t_1, \ldots, t_{i-1}, c_i, x_{i+1}, \ldots, x_n)\sigma \in \Sigma(\mathcal{C}(t))$.

Therefore, $u \in \Sigma(t) \uplus \Sigma(\mathcal{C}(t))$. Here, we can conclude that $\mathcal{T}(\mathcal{F}) \subseteq \Sigma(t) \cup \Sigma(\mathcal{C}(t))$. □

The linearity condition in Lemma 3 is an important limitation of this algorithm and it effects the *pattern separation* technique which will be introduced in the next section.

Example 26. Let signature $\mathcal{F} = \{+^{(2)}, \mathsf{s}^{(1)}, \mathsf{0}^{(0)}\}$. We illustrate the necessity of linear condition. Let $t = \mathsf{s}(x) + x$. Then we obtain:

$$\mathcal{C}(t) = \left\{ \begin{array}{cc} (x+y)+z, & \mathsf{0}+x, \\ \mathsf{s}(x), & \mathsf{0} \end{array} \right\}$$

However,

- $\mathsf{s}(\mathsf{0})+\mathsf{s}(\mathsf{0}) \in \mathcal{T}(\mathcal{F})$,
- $\mathsf{s}(\mathsf{0})+\mathsf{s}(\mathsf{0}) \notin \Sigma(\mathsf{s}(x)+x)$, and
- $\mathsf{s}(\mathsf{0})+\mathsf{s}(\mathsf{0}) \notin \Sigma(\mathcal{C}(\mathsf{s}(x)+x))$.

Therefore $\mathcal{T}(\mathcal{F}) = \Sigma(t) \uplus \Sigma(\mathcal{C}(t))$ does not hold.

4.2 Pattern Separation

Most of termination techniques are base on reduction orders which the order satisfies closure under contexts and closure under substitution. However, innermost rewriting is not closed under substitutions witnessed by the following example.

Example 27. Consider the TRS:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ \mathsf{g}(\mathsf{0}) \to \mathsf{h}(\mathsf{g}(\mathsf{1})) \end{array} \right\}$$

There exists a term which is not closed under substitutions:

$$\mathsf{f}(\mathsf{g}(x)) \text{ for } \sigma = \{x \mapsto \mathsf{0}\}$$

Because $\mathsf{f}(\mathsf{g}(x)) \xrightarrow{i}_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\mathsf{0}))$ but $\mathsf{f}(\mathsf{g}(\mathsf{0})) \not\xrightarrow{i}_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\mathsf{0}))$.

We overcome this situation by resolving ambiguities of rule application caused by overlapping rules. For this sake, we instantiate rewrite rules. The following theorem allows us to focus on ground terms when showing innermost termination.

Theorem 9. *A TRS $\mathcal{R}$ over $\mathcal{F}$ is innermost terminating if and only if the TRS $\mathcal{R}$ over $\mathcal{F} \uplus \{\mathsf{h}^{(1)}, \mathsf{c}^{(0)}\}$ is ground innermost terminating.*

Example 28. Consider the TRS $\mathcal{R}$ in Example 27. We instantiate $\mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{0}))$ to obtain the following rules:

$$\begin{array}{lr} 1.1: & \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ 1.2: & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ 1.3: & \mathsf{f}(\mathsf{g}(\mathsf{h}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ 1.4: & \mathsf{f}(\mathsf{g}(\mathsf{1})) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \end{array}$$

In this instantiated system, only the rule 1.2 is overlapping with rule 2 in TRS $\mathcal{R}$. However, the optimized version of pattern separation can get rid of it.

In order to understand the process, we introduce related theorem, definition, and notation. First, we begin with definition and notation for pattern separation.

Definition 38. Let $\mathsf{ren}(t)$ be an arbitrary but fixed linear term resulting from replacing all variable occurrences in a term t with fresh variables. We define the following notation:

- $s \uparrow t$ if $\mathsf{ren}(s)$ and $\mathsf{ren}(t)$ are unifiable.
- $s \sqcup t$ is a fixed instance of s by a mgu of $\mathsf{ren}(s)$ and $\mathsf{ren}(t)$.
- $S \otimes T = \{s \sqcup t \mid s \in S,\, t \in T, \text{ and } s \uparrow t\}$.

Example 29. Let A and B are set of terms.

- $A = \left\{\ \mathsf{f}(\mathsf{g}(x)),\ \ \mathsf{g}(\mathsf{0})\ \right\}$
- $B = \left\{ \begin{array}{ccc} \mathsf{f}(\mathsf{g}(\mathsf{f}(x)), & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))), & \mathsf{f}(\mathsf{0}), \\ \mathsf{f}(\mathsf{f}(x)), & \mathsf{g}(x), & \mathsf{0} \end{array} \right\}$

We have $\mathsf{f}(\mathsf{g}(x)) \uparrow \mathsf{f}(\mathsf{g}(\mathsf{f}(x))$, $\mathsf{f}(\mathsf{g}(x)) \uparrow \mathsf{f}(\mathsf{g}(\mathsf{g}(x)))$, and $\mathsf{g}(\mathsf{0}) \uparrow \mathsf{g}(x)$ while the other pairs in $A \times B$ are not unifiable. Hence,

$$\begin{aligned} A \otimes B &= \{a \sqcup b \mid a \in S,\ b \in T, \text{ and } a \uparrow b\} \\ &= \{\mathsf{f}(\mathsf{g}(x)) \sqcup \mathsf{f}(\mathsf{g}(\mathsf{f}(x)), \mathsf{f}(\mathsf{g}(x)) \sqcup \mathsf{f}(\mathsf{g}(\mathsf{g}(x))), \mathsf{g}(\mathsf{0}) \sqcup \mathsf{g}(x)\} \\ &= \{\mathsf{f}(\mathsf{g}(\mathsf{f}(x)), \mathsf{f}(\mathsf{g}(\mathsf{g}(x))), \mathsf{g}(\mathsf{0})\} \end{aligned}$$

We use the $\otimes$ operator to perform intersection of ground term in a view of finite representation.

Lemma 6. *The identity $\Sigma(S \otimes T) = \Sigma(S) \cap \Sigma(T)$ holds for all sets of terms S and T.*

Proof. We divide proof of Lemma 6 into two steps.

- First, we show that $\Sigma(S \otimes T) \subseteq \Sigma(S) \cap \Sigma(T)$ holds for all sets of terms S and T. Let $a \in \Sigma(S \otimes T)$ and σ an arbitrary substitution. We can write $a = (s \sqcup t)\sigma$ for some $s \in S$ and $t \in T$ by Definition 38. Suppose τ is the most general unification of s and t. Now we have $s \sqcup t = s\tau = t\tau$. Hence, $a = s\tau\sigma \in \Sigma(S)$ and $a = t\tau\sigma \in \Sigma(T)$. Therefore, $a \in \Sigma(S) \cap \Sigma(T)$.
- Next, we show that $\Sigma(S) \cap \Sigma(T) \subseteq \Sigma(S \otimes T)$ holds for all sets of terms S and T. Let $a \in \Sigma(S) \cap \Sigma(T)$. We can write $a = s\sigma_1 = t\sigma_2$ for some $s \in S$, $t \in T$ and suitable substitution σ_1 and σ_2. Since the identity $s\sigma_1 = t\sigma_2$ holds, there exists τ such that $(s \sqcup t)\tau = s\sigma_1 = t\sigma_2$. Therefore, $a = (s \sqcup t)\tau \in \Sigma(S \otimes T)$.

□

From here on, we define definitions for pattern separation.

Definition 39. We define *non-root overlap* of term ℓ with respect to a TRS $\mathcal{R}$ as follows:

$$\mathsf{O}_{\mathcal{R}}(\ell) = \{\ell[\ell|_p \sqcup \ell']_p \mid \ell' \in \mathit{lhs}(\mathcal{R}),\ p \in \mathcal{P}\mathsf{os}_{\mathcal{F}}(\ell) \setminus \{\epsilon\} \text{ and } \ell|_p \uparrow \ell'\}$$

This function collects terms which are instance of ℓ but rewritable at non-root position by the TRS $\mathcal{R}$.

Example 30. Let TRS $\mathcal{R}$ consist of the following rules:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) \\ \mathsf{g}(\mathsf{0}) \to \mathsf{h}(\mathsf{g}(\mathsf{1})) \end{array} \right\}$$

We compute non-root overlaps of left-hand-side of rules in TRS $\mathcal{R}$, which is resulting as:

- $\mathsf{O}_{\mathcal{R}}(\mathsf{f}(\mathsf{g}(x))) = \{\mathsf{f}(\mathsf{g}(\mathsf{0}))\}$

- $\mathsf{O}_{\mathcal{R}}(\mathsf{g}(0)) = \varnothing$

Next, we define *complement pattern* of term ℓ with respect to a TRS $\mathcal{R}$. This function aims to collect terms which are instances of ℓ but not rewritable at any non-root position.

Definition 40. Let $\mathcal{R}$ be a TRS and let ℓ be a term. We define *complement pattern* $\mathsf{P}_{\mathcal{R}}(\ell)$ as follows:

$$\mathsf{P}_{\mathcal{R}}(\ell) = \begin{cases} \{\ell\} & \text{if } \mathsf{O}_{\mathcal{R}}(\ell) = \varnothing \\ \{\ell\} \otimes \bigotimes_{t \in \mathsf{O}_{\mathcal{R}}(\ell)} \mathcal{C}(t) & \text{otherwise} \end{cases}$$

Lemma 7. *For a left-linear TRS $\mathcal{R}$ over signature $\mathcal{F}$ and linear term ℓ*

$$\Sigma(\mathsf{P}_{\mathcal{R}}(\ell)) = \{ s \in \mathcal{T}(\mathcal{F}) \mid \ell \geq s \text{ and } t \ngeq s \text{ for all } t \in \mathsf{O}_{\mathcal{R}}(\ell) \ \}$$

holds.

Proof. We distinguish two cases.

- If $\mathsf{O}_{\mathcal{R}}(\ell) = \varnothing$, then

$$\begin{aligned} & s \in \Sigma(\mathsf{P}_{\mathcal{R}}(\ell)) \\ \iff & s \in \mathcal{T}(\mathcal{F}) \text{ and } \ell \geq s \text{ and } t \ngeq s \text{ for all } t \in \mathsf{O}_{\mathcal{R}}(\ell) \\ \iff & s \in \Sigma(\ell) \\ \iff & s \in \Sigma(\mathsf{P}_{\mathcal{R}}(\ell)) && \text{by Definition 40} \end{aligned}$$

- If $\mathsf{O}_{\mathcal{R}}(\ell) \neq \varnothing$, then

$$\begin{aligned} & s \in \Sigma(\mathsf{P}_{\mathcal{R}}(\ell)) \\ \iff & s \in \mathcal{T}(\mathcal{F}) \text{ and } \ell \geq s \text{ and } t \ngeq s \text{ for all } t \in \mathsf{O}_{\mathcal{R}}(\ell) \\ \iff & s \in \Sigma(\ell) \text{ and } s \notin \Sigma(t) \text{ for all } t \in \mathsf{O}_{\mathcal{R}}(\ell) && \text{by Definition 37} \\ \iff & s \in \Sigma(\ell) \text{ and } s \in \Sigma(\mathcal{C}(t)) \text{ for all } t \in \mathsf{O}_{\mathcal{R}}(\ell) && \text{by Lemma 3} \\ \iff & s \in \Sigma(\ell) \cap \bigcap_{t \in \mathsf{O}_{\mathcal{R}}(\ell)} \Sigma(\mathcal{C}(t))` \\ \iff & s \in \Sigma(\{\ell\} \otimes \bigotimes_{t \in \mathsf{O}_{\mathcal{R}}(\ell)} \mathcal{C}(t)) && \text{by Lemma 6} \\ \iff & s \in \Sigma(\mathsf{P}_{\mathcal{R}}(\ell)) && \text{by Definition 40} \end{aligned}$$

□

Example 31. Consider TRS $\mathcal{R}$ from the previous example. The complement pattern of left-hand-side of rules in TRS $\mathcal{R}$, which is resulting as:

- $\mathsf{P}_{\mathcal{R}}(\mathsf{f}(\mathsf{g}(x))) = \left\{ \begin{array}{ll} \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \\ \mathsf{f}(\mathsf{g}(\mathsf{h}(x))) & \mathsf{f}(\mathsf{g}(1)) \end{array} \right\}$

- $\mathsf{P}_{\mathcal{R}}(\mathsf{g}(0)) = \{\mathsf{g}(0)\}$

Lastly, we rewrite a collection of term which is collected by complement pattern, with innermost step with respect to a TRS $\mathcal{R}$.

Definition 41. Let $\mathcal{R}$ be a TRS over a signature $\mathcal{F}$. The *pattern separation* version of a TRS $\mathcal{R}$ over signature $\mathcal{F}$, denoted by $\mathcal{S}(\mathcal{R})$ and defined as follows:

$$\mathcal{S}(\mathcal{R}) = \{\ell \to r \mid \ell' \in \mathit{lhs}(\mathcal{R}),\ \ell \in \mathsf{P}_{\mathcal{R}}(\ell'),\ \text{and}\ \ell \xrightarrow{i}_{\mathcal{R}} r\}$$

Note that the complement algorithm compute over signature $\mathcal{F} \uplus \{\mathsf{h}^{(1)}, \mathsf{c}^{(0)}\}$.

