Doctor's Thesis

A Decidable Subclass of Term Rewriting Systems Which Effectively Preserve Recognizability

Toshinori Takai

February 5, 2002

Department of Information Processing
Graduate School of Information Science
Nara Institute of Science and Technology

Doctor's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
DOCTOR of ENGINEERING

Toshinori Takai

Thesis committee:	Hiroyuki Seki, Professor
	Minoru Ito, Professor
	Michiko Inoue, Associate Professor
	Yuichi Kaji, Associate Professor

A Decidable Subclass of Term Rewriting Systems Which Effectively Preserve Recognizability*

Toshinori Takai

Abstract

Term Rewriting system (TRS) is a well-known computational model which operates on terms (or trees). Recently much attention has been paid to TRSs which effectively preserve recognizability (EPR-TRSs). A set L of terms (or a tree language) is recognizable if and only if there exists a tree automaton which accepts L. A TRS R effectively preserves recognizability if and only if for every recognizable set L, we can construct a tree automaton which exactly accepts those terms rewritable from terms in L by R. It is known that some important properties such as local confluence and joinability are decidable for EPR-TRSs. It is undecidable whether a given TRS effectively preserves recognizability or not, and hence decidable subclasses of EPR-TRSs have been proposed. However, there exist EPR-TRSs which do not belong to any of those subclasses.

This thesis proposes a decidable subclass of TRSs, which is called *finitely path overlapping TRSs (FPO-TRSs)*, and shows that every right-linear FPO-TRS effectively preserves recognizability. Also, right-linear FPO-TRSs are shown to properly include other well-known decidable subclasses of EPR-TRSs.

Strongly normalizing (or termination) property is one of the most fundamental properties in the theory of TRSs. However, the property is undecidable, and some subclasses have been proposed for which the property is decidable. Nagaya and Toyama proposed growing TRSs and showed that for an almost orthogonal

*Doctor's Thesis, Department of Information Processing, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DT9861012, February 5, 2002.

growing TRS, strongly normalizing property is decidable by using a tree automata technique.

In the latter half of this thesis, we consider an inverse FPO-TRS (denoted by FPO^{-1}-TRS), which is obtained from an FPO-TRS R by interchanging the left-hand side and the right-hand side of each rewrite rule in R. Following the Nagaya and Toyama's method, we show that for a given almost orthogonal FPO^{-1}-TRS R, it is decidable whether R is strongly normalizing or not.

Keywords:

term rewriting system, tree automaton, decidability, recognizability, strongly normalizing property

論文内容の要旨

博士論文題目:

A Decidable Subclass of Term Rewriting Systems
Which Effectively Preserve Recognizability
(構成的正則保存項書換え系の決定可能な部分クラス)

氏　　名: 高井 利憲

項書換え系は, 木構造 (項) を扱う代表的な計算モデルである. 近年, 項書換え系の部分クラスである, 構成的正則保存項書換え系が注目されている. 木言語 (項の集合) L が正則であるとは, L を受理する木オートマトンが存在することである. 項書換え系 R が構成的正則保存であるとは, 任意の正則木言語 L に対して, R による書換えより L から得られる項全体の集合を受理する木オートマトンを構成できることをいう. 構成的正則保存項書換え系に対しては, 項書換え系に関するいくつかの重要な性質, 例えば, 局所合流性や到達可能性などが決定可能になることが知られている. 与えられた項書換え系が構成的正則保存であるかどうかは決定不能であるため, 多くの決定可能な部分クラスが提案されてきた. しかし, それらのクラスはいくつかの簡単な構成的正則保存項書換え系を含んでおらず, 十分広い部分クラスとはいえない.

本論文では, 決定可能な項書換え系の部分クラス, 有界重なり項書換え系を提案し, 任意の右線形有界重なり項書換え系が構成的正則保存であることを示す. また, 右線形有界重なり項書換え系は, 今までに知られている, 他の決定可能な構成的正則保存の部分クラスを真に含むことも示す.

強正規化性 (停止性) は, 項書換え系の重要な性質の一つである. しかし, 強正規化性は決定不能であるため, 強正規化性が決定可能になるような部分クラスがいくつか提案されてきた. 長谷と外山は, 成長的項書換え系を提案し, 木オートマトンを用いて, 準直交成長的項書換え系に対しては, 強正規化性が決定可能になることを示した.

本論文の後半では, 有界重なり項書換え系の左辺と右辺を入れ換えて得られる逆有界重なり項書換え系について考察する. 長谷と外山の方法を利用し, 与えられた準直交逆有界重なり項書換え系に対して, 強正規化性が決定可能になることを示す.

キーワード:

項書換え系, 木オートマトン, 決定可能性, 正則性, 強正規化性

Acknowledgements

First of all I wish to express my gratitude to Professor Hiroyuki Seki. Without his pertinent advice and constant encouragement during the course of the study I would have never written this thesis. I also thank to Professor Minoru Ito and Associate Professor Michiko Inoue for their willingness to be members of the thesis committee for this thesis. I express my thanks to Associate Professor Yuichi Kaji, who guided me into this study, for his insightful comments and invaluable support throughout the work.

I would like to acknowledge Dr. Yoshiki Kinoshita of National Institute of Advanced Industrial Science and Technology for his kind support and encouragement. I am extremely grateful to Dr. Hitoshi Ohsaki of National Institute of Advanced Industrial Science and Technology for his helpful comments and valuable information on TRSs. The TRS meetings I have attended gave me suggestions and comments; I thank all participants.

I am deeply indebted to Professor Tadao Kasami of Hiroshima City University, Professor Toyoo Takata of Iwate Prefectual University, Dr. Hajime Watanabe of National Institute of Advanced Industrial Science and Technology for their kind supports. I would also like to express my thanks to my collaborators Mr. Kouji Kitaoka and Mr. Takenori Abe for their discussions. I am obligated to entire staff of the Kasami and Seki laboratories, to which I have belonged, of Nara Institute of Science and Technology for their friendliness and interest. I also thank my present colleagues of National Institute of Advanced Industrial Science and Technology for their comments and suggestions.

Finally I am especially grateful to my parents and the rest of the family.

List of Publications

Journal Papers

1. Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: “Termination property of inverse finite path overlapping term rewriting system is decidable,” IEICE Transactions on Information and Systems (to appear).month

2. Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: “Right-linear finite-path overlapping term rewriting systems effectively preserve recognizability,” Scienticae Mathematicae Japonicae (submitted).

International Conference

1. Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: “Right-linear finite-path overlapping term rewriting systems effectively preserve recognizability,” Proceedings of the 11th International Conference on Rewriting Techniques and Applications, Norwich, U.K., Lecture Notes in Computer Science, volume 1833, pages 246–260, Springer-Verlag, July 2000.

Workshops

1. Toshinori Takai, Yuichi Kaji, Takehiko Tanaka and Hiroyuki Seki: “A procedure for solving an order-sorted unification problem — extension for left nonlinear system,” IEICE Technical Report, COMP98-44, October 1998.

2. Kouji Kitaoka, Toshinori Takai, Yuichi Kaji, Takehiko Tanaka and Hiroyuki Seki: “Finite-overlapping term rewriting systems effectively preserve

recognizability," IEICE Technical Report, COMP98-45, October 1998 (in Japanese).

3. Kouji Kitaoka, Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "Finite-overlapping term rewriting systems effectively preserve recognizability," The 14th Term Rewriting Meeting, NAIST, March 1999.

4. Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "Right-linear finite path overlapping term rewriting systems effectively preserve recognizability," The 17th Term Rewriting Meeting, Osaka LERC, November 2000.

5. Takenori Abe, Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "Termination of finite path overlapping term rewriting system," Naoki Kato, editor, New Developments of Theory of Computation and Algorithms, Kyoto, Kokyuroku (research report) of Research Institute for Mathematical Sciences, number 1205, Kyoto University, January 2001 (in Japanese).

6. Toshinori Takai: "Termination of finite path overlapping term rewriting systems," The 18th Term Rewriting Meeting, Sakunami, March 2001.

Technical Reports

1. Toshinori Takai, Yuichi Kaji, Takehiko Tanaka and Hiroyuki Seki: "A procedure for solving an order-sorted unification problem – extension for left-nonlinear system," NAIST Technical Report, NAIST-IS-TR98011, 1998.

2. Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "A sufficient condition for the termination of the procedure for solving an order-sorted unification problem," NAIST Technical Report, NAIST-IS-TR99010, 1999.

Contents

List of Figures

Chapter 1

Introduction

Term Rewriting system is a well-known computational model which operates on terms (or trees). The following is an example of a term rewriting system:

$$\mathcal{R}_0 = \{\ d(x, e(x, y)) \to y\ \}$$

where d and e are function symbols and x and y are variables. In general, a term rewriting system (abbreviated as TRS) is an arbitrary finite relation on first-order terms and defines a *rewrite relation* on terms. The rewrite relation of a TRS $\mathcal{R}$ is an infinite relation and written as $\to^*_{\mathcal{R}}$. For example, the term $m_1(d(k(Alice), e(k(Alice), m_2))$ where m_1 and k are function symbols and $Alice$ and m_2 are constants can be 'rewritten' to a term m by using the TRS $\mathcal{R}_0$, i.e. $m_1(d(k(Alice), e(k(Alice), m_2)) \to^*_{\mathcal{R}_0} m_1(m_2)$. Intuitively saying, for a TRS $\mathcal{R}$, the rewrite relation defined by $\mathcal{R}$ is the minimal relation which contains $\mathcal{R}$ and is closed under contexts and substitutions. An element in a TRS is called a rewrite rule and for a rewrite rule $l \to r$, l is called the left-hand side and r is the right-hand side. The TRS $\mathcal{R}_0$ above consists of only one rewrite rule and defines the characteristics of an encryption function (e) and a decryption function (d) in some cryptographic protocols[13, 25]. The function e encrypts a message y with a key x and the result is $e(x, y)$. The TRS $\mathcal{R}_0$ intuitively means that the function d can decript the result $e(x, y)$ to y if the same key x is also given as the first argument of d.

As in the example presented above, operations to replace a pattern of trees with another pattern appear in various areas in computer science. For example, a set of inference rules in the theory of automated theorem proving defines such operations, which is called derivations, for a given formula. Grammars in formal language theory replace a pattern of the left-hand side of a production rule with the corresponding right-hand side in derivation trees. Functional programming lanugages can directly be regarded as TRSs. Dolev and Yao[13] firstly proposed a mathematical model of a class of cryptographic protocols by using TRSs. Kaji *et al.*[25] presented a method to verify the cryptographic protocol which is specified by using such a TRS as $\mathcal{R}_0$ above. TRSs are investigated as a general framework for treating such replacement operations on tree structured data. More applications can be found in surveys of TRSs[11].

We can easily see that any type 0 grammar in Chomsky's hierarchy can be simulated by a TRS according to the definition that the left- and right-hand sides of a rewrite rule can be an arbitrary term. Especially, it it known that any Turing machine can be simulated by a TRS which consists of one (left-linear) rewrite rule[8]. Due to its Turing-complete computational power, many important properties such as reachability, confluence, unifiability and strongly normalizing property are undecidable in general. Consequently, finding an appropriate class of TRSs which has sufficient computational power as well as favorable properties has been paid attention to for a long time.

On the other hand, *tree automata* are also widely investigated as a mathematical model dealing with terms[16]. A tree automaton is a finite-state machine which accepts terms and tree automata define the class of tree languages (sets of terms) as follows: A tree language L is recognizable if there is a tree automaton which accepts L. Tree automata have some of the useful properties as the traditional finite-state automata on strings have. For example, the class of reconizable tree languages is closed under boolean operations (union, intersection and complement) and membership and emptiness problems are decidable.

Nevertheless TRSs and tree automata have been studied on rather independently since their research histories and motivations are different. Studies on tree automata have strong relation to traditional string automata and formal language theory, while the problems of TRSs are mainly motivated by problems of mathematical logic, universal algebra, automated theorem proving and functional programming. As already mentioned before, tree automata inherit many advantageous properties of finite-state automata on strings[16].

Recently, many researchers have been interested in the relation between TRSs and tree automata[5]. The class of TRSs which effectively preserve recognizability is defined by using tree automata as follows: Let $\mathcal{L}(\mathcal{A})$ be the tree language accepted by a tree automaton $\mathcal{A}$. For a TRS $\mathcal{R}$ and a tree language L, the descendant of L by $\mathcal{R}$, denoted by $(\rightarrow^*_{\mathcal{R}})(L)$, is the image of L by the rewrite relation defined by $\mathcal{R}$. That is, $(\rightarrow^*_{\mathcal{R}})(L) = \{t \mid \exists s \in L\colon s \rightarrow^*_{\mathcal{R}} t\}$. A TRS effectively preserves recognizability if for any tree automaton $\mathcal{A}$ we can construct a tree automaton which accepts $(\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$. It is known that for a TRS which effectively preserves recognizability, reachability problem, joinability problem, local confluence are decidable. In fact, these problems can be translated into problems on tree languages which are decidable according to the properties of tree automata. In this thesis, decidability of weakly normalizing, unification problem and needed redexes is also discussed.

Unfortunately, it is undecidable whether a given TRS effectively preserves recognizability or not. Therefore, some decidable subclasses of recognizability preserving TRSs have been proposed in many papers. Such classes include ground TRS[3], right-linear monadic TRS[29], linear semi-monadic TRS[6] and linear generalized semi-monadic TRS[21]. The class of linear semi-monadic TRS properly includes ground TRSs. The class of linear generalized semi-monadic TRSs properly includes linear semi-monadic TRS but does not include right-linear monadic TRSs. The class of right-linear monadic TRSs does not include ground TRSs. The inclusion relation of them is shown in Figure 1.1.

In the first half part of this thesis, a new class of TRSs, *finitely path overlapping TRSs* (FPO-TRSs) is proposed. A TRS in the class of right-linear FPO-TRSs (RL-FPO-TRSs) effectively preserves recognizability, and the class properly includes all the above mentioned decidable subclasses of TRSs which effectively preserve recognizability. Gyenizse and Vágvölgyi[21] presented the open problem to ask to generalize the class of linear generalized semi-monadic TRSs so that a TRS in the obtained class still effectively preserves recognizability. The proposed class RL-FPO-TRSs is shown to properly includes linear generalized semi-monadic TRSs. The following TRS is an example which is included in RL-FPO-TRS but not in the other decidable subclasses stated above where f and g

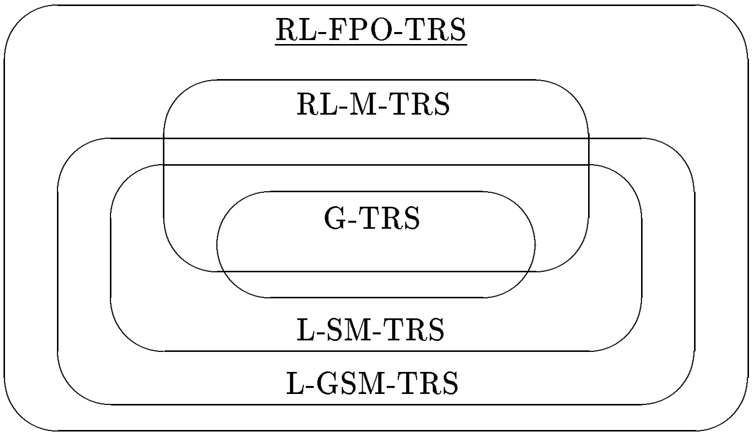


Figure 1.1. Inclusion relation of subclasses of TRSs

are function symbols, a and b are constants and x is a variable:

$$\begin{cases} g(x) \rightarrow f(g(x), b), \\ f(x, a) \rightarrow f(a, x). \end{cases}$$

In order to prove a TRS in RL-FPO-TRSs effectively preserves recognizability, this thesis provides a procedure of which input is a TRS $\mathcal{R}$ and a tree automaton $\mathcal{A}$ and of which output is a tree automaton which accepts $(\rightarrow^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$, the descendant of the accepting language of $\mathcal{A}$ by $\mathcal{R}$. The procedure is a non-trivial extension of Kaji *et al.*'s unification procedure[25]. Dealing with non-linearity by tree automata is very difficult due to the limitation of their recognizing power. While Kaji *et al.*'s procedure can deal with left-non-linearity with some restrictions, the procedure proposed in this thesis can deal with arbitrary left-non-linear rewrite rules. This thesis proves that for an RL-FPO-TRS and an arbitrary tree automaton, the procedure is sound and complete and always terminates.

For a TRS $\mathcal{R}$, the inverse of $\mathcal{R}$ is a TRS obtained from $\mathcal{R}$ by interchanging the left-hand side and the right-hand side of each rewrite rule in $\mathcal{R}$. The class of TRSs whose inverses effectively preserve recognizability has been also investigated. A linear growing TRS[24] has this property, and later, the result was extended to left-linear growing TRSs by Nagaya and Toyama[28]. The inverse of a (linear, left-linear) growing TRS is a (linear, right-linear) semi-monadic TRS and vice versa.

A TRS $\mathcal{R}$ is strongly normalizing if there is no infinite chain by the rewrite relation of $\mathcal{R}$. Strongly normalizing property is one of the most fundamental properties in the theory of TRSs. However, it is undecidable whether a given TRS has the strongly normalizing property or not, and this topic has been extensively studied. Those studies can be divided into two approaches. One approach is to give sufficient conditions to guarantee strongly normalizing property. Multiset ordering[9] is one of the techniques and other well-known (complete) techniques are dependency pair[1] and semantic labelling[36, 27]. The other approach is to propose (decidable) subclasses of TRSs for which the strongly normalizing property is decidable. For ground TRSs[22], right-ground TRSs[10], right-linear monadic TRSs[30], strongly normalizing property has been shown to be decidable. In 1999, Nagaya and Toyama showed that strongly normalizing property is decidable for almost orthogonal growing TRSs[28] as follows: It is well-known that for an almost orthogonal TRS, strongly normalizing is equivalent to inner-most weakly normalizing[20]. Nagaya and Toyama proposed a procedure which constructs a tree automaton accepting the inverse image of inner-most rewrite relation for a given left-linear growing TRS $\mathcal{R}$. Once such a tree automaton is constructed, it is not difficult to decide whether the TRS $\mathcal{R}$ is weakly inner-most normalizing or not by using properties of tree automata.

The latter half part of this thesis discusses the *inverse finitely path overlapping TRSs* (FPO^{-1}*-TRS*). It is shown that strongly normalization is decidable for almost orthogonal FPO^{-1}-TRSs. The proof is along with Toyama and Nagaya's method[28] mentioned above.

The remainder of this thesis is organized as follows. In Chapter 2, the terminologies and notations used throughout this thesis are introduced. After that, we define finitely path overlapping term rewriting systems (FPO-TRSs) and show that the class properly includes other known decidable subclasses of TRSs which effectively preserve recognizability. In Chapter 3, we prove that right-linear FPO-TRSs effectively preserve recognizability. In Chapter 4, we introduce the inverse FPO-TRSs (FPO^{-1}-TRSs) and show that strongly normalizing property is decidable for almost-orthogonal FPO^{-1}-TRSs. Chapter 5 concludes this thesis.

Chapter 2

Preliminaries

The chapter first introduces the terminologies and notations which we will use throughout this thesis. After that we define a new class of TRSs called finitely path overlapping TRSs.

The set of all natural numbers is denoted by $\mathcal{N}$. For a set A, the power set of A, the cardinality of A and the set consisting of all sequences over A are denoted by 2^A, $|A|$ and A^*, respectively. For a sequence A, the length of A is denoted by $|A|$. For two mappings σ and σ', their composition is denoted by $\sigma \circ \sigma'$. The empty set and the empty sequence are denoted by $\emptyset$ and λ, respectively. For a relation R, transitive clouser and the reflexive and transitive clouser of R are denoted by R^+ and R^*, respectively.

2.1. Term Rewriting System

A *signature* is a finite set in which each element is associated with a natural number. An element in a signature is called a *function symbol* and for a function symbol f the associated natural number of f is called the *arity of f* and denoted as $a(f)$. A function symbol f with $a(f) = 0$ is also called a *constant*. A set of *variables* is an enumerable set $\mathcal{V}$ such that $\mathcal{V} \cap \mathcal{F} = \emptyset$. In the following, we assume that $\mathcal{F}$ is a signature and $\mathcal{V}$ is a set of variables.

The set of all *terms on $\mathcal{F}$ and $\mathcal{V}$* is denoted as $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and recursively defined as follows:

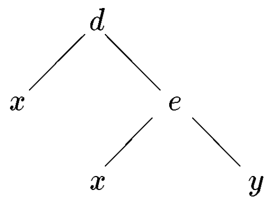


Figure 2.1. Tree representation for term $d(x, e(x, y))$.

1. If $x \in \mathcal{V}$, then $x \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.

2. If $f \in \mathcal{F}$ and $t_1, \ldots, t_{a(f)} \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, then $f(t_1, \ldots, t_{a(f)}) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$.

For a term $f()$ with $a(f) = 0$, we write f. Terms have tree structures. For example, a term $d(x, e(x, y))$ can be regarded as the tree in Figure 2.1 where d and e are function symbols with arity two and x and y are variables. A set of terms may be called a *tree language*. For a term s, a *position of s* is a sequence of natural numbers which indicates a certain subtree of s if we regard s as a tree. All positions of a term s is denoted as $\mathcal{P}os(s)$ and recursively defined as follows:

1. If t is a variable, then $\mathcal{P}os(t) = \{\lambda\}$.

2. If t is of the form $t = f(t_1, \ldots, t_{a(f)})$ where f is a function symbol and $t_1, \ldots, t_{a(f)}$ are terms, then $\mathcal{P}os(t) = \{\lambda\} \cup \bigcup_{1 \leq i \leq a(f)} \{i \cdot o \mid o \in \mathcal{P}os(t_i)\}$.

A *subterm of a term t at a position $o \in \mathcal{P}os(t)$* is denoted as t/o and defined as follows:

1. $t/\lambda = t$.

2. If $o = i \cdot o'$ and $t = f(t_1, \ldots, t_{a(f)})$ with $1 \leq i \leq a(f)$, then $t/o = t_i/o'$.

For a term t, the set $\mathcal{V}ar(t)$ consists of all variables appearing in t, i.e. $\mathcal{V}ar(t) = \{x \in \mathcal{V} \mid t/o = x, \exists o \in \mathcal{P}os(t)\}$. If $\mathcal{V}ar(t) = \emptyset$, then t is called *ground*. The set of all ground terms on signature $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F})$. A term s is *linear* if, for all $x \in \mathcal{V}ar(s)$, $|\{o \in \mathcal{P}os(s) \mid s/o = x\}| = 1$ holds.

The term obtained by replacing a subterm of t at a position o with a term s is denoted by $t[o \leftarrow s]$. A *context* is a term obtained by replacing a subterm of some term t with the special constant $\Box \notin \mathcal{F}$. A term obtained from a context C by replacing $\Box$ with a term s is denoted by $C[s]$. A relation R on terms is *closed under contexts* if for two terms s, t, $(s,t) \in R$ implies $(C[s], C[t]) \in R$ for any context C. A *substitution* σ is a mapping in $\mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ satisfying that $\{x \mid \sigma(x) \neq x\}$ is finite. For a substitution σ, $\{x \mid \sigma(x) \neq x\}$ is called the *domain of* σ. If the domain of a substitution σ is $\{x_1, \ldots, x_n\}$, then we may write $\{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}$ to represent σ. A substitution σ can be extended to a mapping $\sigma' : \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ in the unique way as follows:

1. If a term t is a variable, then $\sigma'(t) = \sigma(t)$.
2. If a term t is of the form $t = f(t_1, \ldots, t_{a(f)})$ where $f \in \mathcal{F}$ and $t_1, \ldots, t_{a(f)}$ are terms, then $\sigma'(t) = f(\sigma'(t_1), \ldots, \sigma'(t_{a(f)}))$.

For a term s and a substitution σ, we may write $s\sigma$ for $\sigma(s)$. For a term s, a term t is an *instance of* s if there is a substitution σ such that $\sigma(s) = t$. For two substitutions σ and σ', $\sigma \preceq \sigma'$ if there is a substitution σ'' such that $\sigma'' \circ \sigma = \sigma'$. For two terms s and t, σ is a *syntactic unifier of s and t* if $\sigma(s) = \sigma(t)$. Two terms s and t are *syntactically unifiable* if there is a syntactic unifier of s and t. A syntactic unifier σ of s and t are *most general* if, for any syntactic unifier σ', $\sigma \preceq \sigma'$.

A relation R on terms is *closed under substitutions* if for two terms s, t, $(s,t) \in R$ implies $(\sigma(s), \sigma(t)) \in R$ for any substitution σ. A relation on terms which is closed under both contexts and substitutions is called a *rewrite relation*.

Definition 2.1 A *term rewriting system* (abbreviated to *TRS*) is a finite relation on terms. For a TRS $\mathcal{R}$, the relation $\to_{\mathcal{R}}$ is the smallest rewrite relation containing $\mathcal{R}$. $\Box$

An element in a TRS is called a *rewrite rule*. A rewrite rule (l, r) is written as $l \to r$. In the following, we sometimes present a TRS as a set of rewrite rules and we assume that $\mathcal{R}$ is a TRS.

In most literatures on TRSs, the *variable restriction* for a rewrite rule $l \to r$ is assumed in the definition of TRSs, i.e.

1. $l \notin \mathcal{V}$ and

2. $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$.

In this thesis, we treat TRSs without the variable restriction unless stated otherwise.

For the inverse relation of $\mathcal{R}$, $\rightarrow_{\mathcal{R}}$, $\rightarrow^*_{\mathcal{R}}$, we may write $\mathcal{R}^{-1}$, $\leftarrow_{\mathcal{R}}$ and $\leftarrow^*_{\mathcal{R}}$, respectively. We denote the relation $\rightarrow_{\mathcal{R}} \cup \leftarrow_{\mathcal{R}}$ by $\leftrightarrow_{\mathcal{R}}$. It is not difficult to see that for two terms s and t, if $s \rightarrow_{\mathcal{R}} t$, then there is a substitution σ, a rewrite rule $l \rightarrow r \in \mathcal{R}$ and a position o such that $s/o = l\sigma$ and $t = s[o \leftarrow r\sigma]$.

