Simple Derivation Systems for Proving Sufficient Completeness of Non-Terminating Term Rewriting Systems

Kentaro Kikuchi
Tohoku University, Sendai, Japan

Takahito Aoto
Niigata University, Niigata, Japan

Abstract

A term rewriting system (TRS) is said to be sufficiently complete when each function yields some value for any input. Proof methods for sufficient completeness of terminating TRSs have been well studied. In this paper, we introduce a simple derivation system for proving sufficient completeness of possibly non-terminating TRSs. The derivation system consists of rules to manipulate a set of guarded terms, and sufficient completeness of a TRS holds if there exists a successful derivation for each function symbol. We also show that variations of the derivation system are useful for proving special cases of local sufficient completeness of TRSs, which is a generalised notion of sufficient completeness.

2012 ACM Subject Classification Theory of computation → Rewrite systems; Theory of computation → Equational logic and rewriting

Keywords and phrases Term rewriting, Sufficient completeness, Local sufficient completeness, Non-termination, Derivation rule, Well-founded induction schema

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.49

Funding This work was partially supported by JSPS KAKENHI Grant Numbers JP17K00005, JP19K11891, JP20H04164 and JP21K11750.

Acknowledgements We are grateful to the anonymous reviewers for valuable comments.

1 Introduction

This paper addresses a kind of reachability problem in transformation of labelled trees (i.e. terms) by rules schematised as term rewriting systems (TRSs). The main concern is whether all ground terms (i.e. terms without variables) can be transformed into terms consisting only of special labels called constructors. When the problem is solved positively, the TRS is said to be *sufficiently complete*. This property is useful in automated inductive theorem proving of TRSs, and has largely been studied. One of the sufficient conditions for sufficient completeness of a TRS is that it is terminating (strongly normalising) and quasi-reducible. For terminating TRSs, various decision procedures of sufficient completeness have been proposed [2, 9, 11].

On the other hand, only a few results [3, 4, 17] have been known about proof methods for sufficient completeness of non-terminating TRSs. In recent work [10], the authors proposed a framework for proving inductive theorems of possibly non-terminating TRSs. It is based on a generalised notion of sufficient completeness, called *local sufficient completeness*, where the problem concerns not all ground terms but only terms of specific form, specific sort, etc. In [10], the authors introduced a derivation system for proving local sufficient completeness, but it involves complicated notations and rules with complicated side conditions. In later work [15], the authors gave a proof method based on a sufficient condition for local sufficient

completeness. The method is applicable to some non-terminating TRSs for which the property is difficult to show by the derivation system of [10]. However, the method of [15] works only for TRSs that consist of functions on natural numbers and lists of natural numbers.

In the present paper, we introduce a simple derivation system for proving sufficient completeness of possibly non-terminating TRSs. The derivation system has rules to manipulate a set of guarded terms, which are pairs of a term and a set of terms. Although the meaning of a guarded term is not easy to grasp, we give its interpretation by introducing a notion of *well-founded induction schema*. This notion plays an important role in the proof of the correctness of our method based on the derivation system. It is easy to apply the method to various non-terminating TRSs, and we give some examples of application of it.

We also introduce variations of the derivation system for proving two particular cases of local sufficient completeness: local sufficient completeness with signature restriction and local sufficient completeness with sort partition. Apart from the simplicity of the systems, our method is different from the method of [10] in that a group of function symbols are simultaneously tested for existence of successful derivations. In this respect, our proof method can be seen as a natural extension of checking quasi-reducibility in the case of terminating TRSs, i.e., it not only checks reducibility of $f(t_1, \ldots, t_n)$ for each function symbol f but also traces results of reduction to terms to which the induction hypothesis can be applied.

Related work. The notion of sufficient completeness was originally introduced in [6, 7]. Since then, numerous works have treated the property in the fields of algebraic specification and term rewriting. In the literature, sufficient completeness has often been defined not w.r.t. reduction but w.r.t. conversion. Sufficient completeness w.r.t. reduction, as in the present paper, was introduced in [11]. In most cases, efforts have been devoted to TRSs consisting of those functions for which transformations by reduction rules are always terminating.