Example 32. Continuing from Example 31. The pattern separation version of a TRS $\mathcal{R}$ is

$$\mathcal{S}(\mathcal{R}) = \left\{ \begin{array}{cc} \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{f}(\mathsf{g}(0)) & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{f}(\mathsf{g}(\mathsf{h}(x))) \to \mathsf{f}(\mathsf{g}(0)) & \mathsf{f}(\mathsf{g}(1)) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{g}(0) \to \mathsf{h}(\mathsf{g}(1)) & \end{array} \right\}$$

Lemma 8. *Let s and t be ground terms. Then $s \xrightarrow{i}_{\mathcal{R}} t$ if and only if $s \xrightarrow{i}_{\mathcal{S}(\mathcal{R})} t$.*

Proof. We separate the proof into two directions.

- First, we show the 'if'-direction. Since Innermost rewrite relation is a subset of rewrite relation, then we have $\xrightarrow{i}_{\mathcal{S}(\mathcal{R})}$ is a subset of $\to_{\mathcal{S}(\mathcal{R})}$. Following from Definition 41, we have $\to_{\mathcal{S}(\mathcal{R})}$ is a subset of $\xrightarrow{i}_{\mathcal{R}}$. Therefore, $s \xrightarrow{i}_{\mathcal{S}(\mathcal{R})} t$ implies $s \xrightarrow{i}_{\mathcal{R}} t$.

- Next, we show the 'only if'-direction. Suppose $s \xrightarrow{i}_{\mathcal{R}} t$ with $s = C[\ell\sigma]$, $t = C[r\sigma]$ for some rule $\ell \to r \in \mathcal{R}$, a context C, and a substitution σ. All proper subterms of $\ell\sigma$ are in normal form. It is obvious that $\ell\sigma \in \Sigma(\ell)$. Due to the normal form condition $\ell\sigma \notin \Sigma(\mathsf{O}_{\mathcal{R}}(\ell))$ follows from Definition 39. Because $\ell\sigma \in \Sigma(\ell)$ and $\ell\sigma \notin \Sigma(\mathsf{O}_{\mathcal{R}}(\ell))$ we can conclude $\ell\sigma \in \Sigma(\mathsf{P}_{\mathcal{R}}(\ell))$ by Lemma 7. There exists substitution σ_1 such that $\ell\sigma_1 \in \mathsf{P}_{\mathcal{R}}(\ell)$ and $\ell\sigma_1\sigma_2 = \ell\sigma$. Since $\ell\sigma_1 \in \mathsf{P}_{\mathcal{R}}(\ell)$ and $\ell\sigma \xrightarrow{i}_{\mathcal{R}} r\sigma$, we have $\ell\sigma_1 \to r\sigma_1 \in \mathcal{S}(\mathcal{R})$ by Definition 41. Because all proper subterms of $\ell\sigma$ are in normal form, $\ell\sigma_1\sigma_2 \xrightarrow{i}_{\mathcal{S}(\mathcal{R})} r\sigma_1\sigma_2$ holds. Hence, $s = C[\ell\sigma_1\sigma_2] \xrightarrow{i}_{\mathcal{S}(\mathcal{R})} C[r\sigma_1\sigma_2] = t$.

$\square$

Theorem 10. *A left-linear TRS $\mathcal{R}$ is innermost terminating if and only if $\mathcal{S}(\mathcal{R})$ is innermost terminating.*

Proof. Let $\mathcal{R}$ be a left-linear TRS over signature $\mathcal{F}$.

$$\begin{array}{lll} \mathcal{R} \text{ is innermost terminating} & & \\ \iff \mathcal{R} \text{ over signature } \mathcal{F}' \text{ is ground innermost terminating} & & \text{by Theorem 9} \\ \iff \mathcal{S}(\mathcal{R}) \text{ is ground innermost terminating} & & \text{by Lemma 8} \\ \iff \mathcal{S}(\mathcal{R}) \text{ is innermost terminating} & & \text{by Theorem 9} \end{array}$$

where, $\mathcal{F}' = \mathcal{F} \uplus \{\mathsf{h}^{(1)}, \mathsf{c}^{(0)}\}$. $\square$

This technique applied to a system and transformed it into another system. The technique itself does not provide a full termination prove. So, we can select any suited termination technique for the transformed system to show a complete proof.

Example 33. Consider the running example TRS $\mathcal{R}^T$ which consisting of the following rules:

$$\begin{array}{rrlrrl}
1: & \triangledown(\mathsf{f}(x)) & \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) & \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & \to \mathsf{f}_1(\triangledown(x)) & 4: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) & \to \mathsf{g}_1(\triangledown(x)) \\
5: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) & \to \blacktriangle(x) & 6: & \underline{\mathsf{g}}(\mathsf{b}) & \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
7: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) & \to \triangle(x) & 8: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) & \to \triangle(x) \\
9: & \mathsf{f}_1(\triangle(x)) & \to \triangle(\mathsf{f}(x)) & 10: & \mathsf{g}_1(\triangle(x)) & \to \triangle(\mathsf{g}(x)) \\
11: & \mathsf{top}(\triangle(x)) & \to \mathsf{top}(\triangledown(x)) & & &
\end{array}$$

We compute $\mathcal{S}(\mathcal{R}^T)$ (i.e. the pattern separation of the TRS $\mathcal{R}^T$), which is resulting as:

$$\left\{\begin{array}{rcrc}
1: & \triangledown(\mathsf{f}(x)) :\to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) :\to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\triangledown(x))) :\to \mathsf{f}_1(\triangledown(\triangledown(x))) & 4: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangledown_{\mathsf{f}}(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangledown_{\mathsf{f}}(x))) \\
5: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\underline{\mathsf{f}}(x))) :\to \mathsf{f}_1(\triangledown(\underline{\mathsf{f}}(x))) & 6: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{g}(x))) \\
7: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangledown_{\mathsf{g}}(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangledown_{\mathsf{g}}(x))) & 8: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\underline{\mathsf{g}}(x))) :\to \mathsf{f}_1(\triangledown(\underline{\mathsf{g}}(x))) \\
9: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}_1(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}_1(x))) & 10: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}_1(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{g}_1(x))) \\
11: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangle(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangle(x))) & 12: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{b})) :\to \mathsf{f}_1(\triangledown(\mathsf{b})) \\
13: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\triangle(x))) :\to \mathsf{f}_1(\triangledown(\triangle(x))) & 14: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{top}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{top}(x))) \\
15: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{h}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{h}(x))) & 16: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{c})) :\to \mathsf{f}_1(\triangledown(\mathsf{c})) \\
17: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\triangledown(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\triangledown(x)))) & 18: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}(x)))) \\
19: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangledown_{\mathsf{f}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangledown_{\mathsf{f}}(x)))) & 20: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\underline{\mathsf{f}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\underline{\mathsf{f}}(x)))) \\
21: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangledown_{\mathsf{g}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangledown_{\mathsf{g}}(x)))) & 22: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\underline{\mathsf{g}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\underline{\mathsf{g}}(x)))) \\
23: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}_1(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}_1(x)))) & 24: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}_1(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{g}_1(x)))) \\
25: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangle(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangle(x)))) & 26: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{b}))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{b}))) \\
27: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\triangle(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\triangle(x)))) & 28: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{top}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{top}(x)))) \\
29: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{h}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{h}(x)))) & 30: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{c}))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{c}))) \\
31: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\triangledown(x))) :\to \mathsf{g}_1(\triangledown(\triangledown(x))) & 32: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{f}(x))) \\
33: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangledown_{\mathsf{f}}(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangledown_{\mathsf{f}}(x))) & 34: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\underline{\mathsf{f}}(x))) :\to \mathsf{g}_1(\triangledown(\underline{\mathsf{f}}(x))) \\
35: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{g}(x))) & 36: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangledown_{\mathsf{g}}(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangledown_{\mathsf{g}}(x))) \\
37: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\underline{\mathsf{g}}(x))) :\to \mathsf{g}_1(\triangledown(\underline{\mathsf{g}}(x))) & 38: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}_1(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{f}_1(x))) \\
39: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}_1(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{g}_1(x))) & 40: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangle(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangle(x))) \\
41: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\triangle(x))) :\to \mathsf{g}_1(\triangledown(\triangle(x))) & 42: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{top}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{top}(x))) \\
43: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{h}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{h}(x))) & 44: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{c})) :\to \mathsf{g}_1(\triangledown(\mathsf{c})) \\
45: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) :\to \blacktriangle(x) & 46: & \underline{\mathsf{g}}(\mathsf{b}) :\to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
47: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) :\to \triangle(x) & 48: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) :\to \triangle(x) \\
49: & \mathsf{f}_1(\triangle(x)) :\to \triangle(\mathsf{f}(x)) & 50: & \mathsf{g}_1(\triangle(x)) :\to \triangle(\mathsf{g}(x)) \\
51: & \mathsf{top}(\triangle(x)) :\to \mathsf{top}(\triangledown(x)) & &
\end{array}\right\}$$

TTT2 fails to handle this system. However, AProVE can prove its innermost termination with in 30.93 seconds. This system name is *ffg.trs*. The result can be found in the appendix.

The result of this technique combines with termination provers are shown in Chapter 5 and the appendix. Lastly, the technique does not guarantee to resolve all ambiguity at overlap positions (i.e. transform to overlay system). However, the optimization version of pattern separation yield better result, details are described in the next section.

4.3 Optimization

In the previous section, we showed that *pattern separation transformation* is capable of supporting innermost termination proof. However, one drawback is termination technique need to handle many additional rules. This section will describe how to further optimize *pattern separation transformation* by using type information. By this optimization, the transformation result will produce less additional rules and may become an overlay system. The optimization derives benefit from the following theorem proposed by Van de Pol [25].

Theorem 11. *A TRS $\mathcal{R}$ is innermost terminating if and only if many-sorted version of the TRS $\mathcal{R}$ is innermost terminating*

Example 34. Let TRS $\mathcal{R}$ consist of the following rules:

$$\mathcal{R} = \left\{ \begin{array}{r} \mathsf{f}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{g}(0) \to \mathsf{h}(\mathsf{g}(1)) \end{array} \right\}$$

As shown in the previous section, the pattern separation version of a TRS $\mathcal{R}$ is

$$\mathcal{S}(\mathcal{R}) = \left\{ \begin{array}{rr} \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \to \mathsf{f}(\mathsf{g}(0)) & \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{f}(\mathsf{g}(\mathsf{h}(x))) \to \mathsf{f}(\mathsf{g}(0)) & \mathsf{f}(\mathsf{g}(1)) \to \mathsf{f}(\mathsf{g}(0)) \\ \mathsf{g}(0) \to \mathsf{h}(\mathsf{g}(1)) & \end{array} \right\}$$

The optimization version of pattern separation $\mathcal{S}_{op}(\mathcal{R})$ will result in the following TRS:

$$\mathcal{S}_{op}(\mathcal{R}) = \left\{ \ \mathsf{f}(\mathsf{g}(1)) \to \mathsf{f}(\mathsf{g}(0)) \quad \mathsf{g}(0) \to \mathsf{h}(\mathsf{g}(1)) \ \right\}$$

The number of rules in pattern separation transformation exposed from the complement pattern process. We reduce the number of rules by considering sort information while performing complement on terms.

Definition 42. Let t be a well-sorted term. The sorted complement set of t is a set of well-sorted terms which an element is not instant of t. The sorted-complement set $\mathcal{C}'(t)$ is defined as follows:

$$\mathcal{C}'(t^\alpha) = \begin{cases} \varnothing & \text{if } t \in \mathcal{V} \\ A_1 \cup \ldots \cup A_n \cup B & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Here, Ai and B stand for the following sets:

$$A_i = \{f(t_1, \ldots, t_{i-1}, c_i, x_{i+1}, \ldots, x_n) \mid c_i \in \mathcal{C}'(t_i)\}$$
$$B = \{g(x_1, \ldots, x_m)^\beta \mid g \in \mathcal{F} \setminus \{f\}, \text{and } \alpha = \beta\}$$

where, $x_1, x_2, \ldots$ are fresh variables.

The *optimization version $\mathcal{S}_{op}(\mathcal{R})$ of pattern separation* is defined in the same way as $\mathcal{S}(\mathcal{R})$ but it uses sorted complement set instead.

Example 35. After performing type introduction on TRS $\mathcal{R}$. The following sorted-signature obtained.

$$\left\{ \begin{array}{ll} \mathsf{0} : \alpha & \mathsf{1} : \alpha \\ \mathsf{g} : \alpha \to \beta & \mathsf{h} : \beta \to \beta \\ \mathsf{f} : \beta \to \gamma & \end{array} \right\}$$

We illustrate computation of $\mathcal{C}'(\mathsf{f}(\mathsf{g}(\mathsf{0})))$. Below we follow notations in Definition 42 to compute the set step by step.