Example 2.1 [25] Let $\mathcal{F} = \{d, e, k, r, m, \mathit{Alice}, \mathit{Bob}, \mathit{Chris}\}$ where $a(d) = a(e) = 2$, $a(k) = 1$ and $r, m, \mathit{Alice}, \mathit{Bob}, \mathit{Chris}$ are constants. Also let $\mathcal{V} = \{x, y, z\}$. Let us consider the following TRS $\mathcal{R}_0$ which appeared in Chapter 1.

$$\mathcal{R}_0 = \{\ d(x, e(x, y)) \rightarrow y\ \}.$$

For a ground term

$$d(d(k(\mathit{Chris}), e(k(\mathit{Chris}), d(k(\mathit{Alice}), e(k(\mathit{Alice}), r)))), e(r, m)),$$

we obtain the following sequence:

$$\begin{array}{rcl} d(d(k(\mathit{Chris}), e(k(\mathit{Chris}), d(k(\mathit{Alice}), e(k(\mathit{Alice}), r)))), e(r, m)) & \rightarrow_{\mathcal{R}} & \\ d(d(k(\mathit{Chris}), e(k(\mathit{Chris}), r)), e(r, m)) & \rightarrow_{\mathcal{R}} & \\ d(r, e(r, m)) & \rightarrow_{\mathcal{R}} & m. \end{array}$$

$\square$

A *redex* (*in* $\mathcal{R}$) is an instance of l for some $l \rightarrow r \in \mathcal{R}$. A *normal form* (*in* $\mathcal{R}$) is a term which has no redex as its subterm. Let $\mathit{NF}_{\mathcal{R}}$ denote the set of all ground normal forms in $\mathcal{R}$. For terms t, t' and a TRS $\mathcal{R}$, if $t = t[o \leftarrow l\sigma] \rightarrow_{\mathcal{R}} t[o \leftarrow r\sigma] = t'$ and t/o' is a normal form for any o' with $o' \in \mathcal{P}os(t)$ and $o \prec o'$, then we write $t \rightarrow_{I,\mathcal{R}} t'$ and the relation is called a *one-step innermost rewrite relation*.

Definition 2.2 For a TRS $\mathcal{R}$ and a term s:

1. s is *strongly normalizing* (*SN*) *in* $\mathcal{R}$ if there exists no infinite sequence $s_0 s_1 s_2 \cdots$ such that $s_0 = s$ and $s_i \rightarrow_{\mathcal{R}} s_{i+1}$ for all $i \geq 0$.

2. s is *weakly normalizing* (*WN*) *in* $\mathcal{R}$ if there exists a normal form t such that $s \rightarrow^*_{\mathcal{R}} t$.

3. s is *weakly innermost normalizing* (*WIN*) *in* $\mathcal{R}$ if there exists a normal form t such that $s \rightarrow^*_{I,\mathcal{R}} t$.

A TRS $\mathcal{R}$ is *strongly normalizing* (*SN*) (respectively *weakly normalizing* (*WN*), *weakly innermost normalizing* (*WIN*)) if every term is SN (respectively WN, WIN) in $\mathcal{R}$. □

The property SN is also called *termination*.

Theorem 2.1 *[22] The following problems are undecidable:*

1. *For a given TRS $\mathcal{R}$ and a term s, is s SN (respectively WN, WIN) in $\mathcal{R}$?*

2. *For a given TRS $\mathcal{R}$, is $\mathcal{R}$ SN (respectively WN, WIN)?* □

A rewrite rule $l \rightarrow r$ is *left-linear*(respectively *right-linear*) if l is linear (respectively r is linear). A TRS $\mathcal{R}$ is *left-linear* (respectively *right-linear*) if every rule in $\mathcal{R}$ is left-linear (respectively right-linear).

For a TRS $\mathcal{R}$, let $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ be (possibly the same) rewrite rules in $\mathcal{R}$ whose variables have been renamed to have no shared variables. If a non-variable subterm of l_1 at a position $o \in \mathcal{P}os(l_1)$ and l_2 are unifiable with a most general unifier σ, then the pair $r_1\sigma$ and $l_1\sigma[o \leftarrow r_2\sigma]$ is called a *critical pair of* $\mathcal{R}$ and is written as $\langle r_1\sigma, l_1\sigma[o \leftarrow r_2\sigma]\rangle$. If $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are the same rewrite rule, then we do not consider the case $o = \lambda$. A critical pair $\langle r_1\sigma, l_1\sigma[o \leftarrow r_2\sigma]\rangle$ is an *overlay* if $o = \lambda$. A critical pair $\langle t, t'\rangle$ is *trivial*, if $t = t'$.

Definition 2.3 [20, 23] A TRS $\mathcal{R}$ is:

1. *orthogonal* if $\mathcal{R}$ is left-linear and has no critical pairs.

2. *almost-orthogonal* (*AO*) if $\mathcal{R}$ is left-linear and every critical pair of $\mathcal{R}$ is a trivial overlay. □

The following lemmas concerning with one-step innermost rewrite relations can be easily understood.

Lemma 2.2 *For a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and a TRS $\mathcal{R}$, if a rewrite step $t[o \leftarrow l\sigma] \rightarrow_{\mathcal{R}} t[o \leftarrow r\sigma]$ is innermost at a position $o \in \mathcal{P}os(t)$ with a rewrite rule $l \rightarrow r \in \mathcal{R}$ and a substitution σ, then $l\sigma \rightarrow_{\mathcal{R}} r\sigma$ is innermost.* $\Box$

Lemma 2.3 *Let $\mathcal{R}$ be an AO-TRS. For two terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and a rewrite rule $l \rightarrow r \in \mathcal{R}$ if $s/o = l\sigma, t = s[o \leftarrow r\sigma]$ where $o \in \mathcal{P}os(s)$, $\sigma = \{x_i \mapsto t_i \mid 1 \leq i \leq n\}$, then $s \rightarrow_{I,\mathcal{R}} t$ if and only if $t_i \in NF_{\mathcal{R}}$ for $1 \leq i \leq n$.* $\Box$

Definition 2.4 For a TRS $\mathcal{R}$ and two terms s and t:

1. *s and t are reachable in $\mathcal{R}$* if $s \rightarrow^*_{\mathcal{R}} t$ or $t \rightarrow^*_{\mathcal{R}} s$.

2. *s and t are joinable in $\mathcal{R}$* if there is a term u such that $s \rightarrow^*_{\mathcal{R}} u$ and $t \rightarrow^*_{\mathcal{R}} u$.

3. A *(semantic) unifier of s and t in $\mathcal{R}$* is a substitution σ such that $\sigma(s) \leftrightarrow^*_{\mathcal{R}} \sigma(t)$. *$s$ and t are unifiable in $\mathcal{R}$* if there is a unifier of s and t in $\mathcal{R}$. $\Box$

A unifier σ of s and t in $\mathcal{R}$ is *most general* if, for any unifier σ' of s and t in $\mathcal{R}$, $\sigma \preceq \sigma'$ holds.

Theorem 2.4 *For a given TRS $\mathcal{R}$ and two terms s and t, the following problems are undecidable:*

1. *Are s and t reachable in $\mathcal{R}$?*

2. *Are s and t joinable in $\mathcal{R}$?*

3. *Are s and t unifiable in $\mathcal{R}$?* $\Box$

Definition 2.5 For a TRS $\mathcal{R}$:

1. $\mathcal{R}$ is *confluent* if, for terms s, t and t', $s \rightarrow^*_{\mathcal{R}} t$ and $s \rightarrow^*_{\mathcal{R}} t'$ then t and t' are joinable.

2. $\mathcal{R}$ is *locally confluent* if, for terms s, t and t', $s \rightarrow_{\mathcal{R}} t$ and $s \rightarrow_{\mathcal{R}} t'$ then t and t' are joinable. $\Box$

Theorem 2.5 *The following problems are undecidable:*

1. *Is a given TRS $\mathcal{R}$ confluent?*

2. *Is a given TRS $\mathcal{R}$ locally confluent?* $\Box$

2.2. Tree Automaton

Tree automata are natural generalization of traditional finite-state automata on strings. A tree automaton accepts terms instead of strings and can be defined as a TRS[5]. A *state* is a special constant not in $\mathcal{F}$. For a finite set $\mathcal{Q}$ of states, ground terms on $\mathcal{F} \cup \mathcal{Q}$, i.e. terms in $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$, are called *$\mathcal{Q}$-terms*.

Definition 2.6 A *tree automaton* (abbreviated to *TA*) is given by a 4-tuple $(\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$ where $\mathcal{F}$ is a signature, $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_{\mathit{final}}$ is a subset of $\mathcal{Q}$ and Δ is a TRS constructed from $\mathcal{Q}$-terms in which each rewrite rule has the form either:

1. $f(q_1, \ldots, q_{a(f)}) \to q$ or
2. $q' \to q$

where f is a function symbol and $q_1, \ldots, q_n, q$ and q' are states. $\square$

An element in $\mathcal{Q}_{\mathit{final}}$ and an element in Δ are called a *final state* and a *transition rule*, respectively. The behaviors of tree automata are defined as follows.

Definition 2.7 Let $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$ be a tree automaton. For a ground term s, *s is accepted by $\mathcal{A}$* if $s \to_\Delta^* q_f$ for some final state $q_f \in \mathcal{Q}_{\mathit{final}}$. The *accepting language of* $\mathcal{A}$ is the set of all ground terms accepted by $\mathcal{A}$. $\square$

For a tree automaton $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$ we call the relation $\to_\Delta$ a *move* and we may write $\vdash_\Delta$ or $\vdash_{\mathcal{A}}$ for $\to_\Delta$. The accepting language of a tree automaton $\mathcal{A}$ is denoted as $\mathcal{L}(\mathcal{A})$. i.e. $\mathcal{L}(\mathcal{A}) = \{t \mid t \vdash_{\mathcal{A}}^* q_f, \exists q_f \in \mathcal{Q}_{\mathit{final}}\}$. Also let $\mathcal{L}_q(\mathcal{A}) = \{t \mid t \vdash_{\mathcal{A}}^* q\}$ for a state q of $\mathcal{A}$.

By using the notion of TAs, we can define a class of sets of terms.

Definition 2.8 A set L of terms is *recognizable* if there is a tree automata $\mathcal{A}$ such that $L = \mathcal{L}(\mathcal{A})$. $\square$

Example 2.2 Let $\mathcal{F}$ and $\mathcal{V}$ be the signature and the variables in Example 2.1, respectively. The TA $\mathcal{B}_1 = (\mathcal{F}, \mathcal{Q}_0, \mathcal{Q}_0, \Delta_0)$ accepts $\mathcal{T}(\mathcal{F})$, the set of all ground terms, where $\mathcal{Q} = \{q\}$ and Δ_0 consists of the following transition rules:

$$\begin{array}{lllllll} d(q,q) & \to & q, & e(q,q) & \to & q \\ k(q) & \to & q, & r & \to & q \\ m & \to & q, & Alice & \to & q \\ Bob & \to & q, & Chris & \to & q. \end{array}$$

For a term $s_1 = d(x, e(y, z))$, let L_1 be the set of all ground instances of s_1, then L_1 is recognizable since the TA $\mathcal{B}_2 = (\mathcal{F}, \mathcal{Q}_0 \cup \mathcal{Q}_1, \{q_f\}, \Delta_0 \cup \Delta_1)$ accepts L_1 where $\mathcal{Q}_0$ and Δ_0 are the same as in $\mathcal{B}_1$, $\mathcal{Q}_1 = \{q_1, q_f\}$ and Δ_1 consists of the following transition rules:

$$e(q,q) \;\to\; q_1, \quad d(q,q_1) \;\to\; q_f.$$

On the other hand, for a term $s_2 = d(x, e(x, y))$, let L_2 be the set of all ground instances of s_2, then L_2 is not recognizable since there is no tree automaton which accepts L_2. □

Recognizable sets inherit some useful properties of regular (string) languages[16].

Lemma 2.6 *The class of recognizable sets is effectively closed under union, intersection and complementation. For a recognizable set L, the following problems are decidable.*

1. *Does a given ground term s belong to L?*
2. *Is L empty?* □

The following lemmas are easily understood.

Lemma 2.7 *The set of all ground instances of a linear term is recognizable.* □

Lemma 2.8 *[15] For a left-linear TRS $\mathcal{R}$, $NF_{\mathcal{R}}$ is recognizable.* □

2.3. TRS which Preserves Recognizability

Let L be a set of terms and $\mathcal{R}$ be a TRS. The *descendant of L by $\mathcal{R}$* is denoted by $(\to^*_{\mathcal{R}})(L)$ and defined as $(\to^*_{\mathcal{R}})(L) = \{t \mid \exists s \in L, s \to^*_{\mathcal{R}} t\}$. The *ancestor of L by $\mathcal{R}$* is denoted by $(\leftarrow^*_{\mathcal{R}})(L)$ and defined as $(\leftarrow^*_{\mathcal{R}})(L) = \{t \mid \exists s \in L, t \to^*_{\mathcal{R}} s\}$.

Definition 2.9 A *TRS* $\mathcal{R}$ *effectively preserves recognizability* if for any tree automaton $\mathcal{A}$ we can effectively construct a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\rightarrow_{\mathcal{R}}^*)(\mathcal{L}(\mathcal{A}))$. □

Remark that Gyenizse and Vágvölgyi[21] introduced the notion *preserving $\mathcal{F}$-recognizability* and showed that there is a difference between the notions preserving $\mathcal{F}$-recognizability and preserving recognizability. Let $\mathcal{F}$ be a signature. A *TRS* $\mathcal{R}$ *effectively preserves $\mathcal{F}$-recognizability* if for any tree automaton $\mathcal{A}$ whose accepting language is over $\mathcal{F}$ we can effectively construct a tree automaton $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\rightarrow_{\mathcal{R}}^*)(\mathcal{L}(\mathcal{A}))$. For example, let a signature $\mathcal{F} = \{f, g, a\}$ with $a(f) = 1, a(g) = 1, a(a) = 0$ and a TRS $\mathcal{R} = \{f(x) \rightarrow g(f(g(x))), f(a) \rightarrow a, a \rightarrow f(a), g(a) \rightarrow a, a \rightarrow g(a)\}$. We can see that the TRS $\mathcal{R}$ effectively preserves $\mathcal{F}$-recognizability since descendants by $\mathcal{R}$ of any tree language on $\mathcal{F}$ is obviously $\mathcal{T}(\mathcal{F})$. On the other hand, let $\mathcal{F}' = \mathcal{F} \cup \{c\}$ with $a(c) = 0$, then the descendant by $\mathcal{R}$ of the tree language $\{f(c)\}$ is $(\rightarrow_{\mathcal{R}}^*)(\{f(c)\}) = \{g^n(f(g^n(c))) \mid n \geq 0\}$; this implies that $\mathcal{R}$ does not effectively preserve recognizability.

In this thesis, we consider the class of TRSs which effectively preserve recognizability and we write the class as EPR-TRSs.

Theorem 2.9 *[19, 21, 28] The following problems are decidable:*

1. *For a given EPR-TRS $\mathcal{R}$ and two terms s and t:*
   (a) *Are s and t reachable in $\mathcal{R}$?*
   (b) *Are s and t reachable in $\mathcal{R}^{-1}$?*
   (c) *Are s and t joinable in $\mathcal{R}$?*
2. *For a given EPR-TRS $\mathcal{R}$, is $\mathcal{R}$ locally confluent?* □

We show some more properties of EPR-TRSs.

Theorem 2.10 *Let $\mathcal{R}$ be a left-linear TRS such that $\mathcal{R}^{-1}$ is an EPR-TRS, then the following problem is decidable:*

1. *For a term s, is s WN in $\mathcal{R}$?*
2. *Is $\mathcal{R}$ WN?*

Proof. It is easily understood that $\mathcal{T}(\mathcal{F}) = (\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})$ if and only if $\mathcal{R}$ is WN. On the other hand, by Lemma 2.8, the set $NF_{\mathcal{R}}$ of normal forms in $\mathcal{R}$ is recognizable since $\mathcal{R}$ is left-linear. Since $(\leftarrow^*_{\mathcal{R}})(L) = (\rightarrow^*_{\mathcal{R}^{-1}})(L)$ for any set L of terms and $\mathcal{R}^{-1}$ is in EPR-TRS, $(\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})$ is recognizable. For the first part, we can see that s is WN if and only if $s \in (\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})$ and the membership problem is decidable by Lemma 2.6. For the second part, note that $\mathcal{T}(\mathcal{F}) = (\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})$ if and only if $\overline{(\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})} = \emptyset$. Hence, $\mathcal{T}(\mathcal{F}) = (\leftarrow^*_{\mathcal{R}})(NF_{\mathcal{R}})$ is decidable by Lemma 2.6. □

Theorem 2.11 *For a confluent $\mathcal{R} \in$ EPR-TRS and linear terms t_1 and t_2 with $\mathcal{V}ar(t_1) \cap \mathcal{V}ar(t_2) = \emptyset$, the following problem is decidable: Are t_1 and t_2 unifiable in $\mathcal{R}$?*

Proof. Since $\mathcal{R}$ is confluent, t_1 and t_2 are unifiable in $\mathcal{R}$ if and only if there exists a substitution σ and a term v such that $t_1\sigma \rightarrow^*_{\mathcal{R}} v$ and $t_2\sigma \rightarrow^*_{\mathcal{R}} v$. For a term t, let $I(t)$ denote the set of ground instances of t, i.e., $I(t) = \{t\sigma \in \mathcal{T}(\mathcal{F}) \mid \sigma \text{ is a substitution}\}$. Then t_1 and t_2 are unifiable in $\mathcal{R}$ if and only if

$$(\rightarrow^*_{\mathcal{R}})(I(t_1)) \cap (\rightarrow^*_{\mathcal{R}})(I(t_2)) \neq \emptyset \tag{2.1}$$

since $\mathcal{V}ar(t_1) \cap \mathcal{V}ar(t_2) = \emptyset$. Moreover, both $I(t_1)$ and $I(t_2)$ are recognizable by Lemma 2.8. Thus $(\rightarrow^*_{\mathcal{R}})(I(t_1))$ and $(\rightarrow^*_{\mathcal{R}})(I(t_2))$ are recognizable since $\mathcal{R} \in$ EPR-TRS. By Lemma 2.6, the condition (2.1) is decidable. □

Theorem 2.12 *[7] For a TRS $\mathcal{R}$, the following problem is undecidable: Does $\mathcal{R}$ effectively preserve recognizability?* □

In the following, we review some classes of TRSs which have been proposed.

Definition 2.10 A rewrite rule is *ground* (respectively *linear*) if both the left- and right-hand sides are ground (respectively linear). A *ground* (respectively *linear*) *TRS* consists of ground (respectively linear) rewrite rules. The class of ground TRSs (respectively linear TRSs) is denoted as G-TRS (respectively L-TRS). □

Example 2.3 Let $\mathcal{F} = \{f, g, h, a, c\}$ be a signature such that $a(f) = 2, a(g) = 1, a(h) = 1$ and a and c are constants. Also let $\mathcal{V} = \{x, y, z\}$ be a set of variables. The TRS $\mathcal{R}_1$ below is ground but $\mathcal{R}_2$ is not ground. Both $\mathcal{R}_1$ and $\mathcal{R}_2$ are linear.

$$\mathcal{R}_1 = \begin{cases} g(c) \to f(g(c), c), \\ f(c, a) \to f(f(f(a, c), a), g(c)), \end{cases}$$

$$\mathcal{R}_2 = \begin{cases} g(x) \to f(g(x), c), \\ f(x, g(y)) \to f(f(x, c), g(y)). \end{cases}$$

□

For a term s, the *depth of* s is denoted by $depth(s)$ $depth(s) = \max\{|o| \mid o \in \mathcal{P}os(s)\}$.

Definition 2.11 [29] A rewrite rule is *monadic* if it satisfies the variable restriction, the depth of the left-hand side is at least one and the depth of the right-hand side is at most one. A TRS is *monadic* if it consists of monadic rewrite rules. □

The class of monadic TRSs (respectively right-linear monadic TRSs) is denoted by M-TRS (respectively RL-M-TRS).

Example 2.4 Consider the signature and variables which are the same as in Example 2.3. The TRS $\mathcal{R}_3$ below is an example of an RL-M-TRS.

$$\mathcal{R}_3 = \begin{cases} g(f(x, y)) \to g(x), \\ f(f(f(x, x), y), g(c)) \to f(x, y). \end{cases}$$

□

Definition 2.12 [6] A rewrite rule is *semi-monadic* if it satisfies the variable restriction, the depth of the left-hand side is at least one and the depth of the right-hand side is zero (i.e. it is a variable or a constant) or the right-hand side is of the form $f(t_1, \ldots, t_{a(f)})$ where t_i $(1 \leq i \leq a(f))$ is either a variable or a ground term. □

The class of semi-monadic TRSs is denoted by SM-TRS and the class of TRSs in L-TRS $\cap$ SM-TRS is denoted by L-SM-TRS.

Example 2.5 Consider the signature and variables which are the same as in Example 2.3. The TRS $\mathcal{R}_4$ below is an example of an L-SM-TRS which is not an M-TRS.

$$\mathcal{R}_4 = \begin{cases} g(f(x,c)) \rightarrow g(a), \\ f(f(f(x,y),z),g(c)) \rightarrow f(g(f(a,c)),x). \end{cases}$$

$\square$

Definition 2.13 [21] A TRS $\mathcal{R}$ is *generalized semi-monadic* if it satisfies the variable restriction and, for any pair of rewrite rules $l_1 \rightarrow r_1, l_2 \rightarrow r_2$ in $\mathcal{R}$, the following holds: For any positions $\alpha \in \mathcal{P}os(r_1)$ and $\beta \in \mathcal{P}os(l_2)$ and for any term $l_3 \in \mathcal{T}(\mathcal{F},\mathcal{V})$ satisfying that there is a substitution θ such that $\theta(l_3) = l_2/\beta$ and $\mathcal{V}ar(l_3) \cap \mathcal{V}ar(l_1) = \emptyset$,

1. $\alpha = \lambda$ or $\beta = \lambda$ and
2. r_1/α and l_3 are syntactically unifiable with most general unifier σ,

then

(a) $l_2/\beta \in \mathcal{V}$ or

(b) for each $\gamma \in \mathcal{P}os(l_3)$, $l_2/\beta \cdot \gamma \in \mathcal{V}$ implies $(l_3/\gamma)\sigma \in \mathcal{V} \cup \mathcal{T}(\mathcal{F})$. $\square$

The class of generalized semi-monadic TRSs is denoted by GSM-TRS and the class of TRS in L-TRS $\cap$ GSM-TRS is denoted by L-GSM-TRS.

Example 2.6 Consider the signature and variables which are the same as in Example 2.3. The TRS $\mathcal{R}_5$ below is an example of an L-GSM-TRS which is not an SM-TRS.

$$\mathcal{R}_5 = \begin{cases} g(g(f(x,a))) \rightarrow g(g(x)), \\ f(x,y) \rightarrow g(f(x,a)). \end{cases}$$

$\square$

Theorem 2.13 *[3, 6, 21, 29] RL-M-TRS $\subset$ EPR-TRS, and G-TRS $\subset$ L-SM-TRS $\subset$ L-GSM-TRS $\subset$ EPR-TRS.* $\square$

Remark that Salomaa[29] proved that any right-linear monadic TRS which consists of possibly infinitely many rewrite rules effectively preserves recognizability.

There is another stream of studies which relate TRSs and recognizability[24, 14, 28].

Definition 2.14 [24] A TRS $\mathcal{R}$ is *growing* if all variables in $\mathcal{V}ar(l) \cap \mathcal{V}ar(r)$ occur at depth 0 or 1 in l for every rewrite rule $l \to r$ in $\mathcal{R}$. $\square$

Jacquemard[24] showed that, for any linear growing TRS $\mathcal{R}$, $\mathcal{R}^{-1}$ effectively preserves recognizability and this result was extended by Nagaya and Toyama[28] as follows.

Theorem 2.14 *[28] For any left-linear growing TRS (LL-GR-TRS) $\mathcal{R}$, $\mathcal{R}^{-1}$ effectively preserves recognizability.* $\square$

Note that in Definition 2.14, the variable restriction is not assumed. It is easy to see the following holds from Definitions 2.12 and 2.14.

Lemma 2.15 *If a TRS $\mathcal{R}$ satisfies the variable restriction then $\mathcal{R}$ is (linear, right-linear) semi-monadic if and only if $\mathcal{R}^{-1}$ is (linear, left-linear) growing and the left-hand side of every rewrite rule in $\mathcal{R}$ is not a constant.* $\square$

As a result, RL-GR^{-1}-TRS (i.e. the class of the inverses of LL-GR-TRSs) properly includes both of RL-M-TRS and L-SM-TRS, and it is incomparable with L-GSM-TRS. By Theorem 2.14 and Lemma 2.15, the following corollary is directly obtained.

Corollary 2.16 *RL-SM-TRS $\subset$ RL-GR^{-1}-TRS $\subset$ EPR-TRS.* $\square$

2.4. Finitely Path Overlapping TRS

A new class of TRS named *finitely path overlapping TRS* (*FPO-TRS*) is proposed in this section. As we will show later, the class of RL-FPO-TRS properly includes the class of RL-GSM-TRS and RL-GR^{-1}-TRS. It will also be shown in the next chapter that an RL-FPO-TRS (without the variable restriction) is an EPR-TRS. To the author's knowledge, the proposed class is the largest decidable subclass of EPR-TRS.

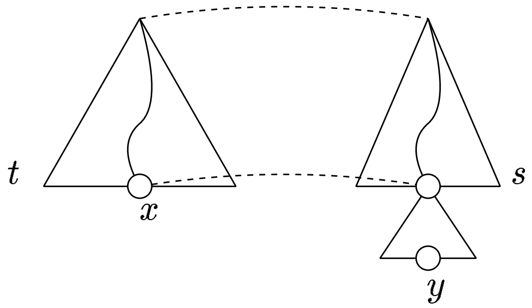


Figure 2.2. s sticks out of t.

2.4.1 Definitions

To define the class, some additional definitions are necessary.

Definition 2.15 For two terms s and t, s *sticks out of* t if t is not a variable and there is a variable position γ ($\neq \lambda$) of t such that

1. for any position o with $\lambda \preceq o \prec \gamma$, we have $o \in \mathcal{P}os(s)$ and the function symbol of s at o and the function symbol of t at o are the same, and

2. $\gamma \in \mathcal{P}os(s)$ and s/γ is not a ground term.

If s sticks out of t at γ and s/γ is not a variable (i.e. s/γ is a non-ground and non-variable term), then s is said to *properly stick out of* t □

When the position γ is of interest, we say that s sticks out of t at γ. The sticking out relation is illustrated in Figure 2.2.