In [17], Toyama studied sufficient completeness w.r.t. reduction in the light of a more general notion of "reachability". He also gave a proof method for reachability, and applied it to some examples of left-linear non-terminating TRSs. Some of those examples are related to local sufficient completeness with signature restriction in terms of the present paper.

In [3, 4], Gnaedig and Kirchner studied sufficient completeness w.r.t. reduction (called $\mathcal{C}$-reducibility) of possibly non-terminating TRSs. They treated only usual sufficient completeness, and did not address any kind of local sufficient completeness. Although their system involves notions of abstract variables and narrowing, the process of proving sufficient completeness has some similarity with ours. We give an example of a TRS for which the method of [3, 4] does not work but our method works.

Organisation of the paper. The paper is organised as follows. In Section 2, we explain basic notions and notations of term rewriting. In Section 3, we introduce a derivation system for proving sufficient completeness of TRSs. In Section 4, we discuss local sufficient completeness with signature restriction. In Section 5, we discuss local sufficient completeness with sort partition. In Section 6, we conclude with suggestions for further work.

2 Preliminaries

In this section, we introduce some notations and notions from the field of term rewriting. For detailed information about term rewriting, see, e.g. [1, 14, 16].

A *many-sorted signature* is given by a non-empty finite set $\mathcal{S}$ of *sorts* and a finite set $\mathcal{F}$ of *function symbols*; each $f \in \mathcal{F}$ is equipped with its *sort declaration* $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha_0$ where $\alpha_0, \ldots, \alpha_n \in \mathcal{S}$ $(n \geq 0)$. We also use $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha_0$ to mean that f is equipped

with the sort declaration, or to denote such a function symbol f itself. We use $\mathcal{V}$ to denote the set of *variables* where $\mathcal{F} \cap \mathcal{V} = \emptyset$ and each $x \in \mathcal{V}$ has a unique sort $\alpha \in \mathcal{S}$. The set of variables with sort α is denoted by $\mathcal{V}^\alpha$. Then the set $T^\alpha(\mathcal{F}, \mathcal{V})$ of *terms of sort* α is defined inductively as follows:

1. If $x \in \mathcal{V}^\alpha$ then $x \in T^\alpha(\mathcal{F}, \mathcal{V})$.
2. If $f \in \mathcal{F}$, $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha$ and $t_i \in T^{\alpha_i}(\mathcal{F}, \mathcal{V})$ for each i $(1 \leq i \leq n)$ then $f(t_1, \ldots, t_n) \in T^\alpha(\mathcal{F}, \mathcal{V})$.

We define $T(\mathcal{F}, \mathcal{V}) = \bigcup_{\alpha \in \mathcal{S}} T^\alpha(\mathcal{F}, \mathcal{V})$, and $sort(t) = \alpha$ for each $t \in T^\alpha(\mathcal{F}, \mathcal{V})$.

For a term $t = f(t_1, \ldots, t_n)$, its *root symbol* f is denoted by $root(t)$. The set of variables in a term t is denoted by $\mathcal{V}(t)$. A term t is *ground* if $\mathcal{V}(t) = \emptyset$; the set of ground terms is denoted by $T(\mathcal{F})$. We write $f(\vec{x})$ for a term $f(x_1, \ldots, x_n)$ where $x_1, \ldots, x_n$ are distinct variables.

A *context* is a term $C \in T(\mathcal{F} \cup \{\Box^\alpha \mid \alpha \in \mathcal{S}\}, \mathcal{V})$ where $\mathcal{F} \cap \{\Box^\alpha \mid \alpha \in \mathcal{S}\} = \emptyset$ and the special symbol $\Box^\alpha$, called a *hole*, is a term of sort α. A context C with only one hole is denoted by $C[\,]$, and $C[t]$ denotes the term obtained by filling the hole with a term t of the same sort. If $s = C[t]$ for some context $C[\,]$, then t is a *subterm* of s, denoted by $t \trianglelefteq s$.