We start with computing $\mathcal{C}'(\mathsf{0})$:

$$\begin{aligned} \mathcal{C}'(\mathsf{0}^{\alpha}) &= \{g(x_1,\ldots,x_m)^{\delta} \mid g \in \mathcal{F},\, \mathsf{0} \neq g \text{ and } \alpha = \delta\} \\ &= \{\mathsf{1}^{\alpha}\} \end{aligned}$$

Then we compute $\mathcal{C}'(\mathsf{g}(\mathsf{0}))$:

$$\begin{aligned} \mathcal{C}'(\mathsf{g}(\mathsf{0}^{\alpha})^{\beta}) &= \{\mathsf{g}(c) \mid c \in \mathcal{C}'(\mathsf{0})\} \cup \{g(x_1,\ldots,x_m)^{\delta} \mid g \in \mathcal{F},\, \mathsf{g} \neq g \text{ and } \beta = \delta\} \\ &= \{\mathsf{g}(c) \mid c \in \mathcal{C}'(\mathsf{0})\} \cup \{\mathsf{h}(x^{\beta})^{\beta}\} \\ &= \{\mathsf{g}(\mathsf{1}^{\alpha})^{\beta}, \mathsf{h}(x^{\beta})^{\beta}\} \end{aligned}$$

Finally, we can compute $\mathcal{C}'(\mathsf{f}(\mathsf{g}(\mathsf{0})))$:

$$\begin{aligned} \mathcal{C}'(\mathsf{f}(\mathsf{g}(\mathsf{0}^{\alpha})^{\beta})\gamma) &= \{\mathsf{f}(c) \mid c \in \mathcal{C}'(\mathsf{g}(\mathsf{0}))\} \cup \{g(x_1,\ldots,x_m)^{\delta} \mid g \in \mathcal{F},\, \mathsf{f} \neq g \text{ and } \gamma = \delta\} \\ &= \{\mathsf{f}(c) \mid c \in \mathcal{C}'(\mathsf{g}(\mathsf{0}))\} \cup \varnothing \\ &= \{\mathsf{f}(\mathsf{g}(\mathsf{1}^{\alpha})^{\beta})^{\gamma}, \mathsf{f}(\mathsf{h}(x^{\beta})^{\beta})^{\gamma}\} \end{aligned}$$

Definition 42 directly effect to the complement pattern which rely on complement process. The new result on complement pattern of left-hand-side of rules in TRS $\mathcal{R}$ are

- $\mathsf{P}_{\mathcal{R}}(\mathsf{f}(\mathsf{g}(x))) = \{\mathsf{f}(\mathsf{g}(\mathsf{1})), \mathsf{f}(\mathsf{h}(x))\}$
- $\mathsf{P}_{\mathcal{R}}(\mathsf{g}(\mathsf{0})) = \{\mathsf{g}(\mathsf{0})\}$

After rewrite with innermost step we finally obtained the system

$$\mathcal{S}_{op}(\mathcal{R}) = \left\{ \begin{array}{ll} \mathsf{f}(\mathsf{g}(\mathsf{1})) \to \mathsf{f}(\mathsf{g}(\mathsf{0})) & \mathsf{g}(\mathsf{0}) \to \mathsf{h}(\mathsf{g}(\mathsf{1})) \end{array} \right\}$$

Here, the term $\mathsf{f}(\mathsf{h}(x))$ disappear because it not rewritable. As mention in the previous section, pattern separation with optimization may result in an overlay system.

Theorem 12. *A left-linear TRS $\mathcal{R}$ is innermost terminating if and only if the TRS $\mathcal{S}_{op}(\mathcal{R})$ is innermost terminating.*

We exploit Theorem 11 and prove Theorem 12 in the same way as Theorem 10.

Example 36 (continued from Example 33)**.** We compute $\mathcal{S}_{op}(\mathcal{R}^T)$ (i.e. the optimized version of pattern separation of the TRS $\mathcal{R}^T$), which is resulting as:

$$\left\{\begin{array}{lrlr}
1: & \triangledown(\mathsf{f}(x)) \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}(x))) \to \mathsf{f}_1(\triangledown(\mathsf{g}(x))) & 4: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{b})) \to \mathsf{f}_1(\triangledown(\mathsf{b})) \\
5: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{h}(x))) \to \mathsf{f}_1(\triangledown(\mathsf{h}(x))) & 6: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{c})) \to \mathsf{f}_1(\triangledown(\mathsf{c})) \\
7: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}(x)))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}(x)))) & 8: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{b}))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{b}))) \\
9: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{h}(x)))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{h}(x)))) & 10: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{c}))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{c}))) \\
11: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{f}(x))) & 12: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{g}(x))) \\
13: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{h}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{h}(x))) & 14: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{c})) \to \mathsf{g}_1(\triangledown(\mathsf{c})) \\
15: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & 16: & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
17: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) \to \triangle(x) & 18: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) \to \triangle(x) \\
19: & \mathsf{f}_1(\triangle(x)) \to \triangle(\mathsf{f}(x)) & 20: & \mathsf{g}_1(\triangle(x)) \to \triangle(\mathsf{g}(x)) \\
21: & \mathsf{top}(\triangle(x)) \to \mathsf{top}(\triangledown(x)) & &
\end{array}\right\}$$

Both TTT2 and AProVE can prove its innermost termination with in 7.95 seconds and 5.63 seconds respectively. The system is about half in a number of rules compared to the normal pattern separation version. The result can be found in the appendix.

Apply this technique to the transformed system mostly result as an overlay system. In the next chapter, we demonstrate pattern separation on transformed systems.

Chapter 5

Experiments

In this chapter, we illustrate pattern separation technique. Unfortunately, type-based reachability analysis cannot make a difference in a number of edges in dependency graphs of our experiment systems. For the pattern separation technique, we divide the experiment into two parts. In the first part, we will experiment on transformed systems by comparing a number of rules in transformed systems between standard and optimized version. In the second part, we use original systems and their transformed version as inputs for termination tools then compare the result provided by tools. All experiments were performed on a PC with a CPU Intel Core i7-7500U with 16GB RAM for all experiments.

5.1 Pattern Separation

We implemented both versions of the pattern separation techniques in OCaml. The program consists of about 1000 lines of OCaml code. Input TRSs are generated from Thiemann's transformation [23] which take input TRSs from 279 outermost termination problems in The Termination Problems Data Base (TPDB) [1].

Given an input TRS, the optimized version of the algorithm first performs type-introduction to the TRS, then pattern separation. Pattern separation can be applied for each rule therefore, in the case of non left-linear TRS, non left-linear rules will be ignored and just transformed the rest.

Here we show the difference between both versions in term of a number of rules in transformed systems. We limit the time for transformation to 60 seconds. The system which transforms longer than 60 seconds considered as timeout in our experiment.

version	`Separation (Basic)`	`Separation (Optimized)`
`Average Rules`	257	92
`Timeout (60s)`	21	6

NOTE 1: We do not include systems which are timeout in the calculation.
NOTE 2: Full table of the result shown in the appendix.

Example 37. Consider the running example TRS $\mathcal{R}$ to illustrate the difference between basic version and optimized version.

$$\left\{\begin{array}{ll} 1: & \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x \\ 2: & \mathsf{g}(\mathsf{b}) \to \mathsf{f}(\mathsf{g}(\mathsf{b})) \end{array}\right\}$$

The following TRS $\mathcal{R}^T$ is resulting from The Thiemann's transformation [23].

$$\left\{\begin{array}{llll}
1: & \triangledown(\mathsf{f}(x)) \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) \to \mathsf{f}_1(\triangledown(x)) & 4: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \to \mathsf{g}_1(\triangledown(x)) \\
5: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & 6: & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
7: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) \to \triangle(x) & 8: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) \to \triangle(x) \\
9: & \mathsf{f}_1(\triangle(x)) \to \triangle(\mathsf{f}(x)) & 10: & \mathsf{g}_1(\triangle(x)) \to \triangle(\mathsf{g}(x)) \\
11: & \mathsf{top}(\triangle(x)) \to \mathsf{top}(\triangledown(x)) & &
\end{array}\right\}$$

Applying the pattern separation on the TRS $\mathcal{R}^T$ results as:

$$\left\{\begin{array}{llll}
1: & \triangledown(\mathsf{f}(x)) :\to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) :\to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\triangledown(x))) :\to \mathsf{f}_1(\triangledown(\triangledown(x))) & 4: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangledown_{\mathsf{f}}(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangledown_{\mathsf{f}}(x))) \\
5: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\underline{\mathsf{f}}(x))) :\to \mathsf{f}_1(\triangledown(\underline{\mathsf{f}}(x))) & 6: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{g}(x))) \\
7: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangledown_{\mathsf{g}}(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangledown_{\mathsf{g}}(x))) & 8: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\underline{\mathsf{g}}(x))) :\to \mathsf{f}_1(\triangledown(\underline{\mathsf{g}}(x))) \\
9: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}_1(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}_1(x))) & 10: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}_1(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{g}_1(x))) \\
11: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\blacktriangle(x))) :\to \mathsf{f}_1(\triangledown(\blacktriangle(x))) & 12: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{b})) :\to \mathsf{f}_1(\triangledown(\mathsf{b})) \\
13: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\triangle(x))) :\to \mathsf{f}_1(\triangledown(\triangle(x))) & 14: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{top}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{top}(x))) \\
15: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{h}(x))) :\to \mathsf{f}_1(\triangledown(\mathsf{h}(x))) & 16: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{c})) :\to \mathsf{f}_1(\triangledown(\mathsf{c})) \\
17: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\triangledown(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\triangledown(x)))) & 18: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}(x)))) \\
19: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangledown_{\mathsf{f}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangledown_{\mathsf{f}}(x)))) & 20: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\underline{\mathsf{f}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\underline{\mathsf{f}}(x)))) \\
21: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangledown_{\mathsf{g}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangledown_{\mathsf{g}}(x)))) & 22: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\underline{\mathsf{g}}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\underline{\mathsf{g}}(x)))) \\
23: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}_1(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}_1(x)))) & 24: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}_1(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{g}_1(x)))) \\
25: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\blacktriangle(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\blacktriangle(x)))) & 26: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{b}))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{b}))) \\
27: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\triangle(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\triangle(x)))) & 28: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{top}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{top}(x)))) \\
29: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{h}(x)))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{h}(x)))) & 30: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{c}))) :\to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{c}))) \\
31: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\triangledown(x))) :\to \mathsf{g}_1(\triangledown(\triangledown(x))) & 32: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{f}(x))) \\
33: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangledown_{\mathsf{f}}(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangledown_{\mathsf{f}}(x))) & 34: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\underline{\mathsf{f}}(x))) :\to \mathsf{g}_1(\triangledown(\underline{\mathsf{f}}(x))) \\
35: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{g}(x))) & 36: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangledown_{\mathsf{g}}(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangledown_{\mathsf{g}}(x))) \\
37: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\underline{\mathsf{g}}(x))) :\to \mathsf{g}_1(\triangledown(\underline{\mathsf{g}}(x))) & 38: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}_1(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{f}_1(x))) \\
39: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}_1(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{g}_1(x))) & 40: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\blacktriangle(x))) :\to \mathsf{g}_1(\triangledown(\blacktriangle(x))) \\
41: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\triangle(x))) :\to \mathsf{g}_1(\triangledown(\triangle(x))) & 42: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{top}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{top}(x))) \\
43: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{h}(x))) :\to \mathsf{g}_1(\triangledown(\mathsf{h}(x))) & 44: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{c})) :\to \mathsf{g}_1(\triangledown(\mathsf{c})) \\
45: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) :\to \blacktriangle(x) & 46: & \underline{\mathsf{g}}(\mathsf{b}) :\to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
47: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) :\to \triangle(x) & 48: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) :\to \triangle(x) \\
49: & \mathsf{f}_1(\triangle(x)) :\to \triangle(\mathsf{f}(x)) & 50: & \mathsf{g}_1(\triangle(x)) :\to \triangle(\mathsf{g}(x)) \\
51: & \mathsf{top}(\triangle(x)) :\to \mathsf{top}(\triangledown(x)) & &
\end{array}\right\}$$

Applying the optimized version of pattern separation on the TRS $\mathcal{R}^T$ results as:

$$\left\{\begin{array}{lrlr}
1: & \triangledown(\mathsf{f}(x)) \to \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(x)) & 2: & \triangledown(\mathsf{g}(x)) \to \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(x)) \\
3: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{g}(x))) \to \mathsf{f}_1(\triangledown(\mathsf{g}(x))) & 4: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{b})) \to \mathsf{f}_1(\triangledown(\mathsf{b})) \\
5: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{h}(x))) \to \mathsf{f}_1(\triangledown(\mathsf{h}(x))) & 6: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{c})) \to \mathsf{f}_1(\triangledown(\mathsf{c})) \\
7: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{f}(x)))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{f}(x)))) & 8: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{b}))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{b}))) \\
9: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{h}(x)))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{h}(x)))) & 10: & \blacktriangledown_{\mathsf{f}}(\underline{\mathsf{f}}(\mathsf{f}(\mathsf{c}))) \to \mathsf{f}_1(\triangledown(\mathsf{f}(\mathsf{c}))) \\
11: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{f}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{f}(x))) & 12: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{g}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{g}(x))) \\
13: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{h}(x))) \to \mathsf{g}_1(\triangledown(\mathsf{h}(x))) & 14: & \blacktriangledown_{\mathsf{g}}(\underline{\mathsf{g}}(\mathsf{c})) \to \mathsf{g}_1(\triangledown(\mathsf{c})) \\
15: & \underline{\mathsf{f}}(\mathsf{f}(\mathsf{g}(x))) \to \blacktriangle(x) & 16: & \underline{\mathsf{g}}(\mathsf{b}) \to \blacktriangle(\mathsf{f}(\mathsf{g}(\mathsf{b}))) \\
17: & \blacktriangledown_{\mathsf{f}}(\blacktriangle(x)) \to \triangle(x) & 18: & \blacktriangledown_{\mathsf{g}}(\blacktriangle(x)) \to \triangle(x) \\
19: & \mathsf{f}_1(\triangle(x)) \to \triangle(\mathsf{f}(x)) & 20: & \mathsf{g}_1(\triangle(x)) \to \triangle(\mathsf{g}(x)) \\
21: & \mathsf{top}(\triangle(x)) \to \mathsf{top}(\triangledown(x)) & &
\end{array}\right\}$$

This example picks up to represent the average difference between basic and optimized version. However, the difference between two version may vary. In this experiment, the highest difference in a number of rules is 2314 rules from *LISTUTILITIES_nosorts-noand_L.trs* and the highest rules reduction from basic version to optimized version is 92 percent from *4.06.trs* which reduce from 1242 rules to 98 rules.