Example 2.7 A term $f(g(x), a)$ sticks out of $f(g(y), b)$ at the position $1 \cdot 1$, and $f(g(g(x)), a)$ properly sticks out of $f(g(y), b)$ at the position $1 \cdot 1$. □

Using the notion of sticking out relatoin, we define a *sticking-out graph* for a TRS.

Definition 2.16 The *sticking-out graph* of a TRS $\mathcal{R}$ is a directed graph $G = (V, E)$ where $V = \mathcal{R}$ (i.e. the vertices are the rewrite rules in $\mathcal{R}$) and E is

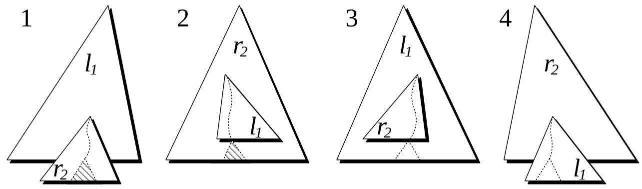


Figure 2.3. The sticking-out relations of rewrite rules.

defined as follows. Let v_1 and v_2 be (possibly identical) vertices which correspond to rewrite rules $l_1 \to r_1$ and $l_2 \to r_2$, respectively. Replace each variable in $\mathcal{V}ar(r_i) \setminus \mathcal{V}ar(l_i)$ with a fresh constant, say $\Diamond$, for $i = 1, 2$.

1. If r_2 properly sticks out of a subterm of l_1, then E contains an edge from v_2 to v_1 with weight one.

2. If a subterm of r_2 properly sticks out of l_1, then E contains an edge from v_2 to v_1 with weight one.

3. If a subterm of l_1 sticks out of r_2, then E contains an edge from v_2 to v_1 with weight zero.

4. If l_1 sticks out of a subterm of r_2, then E contains an edge from v_2 to v_1 with weight zero. □

The four cases are illustrated in Fig. 2.3.

Definition 2.17 A *finitely path overlapping term rewriting system* (*FPO-TRS*) is a TRS $\mathcal{R}$ such that the sticking-out graph of $\mathcal{R}$ does not have a cycle of weight one or more. □

An RL-TRS (right-linear TRS) being FPO is written as RL-FPO-TRS.

Example 2.8 Let $\mathcal{R}_6$ be a TRS consisting of the following rewrite rules p_1 and p_2:

$$\begin{array}{lll} p_1\colon & f(x,a) & \to & f(h(y),x), \\ p_2\colon & g(y) & \to & f(g(y),b). \end{array}$$

Figure 2.4 shows the sticking-out graph of $\mathcal{R}_6$. The right-hand side of p_2 properly sticks out of the left-hand side of p_1 at the position 1, and hence there is an edge

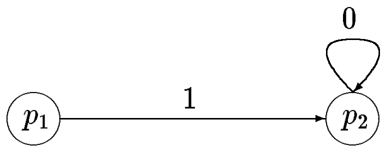


Figure 2.4. The sticking-out graph of $\mathcal{R}_6$.

of weight one from p_2 to p_1. The sticking-out graph also has a self-looping edge of weight zero at p_2 since the left-hand side $g(y)$ of p_2 sticks out of $f(g(y), b)/1 = g(y)$. Since the variable y in p_1 is replaced with a constant $\Diamond$, the right-hand side of p_1 does not stick out of its left-hand side. There is no other edge since there is no other sticking-out relation between subterms of these rewrite rules. The sticking-out graph has a cycle of weight zero, but does not have a cycle of weight one or more, and hence $\mathcal{R}_6$ is finitely path overlapping. Let $\mathcal{R}_7 = \{f(x) \rightarrow g(f(g(x)))\}$. The subterm $f(g(x))$ of the right-hand side of the (unique) rewrite rule properly sticks out of its left-hand side, as in Condition 2 of the definition of sticking-out graph. The sticking-out graph of $\mathcal{R}_7$ consists of one vertex and one cycle with weight one. Therefore, $\mathcal{R}_7$ is not finitely path overlapping. Note that $\mathcal{R}_7 \notin$ EPR-TRS since $(\rightarrow^*_{\mathcal{R}_7})(\{f(a)\}) = \{g^n(f(g^n(a))) \mid n \geq 0\}$ is not recognizable. $\square$

Remark that the sticking-out graph is effectively constructible for a given TRS $\mathcal{R}$, and hence it is decidable whether a given TRS $\mathcal{R}$ is finitely path overlapping or not (in $O(m^2n^2)$ time where m is the maximum size of a term in $\mathcal{R}$ and n is the number of rules in $\mathcal{R}$).

In the following, it is shown that any generalized semi-monadic TRS is an FPO-TRS, which implies that the class of FPO-TRS include the class of generalized semi-monadic TRS. A simple example shows that the inclusion relation is proper.

In the next chapter, it is shown that if a TRS is a right-linear FPO-TRS, then it effectively preserves recognizability. Summarizing these results, the class of right-linear FPO-TRS is a decidable subclass of EPR-TRS and properly contains the class of linear generalized semi-monadic TRS and right-linear monadic TRS.

2.4.2 Hierarchical Relation

Although a generalized semi-monadic TRS (GSM-TRS) was originally defined in [21] with the variable restriction as in Definition 2.13, we give another definition of GSM-TRS without the variable restriction in the following lemma to treat growing TRS, GSM-TRS and FPO-TRS in a uniform way.

Lemma 2.17 *A TRS $\mathcal{R}$ is in GSM-TRS if and only if the sticking-out graph of $\mathcal{R}$ has no edge with weight one. If a TRS $\mathcal{R}$ is generalized semi-monadic, then $\mathcal{R}$ is finitely path overlapping.*

Proof. We show the only if part by contradiction. If part can be shown in a similar way. Assume that $\mathcal{R}$ is a GSM-TRS and contains rules $l_1 \to r_1$ and $l_2 \to r_2$ (each variable in $\mathcal{V}ar(r_i) \setminus \mathcal{V}ar(l_i)$ has been replaced with a constant $\Diamond$ for $i = 1, 2$) which satisfy condition 1 of the definition of sticking-out graph. In this case, there is a position $\alpha \in \mathcal{P}os(l_1)$ such that r_2 properly sticks out of l_1/α. Let γ be the variable position of l_1/α at which r_2 properly sticks out of l_1/α, then $l_1/\alpha \cdot \gamma$ is a variable and r_2/γ is a non-ground and non-variable term. Let l_3 be the term which satisfies the following conditions:

1. For a position o with $\lambda \preceq o \prec \gamma$, l_3 and l_1/α have the same symbol at o,
2. a variable, say x_o, occurs at a position o which is disjoint to γ and is written as $o' \cdot i$ with $o' \prec \gamma$ and
3. a variable x_γ occurs at γ.

It is easily understood that l_1/α is an instance of l_3 and that l_3 and r_2 are syntactically unifiable by an mgu σ which in particular replaces x_γ by r_2/γ. Now we have $(l_3/\gamma)\sigma = r_2/\gamma$, which is neither a variable nor a ground term by assumption. This concludes that $\mathcal{R}$ is not a GSM-TRS. In a similar way, we can show that if any pair of rules in $\mathcal{R}$ satisfy the condition 2 of the definition of sticking-out graph, then $\mathcal{R}$ is not a GSM-TRS. □

Theorem 2.18 *RL-GR^{-1}-TRS $\subset$ RL-GSM-TRS $\subset$ RL-FPO-TRS.*

Proof. The first part is directly obtained from the definitions.

The class of RL-FPO-TRS includes the class of RL-GSM-TRS by Lemma 2.17. TRS $\mathcal{R}_6$ in Example 2.8 is RL-FPO but not GSM. If we take $l_1 = f(x, a)$, $r_2 = f(g(y), b)$, $\alpha = \beta = \lambda$ and $l_3 = f(x, z)$, then r_2 and l_3 are unifiable by an mgu $\sigma = \{x \mapsto g(y), z \mapsto b\}$. Let $\gamma = 1$, then $l_1/\alpha \cdot \gamma = l_1/1$ is a variable x while $(l_3/\gamma)\sigma = g(y)$ is neither a variable nor a ground term. Therefore $\mathcal{R}_6$ is not a GSM-TRS. $\square$

The hierarchical relation among the class of TRSs mentioned in this chapter is illustrated in Figure 1.1 in Chapter 1.

Chapter 3

Recognizability Preserving Property

In this chapter, it is shown that any RL-FPO-TRS effectively preserves recognizability.

3.1. Tree Automata Construction for Descendants

In this section, we will show that every RL-FPO-TRS $\mathcal{R}$ belongs to EPR-TRS by constructing a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$ for a given TA $\mathcal{A}$.

To deal with non-left-linear TRS, we need to construct a kind of product automata whose states are Cartesian products of sets of terms. To represent such a Cartesian product and a usual first-order term in a uniform way, we introduce a packed state. Intuitively, a packed state is an extension of a first-order term such that a finite set of terms, rather than a single term, occurs at a subterm position.

Definition 3.1 For a signature $\mathcal{F}$ and a finite set $\mathcal{Q}$, the set of *packed states*, denoted $\mathcal{P}_{\mathcal{F},\mathcal{Q}}$, is defined as follows:

1. If $q \in \mathcal{Q}$, then $\{q\} \in \mathcal{P}_{\mathcal{F},\mathcal{Q}}$.

2. If $f \in \mathcal{F}$ and $p_1, \ldots, p_{a(f)} \in \mathcal{P}_{\mathcal{F},\mathcal{Q}}$, then $\{f(p_1, \ldots, p_{a(f)})\} \in \mathcal{P}_{\mathcal{F},\mathcal{Q}}$.

3. If $p_1, p_2 \in \mathcal{P}_{\mathcal{F},\mathcal{Q}}$, then $p_1 \cup p_2 \in \mathcal{P}_{\mathcal{F},\mathcal{Q}}$. $\square$

For the readability, a packed state $\{t_1, \ldots, t_n\}$ is written as $\langle t_1, \ldots, t_n \rangle$.

Example 3.1 Let $\mathcal{F}$ the signature in Example 2.3 in Chapter 2 and $\mathcal{Q} = \{q_1, q_2\}$. We can easily verify that $\langle f(\langle q_1 \rangle, \langle q_2 \rangle), g(\langle g(\langle q_1 \rangle), \langle q_2 \rangle \rangle) \rangle$ belongs to $\mathcal{P}_{\mathcal{F},\mathcal{Q}}$. $\square$

Procedure 3.1 (Tree automata Construction)

Input: a TA $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$ and an RL-TRS $\mathcal{R}$

Output: a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\to_{\mathcal{R}}^*)(\mathcal{L}(\mathcal{A}))$

Step 1. Add a new state q_{any} to $\mathcal{Q}$ and add a transition rule $f(q_{any}, \ldots, q_{any}) \to q_{any}$ to Δ for each f in $\mathcal{F}$. Obviously, $t \vdash_{\mathcal{A}}^* q_{any}$ for any $t \in \mathcal{T}(\mathcal{F})$. Let $\mathcal{A}_0 = (\mathcal{F}, \mathcal{Q}_0, \mathcal{Q}_{\mathit{final}}^0, \Delta_0)$ be a "packed" version of $\mathcal{A}$ where $\mathcal{Q}_0 = \{\langle q \rangle \mid q \in \mathcal{Q}\} \subseteq \mathcal{P}_{\mathcal{F},\mathcal{Q}}$, $\mathcal{Q}_{\mathit{final}}^0 = \{\langle q \rangle \mid q \in \mathcal{Q}_{\mathit{final}}\}$, and $\Delta_0 = \{f(\langle q_1 \rangle, \ldots, \langle q_n \rangle) \to \langle q \rangle \mid f(q_1, \ldots, q_n) \to q \in \Delta\} \cup \{\langle q' \rangle \to \langle q \rangle \mid q' \to q \in \Delta\}$.

Step 2. Let $k = 0$. This k is used as a loop counter.

Step 3. Let $\mathcal{Q}_{k+1} = \mathcal{Q}_k$ and $\Delta_{k+1} = \Delta_k$.

Step 4. The set of transition rules is modified in this step. Let $l \to r$ be a rewrite rule in $\mathcal{R}$. Assume l has m variables $x_1, \ldots, x_m$ and x_i $(1 \le i \le m)$ occurs for γ_i times at positions o_{ij} $(1 \le j \le \gamma_i)$ in l. Also assume x_i occurs at o_i in r for $x_i \in \mathcal{V}ar(r)$. If there are states $p_{ij}, p \in \mathcal{Q}_k$ with $1 \le i \le m, 1 \le j \le \gamma_i$,

$$l[o_{ij} \leftarrow p_{ij} \mid 1 \le i \le m, 1 \le j \le \gamma_i] \vdash_k^* p \tag{3.1}$$

and

$$\mathcal{L}_{p_{i1}}(\mathcal{A}_k) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_k) \ne \emptyset \tag{3.2}$$

for $1 \le i \le m$, then add

$$p_i = \bigcup_{1 \le j \le \gamma_i} p_{ij} \quad (1 \le i \le m) \tag{3.3}$$

to $\mathcal{Q}_{k+1}$ as new states and let $\rho = \{x_i \mapsto p_i \mid 1 \le i \le m\} \cup \{x \mapsto \langle q_{any} \rangle \mid x \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)\}$. If r is a variable, then let $t_{r\rho} = r\rho$. Otherwise, let $t_{r\rho} = \langle r\rho \rangle$. Do the following (a) through (c).

(a) Add $t_{r\rho} \to p$ to Δ_{k+1}.

(b) Let $p = \langle t_1, \ldots, t_n \rangle$. Add $t_{r\rho} \to \langle t_i \rangle$ to Δ_{k+1} for $1 \le i \le n$. A transition rule defined in (a) or (b) is called a *rewriting transition rule* of *degree* $k+1$ and if a move of the TA is caused by such a rule, then the move is called a *proper rewriting move* of *degree* $k+1$.

(c) Execute **ADDTRANS**$(t_{r\rho})$. In **ADDTRANS**$(t_{r\rho})$, new states and transition rules are defined so that $r\rho \vdash_{k+1}^{*} t_{r\rho}$.

Simultaneously execute this Step 4 for every rewrite rule and every tuple of states that satisfy conditions (3.1) and (3.2).

Step 5. Continue the loop until $\Delta_{k+1} = \Delta_k$. If $\Delta_{k+1} \neq \Delta_k$, then $k = k+1$ and go to Step 3.

Step 6. Output $\mathcal{A}_k$ as $\mathcal{A}_*$. $\square$

Procedure 3.2 [**ADDTRANS**] This procedure takes a packed state p as an input. If p has already been defined as a state, then the procedure performs nothing. Otherwise, the procedure first defines p as a new state of $\mathcal{Q}_{k+1}$ and also defines transition rules as follows. It is required that if $p = \langle t_1, \ldots, t_n \rangle$ $(n \ge 2)$, then each $\langle t_i \rangle$ has been defined as a state.

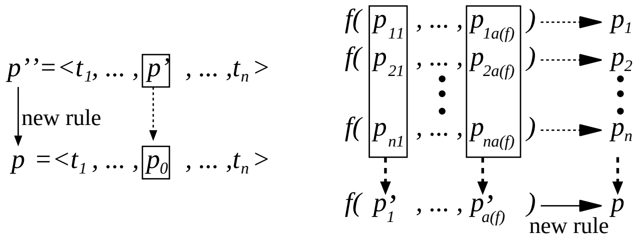
Case 1. If $p = \langle c \rangle$ with c a constant, then define $c \to \langle c \rangle$ as a transition rule.

Case 2. If $p = \langle f(p_1, \ldots, p_{a(f)}) \rangle$ with $f \in \mathcal{F}$, then define $f(p'_1, \ldots, p'_{a(f)}) \to p$ as a transition rule where $p'_i = p_i$ if p_i is a state, otherwise $p'_i = \langle p_i \rangle$ for $1 \le i \le a(f)$ and execute **ADDTRANS**(p'_i) for $1 \le i \le a(f)$.

Case 3. If $p = \langle t_1, \ldots, t_n \rangle$ $(n \ge 2)$, then do the following (i) through (iii).

(i) Define new ε-rules $p \to \langle t_i \rangle$ for $1 \le i \le n$.

(ii) For each transition rule of the form $p' \to p_0$ $(p', p_0 \in \mathcal{Q}_k, p_0 \subseteq p)$, define a new ε-rule $p'' \to p$ and execute **ADDTRANS**(p'') where p'' is the state defined as $p'' = (p \setminus p_0) \cup p'$ (see Figure 3.1(a)). In this case, if $p' \to p_0$ is a rewriting transition rule of degree k', then we call the new rule a *non-proper rewriting transition rule of degree* k'. If a



(a) a new rule defined in (ii)   (b) a new rule defined in (iii)

Figure 3.1. The new rules introduced by **ADDTRANS**.

move of the TA is caused by this new rule, then the move is also called a *non-proper rewriting move of degree* k'.

(iii) If there are states $p_1, \ldots, p_n$ and a function symbol f such that $p = \bigcup_{1 \le i \le n} p_i$ and $f(p_{i1}, \ldots, p_{ia(f)}) \to p_i \in \Delta_k$ for $1 \le i \le n$, then define new rules $f(p'_1, \ldots, p'_{a(f)}) \to p$ and $f(p'_1, \ldots, p'_{a(f)}) \to \langle t_i \rangle$ for $1 \le i \le n$ and execute **ADDTRANS**(p'_j) where $p'_j = \bigcup_{1 \le i \le n} p_{ij}$ for $1 \le j \le a(f)$ (see Figure 3.1(b)). $\square$

Example 3.2 Let $\mathcal{F}$ be the signature in Example 2.3 and $\mathcal{B}_3 = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{final}, \Delta)$ be a TA where $\mathcal{Q} = \{\ q_0,\ q_1,\ q'_0,\ q'_1,\ q'_2,\ q_f\}$, $\mathcal{Q}_{final} = \{q_f\}$ and Δ consists of the following transition rules:

$$\begin{array}{llll} c \to q_0, & h(q_0) \to q_1, & h(q_1) \to q_0, & \\ c \to q'_0, & h(q'_0) \to q'_1, & h(q'_1) \to q'_2, & h(q'_2) \to q'_0, \\ f(q_0, q'_0) \to q_f. & & & \end{array}$$

It can be easily verified that $\mathcal{L}(\mathcal{B}_3) = \{f(h^{2m}(c), h^{3n}(c)) \mid m, n \ge 0\}$. Let $\mathcal{R}_8 = \{f(x, x) \to g(x), g(x) \to x\}$. $\mathcal{R}$ is an RL-FPO-TRS. We apply Procedure 3.1 to $\mathcal{B}_3$ and $\mathcal{R}_8$. Consider the rewrite rule $f(x, x) \to g(x)$ in Step 4 for $\mathcal{A}_0 (k = 0)$. Since a move $f(\langle q_0 \rangle, \langle q'_0 \rangle) \vdash_0 \langle q_f \rangle$ is possible, new transition rules

$$\underline{\langle g(\langle q_0, q'_0 \rangle) \rangle} \to \langle q_f \rangle \quad (3.4)$$

$$g(\underline{\langle q_0, q'_0 \rangle}) \to \langle g(\langle q_0, q'_0 \rangle) \rangle \quad (3.5)$$

$$c \to \langle q_0, q'_0 \rangle \quad (3.6)$$

$$
\begin{array}{rcl}
h(\underline{\langle q_1, q_2' \rangle}) & \to & \langle q_0, q_0' \rangle \\
h(\underline{\langle q_0, q_1' \rangle}) & \to & \langle q_1, q_2' \rangle \\
h(\underline{\langle q_1, q_0' \rangle}) & \to & \langle q_0, q_1' \rangle \\
h(\underline{\langle q_0, q_2' \rangle}) & \to & \langle q_1, q_0' \rangle \\
h(\underline{\langle q_1, q_1' \rangle}) & \to & \langle q_0, q_2' \rangle \\
h(\underline{\langle q_0, q_0' \rangle}) & \to & \langle q_1, q_1' \rangle
\end{array}
$$

are added to Δ_1 where **ADDTRANS** is recursively executed for the underlined subterms. The transition rule (3.4) is defined in Step 4 and (3.5) is added in Case 2 of **ADDTRANS**. When **ADDTRANS**$(\langle q_0, q_0' \rangle)$ is executed, the Case 3(iii) is applied to the input and the rule (3.6) is added by using the rules $c \to q_0$ and $c \to q_0'$. The others are also added in Case 3(iii) of **ADDTRANS**$(\langle q_0, q_0' \rangle)$ and in its recursive execution. Next, consider the rewrite rule $g(x) \to x$ in Step 4 for $\mathcal{A}_1$ $(k = 1)$. Since

$$g(\langle q_0, q_0' \rangle) \vdash_1 \langle g(\langle q_0, q_0' \rangle) \rangle \vdash_1 \langle q_f \rangle,$$

$\langle q_0, q_0' \rangle \to \langle q_f \rangle$ is added to Δ_2. Thus we obtain

$$h(h(h(h(h(h(c)))))) \vdash_2^* \langle q_0, q_0' \rangle \vdash_2 \langle q_f \rangle$$

and hence $h(h(h(h(h(h(c)))))) \in \mathcal{L}(\mathcal{A}_2)$. We can verify that $\mathcal{A}_3 = \mathcal{A}_2$ $(= \mathcal{A}_*)$ and $\mathcal{L}(\mathcal{A}_*) = (\to_{\mathcal{R}_8}^*)(\mathcal{L}(\mathcal{B})) = \{g(h^{6n}(c)) \mid n \geq 0\} \cup \{h^{6n}(c) \mid n \geq 0\} \cup \mathcal{L}(\mathcal{B})$. $\square$

3.2. Correctness of the Construction

3.2.1 Soundness

Lemma 3.1 *For any $k \geq 0$ and a state $q \in \mathcal{Q}_{final}$, $\mathcal{L}_q(\mathcal{A}_k) \subseteq (\to_{\mathcal{R}}^*)(\mathcal{L}_q(\mathcal{A}_0))$.* $\square$

In this subsection, we prove of the soundness lemma, Lemma 3.1, of Procedure 3.1. The procedure can accept some left-non-linear TRSs as an input. Dealing with non-linear terms is beyond the capability of TAs in general. Here we introduce a *conditional linearization* of a non-left-linear TRS in order to deal with non-left-linear TRSs by TAs. The notion of the conditional linearization firstly introduced by De Vrijer[12] and by De Vrijer and Klop[26] to simplify the proof of Chew's theorem. Toyama and Oyamaguchi[35] gave a sufficient condition to guarantee confluence property by using the technique of conditional linearization. The definition of conditional linearization introduced in this thesis is based on Toyma and Oyamaguchi's one[35].

For an RL-TRS $\mathcal{R}$, let $n_{\mathcal{R}}$ be the smallest integer such that, for every rewrite rule $l \to r \in \mathcal{R}$, no variable occurs more than $n_{\mathcal{R}}$ times in l. Let $\wedge_{\mathcal{R}} = \{\wedge_i \mid 2 \leq i \leq n_{\mathcal{R}}\}$ be the set of new function symbols where the arity of $\wedge_i$ is i. Note that if $n_{\mathcal{R}} \leq 1$, then $\wedge_{\mathcal{R}} = \emptyset$ by definition. If the subscript i of the function symbol $\wedge_i$ is clear from the context, then we may write $\wedge$ instead of $\wedge_i$. Also we may write $\wedge$ instead of $\wedge_{\mathcal{R}}$. A term in $\mathcal{T}(\mathcal{F} \cup \wedge)$ is called a $\wedge$-*term*.

Definition 3.2 For an RL-TRS $\mathcal{R}$, α is a TRS which is defined as: $\alpha = \{\wedge_n(x, \ldots, x) \to x \mid \wedge_n \in \wedge_{\mathcal{R}}, n \geq 2\}$. $\square$

Example 3.3 Let $\mathcal{R}_9$ be $\{f(x,x) \to g(x)\}$, then $\alpha = \{\wedge_2(x,x) \to x\}$. For a term $s = f(\wedge(\wedge(a,a),a))$, $s \to_\alpha^* f(a)$. $\square$

Definition 3.3 For an RL-TRS $\mathcal{R}$, a rewrite step $\to_{\mathcal{R}_\alpha}$ is the smallest relation on $\wedge$-terms containing the rewrite relation $\to_{\mathcal{R}}$ on $\mathcal{F}$-terms and closed under contexts on $\wedge$-terms. $\square$

Definition 3.4 For a right-linear rewrite rule $l \to r$, the *conditional linearization* of $l \to r$ is a conditional rewrite rule defined as follows and written as $\wedge_L(l \to r)$:

1. Let $\mathcal{V}ar(l) = \{x_1, \ldots, x_n\}$. Assume x_i occurs at o_{ij} $(1 \leq j \leq \gamma_j)$ in l and if x_i occurs in r then it occurs at o_i.
2. Introduce new variables x_{ij} and y_i for $1 \leq i \leq n$ and $1 \leq j \leq \gamma_j$.
3. Define $\wedge_L(l \to r) = l[o_{ij} \leftarrow x_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq \gamma_j] \to r[o_i \leftarrow \wedge(x_{i1}, \ldots, x_{i\gamma_i}) \mid$ for i such that x_i occurs in $r]$ with the condition $(x_{ij} = y_i$ $(1 \leq i \leq n, 1 \leq j \leq \gamma_i))$.

For an RL-TRS $\mathcal{R}$, define $\wedge(\mathcal{R}) = \{\wedge_L(l \to r) \mid l \to r \in \mathcal{R}\}$. $\square$

Definition 3.5 For an RL-TRS $\mathcal{R}$, a rewrite step $\to_{\wedge(\mathcal{R})}$ is defined as follows:

1. $\to_{\wedge(\mathcal{R})} = \{(s,t) \mid (s,t) \in \to_{\wedge(\mathcal{R}),i}$ for some $i\}$.

2. $\rightarrow_{\wedge(\mathcal{R}),0} = \emptyset$.

3. $\rightarrow_{\wedge(\mathcal{R}),i+1} = \{(C[l\sigma], C[r\sigma]) \mid C$ is a context, $l \rightarrow r(x_1 = y_1, \ldots, x_n = y_n)$ is a conditional rewrite rule in $\wedge(\mathcal{R})$, σ is a substitution such that $y_i\sigma \in \mathcal{T}(\mathcal{F})$ for $1 \leq i \leq n$ and $x_i\sigma\ (\rightarrow_{\wedge(\mathcal{R}),i} \cup \rightarrow_\alpha)^*\ y_i\sigma\ \}$. $\square$

We say $s \rightarrow_{\wedge(\mathcal{R}),i} t$ is a *rewrite step of degree i*.