A *substitution* is a mapping $\theta : \mathcal{V} \to T(\mathcal{F}, \mathcal{V})$ such that $sort(x) = sort(\theta(x))$ for every $x \in \mathcal{V}$, and $dom(\theta) = \{x \in \mathcal{V} \mid \theta(x) \neq x\}$ is finite. A substitution θ is *ground* if $\theta(x) \in T(\mathcal{F})$ for every $x \in dom(\theta)$. The term obtained by applying a substitution θ to a term t is written as $t\theta$. If θ_g is a ground substitution and $\mathcal{V}(t) \subseteq dom(\theta_g)$, the ground term $t\theta_g$ is called a *ground instance* of t. We sometimes write $f(\vec{t})$ for a term $f(\vec{x})\theta$ where $\vec{t}$ is a sequence of terms $x_1\theta, \ldots, x_n\theta$.

A *rewrite rule*, written as $l \to r$, is an ordered pair of terms l and r such that $l \notin \mathcal{V}$, $\mathcal{V}(r) \subseteq \mathcal{V}(l)$ and $sort(l) = sort(r)$. A *term rewriting system* (*TRS*, for short) is a finite set of rewrite rules. For a TRS $\mathcal{R}$, the binary relation $\to_\mathcal{R}$ on $T(\mathcal{F}, \mathcal{V})$ is defined by $s \to_\mathcal{R} t$ iff $s = C[l\theta]$ and $t = C[r\theta]$ for some $l \to r \in \mathcal{R}$, some context $C[\,]$ and some substitution θ. The reflexive transitive closure of $\to_\mathcal{R}$ is denoted by $\overset{*}{\to}_\mathcal{R}$. A term s is in *normal form* if $s \to_\mathcal{R} t$ for no term t. The set of terms in normal form is denoted by $NF(\mathcal{R})$.

Let $\mathcal{R}$ be a TRS. The set $\mathcal{D}$ of *defined symbols* is given by $\mathcal{D} = \{root(l) \mid l \to r \in \mathcal{R}\}$, and the set $\mathcal{C}$ of *constructors* is given by $\mathcal{C} = \mathcal{F} \setminus \mathcal{D}$. Terms in $T(\mathcal{C}, \mathcal{V})$ are called *constructor terms*, and terms in $T(\mathcal{C})$ are called *ground constructor terms*.

Now we define the notion of sufficient completeness w.r.t. reduction.

▶ **Definition 1** (Sufficient completeness). A TRS $\mathcal{R}$ is *sufficiently complete* for a ground term $t_g \in T(\mathcal{F})$, denoted by $SC(t_g)$, if there exists a ground constructor term $s_g \in T(\mathcal{C})$ such that $t_g \overset{*}{\to}_\mathcal{R} s_g$. $\mathcal{R}$ is *(globally) sufficiently complete* if $SC(t_g)$ for every ground term $t_g \in T(\mathcal{F})$.

Let $\mathcal{R}$ be a TRS. $\mathcal{R}$ is *terminating* if there exists no infinite sequence $t_0 \to_\mathcal{R} t_1 \to_\mathcal{R} \cdots$. $\mathcal{R}$ is *quasi-reducible* if $f(t_1, \ldots, t_n) \notin NF(\mathcal{R})$ for every $f(t_1, \ldots, t_n) \in T(\mathcal{F})$ with $f \in \mathcal{D}$ and $t_1, \ldots, t_n \in T(\mathcal{C})$. The next proposition provides a criterion of sufficient completeness of $\mathcal{R}$. (For its proof, see, e.g. Proposition 2.4 of [10].)

▶ **Proposition 2.** *Let $\mathcal{R}$ be a terminating TRS. Then, $\mathcal{R}$ is sufficiently complete if and only if $\mathcal{R}$ is quasi-reducible.*

In this paper we do not use the above proposition, but the proof method by applying our derivation system can be seen as an extension of checking quasi-reducibility.

Next we introduce some notions on orders. A (*strict*) *partial order* is a binary relation that is irreflexive and transitive. A partial order $\succ$ on terms is *monotonic* if it is closed under context, i.e. $s \succ t$ implies $C[s] \succ C[t]$ for every context $C[\,]$; it is *stable* if it is closed

under substitution, i.e. $s \succ t$ implies $s\theta \succ t\theta$ for every substitution θ; it is *well-founded* if there exists no infinite descending chain $t_0 \succ t_1 \succ \cdots$; it has *the subterm property* if $s \trianglelefteq t$ and $s \neq t$ imply $t \succ s$.