5.2 Termination Experiments

We test the innermost termination property of transformed systems compare with original systems on AProVE [9] and TTT2 [16] which is the 1st and the 2nd winners of TermCOMP 2015 [1] respectively.

We limit the run-time to 60 seconds for each system. A system that takes more than 60 seconds to proved or disproved, considered to timeout in our experiments. The result is shown in Table 5.2.

AProVE	Proved	Disproved	Timeout
$\mathcal{R}^T$	0	86	193
$\mathcal{S}(\mathcal{R}^T)$	13	73	193
$\mathcal{S}_{op}(\mathcal{R}^T)$	24	84	171
TTT2	Proved	Disproved	Timeout
$\mathcal{R}^T$	13	74	192
$\mathcal{S}(\mathcal{R}^T)$	20	67	192
$\mathcal{S}_{op}(\mathcal{R}^T)$	30	81	168

NOTE: Full table of the result shown in the appendix.

Lastly, we extract 120 systems which are an overlay and right-linear systems which innermost termination is equivalent to full termination [20] and attempting to proved full termination by NaTT [26] instead.

[1] Termination Problem Database, http://nfa.imn.htwk-leipzig.de/termcomp-2015/competitions/4

NaTT	Proved	Disproved	Timeout
$\mathcal{S}_{op}(\mathcal{R}^T)$	13	48	59

From above results, consider the TRS $\mathcal{R}$ in Example 37. The TRS $\mathcal{R}^T$ cannot handle by both AProVE and TTT2. However, AProVE can prove the innermost termination of TRS $\mathcal{S}(\mathcal{R}^T)$ but TTT2 still cannot. Lastly, the innermost termination of TRS $\mathcal{S}_{op}(\mathcal{R}^T)$ can be proved by both AProVE and TTT2.

Chapter 6

Conclusion

We presented two new techniques on innermost termination analysis, which are *type-based reachability analysis* and *pattern separation*. Our primary target problems are transformed systems especially innermost termination problems obtained from Thiemann's transformation algorithm which is considered as a representation of difficult systems. Type information and complement pattern are exploited to analyze problems. We conclude the thesis by starting with related work.

6.1 Related Work

Type-based reachability analysis is an approximation technique for computing reachability. Another famous approximation technique for computing reachability is $ICAP$ function. However, an approach of those techniques is difference and orthogonal to each other.

The $ICAP$ function [11] is an approximation function which mainly relies on unification. The solid point of this technique is that it can be applied to every system and well performed in the most cases. However, the analysis only inspects from the start term. On the other hand, type-based reachability analysis mainly employs type information to analyze a term and map it to a set. The advantage of this technique is it analyze through rewrite steps. Even though the missing step is hidden in nested rewrite steps, it may detect, and the analysis concludes unreachable. The big drawback of this techniques is that it will be completely useless if after performing type introduction the single-sorted system is obtained. We strongly anticipate that this technique can be used combined with the $ICAP$ technique.

As, reachability analysis is a part of many techniques such as dependency graph, freezing, usable rules and usable replacement maps. The improvement of approximation function will effect many techniques. We anticipate that unification approach and type approach of reachability analysis can be combined. The investigation on how to combine those two techniques are a consideration as future work.

Pattern separation is a transformation technique for innermost termination. We compare this technique in two aspect transformation technique and innermost specific technique.

In transformation techniques aspect, we compare with Narrowing, Instantiation, and Rewriting techniques [22] which are implemented in the AProVE. All of the mention transformation techniques involve unification to compute the instantiation part. On top of that pattern separation uses complement pattern of overlap part to instantiate a rule. In our experiments, considering the proving result with TTT2 on the optimized version of pattern separation of *assoc_f_rhs.trs* and the proving result with AProVE on the original version of *assoc_f_rhs.trs* are failed. However, the proving result with AProVE on the optimized version of pattern separation of *assoc_f_rhs.trs* is succeeded. Since TTT2 does not include Narrowing technique, but the success proof relies on both pattern separation and Narrowing. Therefore, *assoc_f_rhs.trs* is our witness to show the combining power of pattern separation and narrowing technique.

For the innermost specific technique aspect, we compare pattern separation and innermost recursive path ordering (iRPO) [7]. The iRPO is an ordering based on RPO but, specifically designed for innermost termination. This technique is completely different from pattern separation because iRPO directly proves the innermost termination of a TRSs and pattern separation is just a transformation. We do not have experiments on their combinations two technique since there are no existing tools that include iRPO technique. Its evaluation is future work.

6.2 Limitations

The major limitation of our technique is that we heavily rely on type information. We cannot use type based reachability analysis on a single-sorted TRS at all. In the case of pattern separation, the basic version has significantly low performance compared to the optimized version that requires type information. Another bottleneck for pattern separation is complement algorithm which requires to performing on a linear term. Therefore, our technique is shining on a system which yields a nice type information after performing type introduction such as a system from the Thiemanns algorithm. However, it may be harmful to a system which yields single-sorted system after performing type introduction.

As future work, we plan to investigate conditions of type information which is suited to our techniques. Since the performance of our techniques heavily depends on type information. Another direction is to improve our techniques such as relax the linear term condition for the complement algorithm and combine with existing techniques. Also, we consider that simulation of rewrite step is excessive for proving termination. The idea of an unavoidable set [19] which focuses on the termination property may apply to this technique.

Acknowledgements

I am most grateful to my supervisor Prof. Nao Hirokawa for his guidance and valuable advice. His professional guidance has allowed me to progress my research. I would like to thank Prof. Mizuhito Ogawa and Prof. Tachio Terauchi for their importance comments on my research. I also thank Prof. Hajime Ishihara for a beneficial experience on the minor research project. Special thanks go to members in Hirokawa, Ogawa, and Terauchi laboratories for their comments and supports.

Appendix A

Experimental Data

Transformation Experiments

This section reports number of rewrite rules of a particular system in our problem set.

- $\mathcal{S}(\mathcal{R}^T)$: The number of rules in the system resulting from basic version of pattern separation.
- $\mathcal{S}_{op}(\mathcal{R}^T)$: The number of rules in the system resulting from optimized version of pattern separation.
- Difference : The difference in number of rules of both version of pattern separation.
- Percentage : Percentage of reducing the number of rules from basic version to optimized version.

`Problem Name`	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
`001`	54	22	32	59.2593
`003`	116	52	64	55.1724
`2.05`	44	20	24	54.5455
`4.06.trs`	1242	98	1144	92.1095
`4.12a#`	144	54	90	62.5
`4.13#`	21	21	0	0
`4.14#`	131	47	84	64.1221
`4.15#`	409	120	289	70.6601
`4.16#`	131	47	84	64.1221
`4.17#`	193	63	130	67.3575
`4.18#`	40	16	24	60
`4.2#`	193	63	130	67.3575
`4.3#`	16	16	0	0
`4.34`	18	10	8	44.4444
`4.4#`	29	17	12	41.3793
`4.40`	82	34	48	58.5366

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
4.49	40	13	27	67.5
4.54	21	12	9	42.8571
4.7#	103	43	60	58.2524
afbg	300	120	180	60
append-wrong	224	86	138	61.6071
assoc_c_rhs	296	40	256	86.4865
assoc_f_rhs	296	40	256	86.4865
bintree	296	40	256	86.4865
cariboo_add1	30	18	12	40
cariboo_add2	29	18	11	37.931
cariboo_add2a	28	17	11	39.2857
cariboo_add3	25	25	0	0
cariboo_len3	76	32	44	57.8947
cariboo_nl_1	60	27	33	55
cariboo_nl_2	63	33	30	47.619
cariboo_nl_3	68	32	36	52.9412
cariboo_nl_4	15	15	0	0
cariboo_nl_5	12	12	0	0
cariboo_nl_6	25	25	0	
cime4	615	216	399	64.878
countbin	71	32	39	54.9296
countter	110	46	64	58.1818
dupl_rhs	91	39	52	57.1429
even	69	29	40	57.971
ex0	55	22	33	60
ex1	171	61	110	64.3275
Ex14_AEGL02	41	27	14	34.1463
Ex14_AEGL02_FR	79	57	22	27.8481
Ex14_AEGL02_L	12	12	0	0
Ex14_AEGL02_Z	70	50	20	28.5714
Ex14_Luc06_GM	67	47	20	29.8507
Ex14_Luc06_L	31	20	11	35.4839
Ex14_Luc06_Z	68	36	32	47.0588
Ex15_Luc06_L	2	2	0	0
Ex15_Luc98	445	159	286	64.2697
Ex15_Luc98_L	221	102	119	53.8462
Ex16_Luc06_L	6	6	0	0
ex1m	37	19	18	48.6486
Ex1_2_AEL03	811	283	528	65.1048
Ex1_2_AEL03_L	605	221	384	63.4711
Ex1_2_Luc02c	49	25	24	48.9796

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
Ex1_2_Luc02c_L	25	15	10	40
Ex1_GL02a	178	83	95	53.3708
Ex1_GL02a_L	136	71	65	47.7941
Ex1_GL02a_Z	65	65	0	0
Ex1_GM03	251	91	160	63.745
Ex1_GM03_L	181	73	108	59.6685
Ex1_GM03_Z	320	128	192	60
Ex1_GM99	93	39	54	58.0645
Ex1_GM99_GM	44	30	14	31.8182
Ex1_GM99_L	7	7	0	0
Ex1_GM99_Z	133	67	66	49.6241
Ex1_Luc02b	269	99	170	63.197
Ex1_Luc02b_L	257	97	160	62.2568
Ex1_Luc04b_FR	102	66	36	35.2941
Ex1_Luc04b_L	95	50	45	47.3684
Ex1_Luc04b_Z	92	56	36	39.1304
Ex1_Zan97	24	15	9	37.5
Ex1_Zan97_L	4	4	0	0
Ex1_Zan97_Z	39	28	11	28.2051
ex2	12	12	0	0
Ex24_GM04	32	20	12	37.5
Ex24_GM04_GM	55	38	17	30.9091
Ex24_GM04_L	19	12	7	36.8421
Ex24_Luc06_GM	42	28	14	33.3333
Ex24_Luc06_Z	126	60	66	52.381
Ex26_Luc03b	309	117	192	62.1359
ex2m	44	22	22	50
Ex2_8_1ConstSubstFix	82	34	48	58.5366
Ex2_Luc03b	226	86	140	61.9469
ex3	84	36	48	57.1429
ex3m	43	23	20	46.5116
Ex3_12_Luc96a	129	51	78	60.4651
Ex3_12_Luc96a_L	121	49	72	59.5041
Ex3_2_Luc97	357	127	230	64.4258
Ex3_2_Luc97_L	261	108	153	58.6207
Ex3_3_25_Bor03	1536	168	1368	89.0625
Ex3_3_25_Bor03_Z	59	59	0	0
ex4	51	21	30	58.8235
Ex49_GM04_L	56	34	22	39.2857
ex4m	26	16	10	38.4615
Ex4_4_Luc96b	32	14	18	56.25
Ex4_4_Luc96b_FR	26	26	0	0