Definition 3.6 For two $\wedge$-terms s, t and an RL-TRS $\mathcal{R}$:

1. $s \rightarrow_{\alpha,\wedge(\mathcal{R})} t$ if $s \rightarrow_\alpha^* \circ \rightarrow_{\wedge(\mathcal{R})} \circ \rightarrow_\alpha^* t$.

2. $s \rightarrow_{\alpha,\mathcal{R}_\alpha} t$ if $s \rightarrow_\alpha^* \circ \rightarrow_{\mathcal{R}_\alpha} \circ \rightarrow_\alpha^* t$. $\square$

In Definition 3.5, the reason why the domain of $y_i\sigma$ for $1 \leq i \leq n$ is restricted to $\mathcal{T}(F)$ is that if this condition is not assumed, then it may occur that, for two $\mathcal{F}$-terms s and t, $s \not\rightarrow_{\mathcal{R}}^* t$ but $s \rightarrow_{\wedge(\mathcal{R})}^* t$. For example, let $\mathcal{R} = \{f(x_1, x_1, x_2, x_2) \rightarrow g(x_1, x_2), g(x, x) \rightarrow c''\} \cup \mathcal{R}'$ where $\mathcal{R}' = \{a \rightarrow c, a \rightarrow c', b \rightarrow c, d \rightarrow c', e \rightarrow c', e \rightarrow c\}$ and consider two $\mathcal{F}$-terms $f(a, b, d, e)$ and c''. The conditional linearization of $\mathcal{R}$ is $\wedge(\mathcal{R}) = \{f(x_{11}, x_{12}, x_{21}, x_{22}) \rightarrow g(\wedge(x_{11}, x_{12}), \wedge(x_{21}, x_{22}))\ (x_{11} = y_1, x_{12} = y_1, x_{21} = y_2, x_{22} = y_2), g(x_1, x_2) \rightarrow c''\ (x_1 = y, x_2 = y)\} \cup \mathcal{R}'$. If we ignore the condition that the domain of $y_i\sigma$ is restricted to $\mathcal{T}(\mathcal{F})$, then $f(a, b, d, e) \rightarrow_{\wedge(\mathcal{R})} g(\wedge(a, b), \wedge(d, e)) \rightarrow_{\wedge(\mathcal{R})}^* g(\wedge(c', c), \wedge(c', c)) \rightarrow_{\wedge(\mathcal{R})} c''$ holds. On the other hand, we can see that $f(a, b, d, e) \not\rightarrow_{\mathcal{R}}^* c''$.

Definition 3.7 For an RL-TRS $\mathcal{R}$, $\rightarrow_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} = \rightarrow_{\alpha,\mathcal{R}_\alpha} \cup \rightarrow_{\alpha,\wedge(\mathcal{R})}$. $\square$

For two terms s, t, if $s \rightarrow_1 \cdots \rightarrow_n t$ holds where $\rightarrow_i$ is either $\rightarrow_\alpha$ or $\rightarrow_{\mathcal{R}_\alpha}$ or $\rightarrow_{\wedge(\mathcal{R}),d_i}$, then $s \rightarrow_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t$ and we say that $\max\{d_i \mid$ for i such that $\rightarrow_i = \rightarrow_{\wedge(\mathcal{R}),d_i}\}$ is the *maximum degree of* the sequence $s \rightarrow_1 \cdots \rightarrow_n t$.

Example 3.4 Let $\mathcal{R}_{10} = \{f(x) \rightarrow g(x), h(x, x) \rightarrow h'(x)\}$, then we obtain $\wedge(\mathcal{R}_{10}) = \{f(x) \rightarrow g(x), h(x_1, x_2) \rightarrow h'(\wedge(x_1, x_2))\ (x_1 = y, x_2 = y)\}$. For a ground term $h(f(a), g(a))$, $h(f(a), g(a)) \rightarrow_{\alpha,\wedge(\mathcal{R}_{10})} h'(\wedge(f(a), g(a))) \rightarrow_{\alpha,\wedge(\mathcal{R}_{10})}^* h'(g(a))$. $\square$

Lemma 3.2 *For a $\wedge$-term s, an $\mathcal{F}$-term t and an RL-TRS $\mathcal{R}$, $s \rightarrow_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})}^* t$ implies $s \rightarrow_{\alpha,\mathcal{R}}^* t$.*

Proof. The proof is shown by induction on the number of maximum degree of the rewrite sequence $s \rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t$. For the basis, the lemma holds obviously. Assume the lemma holds for every sequence whose maximum degree of rewrite steps is $n-1$ or less and consider the case when the maximum degree is n. The inductive part is shown by another induction on the number of rewrite steps of degree n by $\rightarrow_{\wedge(\mathcal{R})}$. Assume the lemma holds for every sequence whose rewrite steps of degree n by $\rightarrow_{\wedge(\mathcal{R})}$ is $n'-1$ or less and consider the case for n'. A sequence which has n' rewrite steps of degree n by $\rightarrow_{\wedge(\mathcal{R})}$ can be written as:

$$\begin{array}{lcl} s & \rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} & s' \\ & = & s'[o \leftarrow l\sigma] \\ & \rightarrow_{\wedge(\mathcal{R})} & s'[o \leftarrow r\sigma] \\ & = & t' \\ & \rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} & t \end{array} \tag{3.7}$$

where s', t' are $\wedge$-terms, $s'[o \leftarrow l\sigma] \rightarrow_{\wedge(\mathcal{R})} s'[o \leftarrow r\sigma]$ is the first rewrite step of degree n by $\rightarrow_{\wedge(\mathcal{R})}$, $l \rightarrow r$ is a rewrite rule in $\wedge(\mathcal{R})$, σ is a substitution and o is a position in s'. Remark that the sequence $s'[o \leftarrow r\sigma] \rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t$ contains $n'-1$ rewrite steps of degree n by $\rightarrow_{\wedge(\mathcal{R})}$. We assume the following:

1. $\wedge_L(l' \rightarrow r') = l \rightarrow r$ where $l' \rightarrow r' \in \mathcal{R}$.
2. l' has m variables $x_1, \ldots x_m$.
3. For $1 \leq i \leq m$, x_i occurs at positions o_{ij} $(1 \leq i \leq \gamma_i)$ in l' and if x_i occurs in r' then it occurs at o_i.
4. For $1 \leq i \leq m$ and $1 \leq i \leq \gamma_i$, $l/o_{ij} = x_{ij}$, which is a new variable for defining $l \rightarrow r$ from $l' \rightarrow r'$ in Definition 3.4.
5. $\sigma = \{x_{ij} \mapsto t_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i\}$.

In the following, we define an $\mathcal{F}$-term t_k for each x_k $(1 \leq k \leq m)$.

If x_k does not occur in r', then from the definition of $\rightarrow_{\wedge(\mathcal{R})}$ there exists an $\mathcal{F}$-term t_k such that $t_{kj} \rightarrow^*_{\alpha,\wedge(\mathcal{R})} t_k$ for $1 \leq j \leq \gamma_k$ where the degree of each rewrite step $\rightarrow_{\wedge(\mathcal{R})}$ is less than or equal to $n-1$. By the inductive hypothesis for n, we obtain

$$t_{kj} \rightarrow^*_{\alpha,\mathcal{R}} t_k \quad (1 \leq j \leq \gamma_k). \tag{3.8}$$

If x_k occurs in r', then $r\sigma/o_k = \wedge(t_{k1}, \ldots, t_{k\gamma_k})$. Consider how the subterm $\wedge(t_{k1}, \ldots, t_{k\gamma_k})$ of $t = s'[o \leftarrow r\sigma]$ is rewritten in the rewrite sequence (3.7). Since t' is rewritten to a term t in $\mathcal{T}(\mathcal{F})$ (i.e., all the $\wedge$ symbols disappear during the rewriting), there are two cases.

1. The subterms $t_{k1}, \ldots, t_{k\gamma_k}$ of $\wedge(t_{k1}, \ldots, t_{k\gamma_k})$ are rewritten to an identical term t_k in $\mathcal{T}(\mathcal{F})$, i.e. $t_{kj} \rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t_k$ for $1 \leq j \leq \gamma_k$. By applying the inductive hypothesis for n' (when $n' \geq 2$) or the inductive hypothesis for n (when $n' = 1$) to these rewrite sequences, we obtain the relations

$$t_{kj} \quad \rightarrow^*_{\alpha,\mathcal{R}} \quad t_k \quad (1 \leq j \leq \gamma_k). \tag{3.9}$$

2. The term $\wedge(t_{k1}, \ldots, t_{k\gamma_k})$ is rewritten to $\wedge(t'_{k1}, \ldots, t'_{k\gamma_k})$ and disappear in the subsequent rewrite steps, i.e.,

$$\begin{aligned}
t' &= s'[o \leftarrow r\sigma[o_k \leftarrow \wedge(t_{k1}, \ldots, t_{k\gamma_k})]] \\
&\rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t'' \\
&= t''[o' \leftarrow l''\sigma''] \\
&= t''[o' \leftarrow l''\sigma''[o'_1 \leftarrow \wedge(t'_{k1}, \ldots, t'_{k\gamma_k})]] && (3.10) \\
&\rightarrow_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t''[o' \leftarrow r''\sigma''] && (3.11) \\
&\rightarrow^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t
\end{aligned}$$

where t'' is a $\wedge$-term, o' is a position in t'', $l'' \rightarrow r'' \in \wedge(\mathcal{R})$, σ'' is a substitution, o'_1 is a position in $l''\sigma''$ and there is a variable position o''_1 in l'' such that $o''_1 \leq o'_1$ and the variable l''/o''_1 does not occur in r''. The rewrite rule $l'' \rightarrow r''$ must be in $\wedge(\mathcal{R})$ since the co-domain of σ'' contains function symbols in $\wedge$. The position of the subterm $\wedge(t_{k1}, \ldots, t_{k\gamma_k})$ in t' is $o \cdot o_k$. By the rewrite step (3.11) and the definition of $\rightarrow_{\wedge(\mathcal{R})}$, there is an $\mathcal{F}$-term $t_k \in \mathcal{T}(\mathcal{F})$ such that $t'_{kj} \rightarrow^*_{\alpha,\wedge(\mathcal{R})} t_k$ which has only rewrite steps by $\rightarrow_{\wedge(\mathcal{R})}$ of degree $n-1$ or less for $1 \leq j \leq \gamma_k$ by Definition 3.5. By the inductive hypothesis for n, we obtain

$$t'_{kj} \quad \rightarrow^*_{\alpha,\mathcal{R}} \quad t_k. \tag{3.12}$$

Also from the sequence (3.10), it follows that

$$t_{kj} \quad \to^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} \quad t'_{kj} \quad (1 \leq j \leq \gamma_k). \tag{3.13}$$

From (3.12) and (3.13), we obtain the sequences

$$t_{kj} \quad \to^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} \quad t_k \quad (1 \leq j \leq \gamma_k) \tag{3.14}$$

where the numbers of rewrite steps of degree n by $\to_{\wedge(\mathcal{R})}$ are less than or equal to $n' - 1$. By the inductive hypothesis,

$$t_{kj} \quad \to^*_{\alpha,\mathcal{R}} \quad t_k \quad (1 \leq j \leq \gamma_k). \tag{3.15}$$

It follows from (3.8), (3.9) and (3.15) that

$$\begin{aligned} s' \quad &= \quad s'[o \leftarrow l\sigma] \\ &= \quad s'[o \leftarrow l[o_{ij} \leftarrow t_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i]] \\ &\to^*_{\alpha,\mathcal{R}} \quad s'[o \leftarrow l'\sigma'] \end{aligned} \tag{3.16}$$

where $\sigma' = \{x_i \mapsto t_i \mid 1 \leq i \leq m\}$. Since $l' \to r' \in \mathcal{R}$ and $t_i \in \mathcal{T}(\mathcal{F})$ $(1 \leq i \leq m)$, we obtain

$$s'[o \leftarrow l'\sigma'] \to_{\mathcal{R}_\alpha} s'[o \leftarrow r'\sigma'] \tag{3.17}$$

by Definition 3.3. Since $s'[o \leftarrow r\sigma] \to^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t$ and there is no function symbol $\wedge$ in the left-hand side of any rewrite rule in $\mathcal{R} \cup \wedge(\mathcal{R})$,

$$s'[o \leftarrow r'\sigma'] \to^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t, \tag{3.18}$$

which contains $n' - 1$ or less rewrite steps of degree n by $\to_{\wedge(\mathcal{R})}$. By the relations (3.16),(3.17) and (3.18), we obtain $s \to^*_{\alpha,\mathcal{R}\cup\wedge(\mathcal{R})} t$ where the number of rewrite steps of degree n by $\to_{\wedge(\mathcal{R})}$ is less than or equal to $n' - 1$. By the inductive hypothesis, we obtain $s \to^*_{\alpha,\mathcal{R}} t$ and the lemma holds. $\square$

Before proving the soundness of Procedure 3.1, we need some notions concerning with the TA constructed in the procedure.

A state q in $\mathcal{Q}_k$ is *singleton* if $|q| = 1$. A transition rule in Δ_k is *singleton* if its right-hand side is a singleton state. A move caused by a singleton transition rule is called a *singleton move*. For a TA $\mathcal{A}_k$ and a state $q \in \mathcal{Q}_k$, let $\mathcal{A}_{k^-}(q)$ (respectively

$\mathcal{A}_{k^\bullet}(q)$) be the TA obtained from $\mathcal{A}_k(q)$ by removing every rewriting transition rules (respectively non-singleton transition rules). For an $\mathcal{F}$-term s and a state $q \in \mathcal{Q}_k$, if $s \vdash_k^* q$ without any rewriting moves (respectively non-singleton moves), then we write $s \vdash_{k^-}^* q$ (respectively $s \vdash_{k^\bullet}^* q$). Remark that if $s \vdash_{k^\bullet}^* q$, then the move does not contain any non-proper rewriting moves since every non-proper rewriting transition rule is non-sigleton (see Case 3(ii) of **ADDTRANS**).

For a set $\wedge$ and a TA $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$, the extended TA $\wedge(\mathcal{A})$ for $\mathcal{T}(\mathcal{F} \cup \wedge)$ is defined as $\wedge(\mathcal{A}) = (\mathcal{F} \cup \wedge, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta \cup \Delta_\wedge)$ where $\Delta_\wedge = \{\wedge_n(q_1, \ldots, q_n) \to q, \wedge_n(q_1, \ldots, q_n) \to \langle t \rangle \mid \wedge_n \in \wedge, q_1, \ldots, q_n \in \mathcal{Q}, q_1 \cup \cdots \cup q_n = q \in \mathcal{Q}, t \in q\}$. A move caused by a transition rule in $\Delta_\wedge$ is called a $\wedge$-move. For a TA $\mathcal{A}_k$ in the procedure, we write $\vdash_{\wedge,k}$ instead of $\vdash_{\wedge(\mathcal{A}_k)}$. The TAs $\mathcal{A}_{\wedge,k^-}$, $\mathcal{A}_{\wedge,k^\bullet}$ and the relations $\vdash_{\wedge,k^-}, \vdash_{\wedge,k^\bullet}$ are similarly defined and we will use their combinations, e.g. $\vdash_{\wedge,k^{-,\bullet}}$.

Lemma 3.3 *Let $r \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ be a linear term with m variables $x_1, \ldots, x_m$ at $o_1, \ldots, o_m$, respectively, s be a $\wedge$-term and ρ be a substitution $\{x_i \mapsto q_i \mid 1 \le i \le m\}$ where $q_i \in \mathcal{Q}_k$ $(1 \le i \le m)$. If r is a variable, then let $t_{r\rho} = r\rho$. Otherwise, let $t_{r\rho} = \langle r\rho \rangle$. If $t_{r\rho} \in \mathcal{Q}_k$ and $s \vdash_{\wedge,k^-}^* t_{r\rho}$, then the sequence can be written as $s \vdash_{\wedge,k^-}^* s[o_i \leftarrow q_i \mid 1 \le i \le m] \vdash_{\wedge,k^-}^* t_{r\rho}$.* □

The next lemma states that if a $\wedge$-term s is accepted by a state q then there is a $\wedge$-term s' such that s' is accepted by q only with singleton moves and s' can be rewritten to s by rewrite rule $\wedge_i(x, \ldots, x) \to x$ $(i \ge 2)$.

Lemma 3.4 *For a $\wedge$-term s and a state $q \in \mathcal{Q}_k$, if $s \vdash_{\wedge,k}^* q$, then there is a $\wedge$-term s' such that $s' \vdash_{\wedge,k^\bullet}^* q$ and $s' \to_\alpha^* s$.*

Proof. The proof is shown by induction on the number of non-singleton rewriting moves in the sequence β: $s \vdash_{\wedge,k}^* q$. For the base case, the lemma holds obviously. Assume the lemma holds for every sequence which has at most $n-1$ non-singleton rewriting moves and consider the case for n. In this case, β can be written as

$$s \vdash_{\wedge,k^\bullet}^* s[o \leftarrow t] \vdash_k s[o \leftarrow p] \vdash_{\wedge,k}^* q \tag{3.19}$$

where o is in $\mathcal{P}os(s)$, $t \to p$ is a non-singleton transition rule and the move $s[o \leftarrow t] \vdash_k s[o \leftarrow p]$ is the first non-singleton rewriting move in β. Assume

$p = \langle t_1, \ldots, t_m \rangle$ $(m \geq 2)$. There are three cases (1),(2) and (3) for the transition rule $t \to p$:

(1) If t is of the form $f(p_1, \ldots, p_{a(f)})$ where $f \in \mathcal{F}$ and $p_i \in \mathcal{Q}_k$ for $1 \leq i \leq a(f)$, transition rules $t \to \langle t_i \rangle$ for $1 \leq i \leq m$ are also defined in Case 3(iii) of **ADDTRANS**.

(2) If $t \to p$ is a proper rewriting transition rule, then $t \to \langle t_i \rangle$ for $1 \leq i \leq m$ are also defined in Step 4(b) of Procedure 3.1.

Let $s'' = s[o \leftarrow \wedge_m(s/o, \ldots, s/o)]$, then in both cases (1) and (2), we have

$$
\begin{aligned}
s'' &\vdash^*_{\wedge,k\bullet} s[o \leftarrow \wedge_m(t, \ldots, t)] \\
&\vdash^*_{k\bullet} s[o \leftarrow \wedge_m(\langle t_1 \rangle, \ldots, \langle t_m \rangle)] \\
&\vdash_{\wedge} s[o \leftarrow \bigcup_{1 \leq i \leq m} \langle t_i \rangle] \\
&= s[o \leftarrow p] \\
&\vdash^*_{\wedge,k} q.
\end{aligned} \tag{3.20}
$$

(3) If $t \in \mathcal{Q}_k$ and the transition rule $t \to p$ is defined in Case 3(ii) of **ADDTRANS**, there are two cases: Assume that the transition rule $t \to p$ is defined from a transition rule $p' \to p_0$ such that $t = (p \setminus p_0) \cup p'$.

(3a) If $p' \to p_0$ is a singleton transition rule, i.e. $|p_0| = 1$, or $p' \to p_0$ is not a rewriting transition rule, then it is easy to see that there is a singleton transition rule $q' \to q_0$ such that $t = (p \setminus q_0) \cup q'$. (Especially, $q' = p'$ and $q_0 = p_0$ in the former case.) Assume $q' = \langle t'_1, \ldots, t'_{m'} \rangle$ and $q_0 = \langle t_1 \rangle$ without loss of generality. Let $s'' = s[o \leftarrow \wedge_m(\wedge_{m'}(s/o, \ldots, s/o), s/o, \ldots, s/o)]$, then we have

$$
\begin{aligned}
s'' &\vdash^*_{\wedge,k\bullet} s[o \leftarrow \wedge_m(\wedge_{m'}(t, \ldots, t), t, \ldots, t)] \\
&\vdash^*_{\wedge,k\bullet} s[o \leftarrow \wedge_m(\wedge_{m'}(\langle t'_1 \rangle, \ldots, \langle t'_{m'} \rangle), \langle t_2 \rangle, \ldots, \langle t_m \rangle)] \\
&\qquad \text{(by Case 3(i) of } \textbf{ADDTRANS}) \\
&\vdash_{\wedge} s[o \leftarrow \wedge_m(\bigcup_{1 \leq i \leq m'} \langle t'_i \rangle, \langle t_2 \rangle, \ldots, \langle t_m \rangle)] \\
&= s[o \leftarrow \wedge_m(q', \langle t_2 \rangle, \ldots, \langle t_m \rangle)]
\end{aligned}
$$

$$\begin{aligned}
&\vdash_{\wedge,k^\bullet} && s[o \leftarrow \wedge_m(q_0, \langle t_2\rangle, \ldots, \langle t_m\rangle)] \\
&= && s[o \leftarrow \wedge_m(\langle t_1\rangle, \langle t_2\rangle, \ldots, \langle t_m\rangle)] \\
&\vdash_\wedge && s[o \leftarrow \bigcup_{1\le i\le m} \langle t_i\rangle] \\
&= && s[o \leftarrow p] \\
&\vdash^*_{\wedge,k} && q. \qquad (3.21)
\end{aligned}$$

Since there are at most $n-1$ non-singleton rewriting moves in both (3.20) and (3.21), by the inductive hypothesis, there is a term s' such that $s' \vdash^*_{\wedge,k^\bullet} q$ and $s' \to^*_\alpha s''$. Obviously, $s'' \to^*_\alpha s$ and the lemma holds.

(3b) If $p' \to p_0$ is a rewriting transition rule and $|p_0| \ge 2$, then there are two cases: Assume $p' = \langle t'_1, \ldots, t'_{m'}\rangle$ and $p \setminus p_0 = \langle t_{j_1}, \ldots, t_{j_{m''}}\rangle$ where $m'' = |p| - |p_0|$.

(i) If $|p'| = 1$ (i.e., $p' = \langle t'_1\rangle$), then for $s''' = s[o \leftarrow \wedge_{m''+1}(s/o, \ldots, s/o)]$ we have

$$\begin{aligned}
s''' \;&\vdash^*_{\wedge,k^\bullet} && s[o \leftarrow \wedge_{m''+1}(\langle t'_1\rangle, \langle t_{j_1}\rangle \ldots, \langle t_{j_{m''}}\rangle)] \\
&\vdash_k && s[o \leftarrow \wedge_{m''+1}(p_0, \langle t_{j_1}\rangle \ldots, \langle t_{j_{m''}}\rangle)] \\
&\vdash_\wedge && s[o \leftarrow \bigcup_{1\le i\le m''} \langle t_{j_i}\rangle \cup p_0] \\
&= && s[o \leftarrow p] \\
&\vdash^*_{\wedge,k} && q. \qquad (3.22)
\end{aligned}$$

(ii) If $|p'| > 1$, then let

$$s''' = s[o \leftarrow \wedge_{m''+1}(\wedge_{m'}(s/o, \ldots, s/o), s/o, \ldots, s/o)].$$

We have

$$\begin{aligned}
s''' \;&\vdash^*_{\wedge,k^\bullet} && s[o \leftarrow \wedge_{m''+1}(\wedge_{m'}(\langle t'_1\rangle, \ldots, \langle t'_{m'}\rangle), \langle t_{j_1}\rangle, \ldots, \langle t_{j_{m''}}\rangle)] \\
&\vdash_\wedge && s[o \leftarrow \wedge_{m''+1}(\bigcup_{1\le i\le m'} \langle t'_i\rangle, \langle t_{j_1}\rangle, \ldots, \langle t_{j_{m''}}\rangle)] \\
&= && s[o \leftarrow \wedge_{m''+1}(p', \langle t_{j_1}\rangle, \ldots, \langle t_{j_{m''}}\rangle)] \\
&\vdash_k && s[o \leftarrow \wedge_{m''+1}(p_0, \langle t_{j_1}\rangle, \ldots, \langle t_{j_{m''}}\rangle)] \\
&\vdash_\wedge && s[o \leftarrow \bigcup_{1\le i\le m''} \langle t_{j_i}\rangle \cup p_0]
\end{aligned}$$

$$
\begin{aligned}
&= \quad s[o \leftarrow p] \\
&\vdash^*_{\wedge,k} \quad q. && (3.23)
\end{aligned}
$$

In both cases (i) and (ii) of (3b), $s''' \rightarrow^*_\alpha s$ holds. If $p' \rightarrow p_0$ is a proper rewriting transition rule, then both sequences of moves (3.22) and (3.23) are of the form in case (2) in this proof. Otherwise, repeating the same discussion of this case (3), we can finally obtain a proper rewriting transition rule and a sequence of of the form in case (2). Therefore, we can show that there is a term s'' for s''' in the moves such that $s'' \vdash^*_{\wedge,k} q$ which has at most $n-1$ non-singleton rewriting moves and $s'' \rightarrow^*_\alpha s'''$. Thus the lemma holds by the inductive hypothesis. □

Definition 3.8 For a $\wedge$-term s and an RL-TRS $\mathcal{R}$, $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(s)$ is true if and only if there is an $\mathcal{F}$-term s' such that $s \rightarrow^*_{\alpha,\wedge(\mathcal{R})} s'$. □

Example 3.5 Consider the TRS $\mathcal{R}_{10}$ in Example 3.4. Assume

$$s = h'(\wedge(f(a), g(a))),$$

then $\mathcal{E}_{\alpha,\wedge(\mathcal{R}_{10})}(s)$ is true. On the other hand, let $s' = h'(\wedge(f(a),\ g(c)))$, then $\mathcal{E}_{\alpha,\wedge(\mathcal{R}_{10})}(s')$ is false. □

Lemma 3.5 *For $\wedge$-terms s, s' and an RL-TRS $\mathcal{R}$ such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(s)$ is true, if s' is a subterm of s or $s \rightarrow^*_\alpha s'$, then $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(s')$.* □

Lemma 3.6 *For a $\wedge$-term s and an RL-TRS $\mathcal{R}$ such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(s)$ is true, and for a state $q \in \mathcal{Q}_k$, if $s \vdash^*_{\wedge,k} q$, then there exists a $\wedge$-term u' such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(u')$ is true, $u' \rightarrow^*_{\alpha,\wedge(\mathcal{R})} s$ and $u' \vdash^*_{\wedge,k^-,\bullet} q$.*

Proof. By Lemma 3.4, there is a $\wedge$-term s_α such that $s_\alpha \rightarrow^*_\alpha s$ and

$$s_\alpha \vdash^*_{\wedge,k\bullet} q. \tag{3.24}$$

The proof is shown by induction on the maximum degree of rewriting moves in the sequence (3.24). For the basis, let $u' = s_\alpha$ and the lemma holds. Assume the lemma holds for every sequence where the maximum degree of rewriting moves is $k-1$ or less and consider the case for k (≥ 1). The inductive part is shown by another induction on the number of rewriting moves of degree k in the sequence (3.24). Assume the lemma holds for every sequences which has $n-1$ rewriting moves of degree k and consider the case for n. The sequence (3.24) which has n rewriting moves of degree k can be written as

$$s_\alpha \vdash^*_{\wedge,k^\bullet} s_\alpha[o \leftarrow q'] \vdash_{k^\bullet} s_\alpha[o \leftarrow q''] \vdash^*_{\wedge,k^\bullet} q$$

where o is a position in s and the move $s_\alpha[o \leftarrow q'] \vdash_{k^\bullet} s_\alpha[o \leftarrow q'']$ is the first rewriting move of degree k. Remark that the transition rule used in this move is a proper rewriting move since every singleton rewriting transition rule is proper. Also note that $s_\alpha[o \leftarrow q''] \vdash^*_{\wedge,k^\bullet} q$ contains only $n-1$ rewriting moves of degree k. By the definition of TAs, $s_\alpha/o \vdash^*_{\wedge,k^\bullet} q' \vdash_{k^\bullet} q''$. There is no rewriting move of degree k in $s_\alpha/o \vdash^*_{\wedge,k^\bullet} q'$. By the inductive hypothesis on k, there is a term v such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(v)$ is true, $v \to^*_{\alpha,\wedge(\mathcal{R})} s_\alpha/o$ and $v \vdash^*_{\wedge,k^-,\bullet} q'$.