A well-founded monotonic stable partial order with the subterm property is called a *simplification order*, and many methods for constructing such an order are known (cf. [1, 16]).

3 A Simple Derivation System for Sufficient Completeness

In this section, we present a derivation system for proving sufficient completeness of TRSs. We illustrate the proof method by applying it to some non-terminating TRSs.

In the following, we use a lexicographic path order as an order that is required to define the derivation system.

▶ **Definition 3** (Lexicographic path order)**.** Let $>$ be a partial order, called a *precedence*, on the set $\mathcal{F}$ of function symbols. The *lexicographic path order* $>_{lpo}$ on $T(\mathcal{F},\mathcal{V})$ induced by the precedence $>$ is defined inductively as follows: $s >_{lpo} t$ iff

1. $t \in \mathcal{V}(s)$ and $s \neq t$, or
2. $s = f(s_1,\ldots,s_m)$, $t = g(t_1,\ldots,t_n)$, and
   a. there exists i $(1 \leq i \leq m)$ such that $s_i >_{lpo} t$ or $s_i = t$, or
   b. $f > g$ and $s >_{lpo} t_j$ for every j $(1 \leq j \leq n)$, or
   c. $f = g$, $s >_{lpo} t_j$ for every j $(1 \leq j \leq n)$, and there exists i $(1 \leq i \leq m,\ i \leq n)$ such that $s_1 = t_1$, …, $s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

Lexicographic path orders have the following properties.

▶ **Proposition 4.** *Every lexicographic path order $>_{lpo}$ induced by any precedence $>$ on $\mathcal{F}$ is a simplification order. Furthermore, $s >_{lpo} t$ implies $\mathcal{V}(t) \subseteq \mathcal{V}(s)$ for every $s,t \in T(\mathcal{F},\mathcal{V})$.*

▶ **Lemma 5.** *Let $>$ be a precedence on $\mathcal{F}$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$. Then the lexicographic path order $>_{lpo}$ induced by $>$ satisfies $s_g >_{lpo} t_g$ for every $s_g \in T(\mathcal{F}) \setminus T(\mathcal{C})$ and $t_g \in T(\mathcal{C})$.*

Next we introduce the derivation system, which acts on a set of guarded terms.

▶ **Definition 6** (Guarded term)**.** A *guarded term*, denoted by $t|H$, consists of a term t and a set H of terms. We write $H\theta$ for the set $\{u\theta \mid u \in H\}$.

A derivation starts from a singleton set consisting of a guarded term of the form $\{t|\{t\}\}$. Intuitively, it means the premise of well-founded induction for ground instances of t with respect to the order $>_{lpo}$. Derivation rules subsequently transform the set of guarded terms[1], preserving the meaning of the well-founded induction schema (Definition 10).

▶ **Definition 7** (Derivation)**.** Let $\mathcal{R}$ be a TRS, and let $>_{lpo}$ be a lexicographic path order induced by some precedence $>$ on $\mathcal{F}$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$.

- The derivation rules of the system are listed in Figure 1. It derives from a set of guarded terms (given at the upper side) a set of guarded terms (given at the lower side) if the side condition is satisfied.
- For sets Γ, Γ' of guarded terms, we write $\Gamma \rightsquigarrow \Gamma'$ if Γ' is derived from Γ by one of the derivation rules. The reflexive transitive closure of $\rightsquigarrow$ is written as $\overset{*}{\rightsquigarrow}$.

Decompose

$$\frac{\Gamma \cup \{f(t_1, \ldots, t_n)|H\}}{\Gamma \cup \{t_1|H, \ldots, t_n|H\}}\ f \in \mathcal{C}$$

Expand

$$\frac{\Gamma \cup \{t|H\}}{\Gamma \cup \{t\sigma_i|H\sigma_i\}_i}\quad \{\sigma_i\}_i = \{\{x \mapsto f(\vec{x})\} \mid f \in \mathcal{C},\ sort(x) = sort(f(\vec{x}))\}$$
where $x \in \mathcal{V}(t)$ and $\vec{x}$ is a sequence of fresh variables

Simplify

$$\frac{\Gamma \cup \{t|H\}}{\Gamma \cup \{s|H\}}\ t \to_{\mathcal{R}} s$$

Delete

$$\frac{\Gamma \cup \{t|H\}}{\Gamma}\ \exists u \in H.\ t <_{lpo} u$$

Figure 1 Derivation rules for proving sufficient completeness.