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
Ex4_4_Luc96b_Z	20	20	0	0
Ex4_7_15_Bor03	91	39	52	57.1429
Ex4_7_56_Bor03	91	39	52	57.1429
Ex4_7_56_Bor03_L	85	37	48	56.4706
Ex4_7_77_Bor03	24	15	9	37.5
Ex4_7_77_Bor03_L	21	13	8	38.0952
Ex4_DLMMU04_FR	877	360	517	58.951
Ex4_DLMMU04_L	460	200	260	56.5217
Ex4_DLMMU04_Z	877	360	517	58.951
Ex4_Zan97	180	68	112	62.2222
Ex4_Zan97_L	171	66	105	61.4035
ex5	26	16	10	38.4615
ex5m	11	11	0	0
Ex5_7_Luc97	923	311	612	66.3055
Ex5_7_Luc97_L	763	287	476	62.3853
Ex5_DLMMU04_FR	138	105	33	23.913
Ex5_DLMMU04_Z	134	101	33	24.6269
Ex5_Zan97	60	27	33	55
Ex5_Zan97_L	41	21	20	48.7805
ex6	40	20	20	50
ex6m	18	18	0	0
Ex6_15_AEL02	1542	510	1032	66.9261
Ex6_15_AEL02_L	1260	468	792	62.8571
Ex6_15_AEL02_Z	1224	482	742	60.6209
Ex6_9_Luc02c	59	31	28	47.4576
Ex6_9_Luc02c_L	49	25	24	48.9796
Ex6_GM04	20	12	8	40
Ex6_GM04_Z	40	28	12	30
Ex6_Luc98	95	43	52	54.7368
ex7	71	32	39	54.9296
Ex7_BLR02	423	147	276	65.2482
Ex7_BLR02_L	409	145	264	64.5477
ex8	21	12	9	42.8571
Ex8_BLR02	248	88	160	64.5161
Ex8_BLR02_L	238	86	152	63.8655
ex9	49	19	30	61.2245
Ex9_BLR02	220	84	136	61.8182
Ex9_Luc04_GM	44	30	14	31.8182
Ex9_Luc04_L	34	18	16	47.0588
Ex9_Luc04_Z	124	58	66	53.2258
Ex9_Luc06_L	6	6	0	0
Ex9_Luc06_Z	21	21	0	0

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
ExAppendixB_AEL03	861	301	560	65.0407
ExAppendixB_AEL03_L	825	297	528	64
ExConc_Zan97	9	9	0	0
ExConc_Zan97_Z	19	19	0	0
ExIntrod_GM01	145	69	76	52.4138
ExIntrod_GM01_FR	120	78	42	35
ExIntrod_GM01_L	139	67	72	51.7986
ExIntrod_GM01_Z	120	78	42	35
ExIntrod_GM04	141	65	76	53.9007
ExIntrod_GM04_FR	131	85	46	35.1145
ExIntrod_GM04_L	115	51	64	55.6522
ExIntrod_GM04_Z	131	85	46	35.1145
ExIntrod_GM99	553	205	348	62.9295
ExIntrod_GM99_L	435	175	260	59.7701
ExIntrod_GM99_Z	385	169	216	56.1039
ExIntrod_Zan97	360	126	234	65
ExIntrod_Zan97_L	272	104	168	61.7647
ExIntrod_Zan97_Z	363	146	217	59.7796
ExSec11_1_Luc02a	423	153	270	63.8298
ExSec4_2_DLMMU04_L	538	196	342	63.5688
ffb_SL	26	12	14	53.8462
ffg	51	21	30	58.8235
fg	37	17	20	54.0541
from_one	44	22	22	50
from_one_a	44	22	22	50
from_three	108	42	66	61.1111
f_2_1	40	20	20	50
f_2_2	40	20	20	50
f_5	50	22	28	56
f_5_1	85	35	50	58.8235
f_5_2	85	35	50	58.8235
gfb	54	24	30	55.5556
gkg	28	17	11	39.2857
g_2_const	22	13	9	40.9091
g_2_f_var	33	15	18	54.5455
Hamming	28	28	0	0
inn_out	51	21	30	58.8235
jwno1	34	16	18	52.9412
jwno4	34	16	18	52.9412
jwno6	34	16	18	52.9412
jwno9	50	18	32	64
LengthOfFiniteLists_complete_L	1249	484	765	61.249

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
LengthOfFiniteLists_nokinds-noand_FR	1447	535	912	63.027
LengthOfFiniteLists_nokinds-noand_L	614	254	360	58.6319
LengthOfFiniteLists_nokinds-noand_Z	1447	535	912	63.027
LengthOfFiniteLists_nokinds_FR	325	189	136	41.8462
LengthOfFiniteLists_nokinds_L	110	65	45	40.9091
LengthOfFiniteLists_nokinds_Z	325	189	136	41.8462
LengthOfFiniteLists_nosorts-noand_L	93	48	45	48.3871
LengthOfFiniteLists_nosorts-noand_Z	189	89	100	52.9101
LengthOfFiniteLists_nosorts_GM	65	47	18	27.6923
LengthOfFiniteLists_nosorts_L	56	32	24	42.8571
LengthOfFiniteLists_nosorts_noand_GM	83	60	23	27.7108
LengthOfFiniteLists_nosorts_Z	105	57	48	45.7143
LISTUTILITIES_nokinds_L	2147	727	1420	66.1388
LISTUTILITIES_nosorts-noand_L	3318	1004	2314	69.7408
LISTUTILITIES_nosorts_L	753	269	484	64.2762
morse	78	39	39	50
muladd	69	29	40	57.971
MYNAT_complete_L	1311	427	884	67.4294
MYNAT_nokinds-noand_L	1019	329	690	67.7134
MYNAT_nokinds_L	394	142	252	63.9594
MYNAT_nosorts-noand_L	307	107	200	65.1466
MYNAT_nosorts_L	130	50	80	61.5385
n001	71	27	44	61.9718
n002	5	5	0	0
n003	5	5	0	0
n004	15	8	7	46.6667
n005	5	5	0	0
n006	7	7	0	0
n007	10	10	0	0
n008	17	10	7	41.1765
non-lin1	8	8	0	0
non-lin2	15	15	0	0
non-lin3	15	15	0	0
nonterm	34	21	13	38.2353
nonTermF	10	10	0	0
odd	69	29	40	57.971
outermost_gr	250	26	224	89.6
OvConsOS_complete_L	2576	896	1680	65.2174
OvConsOS_nokinds-noand_L	1783	623	1160	65.0589
OvConsOS_nokinds_L	407	182	225	55.2826
OvConsOS_nosorts-noand_FR	1085	354	731	67.3733
OvConsOS_nosorts-noand_L	454	174	280	61.674

Problem Name	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$	Difference	Percentage
OvConsOS_nosorts-noand_Z	1085	354	731	67.3733
OvConsOS_nosorts_FR	141	78	63	44.6809
OvConsOS_nosorts_GM	85	62	23	27.0588
OvConsOS_nosorts_L	174	78	96	55.1724
OvConsOS_nosorts_noand_GM	187	132	55	29.4118
OvConsOS_nosorts_Z	141	78	63	44.6809
PALINDROME_nokinds_L	884	213	671	75.905
PALINDROME_nosorts_L	797	83	714	89.5859
patterns1	146	56	90	61.6438
patterns2	161	61	100	62.1118
PEANO_complete-noand_L	1845	585	1260	68.2927
PEANO_complete_L	525	195	330	62.8571
PEANO_nokinds-noand_L	346	130	216	62.4277
PEANO_nokinds_L	157	67	90	57.3248
PEANO_nosorts-noand_L	103	43	60	58.2524
PEANO_nosorts_L	65	29	36	55.3846
round_nonterm	49	25	24	48.9796
termMonTypes	145	65	80	55.1724
test10	1081	95	986	91.2118
test75	115	43	72	62.6087
test76	272	85	187	68.75
test77	309	100	209	67.6375
test9	194	66	128	65.9794
toyama_out	115	43	72	62.6087
toyama_stop	65	29	36	55.3846
toyama_stop2	981	117	864	88.0734
ttt1	364	124	240	65.9341
ttt2	29	13	16	55.1724
TypeEx5	69	29	40	57.971
yoyo_2	75	33	42	56
yoyo_3	95	39	56	58.9474
yoyo_3a	67	31	36	53.7313
yoyo_3b	81	33	48	59.2593

Termination Experiments via TTT2

This section report the result from TTT2. The report format is "result : time".

- Y : The system can be proved.
- N : The system can be disproved.
- M : The system cannot proved or disproved.
- T : The system cannot proved or disproved with in time limit 60s.
- E : The system cannot transformed with in time limit 60s.

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
001	M:29.36	M:29.25	M:29.32
2.05	M:29.33	M:29.32	M:29.34
003	M:29.33	M:29.33	M:29.35
4.2#	M:29.36	M:29.34	M:29.30
4.3#	M:29.41	M:29.32	M:29.33
4.4#	M:29.44	M:29.32	M:29.38
4.06	M:29.34	M:29.50	M:29.31
4.7#	M:29.33	M:29.32	M:29.39
4.12a#	M:29.34	M:29.32	M:29.32
4.13#	M:29.37	M:29.36	M:29.35
4.14#	M:29.34	M:29.33	M:29.33
4.15#	M:29.34	M:29.35	M:29.32
4.16#	N:1.18	M:29.30	N:4.09
4.17#	M:29.35	M:29.37	M:29.37
4.18#	M:29.72	M:29.38	M:29.38
4.34	M:29.82	M:29.32	M:29.32
4.40	M:29.32	M:29.28	M:29.30
4.49	M:29.37	M:29.32	M:29.31
4.54	M:29.38	M:29.32	M:29.34
Ex1_2_AEL03	M:29.33	M:29.47	M:29.37
Ex1_2_AEL03_L	N:0.19	N:0.26	N:0.21
Ex1_2_Luc02c	M:29.34	M:29.30	M:29.32
Ex1_2_Luc02c_L	N:0.18	N:0.17	N:0.17
Ex1_GL02a	M:29.29	M:29.30	M:29.27
Ex1_GL02a_L	N:5.26	N:5.40	N:5.55
Ex1_GL02a_Z	M:29.29	M:29.34	M:29.31
Ex1_GM03	M:29.27	M:29.32	M:29.36
Ex1_GM03_L	N:0.19	N:0.19	N:0.17
Ex1_GM03_Z	M:29.36	M:29.35	M:29.29
Ex1_GM99	M:29.43	M:29.33	M:29.32

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
Ex1_GM99_GM	M:29.32	M:29.32	M:29.31
Ex1_GM99_L	N:5.44	N:5.22	N:5.22
Ex1_GM99_Z	M:29.43	M:29.29	M:29.26
Ex1_GM99_iGM	M:29.29	E:0.17	M:29.47
Ex1_Luc02b	M:29.36	M:29.32	M:29.28
Ex1_Luc02b_L	N:0.18	N:0.21	N:0.18
Ex1_Luc04b_FR	M:29.29	M:29.30	Y:14.32
Ex1_Luc04b_L	N:0.18	N:0.18	N:0.19
Ex1_Luc04b_Z	M:29.31	M:29.25	M:29.28
Ex1_Zan97	M:29.33	N:5.69	N:5.79
Ex1_Zan97_L	N:5.22	N:5.20	N:5.22
Ex1_Zan97_Z	M:29.36	M:29.28	N:17.85
Ex2_8_1ConstSubstFix	M:29.30	M:29.28	M:29.32
Ex2_Luc03b	M:29.32	M:29.33	M:29.31
Ex3_2_Luc97	M:29.38	M:29.38	M:29.30
Ex3_2_Luc97_L	N:0.18	N:0.23	N:0.20
Ex3_3_25_Bor03	M:29.36	M:29.62	M:29.32
Ex3_3_25_Bor03_Z	M:29.36	M:29.32	M:29.34
Ex3_12_Luc96a	M:29.34	M:29.27	M:29.31
Ex3_12_Luc96a_L	N:0.19	N:0.19	N:0.17
Ex4_4_Luc96b	M:29.39	M:29.19	M:29.28
Ex4_4_Luc96b_FR	M:29.30	Y:3.63	Y:4.40
Ex4_4_Luc96b_Z	M:29.36	M:29.32	M:29.31
Ex4_7_15_Bor03	M:29.34	M:29.33	M:29.39
Ex4_7_37_Bor03	M:29.34	E:0.15	M:29.40
Ex4_7_37_Bor03_L	N:0.18	E:0.15	N:0.26
Ex4_7_56_Bor03	M:29.33	M:29.28	M:29.24
Ex4_7_56_Bor03_L	N:0.18	N:0.18	N:0.18
Ex4_7_77_Bor03	M:29.34	M:29.29	M:29.24
Ex4_7_77_Bor03_L	N:0.18	N:0.22	N:0.18
Ex4_DLMMU04_FR	M:29.39	M:29.38	M:29.37
Ex4_DLMMU04_L	N:0.22	N:0.21	N:0.20
Ex4_DLMMU04_Z	M:29.41	M:29.40	M:29.38
Ex4_Zan97	M:29.32	M:29.32	M:29.35
Ex4_Zan97_L	N:0.17	N:0.19	N:0.18
Ex5_7_Luc97	M:29.34	M:29.34	M:29.36
Ex5_7_Luc97_L	N:0.21	N:0.27	N:0.21
Ex5_DLMMU04_FR	M:29.39	M:29.30	M:29.30
Ex5_DLMMU04_L	N:0.17	E:0.15	N:0.23
Ex5_DLMMU04_Z	M:29.31	M:29.28	M:29.27
Ex5_Zan97	M:29.33	M:29.31	M:29.32
Ex5_Zan97_L	N:0.17	N:0.17	N:0.18