For the sequence β: $v \vdash^*_{\wedge,k^-,\bullet} q' \vdash_{k^\bullet} q''$, without loss of generality, assume that

1. $q' \to q''$ used in the last move in β is defined for a rewrite rule $l \to r \in \mathcal{R}$,
2. l has m variables $x_1, \dots, x_m$,
3. the variable x_i has γ_i positions in l at $o_{ij} \in \mathcal{P}os(l)$ $(1 \le j \le \gamma_i)$, and
4. if the variable x_i occurs in r, then it occurs at $o_i \in \mathcal{P}os(r)$.

Let $l' \to r'$ be $\wedge_L(l \to r)$. Since the last move $q' \vdash_{k^\bullet} q''$ is a rewriting move of degree k, and since it is defined for the rule $l \to r$ at Step 4 of Procedure 3.1, there are states p_{ij} $(1 \le i \le m, 1 \le j \le \gamma_i)$ and q''_0 in $\mathcal{Q}_{k-1}$ such that

$$l[o_{ij} \leftarrow p_{ij} \mid 1 \le i \le m, 1 \le j \le \gamma_i] \vdash^*_{k-1} q''_0, \tag{3.25}$$

$$\mathcal{L}_{p_{i1}}(\mathcal{A}_{k-1}) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_{k-1}) \neq \emptyset \tag{3.26}$$

where $q'' = q''_0$ or $q'' = \langle t \rangle$ for some $t \in q''_0$. Furthermore, for the substitution $\rho = \{x_i \mapsto p_i \mid \text{for } i \text{ such that } x_i \text{ occurs in } r\ \}$ where $p_i = \bigcup_{1 \le j \le \gamma_i} p_{ij}$, if $r \in \mathcal{V}$ then $q' = r\rho$ else $q' = \langle r\rho \rangle$. By Lemma 3.3, we can write the sequence $v \vdash^*_{\wedge,k^-,\bullet} q'$ as

$$v \vdash^*_{\wedge,k^-,\bullet} v[o_i \leftarrow p_i \mid \text{for } i \text{ such that } x_i \text{ occurs in } r] \vdash^*_{\wedge,k^-,\bullet} q'. \tag{3.27}$$

Define substitutions σ and σ' as $\sigma = \{x_i \mapsto u_i \mid 1 \leq i \leq m\}$ and $\sigma' = \{x_{ij} \mapsto u_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i\}$ where u_i and u_{ij} with $1 \leq i \leq m$ and $1 \leq j \leq \gamma_i$ are defined as follows:

1. If x_i occurs in r, then let $u_i = v/o_i$. By (3.27) we have $u_i \vdash^*_{\wedge,k-,\bullet} p_i$. If $\gamma_i > 1$, then the sequence can be written as $u_i \vdash^*_{\wedge,k-,\bullet} \wedge(p_{i1}, \ldots, p_{i\gamma_i}) \vdash_\wedge p_i$. In this case, let $u_{ij} = u_i/j$ for $1 \leq i \leq m, 1 \leq j \leq \gamma_i$. If $\gamma_i = 1$, then let $u_{i1} = u_i$. Remark that $r\sigma = r'\sigma'$. Also, $u_{ij} \vdash^*_{\wedge,k-,\bullet} p_{ij}$ holds for $1 \leq j \leq \gamma_i$ since $u_i \vdash^*_{\wedge,k-,\bullet} p_i$. By the fact that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(v)$ is true and by Lemma 3.5,
$$\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(u_i) \text{ is true.} \tag{3.28}$$

2. If x_i does not occur in r, then u_i is chosen to satisfy $u_i \in \mathcal{L}_{p_{i1}}(\mathcal{A}_{k-1}) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_{k-1})$, that is $u_i \vdash^*_{k-1} p_{ij}$ $(1 \leq j \leq \gamma_i)$. Such u_i exists by (3.26) and can be found effectively. By the inductive hypothesis on k, there are terms u_{ij} for $1 \leq j \leq \gamma_i$ such that $u_{ij} \to^*_{\alpha,\wedge(\mathcal{R})} u_i$ and
$$u_{ij} \vdash^*_{\wedge,k-,\bullet} p_{ij}. \tag{3.29}$$

In either case 1 or 2, we have
$$u_{ij} \vdash^*_{\wedge,k-,\bullet} p_{ij} \quad (1 \leq j \leq \gamma_i). \tag{3.30}$$

Let $v' = l'\sigma'$, then by (3.25) and (3.30), we have
$$\begin{array}{rcl} v' & = & l'\sigma' \\ & \vdash^*_{\wedge,k-,\bullet} & l'[o_{ij} \leftarrow p_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i] \\ & \vdash^*_{k-1} & q''_0. \end{array}$$

In either case $q'' = q''_0$ or $q'' = \langle t \rangle$ for some $t \in q''_0$, we have $v' \vdash^*_{\wedge,k-1} q''$. On the other hand, by (3.28), (3.29) and the fact that $(\to_{\wedge(\mathcal{R}),i} \cup \to_\alpha)^* \subseteq \to^*_{\alpha,\wedge(\mathcal{R})}$ for any i,
$$v' = l'\sigma' \to_{\wedge(\mathcal{R})} r'\sigma' = r\sigma = v. \tag{3.31}$$

That is, $v' \to_{\wedge(\mathcal{R})} v$. By the definition of TAs and the discussions above, we obtain
$$s_\alpha[o \leftarrow v'] \to_{\alpha,\wedge(\mathcal{R})} s_\alpha[o \leftarrow v] \to^*_{\alpha,\wedge(\mathcal{R})} s_\alpha[o \leftarrow s_\alpha/o] = s_\alpha \tag{3.32}$$

and $s_\alpha[o \leftarrow v'] \vdash^*_{\wedge,k-1} s_\alpha[o \leftarrow q''] \vdash^*_{\wedge,k\bullet} q$ where $s[o \leftarrow q''] \vdash^*_{\wedge,k\bullet} q$ contains $n-1$ rewriting moves of degree k. By the inductive hypothesis on the number of rewriting moves of degree k (when $n > 1$) or on the maximum degree of rewriting moves (when $n = 1$), there is a $\wedge$-term u' such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(u')$ is true,

$$u' \rightarrow^*_{\alpha,\wedge(\mathcal{R})} s_\alpha[o \leftarrow v'], \tag{3.33}$$

and

$$u' \vdash^*_{\wedge,k-,\bullet} q. \tag{3.34}$$

By (3.32), (3.33) and the fact that $s_\alpha \rightarrow^*_\alpha s$, we obtain

$$u' \rightarrow^*_{\alpha,\wedge(\mathcal{R})} s. \tag{3.35}$$

By (3.34) and (3.35), the lemma holds. $\square$

Lemma 3.7 *For an $\mathcal{F}$-term s, an RL-TRS $\mathcal{R}$ and a state $q \in \mathcal{Q}_k$, if $s \vdash^*_k q$, then there exists an $\mathcal{F}$-term u such that $u \rightarrow^*_{\mathcal{R}} s$ and $u \vdash^*_{k-} q$.*

Proof. Suppose $s \vdash^*_k q$ for an $\mathcal{F}$-term $s \in \mathcal{T}(\mathcal{F})$ and $q \in \mathcal{Q}_k$ with $|q| = 1$. By Lemma 3.6 and the fact that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(s)$, there is a $\wedge$-term u' such that $\mathcal{E}_{\alpha,\wedge(\mathcal{R})}(u')$ is true, $u' \rightarrow^*_{\alpha,\wedge(\mathcal{R})} s$ and $u' \vdash^*_{\wedge,k-,\bullet} q$. By Lemma 3.2 and the fact $s \in \mathcal{T}(\mathcal{F})$, we obtain $u' \rightarrow^*_{\alpha,\mathcal{R}} s$. In the following, we construct from u' an $\mathcal{F}$-term u such that $u \rightarrow^*_{\mathcal{R}} s$ and $u \vdash^*_{k-} q$ by replacing every subterm of the form $\wedge_m(t_1, \ldots, t_m)$ where $t_i \in \mathcal{T}(\mathcal{F})$ $(1 \leq i \leq m)$ with some term in $\{t_i \mid 1 \leq i \leq m\}$ from the leaves to the root. Assume $u'/o = \wedge_m(t_1, \ldots, t_m)$ and $t_i \in \mathcal{T}(\mathcal{F})$ for $1 \leq i \leq m$. Since $t_i \in \mathcal{T}(\mathcal{F})$ for $1 \leq i \leq m$ and all moves in the sequence $u' \vdash^*_{\wedge,k-,\bullet} q$ are singleton, we have

$$\begin{aligned} u' \quad &\vdash^*_{k-,\bullet} \quad u'[o \leftarrow \wedge_m(\langle t'_1\rangle, \ldots, \langle t'_m\rangle)] \\ &\vdash_\wedge \quad u'[o \leftarrow \langle t'_1, \ldots, t'_m\rangle] \\ &\vdash^*_{\wedge,k-,\bullet} \quad q \end{aligned} \tag{3.36}$$

where $\langle t'_i\rangle \in \mathcal{Q}_k$. Let $t \rightarrow \langle t'\rangle \in \Delta_k$ be the transition rule which is used to consume the state $\langle t'_1, \ldots, t'_m\rangle$ in v in the subsequence of (3.36) from $u'[o \leftarrow \langle t'_1, \ldots, t'_m\rangle]$ to q. There are two cases for t: (1) $t = \langle t'_1, \ldots, t'_m\rangle$ and $t' \in t$ and (2) t is of the form $f(p_1, \ldots, p_{a(f)})$ where $f \in \mathcal{F}$, $p_i \in \mathcal{Q}_k$ with $1 \le i \le a(f)$ and $\langle t'_1, \ldots, t'_m \rangle = p_1$ without loss of generality. For case (1), the subsequence of (3.36) from $u'[o \leftarrow \langle t'_1, \ldots, t'_m \rangle]$ to q can be written as:

$$\begin{aligned} u'[o \leftarrow \langle t'_1, \ldots, t'_m \rangle] \quad & \vdash_{k-,\bullet} \\ u'[o \leftarrow \langle t'_n \rangle] \quad & \vdash^*_{\wedge,k-,\bullet} \quad q \end{aligned} \tag{3.37}$$

for some n $(1 \le n \le m)$. Let $u'' = u'[o \leftarrow t_n]$, then we have $u'' \vdash^*_{k-,\bullet} u'[o \leftarrow \langle t'_n \rangle] \vdash^*_{\wedge,k-,\bullet} q$ by (3.36) and (3.37). For case (2), the subsequence of (3.36) from $u'[o \leftarrow \langle t'_1, \ldots, t'_m \rangle]$ to q can be written as:

$$\begin{aligned} u'[o \leftarrow \langle t'_1, \ldots, t'_m \rangle] \quad \vdash^*_{k-,\bullet} \quad & u'[o' \leftarrow f(\langle t'_1, \ldots, t'_m \rangle, \ldots, p_{a(f)})] \\ = \quad & u'[o' \leftarrow f(p_1, \ldots, p_{a(f)})] \\ \vdash_{\wedge,k-,\bullet} \quad & u'[o' \leftarrow \langle t' \rangle] \\ \vdash^*_{\wedge,k-,\bullet} \quad & q \end{aligned} \tag{3.38}$$

where $o = o' \cdot 1$. From Step 3(iii) of **ADDTRANS**, there is a transition rule of the form $f(\langle t''_1 \rangle, \ldots, \langle t''_{a(f)} \rangle) \to \langle t' \rangle$ where $t''_i \in p_i$ for $1 \le i \le a(f)$. Let n' $(1 \le n' \le m)$ be an integer such that $t''_1 = t'_{n'}$ and assume that $u' = u'[o' \leftarrow f(t^{o'}_1, \ldots, t^{o'}_{a(f)})]$. Then, by the transition rules defined in Step 3(i) of **ADDTRANS** and (3.38), we have

$$t^{o'}_i \vdash^*_{k-,\bullet} p_i \vdash_{k-,\bullet} \langle t''_i \rangle \quad (2 \le i \le a(f)). \tag{3.39}$$

Let $u'' = u'[o \leftarrow t_{n'}]$. By (3.36) and (3.39), we have

$$\begin{aligned} u'' \quad = \quad & u'[o' \leftarrow f(t_{n'}, \ldots, t^{o'}_{a(f)})] \\ \vdash^*_{k-,\bullet} \quad & u'[o' \leftarrow f(\langle t'_{n'} \rangle, \ldots, \langle t''_{a(f)} \rangle)] \\ = \quad & u'[o' \leftarrow f(\langle t''_1 \rangle, \ldots, \langle t''_{a(f)} \rangle)] \\ \vdash_{k-,\bullet} \quad & u'[o' \leftarrow \langle t' \rangle] \\ \vdash^*_{k-,\bullet} \quad & q. \end{aligned}$$

On the other hand, consider the rewrite sequence $u' \to^*_{\alpha,\mathcal{R}} s$. From the fact that no left-hand side has function symbols in $\wedge$ and $s, t_i \in \mathcal{T}(\mathcal{F})$ with $1 \le i \le m$, and from the definition of $\to_{\alpha,\mathcal{R}}$, there is an $\mathcal{F}$-term t_0 such that $u' \to^*_{\mathcal{R}} u'[o \leftarrow \wedge_m(t_0, \ldots, t_0)] \to_\alpha u'[o \leftarrow t_0] \to^*_{\alpha,\mathcal{R}} s$ where $t_0 \in \mathcal{T}(\mathcal{F})$. From this rewrite sequence, for both cases (1) and (2), we obtain $u'' \to_{\mathcal{R}}^* u'[o \leftarrow t_0] \to_{\alpha,\mathcal{R}}^* s$. Repeating the discussions above for every subterm with a function symbol in $\wedge$, we can obtain an $\mathcal{F}$-term u such that $u \to_{\mathcal{R}}^* s$ and $u \vdash_{k^-}^* q$. $\square$

To show Lemma 3.1, it is sufficient to show that for a term $s \in \mathcal{T}(\mathcal{F})$ and a state $q \in \mathcal{Q}_0$, if $s \vdash_k^* q$, then there exists a term $u \in \mathcal{T}(\mathcal{F})$ such that $u \to_{\mathcal{R}}^* s$ and $u \vdash_{k^-}^* q$. The claim holds from Lemma 3.7.

3.2.2 Completeness

First we prove two technical lemmas concerning packed states.

Lemma 3.8 *For a positive integer n and states p_i, $\langle t_i \rangle$ $(1 \le i \le n)$ in $\mathcal{Q}_k$, if there is a state $\langle t_1, \ldots, t_n \rangle$ in $\mathcal{Q}_k$ and $p_i \vdash_k^* \langle t_i \rangle$ for $1 \le i \le n$, then $p \vdash_k^* \langle t_1, \ldots, t_n \rangle$ where $p = \bigcup_{1 \le i \le n} p_i$.*

Proof. If $n = 1$, then the lemma holds obviously. Consider the case $n \ge 2$. Assume that for each $1 \le i \le n$, $p_i = p_{i0} \vdash_k p_{i1} \vdash_k \cdots \vdash_k p_{il_i} = \langle t_i \rangle$ for some $l_i \ge 0$ and $\langle t_1, \ldots, t_n \rangle \in \mathcal{Q}_k$. If $l_i = 0$ for every $1 \le i \le n$, the lemma holds obviously. Assume that $l_i \ge 1$ for a particular i. Then $p_{il_{i-1}} \to \langle t_i \rangle \in \Delta_k$. Since $\langle t_1, \ldots, t_n \rangle \in \mathcal{Q}_k$, **ADDTRANS**$(\langle t_1, \ldots, t_n \rangle)$ has been executed in Procedure 3.1 and a new ε-rule $p' \to \langle t_1, \ldots, t_n \rangle$ is defined in Case 3(ii) where $p' = (\langle t_1, \ldots, t_n \rangle \setminus \langle t_i \rangle) \cup p_{il_{i-1}} = \langle t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \rangle \cup p_{il_{i-1}}$. Hence, the move $\langle t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \rangle \cup p_{il_{i-1}} \vdash \langle t_1, \ldots, t_n \rangle$ is possible and **ADDTRANS**$(\langle t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n \rangle \cup p_{il_{i-1}})$ is recursively executed. Repeating the above argument, we have $p (= \bigcup_{1 \le i \le n} p_i) \vdash^* \langle t_1, \ldots, t_n \rangle$. $\square$

Lemma 3.9 *For an $\mathcal{F}$-term s, and states $\langle t_i \rangle \in \mathcal{Q}_k$ with $1 \le i \le n$, if $s \vdash_k^* \langle t_i \rangle$ for $1 \le i \le n$ and $\langle t_1, \ldots, t_n \rangle \in \mathcal{Q}_k$, then $s \vdash_k^* \langle t_1, \ldots, t_n \rangle$.*

Proof. The lemma is shown by induction on the depth of the term s. If $s = c$ with $a(c) = 0$, then the sequence $s \vdash_k^* \langle t_i \rangle$ can be written as

$$c \vdash_k p_i \vdash_k^* \langle t_i \rangle \quad (1 \le i \le n) \tag{3.40}$$

for some $p_i \in \mathcal{Q}_k$. Since $p_i \vdash_k^* \langle t_i \rangle$ for $1 \leq i \leq n$, we obtain $p \vdash_k^* \langle t_1, \ldots, t_n \rangle$ where $p = \bigcup_{1 \leq i \leq n} p_i$ by Lemma 3.8. Since $c \to p_i \in \Delta_k$ and $p \in \mathcal{Q}_k$, the transition rule $c \to p$ is defined by Case 3(iii) of **ADDTRANS**(p). Therefore $c \vdash_k p \vdash_k^* \langle t_1, \ldots, t_n \rangle$.

Assume that the lemma holds for every term with depth $l - 1$ or less, and consider a term $s = f(s_1, \ldots, s_{a(f)})$ with depth l. The sequence $s \vdash_k^* \langle t_i \rangle$ can be written as

$$s \vdash_k^* f(p_{i1}, \ldots, p_{ia(f)}) \vdash_k p_i \vdash_k^* \langle t_i \rangle \quad (1 \leq i \leq n) \tag{3.41}$$

where p_{ij} $(1 \leq j \leq a(f))$ and p_i are states. This implies $s_j \vdash_k^* p_{ij}$ for $1 \leq i \leq n$ and $1 \leq j \leq a(f)$, and therefore $s_j \vdash_k^* \bigcup_{1 \leq i \leq n} p_{ij}$ for $1 \leq j \leq a(f)$ by the induction hypothesis. Hence, the sequence

$$s \vdash_k^* f(\bigcup_{1 \leq i \leq n} p_{i1}, \ldots, \bigcup_{1 \leq i \leq n} p_{ia(f)}) \tag{3.42}$$

is possible. On the other hand, since all the moves in the sequence $p_i \vdash_k^* \langle t_i \rangle$ for $1 \leq i \leq n$ of (3.41) are ε-moves, transition rules are defined so that the sequence

$$p \vdash_k^* \langle t_1, \ldots, t_n \rangle \tag{3.43}$$

where $p = \bigcup_{1 \leq i \leq n} p_i$ is possible by means of Lemma 3.8. Furthermore, by the move $f(p_{i1}, \ldots, p_{ia(f)}) \vdash_k p_i$ with $1 \leq i \leq n$ of (3.41) and by Case 3(iii) of **ADDTRANS**(p), the transition rule

$$f(\bigcup_{1 \leq i \leq n} p_{i1}, \ldots, \bigcup_{1 \leq i \leq n} p_{ia(f)}) \to p \tag{3.44}$$

is defined. Summarizing (3.42),(3.43) and (3.44), we obtain $s \vdash_k^* \langle t_1, \ldots, t_n \rangle$. $\square$

The next lemma establishes the completeness of Procedure 3.1.

Lemma 3.10 *For a term $s \in (\to_{\mathcal{R}}^*)(\mathcal{L}(\mathcal{A}))$, there is an integer k such that $s \in \mathcal{L}(\mathcal{A}_k)$.*

Proof. It suffices to show that for a state $p \in \mathcal{Q}_0$, if $s' \to_{\mathcal{R}}^* s$ and $s' \in \mathcal{L}_p(\mathcal{A}_0)$, then there is an integer k such that $s \in \mathcal{L}_p(\mathcal{A}_k)$, or equivalently, $s \vdash_k^* p$. The claim is shown by induction on the length of the derivation $s' \to_{\mathcal{R}}^* s$. For the basis $s' = s$, the claim holds obviously. If $s' \to_{\mathcal{R}}^+ s$, then there is a term u such that

$s' \to^*_{\mathcal{R}} u \to_{\mathcal{R}} s$. By induction hypothesis applied to $s' \to^*_{\mathcal{R}} u$, we have an integer k' such that $u \vdash^*_{k'} p$. Moreover, since $u \to_{\mathcal{R}} s$ there is a rewrite rule $l \to r \in R$, a substitution σ, and a position $o \in \mathcal{P}os(u)$ such that $u/o = l\sigma$ and $s = u[o \leftarrow r\sigma]$. Hence, there is a state $p' \in \mathcal{Q}_{k'}$ such that $u = u[o \leftarrow l\sigma] \vdash^*_{k'} u[o \leftarrow p'] \vdash^*_{k'} p$ and we have

$$l\sigma \vdash^*_{k'} p'. \tag{3.45}$$

Now, let us show that $r\sigma \vdash^*_{k'+1} p'$. Assume that l has m variables $x_1, \ldots, x_m$ and the variable x_i has γ_i occurrences in l at $o_{ij} \in \mathcal{P}os(l)$. By (3.45) there are states p_{ij} for $1 \le i \le m$ and $1 \le j \le \gamma_i$ such that

$$x_i\sigma \vdash^*_{k'} p_{ij} \tag{3.46}$$

and

$$l[o_{ij} \leftarrow p_{ij} \mid 1 \le i \le m, 1 \le j \le \gamma_i] \vdash^*_{k'} p'. \tag{3.47}$$

The sequence (3.46) means that $x_i\sigma \in \mathcal{L}_{p_{i1}}(\mathcal{A}_{k'}) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_{k'})$ and we have

$$\mathcal{L}_{p_{i1}}(\mathcal{A}_{k'}) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_{k'}) \neq \emptyset \tag{3.48}$$

for $1 \le i \le m$. By (3.47) and (3.48), a substitution $\rho = \{x_i \mapsto p_i \mid 1 \le i \le m\} \cup \{x \mapsto \langle q_{any}\rangle \mid x \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)\}$ is defined in Step 4 of Procedure 3.1. By Lemma 3.9, each p_i in the co-domain of ρ satisfies

$$\mathcal{L}_{p_{i1}}(\mathcal{A}_{k'+1}) \cap \cdots \cap \mathcal{L}_{p_{i\gamma_i}}(\mathcal{A}_{k'+1}) \subseteq \mathcal{L}_{p_i}(\mathcal{A}_{k'+1}) \tag{3.49}$$

for $1 \le i \le m$ and transition rules are defined by **ADDTRANS** to satisfy that

$$r\rho \vdash^*_{k'+1} p'. \tag{3.50}$$

By (3.46) and (3.49), we have

$$x_i\sigma \vdash^*_{k'+1} p_i \qquad (1 \le i \le m). \tag{3.51}$$

Summarizing (3.50) and (3.51), we have $r\sigma \vdash^*_{k'+1} p'$, and the lemma holds since $s = u[o \leftarrow r\sigma] \vdash^*_{k'+1} u[o \leftarrow p'] \vdash^*_{k'} p$. □

By Lemma 3.1 and Lemma 3.10, we obtain the following theorem, which states the partial correctness of Procedure 3.1.

Theorem 3.11 *For an RL-TRS $\mathcal{R}$, if Procedure 3.1 halts then $\mathcal{L}(\mathcal{A}_*) = (\to^*_{\mathcal{R}})(\mathcal{L}(\mathcal{A}))$.* □

3.3. Termination of the Construction

We show that if an RL-FPO-TRS is given to Procedure 3.1, then there is an upper-bound limit on the number of states which are newly defined. Once the set of states saturates, then the set of transition rules also saturates and the procedure halts. First, as a measure of the size of a state, we introduce the concept of the *layer* of a packed state. Intuitively, the number of layers of a packed state is the number of right-hand sides of rewrite rules which are used for defining the state.