The *Expand* rule substitutes a variable in t by each pattern with a constructor as its root symbol, and yields the same number of guarded terms as the constructors. (The index i ranges over a set isomorphic to $\{f \in \mathcal{C} \mid sort(x) = sort(f(\vec{x}))\}$.) The *Delete* rule removes the guarded term $t|H$ if t is less than some $u \in H$ with respect to $>_{lpo}$.

We have a lemma on preservation of variables occurring in guarded terms.

▶ **Definition 8** (*VP*). *For a set Γ of guarded terms, $VP(\Gamma)$ means that for every $t|H \in \Gamma$ and every $x \in \mathcal{V}(t)$, there exists $u \in H \setminus \mathcal{V}$ such that $x \in \mathcal{V}(u)$.*

▶ **Lemma 9.** *If $\Gamma \rightsquigarrow \Gamma'$ and $VP(\Gamma)$ then $VP(\Gamma')$.*

The predicate about the well-founded induction schema is defined as follows.

▶ **Definition 10** (*WIS*). *For a set Γ of guarded terms, $WIS(\Gamma)$ means that for every $t|H \in \Gamma$ and every ground substitution σ_g, the following holds:*

$$(\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)) \Rightarrow SC(t\sigma_g). \tag{WIS 1}$$

Now we prove a key lemma on the well-founded induction schema.

▶ **Lemma 11.** *Let $\Gamma \rightsquigarrow \Gamma'$ and $VP(\Gamma)$. Then, $WIS(\Gamma')$ implies $WIS(\Gamma)$.*

Proof. Let $\Gamma \rightsquigarrow \Gamma'$ and $VP(\Gamma)$. We prove that if

$$(\forall u \in H'.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)) \Rightarrow SC(t'\sigma_g)$$

for every $t'|H' \in \Gamma'$ and every ground substitution σ_g, then

$$(\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)) \Rightarrow SC(t\sigma_g)$$

for every $t|H \in \Gamma$ and every ground substitution σ_g. The proof is by case analysis depending on the rule used in the derivation step $\Gamma \rightsquigarrow \Gamma'$.

[1] Actually, for any guarded term $t|H$ in a derivation starting with the form $\{t|\{t\}\}$, a singleton set H is sufficient, but we prove lemmas in the general setting for future developments.

(*Decompose*) Then $\Gamma = \Sigma \cup \{f(t_1, \ldots, t_n)|H'\}$ and $\Gamma' = \Sigma \cup \{t_1|H', \ldots, t_n|H'\}$, where $f \in \mathcal{C}$. The case $t|H \in \Sigma$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = f(t_1, \ldots, t_n)|H'$. Let σ_g be a ground substitution such that $\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)$. Then, by the hypothesis for Γ', we have $SC(t_i\sigma_g)$ for every i $(1 \leq i \leq n)$. Thus, since $f \in \mathcal{C}$, we have $SC(f(t_1, \ldots, t_n)\sigma_g)$.

(*Expand*) Then $\Gamma = \Sigma \cup \{t'|H'\}$ and $\Gamma' = \Sigma \cup \{t'\sigma_i|H'\sigma_i\}_i$, where $x \in \mathcal{V}(t')$, $\vec{x}$ is a sequence of fresh variables, and $\{\sigma_i\}_i = \{\{x \mapsto f(\vec{x})\} \mid f \in \mathcal{C},\ sort(x) = sort(f(\vec{x}))\}$. The case $t|H \in \Sigma$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = t'|H'$. Let σ_g be a ground substitution such that (β): $\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)$. Our aim is to show $SC(t\sigma_g)$. For this, we distinguish two cases.