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
Ex6_9_Luc02c	M:29.32	M:29.31	M:29.32
Ex6_9_Luc02c_L	N:0.21	N:0.18	N:0.17
Ex6_15_AEL02	M:29.32	M:29.51	M:29.35
Ex6_15_AEL02_L	N:0.17	N:0.38	N:0.23
Ex6_15_AEL02_Z	M:29.40	M:29.48	M:29.37
Ex6_GM04	M:29.36	M:29.26	M:29.33
Ex6_GM04_Z	M:29.37	M:29.35	M:29.30
Ex6_Luc98	M:29.37	M:29.28	M:29.31
Ex7_BLR02	M:29.26	M:29.36	M:29.31
Ex7_BLR02_L	N:0.21	N:0.24	N:0.19
Ex8_BLR02	M:29.33	M:29.31	M:29.28
Ex8_BLR02_L	N:0.18	N:0.21	N:0.19
Ex9_BLR02	M:29.36	M:29.34	M:29.31
Ex9_Luc04_GM	M:29.41	M:29.33	M:29.29
Ex9_Luc04_L	N:0.68	N:3.02	N:0.81
Ex9_Luc04_Z	M:29.40	M:29.32	M:29.36
Ex9_Luc06_L	N:5.47	N:5.40	N:5.21
Ex9_Luc06_Z	M:29.32	M:29.31	M:29.30
Ex14_AEGL02	M:29.32	M:29.30	M:29.34
Ex14_AEGL02_FR	M:29.36	M:29.30	M:29.31
Ex14_AEGL02_L	M:30.68	M:29.31	M:29.34
Ex14_AEGL02_Z	M:29.35	M:29.32	M:29.31
Ex14_Luc06_GM	M:29.30	M:29.32	M:29.37
Ex14_Luc06_L	M:29.33	N:5.95	N:5.88
Ex14_Luc06_Z	M:29.33	M:29.30	M:29.32
Ex15_Luc06_L	N:3.81	N:4.08	N:3.77
Ex15_Luc98	M:29.31	M:29.37	M:29.33
Ex15_Luc98_L	N:0.19	N:0.20	N:0.18
Ex16_Luc06_L	N:5.47	N:5.21	N:5.41
Ex24_GM04	M:29.43	M:29.19	M:29.30
Ex24_GM04_GM	M:29.36	M:29.33	M:29.33
Ex24_GM04_L	N:5.27	N:5.43	N:5.23
Ex24_Luc06_GM	M:29.32	M:29.33	Y:3.90
Ex24_Luc06_Z	M:29.34	M:29.34	M:29.36
Ex24_Luc06_iGM	M:29.38	E:0.17	M:29.42
Ex26_Luc03b	M:29.39	M:29.36	M:29.31
Ex49_GM04_L	N:0.21	N:0.17	N:0.17
ExAppendixB_AEL03	M:29.40	M:29.42	M:29.36
ExAppendixB_AEL03_L	N:0.23	N:0.30	N:0.27
ExConc_Zan97	M:29.85	M:29.80	M:29.95
ExConc_Zan97_Z	M:30.35	M:29.75	M:29.76
ExIntrod_GM01	M:29.38	M:29.33	M:29.30

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
ExIntrod_GM01_FR	M:29.30	M:29.34	M:29.30
ExIntrod_GM01_L	N:0.21	N:0.19	N:0.18
ExIntrod_GM01_Z	M:29.31	M:29.30	M:29.29
ExIntrod_GM04	M:29.36	M:29.31	M:29.30
ExIntrod_GM04_FR	M:29.35	M:29.32	M:29.30
ExIntrod_GM04_L	N:0.20	N:0.18	N:0.18
ExIntrod_GM04_Z	M:29.46	M:29.35	M:29.29
ExIntrod_GM99	M:29.42	M:29.35	M:29.35
ExIntrod_GM99_L	N:0.17	N:0.22	N:0.19
ExIntrod_GM99_Z	M:29.30	M:29.32	M:29.27
ExIntrod_Zan97	M:29.32	M:29.36	M:29.34
ExIntrod_Zan97_L	N:0.21	N:0.21	N:0.17
ExIntrod_Zan97_Z	M:29.37	M:29.32	M:29.30
ExSec4_2_DLMMU04_L	N:0.18	N:0.26	N:0.20
ExSec11_1_Luc02a	M:29.36	M:29.35	M:29.33
Hamming	M:29.37	M:29.31	M:29.34
LISTUTILITIES_complete-noand_L	N:0.28	E:0.17	E:0.18
LISTUTILITIES_complete_L	N:0.23	E:0.15	N:0.56
LISTUTILITIES_nokinds-noand_L	N:0.24	E:0.14	N:0.51
LISTUTILITIES_nokinds_L	N:0.20	N:0.38	N:0.23
LISTUTILITIES_nosorts-noand_L	N:0.21	N:0.51	N:0.24
LISTUTILITIES_nosorts_L	N:0.20	N:0.23	N:0.19
LengthOfFiniteLists_complete-noand_FR	M:29.39	E:0.17	E:0.15
LengthOfFiniteLists_complete-noand_L	N:0.23	N:0.61	N:0.28
LengthOfFiniteLists_complete-noand_Z	M:29.44	E:0.14	E:0.14
LengthOfFiniteLists_complete_L	N:0.17	N:0.26	N:0.21
LengthOfFiniteLists_nokinds-noand_FR	M:29.37	M:29.47	M:29.36
LengthOfFiniteLists_nokinds-noand_L	N:0.23	N:0.24	N:0.19
LengthOfFiniteLists_nokinds-noand_Z	M:29.41	M:29.50	M:29.36
LengthOfFiniteLists_nokinds_FR	M:29.40	M:29.36	M:29.34
LengthOfFiniteLists_nokinds_L	N:0.17	N:0.19	N:0.18
LengthOfFiniteLists_nokinds_Z	M:29.36	M:29.35	M:29.32
LengthOfFiniteLists_nosorts-noand_L	N:0.17	N:0.18	N:0.18
LengthOfFiniteLists_nosorts-noand_Z	M:29.36	M:29.38	M:29.37
LengthOfFiniteLists_nosorts_GM	M:29.34	M:29.31	M:29.32
LengthOfFiniteLists_nosorts_L	N:0.20	N:0.18	N:0.18
LengthOfFiniteLists_nosorts_Z	M:29.41	M:29.36	M:29.29
LengthOfFiniteLists_nosorts_iGM	M:29.43	E:0.17	M:29.43
LengthOfFiniteLists_nosorts_noand_GM	M:29.31	M:29.31	M:29.33
MYNAT_complete-noand_L	N:0.23	E:0.17	N:0.31
MYNAT_complete_L	N:0.20	N:0.27	N:0.20
MYNAT_nokinds-noand_L	N:0.23	N:0.28	N:0.18

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
MYNAT_nokinds_L	N:0.18	N:0.19	N:0.16
MYNAT_nosorts-noand_L	N:0.17	N:0.18	N:0.17
MYNAT_nosorts_L	N:0.19	N:0.17	N:0.15
OvConsOS_complete-noand_L	N:0.17	E:0.16	N:0.47
OvConsOS_complete_L	N:0.17	N:0.47	N:0.23
OvConsOS_nokinds-noand_FR	M:29.36	E:0.16	M:29.51
OvConsOS_nokinds-noand_L	N:0.21	N:0.36	N:0.25
OvConsOS_nokinds-noand_Z	M:29.47	E:0.17	M:29.48
OvConsOS_nokinds_L	N:0.21	N:0.22	N:0.20
OvConsOS_nosorts-noand_FR	M:29.37	M:29.51	M:29.40
OvConsOS_nosorts-noand_L	N:0.17	N:0.23	N:0.20
OvConsOS_nosorts-noand_Z	M:29.32	M:29.55	M:29.41
OvConsOS_nosorts_FR	M:29.33	M:29.37	M:29.33
OvConsOS_nosorts_GM	M:29.35	M:29.33	Y:12.69
OvConsOS_nosorts_L	N:0.21	N:0.18	N:0.18
OvConsOS_nosorts_Z	M:29.34	M:29.37	M:29.29
OvConsOS_nosorts_noand_GM	M:29.40	M:29.34	M:29.36
PALINDROME_complete-noand_L	M:29.38	E:0.18	E:0.16
PALINDROME_complete_L	N:0.18	E:0.14	N:0.40
PALINDROME_nokinds-noand_L	M:29.32	E:0.14	M:29.46
PALINDROME_nokinds_L	N:0.21	N:0.26	N:0.22
PALINDROME_nosorts_L	N:0.20	N:0.24	N:0.20
PEANO_complete-noand_L	N:0.22	N:0.36	N:0.24
PEANO_complete_L	N:0.20	N:0.20	N:0.19
PEANO_nokinds-noand_L	N:0.17	N:0.19	N:0.19
PEANO_nokinds_L	N:0.17	N:0.18	N:0.19
PEANO_nosorts-noand_L	N:0.16	N:0.17	N:0.18
PEANO_nosorts_L	N:0.16	N:0.17	N:0.18
TypeEx3	M:29.37	E:0.16	M:29.44
TypeEx5	N:5.29	M:29.37	N:6.85
afbg	M:29.32	M:29.48	M:29.41
append-wrong	M:29.36	M:29.37	M:29.32
assoc_c_rhs	M:29.34	M:29.35	Y:7.49
assoc_f_rhs	M:29.32	M:29.38	M:29.38
bintree	M:29.35	Y:14.45	Y:6.39
cariboo_add1	M:29.32	M:29.32	M:29.32
cariboo_add2	N:22.42	Y:5.45	Y:1.80
cariboo_add2a	N:12.88	Y:8.83	Y:2.40
cariboo_add3	M:29.32	M:29.34	M:29.32
cariboo_len3	M:29.34	M:29.32	M:29.31
cariboo_nl_1	M:29.31	M:29.33	M:29.40
cariboo_nl_2	M:29.39	M:29.33	M:29.29

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
cariboo_nl_3	M:29.33	M:29.34	M:29.33
cariboo_nl_4	M:29.35	M:29.32	M:29.29
cariboo_nl_5	M:29.37	Y:1.53	Y:1.54
cariboo_nl_6	M:29.34	M:29.31	M:29.32
cime4	M:29.37	M:29.39	M:29.34
countbin	M:29.34	M:29.30	M:29.26
countter	M:29.34	M:29.33	M:29.34
dupl_rhs	M:29.39	M:29.34	M:29.33
even	M:29.37	M:29.33	M:29.31
ex0	M:29.42	M:29.31	M:29.32
ex1	M:29.32	M:29.38	M:29.34
ex1m	M:29.34	M:29.35	Y:2.12
ex2	N:16.23	N:16.17	N:16.43
ex2m	M:29.37	M:29.32	M:29.37
ex3	N:17.49	M:29.38	M:29.39
ex3m	M:29.38	M:29.35	Y:3.15
ex4	M:29.33	M:29.35	M:29.26
ex4m	M:29.36	Y:4.87	Y:1.51
ex5	M:29.35	Y:10.14	Y:7.36
ex5m	M:29.29	Y:0.53	Y:0.53
ex6	M:29.30	M:29.31	M:29.37
ex6m	M:29.33	Y:18.77	Y:17.80
ex7	M:29.31	M:29.30	M:29.24
ex8	M:29.97	Y:3.86	Y:1.02
ex9	M:29.38	M:29.35	M:29.28
f_2_1	M:29.36	M:29.26	Y:7.73
f_2_2	M:29.35	M:29.30	Y:8.65
f_5	M:29.34	M:29.35	Y:10.20
f_5_1	M:29.32	M:29.34	M:29.32
f_5_2	M:29.35	M:29.37	M:29.29
ffb_SL	N:4.75	N:10.17	N:5.52
ffg	M:29.39	M:29.38	Y:7.95
fg	M:29.37	Y:9.06	Y:2.65
from_one	M:29.35	M:29.33	M:29.37
from_one_a	M:29.35	M:29.31	M:29.34
from_three	M:29.38	M:29.31	M:29.29
g_2_const	M:29.28	Y:3.00	Y:1.59
g_2_f_var	M:29.88	Y:19.45	Y:2.53
gfb	M:29.41	M:29.31	Y:4.93
gkg	N:5.55	N:5.55	N:5.67
inn_out	N:5.44	N:6.33	N:5.44
jwno1	M:29.32	M:29.37	M:29.31