Definition 3.9 For a packed state $p \in \mathcal{Q}_k$, define the number of *layers* of p, denoted $\mathrm{layer}(p)$, as follows:

(1) if $p \in \mathcal{Q}_0$ or $p = \langle t \rangle$ with t a ground subterm of a rewrite rule in $\mathcal{R}$, then $\mathrm{layer}(p) = 0$,

(2) if $p = p_1 \cup p_2$, then $\mathrm{layer}(p) = \max\{\mathrm{layer}(p_1), \mathrm{layer}(p_2)\}$, and

(3) if $p = \langle r\sigma/o \rangle$ with $l \to r \in \mathcal{R}$, $o \in \mathcal{P}os(r)$, r/o is not a variable, $\mathcal{V}ar(r/o) = \{x_1, \ldots, x_n\}$ and $\sigma = \{x_i \mapsto p_i \mid 1 \leq i \leq n\}$, then $\mathrm{layer}(p) = 1 + \max\{\mathrm{layer}(p_i) \mid 1 \leq i \leq n\}$. □

Remark that $\mathrm{layer}(p)$ is not defined for all packed states, but all packed states introduced in Procedure 3.1 are of the form (1), (2) or (3). Also remark that $\mathrm{layer}(p)$ is not always uniquely determined by this definition. If different values are defined as $\mathrm{layer}(p)$, then we choose the minimum among the values as $\mathrm{layer}(p)$. We note that in (3) above if $x_i \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)$, then $p_i = \langle q_{any} \rangle$ and $\mathrm{layer}(p_i) = 0$. This means that variables which occurs only in the right-hand side are ignored for defining the number of layers.

Example 3.6 Consider the states of the TAs in Example 3.2. Let $l \to r = f(x, x) \to g(x) \in \mathcal{R}_8$, $o = \lambda$ and $\sigma = \{x \mapsto \langle q_0, q_0' \rangle\}$ in the above definition (3). Then, $p = \langle r\sigma/o \rangle = \langle g(\langle q_0, q_0' \rangle) \rangle$ and $\mathrm{layer}(p) = \mathrm{layer}(\langle q_0, q_0' \rangle) + 1 = \max\{\mathrm{layer}(\langle q_0 \rangle), \mathrm{layer}(\langle q_0' \rangle)\} + 1 = 1$. □

Lemma 3.12 *For any non-negative integer j, the number of packed states which have j or less layers is finite.*

Proof. The lemma will be shown by induction on j. For the base case, the number of the states that have 0 layer is finite, since the number of the states of $\mathcal{Q}_0$ and the number of the states that are made from ground subterms of the right-hand sides of a given TRS are finite.

Assume that the number of states that have $n-1$ or less layers is finite and show it is also true for the case that $j = n$. In Procedure 3.1, there are four cases when a new state which has n layers is added.

1. In Step 4 of Procedure 3.1, a state which is defined as $p_i = \bigcup_{1 \le j \le \gamma_i} p_{ij}$ in (3.3) is added.

2. In Step 4(c) of Procedure 3.1, a new state t_{r_ρ} is added.

3. In Case 2 of Procedure 3.2, a new state p_i' is added.

4. In Case 3(ii) of Procedure 3.2, a new state $p'' = (p' \setminus p_0) \cup p'$ is added.

5. In Case 3(iii) of Procedure 3.2, a new state $p_j' = \bigcup_{1 \le i \le n} p_{ij}$ is added.

From the inductive hypothesis and the definition (3) of the number of layers, there exists a number k' such that case 2 does not take place at any loop counter k'' for $k'' \ge k'$ in Procedure 3.1. Let $\widetilde{\mathcal{Q}}_{k'} = \{t \mid t \in p, p \in \mathcal{Q}_{k'}\}$. (Note that a packed state itself is a set.) A new state which is added in case 1, 3, 4, or 5 is a subset of $\widetilde{\mathcal{Q}}_{k'}$. Since $\mathcal{Q}_{k'}$ is finite, the number of subsets of $\widetilde{\mathcal{Q}}_{k'}$ is also finite. Hence the lemma holds. $\square$

In the following, it is shown that if $\mathcal{R}$ is an RL-FPO-TRS, then $\text{layer}(p) \le |\mathcal{R}|$ for any state p defined by Procedure 3.1 where $|\mathcal{R}|$ is the number of rewrite rules in $\mathcal{R}$. An outline of the proof is as follows. First we associate each rule in $\mathcal{R}$ with a non-negative integer called a *rank*. If $\mathcal{R}$ is finite path overlapping, then the rank is well-defined and is less than $|\mathcal{R}|$. Next, it is shown that if a rewrite rule with rank j is used in Step 4 of Procedure 3.1, then $\text{layer}(p) \le j+1$ for any state p defined in the same step. The *rank* of a rule in $\mathcal{R}$ is defined based on the sticking-out graph $G = (V, E)$ of $\mathcal{R}$. Let v be the vertex of G which corresponds to a rewrite rule $l \to r$ in $\mathcal{R}$. The *rank* of $l \to r$ is the maximum weight of a path to v from any vertex in V. If $\mathcal{R}$ is finite path overlapping, then the rank of any rewrite rule is a non-negative integer less than $|\mathcal{R}|$. For $\mathcal{R}_6$ in Example 2.8, the ranks of p_1 and p_2 are one and zero, respectively, since there is an edge with weight one from p_2 to p_1.

Lemma 3.13 *Let $l \to r$ be a rewrite rule and $\rho = \{x_i \mapsto p_i \mid 1 \leq i \leq m\} \cup \{x \mapsto \langle q_{any}\rangle \mid x \in \mathcal{V}ar(r) \setminus \mathcal{V}ar(l)\}$ be a substitution which are used in Step 4 of Procedure 3.1. If the rank of $l \to r$ is j or less, then layer$(p_i) \leq j$ for each $1 \leq i \leq m$.* $\square$

Before presenting a proof of the lemma, we first see how the number of layers of the state changes by a move of the TA. A transition rule of the TA is either an ε-rule or a non-ε-rule. An ε-rule is either an ε-rule of the original TA $\mathcal{A}_0$ or a rule defined in Step 4(a) or (b) of Procedure 3.1, or a rule defined in Case 3(i) or (ii) of **ADDTRANS** procedure. If an ε-rule of the original automaton is used at a move, then the number of layer does not change at the move. A non-ε-rule is either a non-ε-rule of $\mathcal{A}_0$, or a rule defined in Cases 1, 2 or 3(iii) of **ADDTRANS**. In all cases, the maximum number of layers in a state is increased by one or not changed by a move (Lemma 3.14). Hence, if the number of layers decreases at a move, then the rule is an ε-rule defined in Step 4(a) or (b) of Procedure 3.1 or in Case 3(ii) of **ADDTRANS**.

Lemma 3.14 *For a non-ε rule $f(p_1, \ldots, p_{a(f)}) \to p \in \Delta_k$ ($a(f) \geq 1$), let $m =$* $\max\{\text{ layer}(p_j) \mid 1 \leq j \leq a(f)\}$. *Then, $m \leq$* layer$(p) \leq m + 1$.

Proof. By induction on k. A non-ε rule is introduced either Step 1 of Procedure 3.1, or Case 1, Case 2, or Case 3(iii) of **ADDTRANS**. If $f(p_1, \ldots, p_{a(f)}) \to p$ is introduced in Step 1, then $\max\{\text{layer}(p_i) \mid 1 \leq i \leq a(f)\} = 0$ and layer$(p) = 0$. Thus the lemma holds. If $c \to \langle c\rangle$ is introduced in Case 1 of **ADDTRANS**, then the lemma holds vacuously. Assume that $f(p_1, \ldots, p_{a(f)}) \to p = \langle f(p_1, \ldots, p_{a(f)})\rangle$ is introduced in Case 2. Then there exists a rewrite rule $l \to r$ and a $\mathcal{Q}_k$-substitution ρ which satisfies (3.1) and (3.2) such that $(r/o)\rho = f(p_1, \ldots, p_{a(f)})$ for some $o \in \mathcal{P}os(r)$. Let $m = \max\{\text{layer}(p_j) \mid 1 \leq j \leq a(f)\}$. By definition of layer$(\cdot)$, layer$(p) = m$. Assume that $f(p_1, \ldots, p_{a(f)}) \to p$ is introduced in Case 3(iii). Let

$$\begin{aligned} m &= \max\{\text{layer}(p_j) \mid 1 \leq j \leq a(f)\} \\ &= \max\{\text{layer}(p_{ij}) \mid 1 \leq i \leq n, 1 \leq j \leq a(f)\}. \end{aligned}$$

There are two cases for each $1 \le i \le n$. If layer$(p_{ij}) = m$ for some j $(1 \le j \le a(f))$, then $m \le$ layer$(\langle t_i \rangle) \le m+1$ by the inductive hypothesis. If layer$(p_{ij}) < m$ for each j $(1 \le j \le a(f))$, then layer$(\langle t_i \rangle) \le m$ by the inductive hypothesis. Hence, $m \le$ layer$(p) = \max\{$layer$(\langle t_i \rangle) \mid 1 \le i \le n\} \le m+1$. □

Proof of Lemma 3.13 The proof is by induction on the loop variable k of Procedure 3.1. When $k = 0$, every state belongs to $\mathcal{Q}_0$ and layer$(p_i) = 0$ for $1 \le i \le n$, and the lemma holds for any j. Assume that the lemma holds for $k \le n-1$, and consider the case with $k = n$. The inductive part is shown by contradiction. Without loss of generality, let p_1 be a state such that layer$(p_1) \ge j+1$. Since $p_1 = \bigcup_{1 \le l \le \gamma_1} p_{1l}$, layer$(p_1) = \max\{$layer$(p_{1l}) \mid 1 \le l \le \gamma_1\}$ by the definition of layer$(\cdot)$. We can assume p_{11} is the state such that layer$(p_{11}) \ge j+1$ without loss of generality. Let us consider the sequence (3.1) in Step 4 of Procedure 3.1 and observe how the number of layers of the state changes as the head of $\mathcal{A}_k$ moves from o_{11} to the root in the sequence (3.1) of moves. There are four different cases:

1. A rewriting move is caused at a certain position. Let o be the innermost position among such positions. There are two different subcases:
   (a) The number of layers does not increase at any o' with $o \prec o' \prec o_{11}$.
   (b) There is a position o' with $o \prec o' \prec o_{11}$ such that the number of layers increases at o'.
2. There are no rewriting moves in the sequence. There are two subcases:
   (a) The number of layers does not increase at any o' with $\lambda \prec o' \prec o_{11}$.
   (b) There is a position o' with $\lambda \prec o' \prec o_{11}$ such that the number of layers increases at o'.

These four cases are illustrated in Fig. 3.2.

Assume that the number of layers changes as in case 1(a) above. In this case we can derive a contradiction as follows. First we assume a rewriting move at position o is proper and let $l' \to r'$ be the rewrite rule used for defining this transition rule in Step 4 of Procedure 3.1. Then, the state just before this rewriting move occurs at o can be written as $\langle r'\rho' \rangle$. Remark that layer$(\langle r'\rho' \rangle) =$ layer$(p_{11}) \ge j+1$ since

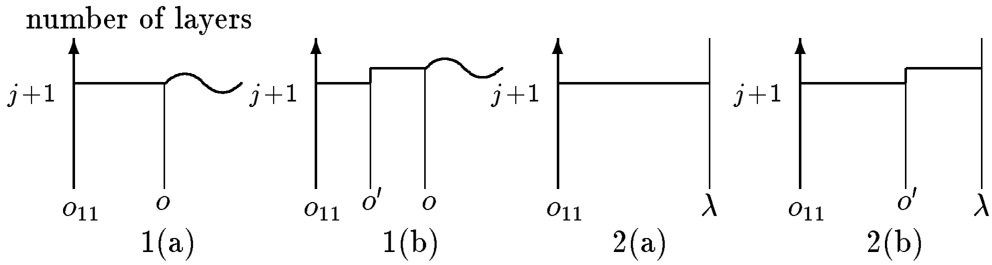


Figure 3.2. The number of layers of a state of $\mathcal{A}_k$ in the sequence (3.1).

the number of layers does not change at any o' ($o \prec o' \prec o_{11}$). This implies that the $\mathcal{Q}_k$-substitution ρ' replaces a variable in r' with a state which has j or more layers (see the definition (3) of the number of layers). Therefore, by using the inductive hypothesis, the rule $l' \to r'$ must have rank j or more. On the other hand, the fact that the number of layers does not increase at o' with $o \prec o' \prec o_{11}$ implies that r' properly sticks out of l/o as follows.

Consider the moves of the TA from the position o_{11} to o. Since o is the inner most position among the positions where rewriting moves are caused, all moves at o' ($o \prec o' \preceq o_{11}$) are defined by **ADDTRANS**. By the construction of transition rules in **ADDTRANS**, it follows that the function symbol of l at the position $o \cdot o''$ is the same as the function symbol of r' at o for every o such that $o \cdot o'' \prec o_{11}$. Furthermore, it can be easily shown that when the head visits the position $o \cdot o''(o \cdot o'' \preceq o_{11})$ of l, the state $\langle r'\rho'/o''\rangle$ is attached to that head. Thereby, at the variable position o_{11}, $\langle r'\rho'/o''\rangle$ was attached where o'' is such that $o \cdot o'' = o_{11}$, and this is the state p_{11}. Intuitively saying, the head goes up l along the path from o_{11} to o by changing the state from p_{11} to $\langle f(\ldots, p_{11}, \ldots)\rangle$ where f is the scanned symbol. This implies that r' properly sticks out of l/o by Case 1 of the definition of the sticking-out graph (Definition 2.16). We have observed that the rank of $l' \to r'$ is j or more, and thus the rank of $l \to r$ must be defined to be $j+1$ or more, a contradiction.

Next consider the case that the first rewriting transition rule used at o is defined in Case 3(ii) of **ADDTRANS** and let the rule be $p'' \to p$ such that $p'' = (p \setminus p_0) \cup p'$ and $p_0 \subseteq p$ for some rewriting transition rule $p' \to p_0$. The rewriting transition rule $p' \to p_0$ is either proper or non-proper. Assume $p' \to p_0$

is proper, then p' can be written as $\langle r'\rho'\rangle$ for some rewrite rule $l' \to r'$ and some $\mathcal{Q}_k$-substitution ρ'. From the fact that there is no rewriting move from o_{11} to o we can see that for every position o'' with $o{\cdot}o'' \preceq o_{11}$ when the head visits the position $o{\cdot}o''$ in l, a packed state which has $r'\rho'/o''$ as an element is attached to that head. Moreover from the construction of a non-ε rule whose right-hand side has more than one element (Case 3(iii) of **ADDTRANS**), the function symbol at $o \cdot o''$ ($o \cdot o'' \prec o_{11}$) in l coincides with the one in r' at o''. This implies that r' properly sticks out of l/o by Case 1 of the definition of sticking-out graph (Definition 2.16). By using this fact, we can derive a contradiction in the same way as in the case when the rule used at o is proper. Even if $p' \to p_0$ is non-proper, it is easy to see that there is a proper rewriting transition rule whose left-hand side is included in p'' and again a contradiction can be derived.

For other Cases 1(b), 2(a) and 2(b), we can derive a contradiction in a similar way (See the appendix). Thereby, it cannot happen that layer$(p_1) \geq j+1$ and the induction completes. $\square$

For an RL-FPO-TRS $\mathcal{R}$, the rank of every rule is less than $|\mathcal{R}|$ and hence the number of layers of any packed state is $|\mathcal{R}|$ or less by Lemma 3.13. By Lemma 3.12, the number of packed states is finite and the following theorem holds.

Theorem 3.15 *Procedure 3.1 halts for an RL-FPO-TRS.* $\square$

In general, the running time of Procedure 3.1 is exponential to both of the size of a TRS $\mathcal{R}$ and the size of a TA $\mathcal{A}$.

Corollary 3.16 *RL-GR^{-1}-TRS $\subset$ RL-GSM-TRS $\subset$ RL-FPO-TRS $\subset$ EPR-TRS.*

Proof. RL-GR^{-1}-TRS $\subset$ RL-GSM-TRS and RL-GSM-TRS $\subset$ RL-FPO-TRS are shown by Theorem 2.18. RL-FPO-TRS $\subset$ EPR-TRS is by Theorems 3.11 and 3.15. $\square$

Corollary 3.16 answeres that the open problem presented by Gyenizse and Vágvölgyi[21] positively, which asks to generalize the class of linear generalized semi-monadic TRSs so that a TRS in the obtained class still effectively preserves recognizability.

3.4. Decidable Approximations

In this section, we investigate decidable approximations of TRS along the lines of [14, 24, 28].

Definition 3.10 For a TRS $\mathcal{R}$, a TRS $\mathcal{R}'$ is an *approximation of a* $\mathcal{R}$ if $\to_{\mathcal{R}}^{*} \subseteq \to_{\mathcal{R}'}^{*}$ and $NF_{\mathcal{R}} = NF_{\mathcal{R}'}$. An *approximation mapping* α is a mapping from TRSs to TRSs such that $\alpha(\mathcal{R})$ is an approximation of $\mathcal{R}$ for any TRS $\mathcal{R}$. For a class C of TRSs, a C *approximation mapping* is an approximation mapping such that $\alpha(\mathcal{R}) \in C$ for every TRS $\mathcal{R}$. $\square$

In 1996, Jacquemard[24] introduced a linear growing approximation mapping. Later Nagaya and Toyama[28] introduced a better approximation called a left-linear growing approximation mapping and presented decidable results on them.

Definition 3.11 An *LL-FPO^{-1}-TRS approximation mapping* α is such that for a TRS $\mathcal{R}$, α replaces some variables in the right-hand side r_2 of a rewrite rule $l_2 \to r_2$ in $\mathcal{R}^{-1}$ with a new variable which is not in $\mathcal{V}ar(l_2)$, so that r_2 cannot contribute to an edge in the sticking-out graph of $\alpha(\mathcal{R}^{-1})$. $\square$

For example, replacing variable x with x' in the right-hand side of the rule in $\mathcal{R}_7$ of Example 2.8 yields an LL-FPO^{-1}-TRS approximation of $\mathcal{R}_7^{-1}$. The following results are a generalization of Nagaya and Toyama's results[28].

Definition 3.12 [14] Let α be an approximation mapping and Ω be a fresh constant. A redex at a position o in $t \in \mathcal{T}(\mathcal{F})$ is *α-needed* if there exists no $s \in NF_{\mathcal{R}}$ such that $t[o \leftarrow \Omega] \to_{\alpha(\mathcal{R})}^{*} s$ and s contains no Ω. $\square$

If $\mathcal{R}$ is orthogonal, then every α-needed redex is a needed redex in the sense of Huet and Lévy [23]. Let CBN-NF$_{\alpha}$ = $\{\mathcal{R} \mid$ every term $t \notin NF_{\mathcal{R}}$ has an α-needed redex $\}$. By Theorems 15 and 29 in the reference[14] and Lemma 2.6 of this thesis, the following theorem holds.

Theorem 3.17 *Let $\mathcal{R}$ be a left-linear TRS and α be an EPR^{-1}-TRS approximation mapping. Then the following problems are decidable:*

1. *Is a given redex in a given term α-needed?*

2. Is $\mathcal{R}$ in CBN-NF$_\alpha$? $\square$

Corollary 3.18 *Let $\mathcal{R}$ be an orthogonal TRS in EPR^{-1}-TRS which satisfies the variable restriction such that l is not a variable and $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ for every $l \to r \in \mathcal{R}$.*

1. *Every term $t \notin NF_{\mathcal{R}}$ has a needed redex.*

2. *It is decidable whether or not a given redex in a given term is needed.* $\square$

To conclude this section, we provide an orthogonal TRS $\mathcal{R}$ in FPO^{-1}-TRS such that there exists no left-linear growing approximation mapping β which satisfies $\mathcal{R} \in$ CBN-NF$_\beta$.

Example 3.7 Let $\mathcal{R}_{11} = \{g(h(x)) \to f(x,x,x)\} \cup \mathcal{R}'$ be an orthogonal TRS where $\mathcal{R}'$ consists of the following five rewrite rules:

$$f(a,b,x) \to a,\ \ f(b,x,a) \to a,\ \ f(x,a,b) \to a,$$
$$f(a,a,a) \to a,\ \ f(b,b,b) \to b.$$

It can be easily verified that $\mathcal{R}_{11}$ is in FPO^{-1}-TRS. Every term $t \notin NF_{\mathcal{R}_{11}}$ has a needed redex in $\mathcal{R}_{11}$ by Corollary 3.16 and Corollary 3.18-1. On the other hand, a left-linear growing approximation mapping β should be $\beta(\mathcal{R}_{11}) = \{g(h(y)) \to f(x,x,x)\} \cup \mathcal{R}'$ for some variable $y \neq x$. Consider a term $t = f(g(h(a)), g(h(a)), g(h(a)))$. Obviously, $g(h(a)) \to^*_{\beta(\mathcal{R}_{11})} a$ and $g(h(a)) \to^*_{\beta(\mathcal{R}_{11})} b$. Hence, t has no β-needed redex. Thus, $\mathcal{R}_{11} \notin$ CBN-NF$_\beta$. $\square$

Chapter 4

Strongly Normalizing Property

In this chapter, we show that for almost-orthogonal inverse FPO-TRSs, strongly normalizing property is decidable.

4.1. Nagaya and Toyama's method

Nagaya and Toyama[28] showed that SN is decidable for the class AO-GR-TRSs (almost orthogonal growing TRSs) in the following way.

Theorem 4.1 *[20] Let $\mathcal{R}$ be an AO-TRS (almost-orthogonal TRS, see Definition 2.3).*

1. *$\mathcal{R}$ is SN if and only if $\mathcal{R}$ is WIN.*
2. *A term s is SN in $\mathcal{R}$ if and only if s is WIN in $\mathcal{R}$.* $\square$

Definition 4.1 For a TRS $\mathcal{R}$ and a set L of terms, the *innermost $\mathcal{R}$-ancestor of* L is defined as $(\leftarrow^*_{I,\mathcal{R}})(L) = \{t \mid \exists s \in L, t \rightarrow^*_{I,\mathcal{R}} s\}$. $\square$

By using the notion of the innermost $\mathcal{R}$-ancestor, the property WIN can be represented as follows.

Lemma 4.2 *For a TRS $\mathcal{R}$:*

1. *$\mathcal{R}$ is WIN if and only if $(\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}}) = \mathcal{T}(\mathcal{F})$.*

2. A term s is WIN in $\mathcal{R}$ if and only if $s \in (\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}})$.

Proof. We only prove the first part as follows.

$$\begin{aligned}
\mathcal{R} \text{ is WIN} &\Leftrightarrow \forall t \in \mathcal{T}(\mathcal{F}).\ \exists t' \in NF_{\mathcal{R}}.\ t \rightarrow^*_{I,\mathcal{R}} t' \\
&\Leftrightarrow \mathcal{T}(\mathcal{F}) \subseteq \{t \mid \exists t' \in NF_{\mathcal{R}}.\ t \rightarrow^*_{I,\mathcal{R}} t'\} = (\rightarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}}) \\
&\Leftrightarrow \mathcal{T}(\mathcal{F}) = (\rightarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}}).
\end{aligned}$$

□

By Lemmas 2.6 and 4.2 and Theorem 4.1 we obtain the following lemma.

Lemma 4.3 *For an AO-TRS $\mathcal{R}$, if we can effectively construct a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}})$, then the following problems are decidable:*

1. Is $\mathcal{R}$ SN?

2. For a given term s, is s SN in $\mathcal{R}$? □

Nagaya and Toyama showed that for a left-linear growing TRS (LL-GR-TRS) $\mathcal{R}$ (see Definition 2.14), the set $(\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}})$ is always recognizable and thus whether $\mathcal{R}$ is WIN or not is decidable. If $\mathcal{R}$ is also an AO-TRS, then we can decide whether $\mathcal{R}$ is SN or not by Lemma 4.3.

Theorem 4.4 *[28] For an AO-GR-TRS $\mathcal{R}$, SN is decidable.* □

In the next section, we show if $\mathcal{R}$ is an AO-FPO^{-1}-TRS, then a TA accepting $(\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}})$ can be effectively constructed.

4.2. Tree Automata Construction for Inner-Most Ancestors

For an LL-TRS $\mathcal{R}$, Comon showed a complete and deterministic TA, denoted by $\mathcal{A}_{NF_{\mathcal{R}}}$, which accepts the set of all ground normal forms, i.e. $\mathcal{L}(\mathcal{A}_{NF_{\mathcal{R}}}) = NF_{\mathcal{R}}$ in [4]. We start with this TA $\mathcal{A}_{NF_{\mathcal{R}}}$.

Procedure 4.1 This procedure takes an AO-TRS $\mathcal{R}$ as an input and outputs a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\leftarrow^*_{I,\mathcal{R}})(NF_{\mathcal{R}})$. The algorithm does not always halt in general. We will show later that if $\mathcal{R}$ is an FPO^{-1}-TRS, then the procedure always halts. Let $\mathcal{A}_{NF_{\mathcal{R}}} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_{\mathit{final}}, \Delta)$ be the complete and deterministic TA with $\mathcal{L}(\mathcal{A}_{NF_{\mathcal{R}}}) = NF_{\mathcal{R}}$[4]. We will construct TAs whose states are represented by terms in $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ where elements in $\mathcal{Q}$ are regarded as constants. A term in $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ is called a $\mathcal{Q}$-term, and, to avoid confusion, a $\mathcal{Q}$-term $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ is written as $\langle t \rangle$ when it is used to represent a state of TAs.

Step 1. Let $\mathcal{A}_0 = (\mathcal{F}, \mathcal{Q}_0, \mathcal{Q}^0_{\mathit{final}}, \Delta_0)$ where $\mathcal{Q}_0 = \{\langle p \rangle \mid p \in \mathcal{Q}\}$, $\mathcal{Q}^0_{\mathit{final}} = \{\langle p \rangle \mid p \in \mathcal{Q}_{\mathit{final}}\}$, and $\Delta_0 = \{f(\langle p_1 \rangle, \ldots, \langle p_{a(f)} \rangle) \to \langle p \rangle \mid f(p_1, \ldots, p_{a(f)}) \to p \in \Delta\}$. In Steps 3 to 5, $\mathcal{A}_{k+1} = (\mathcal{F}, \mathcal{Q}_{k+1}, \mathcal{Q}^0_{\mathit{final}}, \Delta_{k+1})$ $(k \geq 0)$ is constructed from $\mathcal{A}_k = (\mathcal{F}, \mathcal{Q}_k, \mathcal{Q}^0_{\mathit{final}}, \Delta_k)$ by adding states and transition rules to $\mathcal{Q}_k$ and Δ_k, respectively. We abbreviate $\vdash_{\mathcal{A}_k}$ and $\vdash^*_{\mathcal{A}_k}$ as $\vdash_k$ and $\vdash^*_k$, respectively.

Step 2. Let $k = 0$.

Step 3. Let $\mathcal{Q}_{k+1} = \mathcal{Q}_k$ and $\Delta_{k+1} = \Delta_k$.