1. Suppose that there exists an index i such that $t\sigma_g = (t\sigma_i)\sigma'_g$ for some σ'_g. Then, by $t\sigma_i|H\sigma_i \in \Gamma'$, we know from the hypothesis for Γ' that if $\forall u \in H\sigma_i.\ \forall w_g <_{lpo} u\sigma'_g.\ SC(w_g)$ then $SC((t\sigma_i)\sigma'_g)$. Since $t\sigma_g = (t\sigma_i)\sigma'_g$, it remains to show $\forall u \in H\sigma_i.\ \forall w_g <_{lpo} u\sigma'_g.\ SC(w_g)$. Suppose $u \in H\sigma_i$ and $w_g <_{lpo} u\sigma'_g$. Then, there exists $\hat{u} \in H$ such that $u = \hat{u}\sigma_i$, and we have $w_g <_{lpo} u\sigma'_g = (\hat{u}\sigma_i)\sigma'_g = \hat{u}\sigma_g$. Thus, $SC(w_g)$ holds by our assumption (β).
2. Otherwise. Then we have $t\sigma_g = (t\theta)\sigma'_g$ for some $\theta = \{x \mapsto f(\vec{y})\}$ with $f \in \mathcal{D}$. By $VP(\Gamma)$ and the subterm property of $>_{lpo}$, we have $x <_{lpo} u$ for some $u \in H$, and so by the stability of $>_{lpo}$, we have $x\sigma_g <_{lpo} u\sigma_g$ for some $u \in H$. Hence by our assumption (β), we obtain $SC(x\sigma_g)$. Then by the definition, we know there exists a ground constructor term $g(\vec{w}_g) \in T(\mathcal{C})$ such that $x\sigma_g \xrightarrow{*}_{\mathcal{R}} g(\vec{w}_g)$. From $g \in \mathcal{C}$, there is an index i such that $\sigma_i = \{x \mapsto g(\vec{x})\}$. As $\vec{x}$ is fresh, we may assume $x\sigma_g \xrightarrow{*}_{\mathcal{R}} g(\vec{w}_g) = (x\sigma_i)\sigma'_g$. Hence $t\sigma_g \xrightarrow{*}_{\mathcal{R}} (t\sigma_i)\sigma'_g$. Also, we have $x\sigma_g >_{lpo} (x\sigma_i)\sigma'_g$ by Lemma 5.

   The reminder of the proof proceeds in a similar way to 1 except that $t\sigma_g \xrightarrow{*}_{\mathcal{R}} (t\sigma_i)\sigma'_g$ instead of $t\sigma_g = (t\sigma_i)\sigma'_g$. By $t\sigma_i|H\sigma_i \in \Gamma'$, we know from the hypothesis for Γ' that if $\forall u \in H\sigma_i.\ \forall w_g <_{lpo} u\sigma'_g.\ SC(w_g)$ then $SC((t\sigma_i)\sigma'_g)$, which implies $SC(t\sigma_g)$ since $t\sigma_g \xrightarrow{*}_{\mathcal{R}} (t\sigma_i)\sigma'_g$. Thus, it remains to show $\forall u \in H\sigma_i.\ \forall w_g <_{lpo} u\sigma'_g.\ SC(w_g)$. Suppose $u \in H\sigma_i$ and $w_g <_{lpo} u\sigma'_g$. Then, there exists $\hat{u} \in H$ such that $u = \hat{u}\sigma_i$, and we have $w_g <_{lpo} u\sigma'_g = (\hat{u}\sigma_i)\sigma'_g \leq_{lpo} \hat{u}\sigma_g$, where the last part follows from $x\sigma_g >_{lpo} (x\sigma_i)\sigma'_g$ by the monotonicity of $>_{lpo}$ (or $(\hat{u}\sigma_i)\sigma'_g = \hat{u}\sigma_g$ if $x \notin \mathcal{V}(\hat{u})$). Hence $SC(w_g)$ holds by our assumption (β).

(*Simplify*) Then $\Gamma = \Sigma \cup \{t'|H'\}$ and $\Gamma' = \Sigma \cup \{s'|H'\}$, where $t' \to_{\mathcal{R}} s'$. The case $t|H \in \Sigma$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = t'|H'$. Let σ_g be a ground substitution such that $\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)$. Then, by the hypothesis for Γ', we have $SC(s'\sigma_g)$. Since $t\sigma_g = t'\sigma_g \to_{\mathcal{R}} s'\sigma_g$, $SC(t\sigma_g)$ clearly holds.