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
jwno4	M:29.31	M:29.37	M:29.29
jwno6	M:29.32	M:29.32	M:29.30
jwno9	M:29.33	M:29.34	M:29.32
morse	M:29.28	M:29.38	M:29.32
muladd	M:29.35	M:29.32	M:29.34
n001	M:29.96	M:29.36	M:29.38
n002	N:0.26	N:0.25	N:0.24
n003	N:5.47	N:5.20	N:5.42
n004	N:5.43	N:5.46	N:5.21
n005	N:0.34	N:0.27	N:0.28
n006	N:0.34	N:0.24	N:0.25
n007	N:0.77	N:0.56	N:0.56
n008	N:5.47	N:5.49	N:5.21
non-lin1	M:29.36	M:29.26	M:29.26
non-lin2	M:29.37	M:29.30	M:29.32
non-lin3	M:29.39	M:29.30	M:29.32
nonTermF	M:29.36	M:29.27	M:29.29
nonterm	M:29.36	M:29.31	M:29.34
odd	M:29.39	M:29.37	M:29.31
outermost_gr	N:5.81	M:29.39	N:6.03
patterns1	M:29.36	M:29.34	M:29.32
patterns2	M:29.38	M:29.37	M:29.34
round_nonterm	M:29.35	M:29.34	M:29.32
termMonTypes	M:29.40	M:29.32	M:29.33
test9	M:29.41	M:29.34	M:29.23
test10	M:29.30	M:29.52	M:29.30
test75	M:29.36	M:29.36	M:29.32
test76	M:29.36	M:29.39	M:29.30
test77	M:29.41	M:29.41	M:29.32
thiemann28	M:29.44	E:0.17	M:29.46
toyama_out	M:29.43	M:29.38	M:29.35
toyama_stop2	M:29.39	M:29.55	M:29.33
toyama_stop	M:29.38	M:29.36	M:29.33
ttt1	M:29.34	M:29.38	M:29.39
ttt2	N:6.32	M:29.29	M:29.26
yoyo_2	M:29.34	M:29.35	M:29.32
yoyo_3	M:29.38	M:29.33	M:29.33
yoyo_3a	M:29.28	M:29.34	M:29.42
yoyo_3b	M:30.00	M:29.33	M:29.43

Termination Experiments via AProVE

This section report the result from AProVE. The report format is "result : time".

- Y : The system can be proved.
- N : The system can be disproved.
- M : The system cannot proved or disproved.
- T : The system cannot proved or disproved with in time limit 60s.
- E : The system cannot transformed with in time limit 60s.

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
001	T:60.00	T:60.00	T:60.00
2.05	T:60.00	T:60.00	T:60.00
003	T:60.00	T:60.00	T:60.00
4.2#	T:60.00	T:60.00	T:60.00
4.3#	T:60.00	T:60.00	T:60.00
4.4#	T:60.00	T:60.00	T:60.00
4.06	T:60.00	T:60.00	T:60.00
4.7#	T:60.00	T:60.00	T:60.00
4.12a#	T:60.00	T:60.00	T:60.00
4.13#	Y:13.90	Y:15.53	Y:13.04
4.14#	T:60.00	T:60.00	T:60.00
4.15#	T:60.00	T:60.00	T:60.00
4.16#	T:60.00	N:33.02	N:15.64
4.17#	T:60.00	T:60.00	T:60.00
4.18#	T:60.00	T:60.00	T:60.00
4.34	T:60.00	T:60.00	T:60.00
4.40	T:60.00	T:60.00	T:60.00
4.49	N:4.97	N:28.83	N:8.51
4.54	T:60.00	T:60.00	T:60.00
Ex1_2_AEL03	T:60.00	T:60.00	T:60.00
Ex1_2_AEL03_L	N:1.28	N:1.45	N:0.93
Ex1_2_Luc02c	T:60.00	T:60.00	T:60.00
Ex1_2_Luc02c_L	N:1.44	N:1.10	N:0.94
Ex1_GL02a	T:60.00	T:60.00	T:60.00
Ex1_GL02a_L	T:60.00	T:60.00	T:60.00
Ex1_GL02a_Z	Y:3.70	Y:3.71	Y:2.96
Ex1_GM03	T:60.00	T:60.00	T:60.00
Ex1_GM03_L	N:1.30	N:1.29	N:0.98
Ex1_GM03_Z	T:60.00	T:60.00	T:60.00
Ex1_GM99	T:60.00	T:60.00	T:60.00

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
Ex1_GM99_GM	T:60.00	T:60.00	T:60.00
Ex1_GM99_L	N:2.07	N:2.14	N:1.81
Ex1_GM99_Z	T:60.00	T:60.00	T:60.00
Ex1_GM99_iGM	T:60.00	E:0.19	T:60.00
Ex1_Luc02b	T:60.00	T:60.00	T:60.00
Ex1_Luc02b_L	N:1.22	N:1.19	N:1.00
Ex1_Luc04b_FR	T:60.00	T:60.00	T:60.00
Ex1_Luc04b_L	N:1.56	N:1.35	N:0.99
Ex1_Luc04b_Z	T:60.00	T:60.00	T:60.00
Ex1_Zan97	T:60.00	T:60.00	N:2.29
Ex1_Zan97_L	N:2.12	N:2.98	N:1.55
Ex1_Zan97_Z	T:60.00	T:60.00	T:60.00
Ex2_8_1ConstSubstFix	T:60.00	T:60.00	T:60.00
Ex2_Luc03b	T:60.00	T:60.00	T:60.00
Ex3_2_Luc97	T:60.00	T:60.00	T:60.00
Ex3_2_Luc97_L	N:1.29	N:1.97	N:0.96
Ex3_3_25_Bor03	T:60.00	T:60.00	T:60.00
Ex3_3_25_Bor03_Z	T:60.00	T:60.00	T:60.00
Ex3_12_Luc96a	T:60.00	T:60.00	T:60.00
Ex3_12_Luc96a_L	N:1.17	N:1.78	N:0.96
Ex4_4_Luc96b	T:60.00	T:60.00	T:60.00
Ex4_4_Luc96b_FR	T:60.00	T:60.00	T:60.00
Ex4_4_Luc96b_Z	T:60.00	T:60.00	T:60.00
Ex4_7_15_Bor03	T:60.00	T:60.00	T:60.00
Ex4_7_37_Bor03	T:60.00	E:0.31	T:60.00
Ex4_7_37_Bor03_L	N:1.36	E:0.23	N:1.03
Ex4_7_56_Bor03	T:60.00	T:60.00	T:60.00
Ex4_7_56_Bor03_L	N:1.49	N:1.64	N:0.97
Ex4_7_77_Bor03	T:60.00	T:60.00	T:60.00
Ex4_7_77_Bor03_L	N:1.27	N:1.70	N:0.96
Ex4_DLMMU04_FR	T:60.00	T:60.00	T:60.00
Ex4_DLMMU04_L	N:1.32	N:2.04	N:0.99
Ex4_DLMMU04_Z	T:60.00	T:60.00	T:60.00
Ex4_Zan97	T:60.00	T:60.00	T:60.00
Ex4_Zan97_L	N:1.25	N:1.95	N:0.92
Ex5_7_Luc97	T:60.00	T:60.00	T:60.00
Ex5_7_Luc97_L	N:1.23	N:2.37	N:1.03
Ex5_DLMMU04_FR	T:60.00	T:60.00	T:60.00
Ex5_DLMMU04_L	N:1.19	E:0.25	N:1.24
Ex5_DLMMU04_Z	T:60.00	T:60.00	T:60.00
Ex5_Zan97	T:60.00	T:60.00	T:60.00
Ex5_Zan97_L	N:1.49	N:1.96	N:0.97

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
Ex6_9_Luc02c	T:60.00	T:60.00	T:60.00
Ex6_9_Luc02c_L	N:1.16	N:1.81	N:1.03
Ex6_15_AEL02	T:60.00	T:60.00	T:60.00
Ex6_15_AEL02_L	N:1.52	N:2.48	N:0.96
Ex6_15_AEL02_Z	T:60.00	T:60.00	T:60.00
Ex6_GM04	T:60.00	T:60.00	T:60.00
Ex6_GM04_Z	T:60.00	T:60.00	T:60.00
Ex6_Luc98	T:60.00	T:60.00	T:60.00
Ex7_BLR02	T:60.00	T:60.00	T:60.00
Ex7_BLR02_L	N:1.20	N:1.96	N:1.02
Ex8_BLR02	T:60.00	T:60.00	T:60.00
Ex8_BLR02_L	N:1.63	N:1.72	N:0.86
Ex9_BLR02	T:60.00	T:60.00	T:60.00
Ex9_Luc04_GM	T:60.00	T:60.00	T:60.00
Ex9_Luc04_L	N:3.96	N:9.21	N:13.95
Ex9_Luc04_Z	T:60.00	T:60.00	T:60.00
Ex9_Luc06_L	N:1.78	N:2.90	N:1.67
Ex9_Luc06_Z	T:60.00	T:60.00	T:60.00
Ex14_AEGL02	T:60.00	T:60.00	T:60.00
Ex14_AEGL02_FR	T:60.00	T:60.00	T:60.00
Ex14_AEGL02_L	T:60.00	T:60.00	T:60.00
Ex14_AEGL02_Z	T:60.00	T:60.00	T:60.00
Ex14_Luc06_GM	T:60.00	T:60.00	T:60.00
Ex14_Luc06_L	T:60.00	T:60.00	N:2.93
Ex14_Luc06_Z	T:60.00	T:60.00	T:60.00
Ex15_Luc06_L	N:2.58	N:7.05	N:2.02
Ex15_Luc98	T:60.00	T:60.00	T:60.00
Ex15_Luc98_L	N:2.17	N:5.13	N:1.29
Ex16_Luc06_L	N:2.62	N:7.00	N:2.13
Ex24_GM04	T:60.00	T:60.00	T:60.00
Ex24_GM04_GM	T:60.00	T:60.00	T:60.00
Ex24_GM04_L	N:3.27	N:8.56	N:2.29
Ex24_Luc06_GM	Y:5.30	Y:11.40	Y:4.62
Ex24_Luc06_Z	T:60.00	T:60.00	T:60.00
Ex24_Luc06_iGM	T:60.00	E:0.67	T:60.00
Ex26_Luc03b	T:60.00	T:60.00	T:60.00
Ex49_GM04_L	N:1.92	N:4.29	N:1.25
ExAppendixB_AEL03	T:60.00	T:60.00	T:60.00
ExAppendixB_AEL03_L	N:1.94	N:5.99	N:1.24
ExConc_Zan97	T:60.00	T:60.00	T:60.00
ExConc_Zan97_Z	T:60.00	T:60.00	T:60.00
ExIntrod_GM01	T:60.00	T:60.00	T:60.00

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
ExIntrod_GM01_FR	T:60.00	T:60.00	T:60.00
ExIntrod_GM01_L	N:3.15	N:4.05	N:1.31
ExIntrod_GM01_Z	T:60.00	T:60.00	T:60.00
ExIntrod_GM04	T:60.00	T:60.00	T:60.00
ExIntrod_GM04_FR	T:60.00	T:60.00	T:60.00
ExIntrod_GM04_L	N:2.95	N:4.23	N:1.27
ExIntrod_GM04_Z	T:60.00	T:60.00	T:60.00
ExIntrod_GM99	T:60.00	T:60.00	T:60.00
ExIntrod_GM99_L	N:2.98	N:4.97	N:1.41
ExIntrod_GM99_Z	T:60.00	T:60.00	T:60.00
ExIntrod_Zan97	T:60.00	T:60.00	T:60.00
ExIntrod_Zan97_L	N:3.24	N:5.09	N:1.30
ExIntrod_Zan97_Z	T:60.00	T:60.00	T:60.00
ExSec4_2_DLMMU04_L	N:3.05	N:4.62	N:1.51
ExSec11_1_Luc02a	T:60.00	T:60.00	T:60.00
Hamming	T:60.00	T:60.00	T:60.00
LISTUTILITIES_complete-noand_L	N:3.69	E:0.67	E:0.18
LISTUTILITIES_complete_L	N:3.22	E:0.73	N:3.96
LISTUTILITIES_nokinds-noand_L	N:3.01	E:0.71	N:3.45
LISTUTILITIES_nokinds_L	N:2.99	N:8.76	N:1.46
LISTUTILITIES_nosorts-noand_L	N:2.92	N:14.43	N:1.76
LISTUTILITIES_nosorts_L	N:3.03	N:5.58	N:1.37
LengthOfFiniteLists_complete-noand_FR	T:60.00	E:0.65	E:0.15
LengthOfFiniteLists_complete-noand_L	N:3.40	N:20.32	N:1.93
LengthOfFiniteLists_complete-noand_Z	T:60.00	E:0.65	E:0.17
LengthOfFiniteLists_complete_L	N:3.21	N:6.04	N:1.50
LengthOfFiniteLists_nokinds-noand_FR	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nokinds-noand_L	N:2.75	N:6.01	N:1.35
LengthOfFiniteLists_nokinds-noand_Z	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nokinds_FR	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nokinds_L	N:3.06	N:4.09	N:1.11
LengthOfFiniteLists_nokinds_Z	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nosorts-noand_L	N:2.98	N:4.63	N:1.13
LengthOfFiniteLists_nosorts-noand_Z	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nosorts_GM	Y:6.79	Y:15.65	Y:3.74
LengthOfFiniteLists_nosorts_L	N:2.95	N:4.29	N:1.20
LengthOfFiniteLists_nosorts_Z	T:60.00	T:60.00	T:60.00
LengthOfFiniteLists_nosorts_iGM	T:60.00	E:0.79	T:60.00
LengthOfFiniteLists_nosorts_noand_GM	Y:8.13	Y:21.00	Y:3.71
MYNAT_complete-noand_L	N:3.17	E:0.82	N:1.92
MYNAT_complete_L	N:3.27	N:6.56	N:1.35
MYNAT_nokinds-noand_L	N:2.86	N:5.76	N:1.20