Step 4. New states and transition rules are introduced in this step. Let $l \to r$ be a rewrite rule in $\mathcal{R}$ and let $Y = \mathcal{V}ar(l) \setminus \mathcal{V}ar(r)$. It is assumed that r has $m (\geq 0)$ variables $x_i (1 \leq i \leq m)$ and x_i occurs at positions o_{ij} in $r (1 \leq i \leq m, 1 \leq j \leq \gamma_i)$. Assume there are states $q, q_{ij} \in \mathcal{Q}_k (1 \leq i \leq m, 1 \leq j \leq \gamma_i)$ and $q_{i0} \in \mathcal{Q}^0_{\mathit{final}} (1 \leq i \leq m)$ such that

$$r[o_{ij} \leftarrow q_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i] \vdash^*_k q \tag{4.1}$$

and $\mathbf{NO} \neq \mathbf{LUB}(\{q_{ij} \mid 0 \leq j \leq \gamma_i\})$. The function **LUB**, which is defined later, constructs a state which accepts terms accepted by every q_{ij} $(0 \leq j \leq \gamma_i)$. Then for any substitution $\rho' : Y \to \mathcal{Q}_{\mathit{final}}$, let $\rho = \{x_i \mapsto t_i \mid 1 \leq i \leq m\} \cup \rho'$ where $t_i = \mathbf{LUB}(\{q_{ij} \mid 0 \leq j \leq \gamma_i\})$, and do the following 1 and 2.

1. Add $\langle l\rho \rangle \to q$ to Δ_{k+1}. If $\langle l\rho \rangle \to q \in \Delta_{k+1} \setminus \Delta_k$, then the rule is called a *rewriting transition rule* of *degree* $k+1$ and if a move of the TA is caused by this rule, then the move is called a *rewriting move* of *degree* $k+1$.
2. Execute **ADDTRANS**($\langle l\rho \rangle$). In **ADDTRANS**($\langle l\rho \rangle$), new transition rules are defined so that $l\rho \vdash^*_{k+1} \langle l\rho \rangle$.

Simultaneously execute this Step 4 for every rewrite rule and every tuple of states that satisfy the condition (4.1) and every substitution $\rho': Y \to \mathcal{Q}^0_{final}$.

Step 5. If $\Delta_{k+1} = \Delta_k$ then output $\mathcal{A}_k$ as $\mathcal{A}_*$ and halt. Otherwise, let $k = k + 1$ and go to Step 3. $\square$

Procedure 4.2 [ADDTRANS] This procedure takes a state $\langle t \rangle$ as an input. If $\langle t \rangle$ already exists in $\mathcal{Q}_k$, then the procedure performs nothing. Otherwise, the procedure adds $\langle t \rangle$ to $\mathcal{Q}_k$ and defines new transition rules as follows.

Case 1. If $t = c$ with c a constant, then define $c \to \langle c \rangle$ as a transition rule.

Case 2. If $t = f(t_1, \ldots, t_{a(f)})$ with $f \in \mathcal{F}$, then define $f(\langle t_1 \rangle, \ldots, \langle t_{a(f)} \rangle) \to \langle t \rangle$ as a transition rule and execute **ADDTRANS**$(\langle t_i \rangle)$ for $1 \leq i \leq a(f)$. $\square$

In the following, we will use $t, t', t_1, t_2, \ldots$ to denote $\mathcal{Q}$-terms in $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$, $s, s', u, u', u_1, u_2, \ldots$ to denote ground terms in $\mathcal{T}(\mathcal{F})$, $f, g, \ldots$ to denote function symbols. Also $q, q_1, q_2, \ldots$ are states in $\mathcal{Q}_k$ for some $k \geq 0$ and $p, p_1, p_2 \ldots$ are states in $\mathcal{Q}$. If we write $f(t_1, \ldots, t_{a(f)})$, then we implicitly include the case when $a(f) = 0$.

In order to define the function **LUB**, we introduce a partial order on $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$. For a $\mathcal{Q}$-term t, let $t\langle\rangle$ denote the term obtained from t by replacing every $p \in \mathcal{Q}$ in t with $\langle p \rangle$. For example, if $t = f(g(p_1), p_2)$ where $p_1, p_2 \in \mathcal{Q}$, then $t\langle\rangle = f(g(\langle p_1 \rangle), \langle p_2 \rangle)$. Note that if $s \in \mathcal{T}(\mathcal{F})$ then $s\langle\rangle = s$. The relation $\leq$ on $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ is defined as follows:

(1) For $p \in \mathcal{Q}$ and $t' \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q})$, if $t'\langle\rangle \vdash_0^* \langle p \rangle$, then $p \leq t'$.

(2) For $f(t_1, \ldots, t_{a(f)}), f(t'_1, \ldots, t'_{a(f)}) \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q})$, if $t_i \leq t'_i$ $(1 \leq i \leq a(f))$ then $f(t_1, \ldots, t_{a(f)}) \leq f(t'_1, \ldots, t'_{a(f)})$.

Note that if $p \in \mathcal{Q}$, then $p \leq p$ by (1). If $p, p' \in \mathcal{Q}$ and $p \neq p'$ then $p\langle\rangle = \langle p \rangle \not\vdash_0^* \langle p' \rangle$ (since A_0 is deterministic) and hence $p \not\leq p'$ by (1).

For two $\mathcal{Q}$-terms t and t' if there is the least upper bound of t and t' on $\leq$, then it is denoted by $t \sqcup t'$. It is easily shown that $t \sqcup t'$ is represented as follows:

$$
\begin{array}{l}
t \sqcup t' = \\
t \quad \text{if} \quad t = t' \in \mathcal{Q} \\
t \quad \text{if} \quad t \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}, t' \in \mathcal{Q}, t\langle\rangle \vdash_0^* \langle t' \rangle \\
t' \quad \text{if} \quad t \in \mathcal{Q}, t' \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}, t'\langle\rangle \vdash_0^* \langle t \rangle \\
f(t_1 \sqcup t'_1, \ldots, t_{a(f)} \sqcup t'_{a(f)}) \\
\quad \text{if} \quad t = f(t_1, \ldots, t_{a(f)}) \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}, \\
\qquad t' = f(t'_1, \ldots, t'_{a(f)}) \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}, \\
\qquad t_i \sqcup t'_i \text{ defined } (1 \leq i \leq a(f)) \\
\text{undefined otherwise.}
\end{array}
\tag{4.2}
$$

For $k \geq 0$, let $\mathcal{A}_{k^-}$ be the TA obtained from $\mathcal{A}_k$ by removing every rewriting transition rule.

Function 4.1 [**LUB**] This function takes a set of states $\{\langle t_1 \rangle, \ldots, \langle t_n \rangle\}$ as an input and returns a $\mathcal{Q}$-term $t = t_1 \sqcup \cdots \sqcup t_n$ if it is defined. Also the function adds new transition rules and states so that $\mathcal{L}_{\langle t_1 \rangle}(\mathcal{A}_{k^-}) \cap \cdots \cap \mathcal{L}_{\langle t_n \rangle}(\mathcal{A}_{k^-}) = \mathcal{L}_{\langle t \rangle}(\mathcal{A}_{(k+1)^-})$.

Step 1. Decide whether $t_1 \sqcup \cdots \sqcup t_n$ is defined by using (4.2). If defined then let $t = t_1 \sqcup \cdots \sqcup t_n$ and go to Step 2. Otherwise, return **NO**.

Step 2. Execute **ADDTRANS**($\langle t \rangle$) and return t. $\square$

Example 4.1 We apply Procedure 4.1 to the AO-FPO^{-1}-TRS $\mathcal{R}_1$ in Example 2.8. First, we construct the deterministic and complete TA $\mathcal{A}_0$ accepting $NF_{\mathcal{R}_1}$ as $\mathcal{A}_0 = (\mathcal{F}, \mathcal{Q}_0, \mathcal{Q}^0_{final}, \Delta_0)$ where $\mathcal{Q}_0 = \{\langle p_r \rangle, \langle p_0 \rangle, \langle p_1 \rangle\}$, $\mathcal{Q}^0_{final} = \{\langle p_0 \rangle, \langle p_1 \rangle\}$ and $\Delta_0 = \{$

$$
\begin{array}{rcllrcll}
a & \to & \langle p_0 \rangle, & & g(\langle p_0 \rangle) & \to & \langle p_1 \rangle & \\
g(\langle p_1 \rangle) & \to & \langle p_1 \rangle, & & g(\langle p_r \rangle) & \to & \langle p_r \rangle & \\
h(\langle p_0 \rangle) & \to & \langle p_0 \rangle, & & h(\langle p_1 \rangle) & \to & \langle p_r \rangle & \\
h(\langle p_r \rangle) & \to & \langle p_r \rangle, & & f(\langle p_r \rangle, \langle p_r \rangle) & \to & \langle p_r \rangle & \\
f(\langle p_0 \rangle, \langle p_r \rangle) & \to & \langle p_r \rangle, & & f(\langle p_1 \rangle, \langle p_r \rangle) & \to & \langle p_r \rangle & \\
f(\langle p_r \rangle, \langle p_0 \rangle) & \to & \langle p_r \rangle, & & f(\langle p_r \rangle, \langle p_1 \rangle) & \to & \langle p_r \rangle & \\
f(\langle p_1 \rangle, \langle p_0 \rangle) & \to & \langle p_r \rangle, & & f(\langle p_1 \rangle, \langle p_1 \rangle) & \to & \langle p_r \rangle & \\
f(\langle p_0 \rangle, \langle p_0 \rangle) & \to & \langle p_0 \rangle, & & f(\langle p_0 \rangle, \langle p_1 \rangle) & \to & \langle p_0 \rangle & \}.
\end{array}
$$

Consider the rewrite rule $h(g(x)) \to f(x, x)$ in Step 4 for $\mathcal{A}_0$ ($k = 0$). Since a move $f(\langle p_0\rangle, \langle p_0\rangle) \vdash_0 \langle p_0\rangle$ is possible and $\mathbf{LUB}(\{\langle p_0\rangle, \langle p_0\rangle\}) = p_0$, the substitution ρ in Step 4 is $\rho = \{x \mapsto p_0\}$ and new transition rules $\langle h(g(p_0))\rangle \to \langle p_0\rangle$, $h(\langle g(p_0)\rangle) \to \langle h(g(p_0))\rangle$, $g(\langle p_0\rangle) \to \langle g(p_0)\rangle$ are added to Δ_1. The last two rules are added in $\mathbf{ADDTRANS}(\langle h(g(p_0))\rangle)$. Next, consider the rewrite rule $f(g(x), y) \to h(f(a, y))$ in Step 4 for $\mathcal{A}_0$. In this case, we need to consider two substitutions $\{x \mapsto p_0\}$ and $\{x \mapsto p_1\}$ as ρ'. Since $h(f(a, \langle p_0\rangle)) \vdash_0^* \langle p_0\rangle$ is possible, $\langle f(g(p_0), p_0)\rangle \to \langle p_0\rangle$, $\langle f(g(p_1), p_0)\rangle \to \langle p_0\rangle$ are added to Δ_1 and $\mathbf{ADDTRANS}(\langle f(g(p_0), p_0)\rangle)$ and $\mathbf{ADDTRANS}(\langle f(g(p_1), p_0)\rangle)$ are executed. We also have $h(f(a, \langle p_1\rangle)) \vdash_0^* \langle p_0\rangle$ and hence we define $\langle f(g(p_0), p_1)\rangle \to \langle p_0\rangle$, $\langle f(g(p_1), p_1)\rangle \to \langle p_0\rangle$ as new transition rules in Δ_1 and both **ADDTRANS** $(\langle f(g(p_0), p_1)\rangle)$ and **ADDTRANS** $(\langle f(g(p_1), p_1)\rangle)$ are executed. Again consider the rewrite rule $h(g(x)) \to f(x, x)$ in Step 4 for $\mathcal{A}_1$. Since a move $f(\langle g(p_0)\rangle, \langle p_1\rangle)$ $\vdash_1^* \langle p_0\rangle$ is possible and $g(p_0) \sqcup p_1 = g(p_0)$, a new transition rule $\langle h(g(g(p_0)))\rangle \to$ $\langle p_0\rangle$ is added to Δ_2 and $\mathbf{ADDTRANS}(\langle h(g(g(p_0)))\rangle)$ is executed. We can easily verify that $\mathcal{A}_2$ accepts $\mathcal{T}(\mathcal{F})$. $\square$

4.3. Correctness of the Construction

Lemma 4.5 *For a state $q \in \mathcal{Q}_k$ ($k \geq 0$), $\mathcal{L}_q(\mathcal{A}_{k^-}) = \mathcal{L}_q(\mathcal{A}_{k'^-})$ for any $k' \geq k$. (Especially, for a state $q \in \mathcal{Q}_0$, $\mathcal{L}_q(\mathcal{A}_0) = \mathcal{L}_q(\mathcal{A}_{k^-})$ for any $k \geq 0$.)*

Proof. $\mathcal{L}_q(\mathcal{A}_{k^-}) \subseteq \mathcal{L}_q(\mathcal{A}_{k'^-})$ is obvious since the sets of states and rules are enlarged monotonically. Assume $\exists t \in \mathcal{L}_q(\mathcal{A}_{k'^-}) \setminus \mathcal{L}_q(\mathcal{A}_{k^-})$. Then there exists an outermost position $\exists o \in \mathcal{P}os(t)$ where a rule $f(q'_1, \ldots, q'_{a(f)}) \to q'$ in $\Delta_{k'} \setminus \Delta_k$ is used. Since o is an outermost among such positions, $q' \in \mathcal{Q}_k$. However, to define a new transition rule whose right-hand side is q', $\mathbf{ADDTRANS}(q')$ must be executed. Since q' has been already included in $\mathcal{Q}_k$, **ADDTRANS** does not introduce any rules, a contradiction. $\square$

Lemma 4.6 *Assume $\langle f(t_1, \ldots, t_{a(f)})\rangle \in \mathcal{Q}_k$.*

1. *For $u \in \mathcal{T}(\mathcal{F})$, if $u \vdash_{k^-}^* \langle f(t_1, \ldots, t_{a(f)})\rangle$, then $u = f(u_1, \ldots, u_{a(f)}) \vdash_{k^-}^*$ $f(\langle t_1\rangle, \ldots, \langle t_{a(f)}\rangle) \vdash_{k^-} \langle f(t_1, \ldots, t_{a(f)})\rangle$ for some $u_i \in \mathcal{T}(\mathcal{F})$ ($1 \leq i \leq a(f)$).*

2. $\mathcal{L}_{\langle f(t_1,\ldots,t_{a(f)})\rangle}(\mathcal{A}_{k^-}) = \{f(u_1,\ldots,u_{a(f)}) \mid u_i \in \mathcal{L}_{\langle t_i\rangle}(\mathcal{A}_{k^-}), 1 \le i \le a(f)\}$. *(Especially,* $\mathcal{L}_{\langle c\rangle}(\mathcal{A}_{k^-}) = \{c\}$.*)*

Proof. The non-rewriting transition rule defined in Procedures 4.1 and 4.2 whose right-hand side is $\langle f(t_1,\ldots,t_{a(f)})\rangle$ must be $f(\langle t_1\rangle,\ldots,\langle t_{a(f)}\rangle) \to \langle f(t_1,\ldots,t_{a(f)})\rangle$. Those transition rules are defined in **ADDTRANS**. $\Box$

Lemma 4.7 *For a state* $\langle t\rangle \in \mathcal{Q}_k$, $t \le u$ *for any term* u *in* $\mathcal{L}_{\langle t\rangle}(\mathcal{A}_{k^-})$.

Proof. For a state $\langle t\rangle \in \mathcal{Q}_0$, the lemma holds obviously since $u = u\langle\rangle \vdash_0^* \langle t\rangle$ by Lemma 4.5 and hence $t \le u$ by (1) in the definition of $\le$. For a state $\langle t\rangle \in \mathcal{Q}_k \setminus \mathcal{Q}_0$, **ADDTRANS**$(\langle t\rangle)$ must have been executed. We show the lemma holds for $\langle t\rangle$ by structural induction on the term t. For $t = f(t_1,\ldots,t_{a(f)})$, $\mathcal{L}_{\langle t\rangle}(\mathcal{A}_{k^-}) = \{f(u_1,\ldots,u_{a(f)}) \mid u_i \in \mathcal{L}_{\langle t_i\rangle}(\mathcal{A}_{k^-}), 1 \le i \le a(f)\}$ by Lemma 4.6(2). By the inductive hypothesis, for any $u_i \in \mathcal{L}_{\langle t_i\rangle}(\mathcal{A}_{k^-})$ $t_i \le u_i$ holds. Thus, for any term $u \in \mathcal{L}_{\langle t\rangle}(\mathcal{A}_{k^-})$, $t \le u$ holds by (2) in the definition of $\le$. $\Box$

Lemma 4.8 *For a term* $u \in \mathcal{T}(\mathcal{F})$ *and a state* $\langle l\rho\rangle$ *where* l *is a linear term in* $\mathcal{T}(\mathcal{F},\mathcal{V})$ *and* ρ *is a substitution* $\rho = \{x_i \mapsto t_i \mid 1 \le i \le n, t_i \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q})\}$, *if* $u \vdash_{k^-}^* \langle l\rho\rangle$, *then there is a substitution* $\sigma\colon \mathcal{V}ar(l) \to \mathcal{T}(\mathcal{F})$ *such that* $u = l\sigma$ *and the sequence* $u \vdash_{k^-}^* \langle l\rho\rangle$ *can be written as* $u \vdash_{k^-}^* l\rho' \vdash_{k^-}^* \langle l\rho\rangle$ *where* $\rho' = \{x_i \mapsto \langle t_i\rangle \mid 1 \le i \le n\}$.

Proof. We need to show that (1) there is σ with $u = l\sigma$, (2) $u \vdash_{k^-}^* l\rho'$ and (3) $l\rho' \vdash_{k^-}^* \langle l\rho\rangle$. For (1), assume that x_i occurs at o_i in l for $1 \le i \le n$. Using Lemma 4.7, $u \vdash_{k^-}^* \langle l\rho\rangle$ implies $l\rho \le u$, and therefore $o_i \in \mathcal{P}os(u)$. Define $\sigma = \{x_i \mapsto u/o_i \mid 1 \le i \le n\}$, then $u = l\sigma$. (3) is rather obvious from the construction of transition rules in **ADDTRANS**, and hence (2) is shown hereafter. For the proof, it suffices to show that $u/o \vdash_{k^-}^* \langle l\rho/o\rangle$ for all $o \in \mathcal{P}os(l)$, since this will imply $u/o_i \vdash_{k^-}^* \langle l\rho/o_i\rangle = \langle t_i\rangle$ and therefore $u \vdash_{k^-}^* l\rho'$. The proof is by induction on the length of o. The claim holds for the case $|o| = 0$ by the assumption $u \vdash_{k^-}^* \langle l\rho\rangle$. Assume that $l\rho/o = f(t_1,\ldots,t_{a(f)})$ and

$$u/o \vdash_{k^-}^* \langle l\rho/o\rangle \tag{4.3}$$

as an inductive hypothesis. By Lemma 4.6(1), (4.3) can be written as

$$u/o \vdash_{k^-}^* f(\langle t_1\rangle, \ldots, \langle t_{a(f)}\rangle) \vdash_{k^-} \langle l\rho/o\rangle.$$

Hence, $u/o\cdot i \vdash_{k^-}^* \langle t_i\rangle = \langle l\rho/o\cdot i\rangle$ for $1 \le i \le a(f)$. □

For example, assume that $u = f(g(c), h(a))$, $l = f(x, h(y))$, $\rho = \{x \mapsto g(p_1), y \mapsto p_2\}$ and $u \vdash_{k^-}^* \langle f(g(p_1), h(p_2))\rangle (= \langle l\rho\rangle)$. Lemma 4.8 states that

$$\begin{array}{rcl} u & = & f(g(c), h(a)) \\ & \vdash_{k^-}^* & f(\langle g(p_1)\rangle, h(\langle p_2\rangle)) \vdash_{k^-}^* \langle f(g(p_1), h(p_2))\rangle. \end{array}$$

Lemma 4.8 implies the following corollary as a special case.

Corollary 4.9 *For a ground term $u \in \mathcal{T}(\mathcal{F})$ and $\langle t\rangle \in \mathcal{Q}_k$, if $u \vdash_{k^-}^* \langle t\rangle$ then $u \vdash_{k^-}^* t\langle\rangle \vdash_{k^-}^* \langle t\rangle$.* □

The following two lemmas show the correctness of the function **LUB**.

Lemma 4.10 *For states $\langle t_1\rangle, \ldots, \langle t_n\rangle$ in $\mathcal{Q}_k$, if $\mathcal{L}_{\langle t_1\rangle}(\mathcal{A}_{k^-}) \cap \cdots \cap \mathcal{L}_{\langle t_n\rangle}(\mathcal{A}_{k^-}) \neq \emptyset$, then $t_1 \sqcup \cdots \sqcup t_n$ is defined.*

Proof. For simplicity, we prove the lemma only for $n = 2$. (An inductive argument can apply to the case when $n \ge 3$.) Assume

$$\mathcal{L}_{\langle t\rangle}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{\langle t'\rangle}(\mathcal{A}_{k^-}) \neq \emptyset. \tag{4.4}$$

We prove the lemma by the structural induction on t. There are four cases to consider.

- If $t, t' \in \mathcal{Q}$ then $t = t'$ by (4.4), Lemma 4.5 and the fact that A_0 is deterministic. Hence, $t \sqcup t'$ is defined as t by (4.2).
- Assume $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}$ and $t' \in \mathcal{Q}$. By (4.4), there exists a term $u \in \mathcal{L}_{\langle t\rangle}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{\langle t'\rangle}(\mathcal{A}_{k^-})$. Thus $u \vdash_{k^-}^* \langle t\rangle$ and $u \vdash_{k^-}^* \langle t'\rangle$. By Corollary 4.9,

$$u \vdash_{k^-}^* t\langle\rangle \vdash_{k^-}^* \langle t\rangle. \tag{4.5}$$

  Since $\mathcal{A}_0$ is complete, there exists a state $\langle p\rangle \in \mathcal{Q}_0$ such that $t\langle\rangle \vdash_0^* \langle p\rangle$, which implies $u \vdash_{k^-}^* \langle p\rangle$ by (4.5). Since $u \vdash_{k^-}^* \langle t'\rangle$ and $u \vdash_{k^-}^* \langle p\rangle$, we see that $u \vdash_0^* \langle t'\rangle$ and $u \vdash_0^* \langle p\rangle$ by Lemma 4.5, which implies $t' = p$ by the determinicity of $\mathcal{A}_0$. Hence, $t\langle\rangle \vdash_0^* \langle t'\rangle$ and $t \sqcup t'$ is defined as t by (4.2).

- For the case when $t \in \mathcal{Q}$ and $t' \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}$, the claim can be proved in a similar way.

- Assume that $t = f(t_1, \ldots, t_{a(f)})$ and $t' = f(t'_1, \ldots, t'_{a(f)})$. It follows from Lemma 4.6(2) that (4.4) implies $\mathcal{L}_{\langle t_i \rangle}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{\langle t'_i \rangle}(\mathcal{A}_{k^-}) \neq \emptyset$ $(1 \leq i \leq a(f))$. By the inductive hypothesis, $t_i \sqcup t'_i$ is defined for $1 \leq i \leq a(f)$. Hence, $t \sqcup t'$ is defined by (4.2).

$\square$

Although Lemma 4.10 and the following lemma have duality, we divide them because of some technical reasons.

Lemma 4.11 *For states $\langle t_1 \rangle, \ldots, \langle t_n \rangle$ in $\mathcal{Q}_k$, if $t = t_1 \sqcup \cdots \sqcup t_n$ is defined and $\langle t \rangle \in \mathcal{Q}_k$, then $\mathcal{L}_{\langle t \rangle}(\mathcal{A}_{k^-}) = \mathcal{L}_{\langle t_1 \rangle}(\mathcal{A}_{k^-}) \cap \cdots \cap \mathcal{L}_{\langle t_n \rangle}(\mathcal{A}_{k^-})$.*

Proof. Again, we prove the lemma only for $n = 2$ by the structural induction. Assume $t \sqcup t'$ is defined. We perform case analysis according to (4.2).

- If $t = t' \in \mathcal{Q}$, then clearly the lemma holds.

- If $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}$, $t' \in \mathcal{Q}$, then $t \sqcup t'$ must be t and $t\langle\rangle \vdash_0^* \langle t' \rangle$. For any term $u \in \mathcal{L}_{\langle t \rangle}(\mathcal{A}_{k^-})$ (i.e. $u \vdash_{k^-}^* \langle t \rangle$), $u \vdash_{k^-}^* t\langle\rangle \vdash_{k^-}^* \langle t \rangle$ by Corollary 4.9 and thus $u \vdash_{k^-}^* \langle t' \rangle$ by the assumption. Hence, $\mathcal{L}_{\langle t \rangle}(\mathcal{A}_{k^-}) \subseteq \mathcal{L}_{\langle t' \rangle}(\mathcal{A}_{k^-})$ and the lemma holds.

- The case when $t \in \mathcal{Q}$, $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}) \setminus \mathcal{Q}$ and $t'\langle\rangle \vdash_0^* \langle t \rangle$ is similar.