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
MYNAT_nokinds_L	N:3.08	N:4.75	N:1.14
MYNAT_nosorts-noand_L	N:2.74	N:4.76	N:1.12
MYNAT_nosorts_L	N:3.06	N:4.80	N:1.26
OvConsOS_complete-noand_L	N:3.10	E:0.76	N:3.76
OvConsOS_complete_L	N:3.01	N:10.73	N:1.76
OvConsOS_nokinds-noand_FR	T:60.00	E:0.66	T:60.00
OvConsOS_nokinds-noand_L	N:3.12	N:8.17	N:1.36
OvConsOS_nokinds-noand_Z	T:60.00	E:0.74	T:60.00
OvConsOS_nokinds_L	N:3.37	N:4.97	N:1.24
OvConsOS_nosorts-noand_FR	T:60.00	T:60.00	T:60.00
OvConsOS_nosorts-noand_L	N:3.18	N:5.17	N:1.22
OvConsOS_nosorts-noand_Z	T:60.00	T:60.00	T:60.00
OvConsOS_nosorts_FR	T:60.00	T:60.00	T:60.00
OvConsOS_nosorts_GM	T:60.00	T:60.00	T:60.00
OvConsOS_nosorts_L	N:2.52	N:4.92	N:1.29
OvConsOS_nosorts_Z	T:60.00	T:60.00	T:60.00
OvConsOS_nosorts_noand_GM	T:60.00	T:60.00	T:60.00
PALINDROME_complete-noand_L	T:60.00	E:0.67	E:0.19
PALINDROME_complete_L	N:3.52	E:0.70	N:2.31
PALINDROME_nokinds-noand_L	T:60.00	E:0.77	T:60.00
PALINDROME_nokinds_L	N:2.97	N:5.61	N:1.39
PALINDROME_nosorts_L	N:3.24	N:5.61	N:1.28
PEANO_complete-noand_L	N:2.93	N:8.17	N:1.56
PEANO_complete_L	N:3.08	N:5.33	N:1.33
PEANO_nokinds-noand_L	N:3.03	N:5.08	N:1.31
PEANO_nokinds_L	N:2.82	N:4.52	N:1.25
PEANO_nosorts-noand_L	N:3.01	N:5.23	N:1.23
PEANO_nosorts_L	N:2.71	N:4.45	N:1.29
TypeEx3	T:60.00	E:0.77	T:60.00
TypeEx5	T:60.00	T:60.00	T:60.00
afbg	T:60.00	T:60.00	T:60.00
append-wrong	T:60.00	T:60.00	T:60.00
assoc_c_rhs	M:16.86	T:60.00	Y:6.45
assoc_f_rhs	M:40.14	T:60.00	Y:6.44
bintree	Y:6.69	T:60.00	Y:6.42
cariboo_add1	T:60.00	T:60.00	T:60.00
cariboo_add2	T:60.00	Y:12.74	Y:2.80
cariboo_add2a	T:60.00	Y:10.72	Y:2.96
cariboo_add3	T:60.00	T:60.00	T:60.00
cariboo_len3	T:60.00	T:60.00	T:60.00
cariboo_nl_1	T:60.00	T:60.00	T:60.00
cariboo_nl_2	T:60.00	T:60.00	T:60.00

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
cariboo_nl_3	T:60.00	T:60.00	T:60.00
cariboo_nl_4	T:60.00	T:60.00	T:60.00
cariboo_nl_5	Y:3.93	Y:6.17	Y:1.86
cariboo_nl_6	Y:48.34	T:60.00	Y:34.64
cime4	T:60.00	T:60.00	T:60.00
countbin	T:60.00	T:60.00	T:60.00
countter	T:60.00	T:60.00	T:60.00
dupl_rhs	T:60.00	T:60.00	T:60.00
even	T:60.00	T:60.00	T:60.00
ex0	T:60.00	T:60.00	T:60.00
ex1	T:60.00	T:60.00	T:60.00
ex1m	T:60.00	T:60.00	Y:1.42
ex2	T:60.00	T:60.00	T:60.00
ex2m	T:60.00	T:60.00	T:60.00
ex3	T:60.00	T:60.00	T:60.00
ex3m	T:60.00	T:60.00	Y:10.45
ex4	T:60.00	T:60.00	T:60.00
ex4m	T:60.00	Y:11.58	Y:2.81
ex5	Y:7.91	Y:17.54	Y:5.70
ex5m	Y:4.06	Y:4.10	Y:1.57
ex6	T:60.00	T:60.00	Y:7.09
ex6m	Y:11.11	Y:15.61	Y:6.20
ex7	T:60.00	T:60.00	T:60.00
ex8	Y:17.51	Y:4.54	Y:1.51
ex9	T:60.00	T:60.00	T:60.00
f_2_1	T:60.00	T:60.00	Y:5.20
f_2_2	T:60.00	Y:51.07	Y:21.46
f_5	T:60.00	T:60.00	T:60.00
f_5_1	T:60.00	T:60.00	T:60.00
f_5_2	T:60.00	T:60.00	T:60.00
ffb_SL	T:60.00	T:60.00	N:2.33
ffg	T:60.00	Y:30.93	Y:5.63
fg	T:60.00	Y:10.36	Y:2.77
from_one	T:60.00	T:60.00	T:60.00
from_one_a	T:60.00	T:60.00	T:60.00
from_three	T:60.00	T:60.00	T:60.00
g_2_const	T:60.00	Y:5.05	Y:1.19
g_2_f_var	T:60.00	Y:21.72	Y:3.06
gfb	T:60.00	Y:41.07	Y:10.60
gkg	N:37.76	N:41.02	N:13.79
inn_out	T:60.00	T:60.00	N:2.78
jwno1	T:60.00	T:60.00	T:60.00

Problem Name	$\mathcal{R}^T$	$\mathcal{S}(\mathcal{R}^T)$	$\mathcal{S}_{op}(\mathcal{R}^T)$
jwno4	T:60.00	T:60.00	T:60.00
jwno6	T:60.00	T:60.00	T:60.00
jwno9	T:60.00	T:60.00	T:60.00
morse	T:60.00	T:60.00	T:60.00
muladd	T:60.00	T:60.00	T:60.00
n001	T:60.00	T:60.00	N:8.66
n002	N:7.38	N:6.23	N:1.96
n003	N:7.66	N:6.24	N:2.14
n004	T:60.00	T:60.00	N:2.23
n005	N:7.21	N:6.28	N:2.03
n006	N:8.01	N:6.58	N:2.06
n007	N:8.36	N:7.46	N:2.70
n008	T:60.00	T:60.00	N:2.31
non-lin1	Y:6.50	Y:4.97	Y:1.69
non-lin2	T:60.00	T:60.00	T:60.00
non-lin3	T:60.00	T:60.00	T:60.00
nonTermF	T:60.00	T:60.00	T:60.00
nonterm	T:60.00	T:60.00	T:60.00
odd	T:60.00	T:60.00	Y:14.34
outermost_gr	T:60.00	T:60.00	T:60.00
patterns1	T:60.00	T:60.00	T:60.00
patterns2	T:60.00	T:60.00	T:60.00
round_nonterm	T:60.00	T:60.00	T:60.00
termMonTypes	T:60.00	T:60.00	T:60.00
test9	T:60.00	T:60.00	T:60.00
test10	T:60.00	T:60.00	T:60.00
test75	T:60.00	T:60.00	T:60.00
test76	T:60.00	T:60.00	T:60.00
test77	T:60.00	T:60.00	T:60.00
thiemann28	T:60.00	E:0.61	T:60.00
toyama_out	T:60.00	T:60.00	T:60.00
toyama_stop2	T:60.00	T:60.00	T:60.00
toyama_stop	T:60.00	T:60.00	Y:42.61
ttt1	T:60.00	T:60.00	T:60.00
ttt2	T:60.00	T:60.00	T:60.00
yoyo_2	T:60.00	T:60.00	T:60.00
yoyo_3	T:60.00	T:60.00	T:60.00
yoyo_3a	T:60.00	T:60.00	T:60.00
yoyo_3b	T:60.00	T:60.00	T:60.00

Bibliography

[1] Termination problem database. `http://termination-portal.org/wiki/TPDB`. Accessed: 2016-10-30.

[2] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.

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

[4] M. Dauchet. Simulation of turing machines by a regular rewrite rule. *Theoretical Computer Science*, 103:409–420, 1992.

[5] J. Endrullis and D. Hendriks. From outermost to context-sensitive rewriting. In *Proceedings of the 20th International Conference on Rewriting Techniques and Applications*, volume 5595 of *Lecture Notes in Computer Science*, pages 305–319, 2009.

[6] J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. In *Proceedings of the 3rd International Conference on Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 574–588, 2006.

[7] ML. Fernndez, G. Godoy, and A. Rubio. Orderings for innermost termination. In *Proceedings of the 16th International Conference on Rewriting Techniques and Applications*, volume 3467 of *Lecture Notes in Computer Science*, pages 17–31, 2005.

[8] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, , and H. Zankl. Sat solving for termination analysis with polynomial interpretations. In *In Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing*, pages 177–193, 2007.

[9] J. Giesl, M. Brockschmidt, F. Emmes, F. Frohn, C. Fuhs, C. Otto, M. Plcker, P. Schneider-Kamp, T. Strder, S. Swiderski, and R. Thiemann. Proving termination of programs automatically with AProVE. In *Proceedings of the 7th International Joint Conference on Automated Reasoning*, volume 8562 of *Lecture Notes in Computer Science*, pages 184–191, 2014.

[10] J. Giesl and A. Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14:329–427, 2004.

[11] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In *Proceedings of the 5th International Conference on Frontiers of Combining Systems*, volume 3717 of *Lecture Notes in Computer Science*, pages 216–231, 2005.

[12] B. Gramlich. On proving termination by innermost termination. In *Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 93–107, 1996.

[13] N. Hirokawa and A. Middeldorp. Tyrolean termination tool: Techniques and features. *Information and Computation*, 205:474–511, 2007.

[14] N. Hirokawa, A. Middeldorp, and H. Zankl. Uncurrying for termination and complexity. *Journal of Automated Reasoning*, 50:279315, 1990.

[15] N. Hirokawa, A. Middeldorp, and H. Zankl. Modular termination proofs for rewriting using dependency pairs. *Journal of Symbolic Computation*, 34:21–58, 2002.

[16] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean termination tool 2. In *Proceedings of the 20th International Conference on Rewriting Techniques and Applications*, volume 5595 of *Lecture Notes in Computer Science*, pages 295–304, 2009.

[17] A. Lazrek, P. Lescanne, , and J. J. Thiel. Tools for proving inductive equalities, relative completeness, and omega-completeness. *Information and Computation*, 84:47–70, 1990.

[18] E. Ohlebusch. *Advanced Topics in Term Rewriting*, volume 1. Springer-Verlag New York, 2002.

[19] L. Puel. Using unavoidable set of trees to generalize kruskal's theorem. *Journal of Symbolic Computation*, 8:335–382, 1989.

[20] M. Sakai. Innermost terminating right-linear overlay term rewriting systems are terminating. Technical report, 2003.

[21] TeReSe. *Term Rewriting Systems*, volume 55. Cambridge University Press, 2003.

[22] R. Thiemann. *The DP Framework for Proving Termination of Term Rewriting*. PhD thesis, RWTH Aachen University, 2007.

[23] R. Thiemann. From outermost termination to innermost termination. In *Proceedings of the 35th Conference on Current Trends in Theory and Practice of Computer Science*, volume 5404 of *Lecture Notes in Computer Science*, pages 533–545, 2009.

[24] R. Thiemann and A. Middeldorp. Innermost termination of rewrite systems by labeling. In *Proceedings of the 7th International Workshop on Reduction Strategies in Rewriting and Programming*, volume 204 of *Electronic Notes in Theoretical Computer Science*, pages 3–19, 2008.

[25] J. van de Pol. Modularity in many-sorted term rewriting systems. Technical report, 1993.

[26] A. Yamada, K. Kusakari, , and T. Sakabe. Nagoya termination tool. In *Proceedings of the 25th International Conference on Rewriting Techniques and Applications*, volume 8560 of *Lecture Notes in Computer Science*, pages 466–475, 2014.

[27] H. Zantema. Termination of term rewriting. *Journal of Symbolic Computation*, 17:23–50, 1994.