- Assume $t = f(t_1, \ldots, t_{a(f)}), t' = f(t'_1, \ldots, t'_{a(f)})$ and $t_i \sqcup t'_i$ are defined for $1 \leq i \leq a(f)$. Then, $t \sqcup t' = f(t_1 \sqcup t'_1, \ldots, t_{a(f)} \sqcup t'_{a(f)})$. By the inductive hypothesis,
$$\mathcal{L}_{\langle t_i \sqcup t'_i \rangle}(\mathcal{A}_{k^-}) = \mathcal{L}_{\langle t_i \rangle}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{\langle t'_i \rangle}(\mathcal{A}_{k^-}). \tag{4.6}$$

for $1 \le i \le a(f)$. Thus,

$$\begin{aligned}
&\mathcal{L}_{\langle t \sqcup t' \rangle}(\mathcal{A}_{k^-}) \\
&= \mathcal{L}_{\langle f(t_1 \sqcup t'_1, \ldots, t_{a(f)} \sqcup t'_{a(f)}) \rangle}(\mathcal{A}_{k^-}) \\
&= \{f(u_1, \ldots, u_{a(f)}) \mid u_i \in \mathcal{L}_{\langle t_i \sqcup t'_i \rangle}(\mathcal{A}_{k^-})\} \quad \text{(by Lemma 4.6(2))} \\
&= \{f(u_1, \ldots, u_{a(f)}) \mid \\
&\qquad u_i \in \mathcal{L}_{\langle t_i \rangle}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{\langle t'_i \rangle}(\mathcal{A}_{k^-})\} \quad \text{(by (4.6))} \\
&= \{f(u_1, \ldots, u_{a(f)}) \mid u_i \in \mathcal{L}_{\langle t_i \rangle}(\mathcal{A}_{k^-})\} \\
&\qquad \cap \{f(u_1, \ldots, u_{a(f)}) \mid u_i \in \mathcal{L}_{\langle t_i \rangle}(\mathcal{A}_{k^-})\} \\
&= \mathcal{L}_{\langle f(t_1, \ldots, t_{a(f)}) \rangle}(\mathcal{A}_{k^-}) \\
&\qquad \cap \mathcal{L}_{\langle f(t'_1, \ldots, t'_{a(f)}) \rangle}(\mathcal{A}_{k^-}) \quad \text{(by Lemma 4.6(2)).}
\end{aligned}$$

□

4.3.1 Soundness

Lemma 4.12 *For a term $s \in \mathcal{T}(\mathcal{F})$ and states $q, q_0 \in \mathcal{Q}_k$, if $s \vdash^*_{k^-} q \vdash_k q_0$ where the move $q \vdash_k q_0$ is a rewriting move, then there is a term $s' \in \mathcal{T}(\mathcal{F})$ such that $s \to_{I,\mathcal{R}} s'$ and $s' \vdash^*_{k-1} q_0$.*

Proof. Assume that the move $q \vdash_k q_0$ is caused by the rewriting transition rule $q \to q_0$ of degree d $(\le k)$ and $q \to q_0$ is defined for a rewrite rule $l \to r$ in Step 4 of Procedure 4.1. Therefore q can be written as $q = \langle l\rho \rangle \in \mathcal{Q}_d$ where $\rho = \{x_i \mapsto t_i \mid 1 \le i \le n\}$. Also assume that r has m variables $x_1, \ldots, x_m$ and the variable x_i occurs at o_{ij} in r $(1 \le i \le m, 1 \le j \le \gamma_i)$ and at o_i in l $(1 \le i \le n)$. By applying Lemmas 4.5 and 4.8 to $s \vdash^*_{k^-} \langle l\rho \rangle$, there is a substitution σ such that $s = l\sigma$ and

$$x_i\sigma \vdash^*_{d^-} \langle t_i \rangle \quad (1 \le i \le n). \tag{4.7}$$

By Step 4 of Procedure 4.1, there are states $q_{ij} \in \mathcal{Q}_{d-1}$ and $q_{i0} \in \mathcal{Q}^0_{\mathit{final}}$ such that

$$r[o_{ij} \leftarrow q_{ij} \mid 1 \le i \le m, 1 \le j \le \gamma_i] \vdash^*_{d-1} q_0, \tag{4.8}$$

and $t_i = \mathbf{LUB}(\{q_{ij} \mid 0 \le j \le \gamma_i\})$. By (4.7) and Lemmas 4.5, 4.11, the following moves are possible:

$$x_i\sigma \vdash^*_{(d-1)^-} q_{ij} \quad (1 \le i \le m, 0 \le j \le \gamma_i). \tag{4.9}$$

Since $q_{i0} \in \mathcal{Q}^0_{\mathit{final}}$ and also $x_i\sigma \vdash^*_0 q_{i0}$ by Lemma 4.5 and (4.9),

$$x_i\sigma \in NF_{\mathcal{R}} \tag{4.10}$$

for $1 \leq i \leq n$ (For the case $x_i \in \mathcal{V}ar(l) \setminus \mathcal{V}ar(r)$, (4.10) trivially holds since $\langle t_i \rangle \in \mathcal{Q}^0_{\mathit{final}}$ by the definition of ρ in Step 4 of Procedure 4.1). Let $s' = r\sigma$, then $s \rightarrow_{I,\mathcal{R}} s'$ from (4.10) and Lemma 2.3. On the other hand, by (4.8), (4.9) and Lemma 4.5, $r\sigma \vdash^*_{k-1} q_0$ and the lemma holds. □

The next lemma shows the soundness of Procedure 4.1.

Lemma 4.13 *For a term s and a state $q \in \mathcal{Q}_k$, if $s \vdash^*_k q$, then there is a term s' such that $s \rightarrow^*_{I,\mathcal{R}} s'$ and $s' \vdash^*_{k^-} q$.*

Proof. The proof is shown by induction on the highest degree d of rewriting moves in $s \vdash^*_k q$. For the base case ($d = 0$, which means $s \vdash^*_{k^-} q$), let $s' = s$ and the lemma holds. Assume the lemma holds for the highest degree less than d and consider the case with d. The inductive part is shown by induction on the number m (≥ 1) of rewriting moves of degree d. The sequence that has m rewriting moves of degree d can be written as

$$s \vdash^*_k s[o \leftarrow q'] \vdash_k s[o \leftarrow q_0] \vdash^*_k q \tag{4.11}$$

where $o \in \mathcal{P}os(s)$, $q', q \in \mathcal{Q}_k$ and the move $s[o \leftarrow q'] \vdash_k s[o \leftarrow q_0]$ is the first rewriting move of degree d in the sequence (that is, o is one of the innermost position of the rewriting move of degree d). From the definition of TAs, we have

$$s/o \vdash^*_k q' \vdash_k q_0 \tag{4.12}$$

and by the inductive hypothesis for d, there is a term u such that

$$s/o \rightarrow^*_{I,\mathcal{R}} u \tag{4.13}$$

and

$$u \vdash^*_{k^-} q'. \tag{4.14}$$

From (4.12) and (4.14), we have

$$u \vdash^*_{k^-} q' \vdash_k q_0. \tag{4.15}$$

Applying Lemma 4.12 to (4.15), we can see that there is a term v such that

$$u \to_{I,\mathcal{R}} v \tag{4.16}$$

and

$$v \vdash^*_{k-1} q_0. \tag{4.17}$$

From (4.11) and (4.17), we obtain $s[o \leftarrow v] \vdash^*_{k-1} s[o \leftarrow q_0] \vdash^*_k q$ which have only $m-1$ rewriting moves of degree d. By the inductive hypothesis for m (if $m-1 \geq 1$) or by the inductive hypothesis for d (if $m-1=0$), there is a term s' such that

$$s[o \leftarrow v] \to^*_{I,\mathcal{R}} s' \tag{4.18}$$

and $s' \vdash^*_{k-} q$. From (4.13),(4.16) and (4.18), we have $s[o \leftarrow s/o] = s \to^*_{I,\mathcal{R}} s[o \leftarrow u] \to_{I,\mathcal{R}} s[o \leftarrow v] \to^*_{I,\mathcal{R}} s'$ and the lemma holds. □

4.3.2 Completeness

Lemma 4.14 *For a rewrite rule $l \to r \in \mathcal{R}$, a substitution σ and a state $q \in \mathcal{Q}_k$, if $l\sigma \to_{I,\mathcal{R}} r\sigma$ and $r\sigma \vdash^*_k q$, then $l\sigma \vdash^*_{k+1} q$ holds.*

Proof. Assume l has variables $x_1, \ldots, x_n$ and each x_i occurs at o_{ij} in r for $1 \leq i \leq m$ and $1 \leq i \leq \gamma_i$. From the assumption $l\sigma \to_{I,\mathcal{R}} r\sigma$,

$$x_i\sigma \in NF_{\mathcal{R}} \quad (1 \leq i \leq n). \tag{4.19}$$

The sequence $r\sigma \vdash^*_k q$ can be written as

$$\begin{aligned} r\sigma \ &\vdash^*_k \ r[o_{ij} \leftarrow q_{ij} \mid 1 \leq i \leq m, 1 \leq j \leq \gamma_i] \\ &\vdash^*_k \ q. \end{aligned} \tag{4.20}$$

From (4.20), we obtain

$$x_i\sigma \vdash^*_k q_{ij}. \tag{4.21}$$

By applying Lemma 4.13 to (4.21), we can see that there are terms u_{ij} with $1 \leq i \leq m$ and $1 \leq j \leq \gamma_i$ such that $u_{ij} \vdash^*_{k-} q_{ij}$ and $x_i\sigma \to^*_{I,\mathcal{R}} u_{ij}$. Since $x_i\sigma$ is in normal form, $x_i\sigma = u_{ij}$ and hence

$$x_i\sigma \vdash^*_{k-} q_{ij}. \tag{4.22}$$

By (4.19), there are states $q_{i0} \in \mathcal{Q}^0_{\mathit{final}}$ such that

$$x_i\sigma \vdash_0^* q_{i0} \quad (1 \leq i \leq n). \tag{4.23}$$

By (4.22), (4.23) and Lemma 4.5, $\mathcal{L}_{q_{i1}}(\mathcal{A}_{k^-}) \cap \cdots \cap \mathcal{L}_{q_{i\gamma_i}}(\mathcal{A}_{k^-}) \cap \mathcal{L}_{q_{i0}}(\mathcal{A}_{k^-}) \neq \emptyset$ $(1 \leq i \leq m)$ holds and hence $t_i = \mathbf{LUB}(\{q_{ij} \mid 0 \leq j \leq \gamma_i\})$ for some $t_i \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ by Lemma 4.10. Thus, in Step 4 in Procedure 4.1, substitution ρ is defined as $\rho = \{x_i \mapsto t_i \mid 1 \leq i \leq m\} \cup \{x_i \mapsto t_i \mid x_i \in \mathcal{V}ar(l) \setminus \mathcal{V}ar(r), \langle t_i \rangle \in \mathcal{Q}^0_{\mathit{final}}\}$. Moreover, in **ADDTRANS** new states and transition rules are defined so that

$$l\rho' \vdash^*_{(k+1)^-} \langle l\rho \rangle \vdash_{k+1} q \tag{4.24}$$

where $\rho' = \{x_i \mapsto \langle t_i \rangle \mid 1 \leq i \leq n\}$. On the other hand, by Lemmas 4.5, 4.11 and (4.22), we have

$$x_i\sigma \vdash^*_{(k+1)^-} \langle t_i \rangle. \tag{4.25}$$

Summarizing (4.24) and (4.25), we obtain $l\sigma \vdash^*_{(k+1)^-} l\rho' \vdash^*_{k+1} q$ and the lemma holds. □

The next lemma shows the completeness of Procedure 4.1.

Lemma 4.15 *For two terms $s, s' \in \mathcal{T}(\mathcal{F})$ and a state $q \in \mathcal{Q}_0$, if $s' \vdash_0^* q$ and $s \to^*_{I,\mathcal{R}} s'$, then there is an integer k such that $s \vdash_k^* q$.*

Proof. The proof is shown by induction on the number n of rewriting steps in $s \to^*_{I,\mathcal{R}} s'$. For the base case $(s = s')$, let $k = 0$, then the lemma holds. Assume the lemma holds for $n-1$ and consider the case with $n(\geq 1)$. The rewrite sequence of length n can be written as

$$s[o \leftarrow l\sigma] = s \to_{I,\mathcal{R}} s[o \leftarrow r\sigma] \to^*_{I,\mathcal{R}} s' \tag{4.26}$$

where $o \in \mathcal{P}os(t)$, σ is a substitution and $l \to r \in \mathcal{R}$. By the inductive hypothesis, there is an integer k such that $s[o \leftarrow r\sigma] \vdash_k^* q$ and hence there is a state q' such that

$$r\sigma \vdash_k^* q' \tag{4.27}$$

and

$$s[o \leftarrow q'] \vdash_k^* q. \tag{4.28}$$

From (4.26) and Lemma 2.2, we have

$$l\sigma \to_{I,\mathcal{R}} r\sigma. \tag{4.29}$$

By applying Lemma 4.14 to (4.27) and (4.29), it is possible that

$$l\sigma \vdash_{k+1}^{*} q'. \tag{4.30}$$

Summarizing (4.28) and (4.30), we have $s = s[o \leftarrow l\sigma] \vdash_{k+1}^{*} s[o \leftarrow q'] \vdash_{k}^{*} q$ and the lemma holds. □

Summarizing the lemmas in Section 4.3.1 and 4.3.2, the following theorem holds.

Theorem 4.16 *For an AO-TRS $\mathcal{R}$, if Procedure 4.1 halts with an output $\mathcal{A}_*$, then $\mathcal{L}(\mathcal{A}_*) = (\leftarrow_{I,\mathcal{R}}^{*})(NF_{\mathcal{R}})$.* □

Example 4.2 Consider the FPO^{-1}-TRS $\mathcal{R}_1$ in Example 2.8 again. $\mathcal{R}_1$ is an AO-TRS as well. Since $\mathcal{L}(\mathcal{A}_*) \supseteq \mathcal{L}(\mathcal{A}_2) = \mathcal{T}(\mathcal{F})$ by Example 4.1, we know $\mathcal{R}_1$ is SN by Theorem 4.1, Fact 4.2 and Theorem 4.16. □

4.4. Termination of the Construction

Procedure 4.1 and Procedure 4.1 are essentially the same where Procedure 4.1 adds new states and transition rules for only inner-most rewrite relation whereas Procedure 4.1 does for any rewrte relation. Hence the following lemma can easily be seen.

Lemma 4.17 *For a right-linear TRS $\mathcal{R}$, if Procedure 3.1 always halts for $\mathcal{R}$, then Procedure 4.1 always halts for $\mathcal{R}^{-1}$.* □

By Theorem 3.11 and Lemma 4.17 the following lemma holds.

Lemma 4.18 *Procedure 4.1 halts if the input is an AO-FPO^{-1}-TRS.* □

In general, the running time of Procedure 4.1 is exponential to both of the size of the given TRS $\mathcal{R}$ and the size of the given TA $\mathcal{A}$.

Lemma 4.19 *For an AO-FPO^{-1}-TRS $\mathcal{R}$, we can effectively construct a TA $\mathcal{A}_*$ such that $\mathcal{L}(\mathcal{A}_*) = (\leftarrow_{I,\mathcal{R}}^{*})(NF_{\mathcal{R}})$.*

Proof. By Theorem 4.16 and Lemma 4.18. $\square$

By using this lemma, we can show the following main theorem of this chapter.

Theorem 4.20 *For an AO-FPO^{-1}-TRS $\mathcal{R}$ the following problems are decidable:*

1. *Is $\mathcal{R}$ SN?*
2. *For a term s, is s SN in $\mathcal{R}$?*

Proof. By Lemmas 4.3 and 4.19. $\square$

Chapter 5

Conclusions

Properties of finitely path overlapping TRSs (FPO-TRSs) are discussed in this thesis.

In Chapter 2, FPO-TRS is defined (Definition 2.17) and the class of right-linear FPO-TRS is shown to properly include the other known decidable classes of TRSs which effectively preserve recognizability (Theorem 2.18). We also prove some properties of TRSs which effectively preserve recognizability in Chapter 2 (Theorems 2.10 and 2.11).

In Chapter 3, it is shown that any right-linear FPO-TRS (RL-FPO-TRS) effectively preserves recognizability (Theorem 3.11). The result provides a positive answer for an open problem proposed by Gyenizse and Vágvölgyi[21] which asks to generalize the class of linear generalized semi-monadic TRSs so that a TRS in the obtained class still effectively preserves recognizability. Also a new decidable approximation is investigated in order to decide whether or not a given term has needed redex (Definition 3.11 and Corollary 3.18).

In Chapter 4, a new subclass AO-FPO^{-1}-TRS of TRSs is proposed and it is shown that SN property of the class is decidable (Theorem 4.20). In the proof, we adopted tree automata technique similar to the one in [28]. The class of AO-FPO^{-1}-TRSs properly includes AO-GR-TRSs(Theorem 2.18).

The followings are directions to which the study will be developed for the future work.

The one is to restrict tree automata in the definition of recognizability preservation to obtain a wider class of TRSs which still have some appropriate properties. As already mentioned in Chapter 2, Gyenizse and Vágvölgyi[21] introduced the notion preserving $\mathcal{F}$-recognizability and showed that there is a difference between the notions preserving $\mathcal{F}$-recognizability and preserving recognizability. If we consider the property of preserving $\mathcal{F}$-recognizability, we might obtain a wider class of TRSs. Réty[31] proposed a decidable subclass of TRSs which effectively preserve recognizability for recognizable languages each of which consists of ground instances of a linear term. For example, consider $\mathcal{R} = \{f(g(x)) \to g(f(x))\}$ and the tree automaton $\mathcal{A} = (\{f, g, c\}, \{q_1, q_f\}, \{q_f\}, \Delta)$ where $\Delta = \{c \to q_f, g(q_f) \to q_1, f(q_1) \to q_f\}$. It is easy to see that $(\to_{\mathcal{R}}^{*})(\mathcal{L}(\mathcal{A})) \cap NF_{\mathcal{R}} = \{g^n(f^n(c)) \mid n \geq 0\}$. Since $\mathcal{R}$ is left-linear, $NF_{\mathcal{R}}$ is recognizable by Lemma 2.8. This implies that $(\to_{\mathcal{R}}^{*})(\mathcal{L}(\mathcal{A}))$ is not recognizable and hence $\mathcal{R}$ is not in EPR-TRS. However Réty[31] showed that if a TA is restricted to accept only instances of a linear term, then the TRS $\mathcal{R}$ effectively preserves recognizability. Réty also showed that for a TRS in the class reachability problem is decidable.

Another one is to use extensions of tree automata to obtain a wider class of TRSs which still have some appropriate properties. Main properties of tree automata which are used to show properties of EPR-TRS are that (1) the class of recognizable languages is closed under intersection and (2) the emptiness problem is decidable. Several extensions of tree automata which still have the properties (1) and (2) above are proposed and some of those extensions can be found in the survey of TAs[5]. For example, Bogaert and Tison[2] introduced *automata with equality and disequality constraints* (abbreviated to *AWEDC*), in which a transition rule consists of a kind of conditional rewrite rules. The class AWEDC can deal with some restricted class of tree languages of instances of non-linear terms. By using AWEDC, we may define a subclass of right-non-linear TRSs which have some useful properties.

Kaji *et al.*[25] presented a method to verify cryptographic protocols by computing descendants by a TRS of some recognizable languages. On the other hand, Genet and Klay[17] showed that for verifying the safety property of some cryptographic protocols, sometimes it is enough to compute approximations of descendants by a TRS and presented a procedure to compute approximations of descendants for any left-linear TRSs. We may be able to extend Procedure 3.1

in this thesis to compute approximations for wider classes of TRSs.

References

[1] Thomas Arts and Jürgen Giesl: "Termination of term rewriting using dependency pairs," Theoretical Computer Science, volume 236, numbers 1–2, pages 133–178, April 2000.

[2] Bruno Bogaert and Sophie Tison: "Equality and disequality constraints on direct subterms in tree automata," Proceedings of the 9th Annual Symposion on Theoretical Aspects of Computer Science, Cachan, France, Lecture Notes in Computer Science, volume 577, pages 161–172, Springer-Verlag, February 1992.

[3] Walter S. Brainerd: "Tree generating regular systems," Information and Control, volume 14, number 5, pages 217–231, Febuary 1969.

[4] Hubert Comon: "Sequentiality, second order monadic logic and tree automata," Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, pages 508–517, IEEE Computer Society Press, June 1995.

[5] Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Sophie Tison and Marc Tommasi: *Tree Automata Techniques and Applications*, Draft, 1997. Available from `http://l3ux02.univ-lille3.fr/tata/`

[6] Jean-Luc Coquidé, Max Dauchet, Rémi Gilleron and Sándor Vágvölgyi: "Bottom-up tree pushdown automata: classification and connection with rewrite systems," Theoretical Computer Science, volume 127, number 1, pages 69–98, May 1994.

[7] Jean-Luc Coquidé, Max Dauchet and Sophie Tison: "About connections between syntactical and computational complexity," Proceedings of the International Conference on Fundamentals of Computation Theory, Lecture Notes in Computer Science, volume 380, pages 105–115, Springer-Verlag, August 1989.

[8] Max Dauchet: "Simulation of a Turing machine by a left-linear rewrite rule," Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, North Carolina, Lecture Notes in Computer Science, volume 355, pages 109–120, Springer-Verlag, April 1989.

[9] Nachum Dershowitz and Zohar Manna: "Proving termination with multiset orderings," Communications of the ACM, volume 22, number 8, pages 465–476, August 1979.

[10] Nachum Dershowitz: "Termination of linear rewriting systems (preliminary version)," Proceedings of the 8th International Colloquium on Automata, Languages and Programming, Acre (Akko), Israel, Lecture Notes in Computer Science, volume 115, pages 448–458, Springer-Verlag, July 1981.

[11] Nachum Dershowitz and Jean-Pierre Jouannaud: "Rewrite Systems," Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science, volume B, Formal Models and Semantics*, chapter 6, pages 243–320, Elsevier Science Publishers, North-Holland, 1990.

[12] Roel C. de Vrijer: "Unique normal forms for combinatory logic with parallel conditional, a case study in conditional rewriting," Technical Report, Free University, 1990.

[13] Danny Dolev and Andrew C. Yao: "On the security of public key protocols," IEEE Transactions on Information Theory, volume IT-29, number 2, pages 198–208, March 1983.

[14] Irène Durand and Aart Middeldorp: "Decidable call by need computations in term rewriting (extended abstract)," Proceedings of the 14th International Conference on Automated Deduction, North Queensland, Australia, Lecture

Notes in Artificial Intelligence, volume 1249, pages 4–18, Springer-Verlag, July 1997.

[15] Jean H. Gallier and Ronald V. Book: "Reductions in tree replacement systems," Theoretical Computer Science, volume 37, number 2, pages 123–150, November 1985.

[16] Ferenc Gécseq and Magnus Steinby: *Tree Automata*, Académiai Kiadó, Budapest, 1984.

[17] Thomas Genet and Francis Klay: "Rewriting for cryptographic protocol verification," Proceedings of the 17th International Conference on Automated Deduction, Pittsburgh, Pennsylvania, Lecture Notes in Artificial Intelligence, volume 1831, Springer-Verlag, June 2000.

[18] Rémi Gilleron: "Decision problems for term rewriting systems and recognizable tree languages," Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science, Hamburg, Germany, Lecture Notes in Computer Science, volume 480, pages 148–159, Springer-Verlag, February 1991.

[19] Rémi Gilleron and Sophie Tison: "Regular tree languages and rewrite systems," Fundamenta Informaticae, volume 24, pages 157–175, 1995.

[20] Bernhard Gramlich: "Abstract relations between restricted termination and confluence properties of rewrite system," Fundamenta Informaticae, volume 24, pages 2–23, 1995.

[21] Pál Gyenizse and Sándor Vágvölgyi: "Linear generalized semi-monadic rewrite systems effectively preserve recognizability," Theoretical Computer Science, volume 194, numbers 1–2, pages 87–122, March 1998.

[22] Gérard Huet and Dalllas S. Lankford: "On the uniform halting problem for term rewriting systems," INRIA Technical Report 283, 1978.

[23] Gérard Huet and Jean-Jacques Lévy: "Computations in orthogonal rewriting systems, I and II," Jean-Loius Lassez and Gordon Plotkin, editors, *Computa-*

tional Logic: Essays in Honor of Alan Robinson, pages 396–443, MIT Press, 1991.

[24] Florent Jacquemard: "Decidable approximations of term rewriting systems," Proceedings of the 7th International Conference on Rewriting Techniques and Applications, New Brunswick, New Jersey, Lecture Notes in Computer Science, volume 1103, pages 362–376, Springer-Verlag, July 1996.

[25] Yuichi Kaji, Toru Fujiwara and Tadao Kasami: "Solving a unification problem under constrained substitutions using tree automata," Journal of Symbolic Computation, volume 23, number 1, pages 79–117, 1997.

[26] Jan Willem Klop and Roel C. de Vrijer: "Unique normal forms for lambda calculus with surjective pairing," Information and Computation, volume 80, number 2, pages 97–113, February 1989.

[27] Aart Middeldorp, Hitoshi Ohsaki and Hans Zantema: "Transforming termination by self-labelling", Proceedings of the 13th International Conference on Automated Deduction, New Brunwick, New Jersey, Lecture Notes in Artificial Intelligence, volume 1104, pages 373–387, Springer-Verlag, July 30–August 3 1996.

[28] Takashi Nagaya and Yoshihito Toyama: "Decidability for left-linear growing term rewriting systems," Proccedings of the 10th International Conference on Rewriting Techniques and Applications, Trento, Italy, Lecture Notes in Computer Science, volume 1631, pages 256–270, Springer-Verlag, July 1999.

[29] Kai Salomaa: "Deterministic tree pushdown automata and monadic tree rewriting systems," Journal of Computer System Science, volume 37, pages 367–394, December 1988.

[30] Kai Salomaa: "Decidability of confluence and termination of monadic term rewriting systems," Proceedings of the 4th International Conference on Rewriting Techniques and Applications, Como, Italy, Lecture Notes in Computer Science, volume 488, pages 275–286, Springer-Verlag, April 1991.

[31] Pierre Réty: "Regular sets of descendants for constructor-based rewrite systems," Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning, Lecture Notes in Artificial Intelligence, volume 1705, pages 148–160, Springer-Verlag, September 1999.

[32] Toshinori Takai, Yuichi Kaji, Takehiko Tanaka and Hiroyuki Seki: "A procedure for solving an order-sorted unification problem – extension for left-nonlinear system," Technical Report of NAIST, NAIST-IS-TR98011, 1998. Available from `http://www.aist-nara.ac.jp/`

[33] Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "A sufficient condition for the termination of the procedure for solving an order-sorted unification problem," Technical Report of NAIST, NAIST-IS-TR99010, 1999. Available from `http://www.aist-nara.ac.jp/`

[34] Toshinori Takai, Yuichi Kaji and Hiroyuki Seki: "Right-linear finite-path overlapping term rewriting systems effectively preserve recognizability," Proceedings of the 11th International Conference on Rewriting Techniques and Applications, Norwich, U.K., Lecture Notes in Computer Science, volume 1833, pages 246–260, Springer-Verlag, July 2000.

[35] Yoshihito Toyama and Michio Oyamaguchi: "Church-Rosser property and unique normal form property of non-duplicating term rewriting systems," Proccedings of the 4th International Workshop on Conditional Term Rewriting Systems, Jerusalem, Israel, Lecture Notes in Computer Science, volume 968, pages 316–331, Springer-Verlag, July 1994.

[36] Hans Zantema: "Termination of term rewriting by semantic labelling," Fundamenta Informaticae, volume 24, numbers 1–2, pages 89–105, 1995.