(*Delete*) Then $\Gamma = \Gamma' \cup \{t'|H'\}$, where $t' <_{lpo} u$ for some $u \in H'$. The case $t|H \in \Gamma'$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = t'|H'$. Let σ_g be a ground substitution such that $\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)$. Since $t = t' <_{lpo} u$ for some $u \in H' = H$, we have $t\sigma_g <_{lpo} u\sigma_g$ for some $u \in H$. Hence $SC(t\sigma_g)$ holds. ◀

Now we are ready to show the theorem on global sufficient completeness of a TRS. In the following proof of the theorem, we use the fact that every ground term t_g has the form $h(\vec{x})\theta_g$ for some $h \in \mathcal{F}$ and some ground substitution θ_g.

▶ **Theorem 12.** *Let $\mathcal{R}$ be a TRS, and let $>_{lpo}$ be a lexicographic path order induced by some precedence $>$ on $\mathcal{F}$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$. If $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow} \{\,\}$ for every $h \in \mathcal{F}$, then $\mathcal{R}$ is sufficiently complete.*

Proof. Let $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow} \{\,\}$ for every $h \in \mathcal{F}$. We prove $SC(h(\vec{x})\theta_g)$ for every $h \in \mathcal{F}$ and every ground instance $h(\vec{x})\theta_g$ by induction on $T(\mathcal{F})$ with the lexicographic order $>_{lpo}$. Let $h \in \mathcal{F}$ and consider the derivation $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow} \{\,\}$. Then by Lemma 9, $VP(\Gamma)$ holds for every Γ appearing in the derivation. Since $WIS(\{\,\})$ vacuously holds, we have by Lemma 11 $WIS(\{h(\vec{x})|\{h(\vec{x})\}\})$, i.e.,

$$(\forall w_g <_{lpo} h(\vec{x})\theta_g.\ SC(w_g)) \Rightarrow SC(h(\vec{x})\theta_g) \tag{WIS 2}$$

for each ground instance $h(\vec{x})\theta_g$. To prove $SC(h(\vec{x})\theta_g)$, it suffices to show $\forall w_g <_{lpo} h(\vec{x})\theta_g.\ SC(w_g)$. Let $w_g <_{lpo} h(\vec{x})\theta_g$. Since $w_g = f(\vec{y})\rho_g$ for some $f \in \mathcal{F}$ and some ground substitution ρ_g, we have $SC(w_g)$ by the induction hypothesis. Hence $\forall w_g <_{lpo} h(\vec{x})\theta_g.\ SC(w_g)$, and we obtain $SC(h(\vec{x})\theta_g)$. ◀

We give some examples of application of the theorem.

▶ **Example 13** ([4, Example 8.1])**.** Consider a signature with $\mathcal{S} = \{B\}$ and

$$\mathcal{F} = \left\{ \begin{array}{lll} \mathsf{and} : B \times B \to B, & \mathsf{or} : B \times B \to B, \\ \mathsf{not} : B \to B, & \mathsf{0} : B, & \mathsf{1} : B \end{array} \right\}$$

where $\mathcal{C} = \{\mathsf{0} : B,\ \mathsf{1} : B\}$. Let $\mathcal{R}_1$ be the following TRS:

$$\mathcal{R}_1 = \left\{ \begin{array}{llll} (1) & \mathsf{and}(\mathsf{1}, x) & \to & x \\ (2) & \mathsf{and}(\mathsf{0}, x) & \to & \mathsf{0} \\ (3) & \mathsf{or}(\mathsf{1}, x) & \to & \mathsf{1} \\ (4) & \mathsf{or}(\mathsf{0}, x) & \to & x \\ (5) & \mathsf{and}(\mathsf{1}, x) & \to & \mathsf{not}(\mathsf{not}(\mathsf{and}(\mathsf{1}, x))) \\ (6) & \mathsf{not}(\mathsf{1}) & \to & \mathsf{0} \\ (7) & \mathsf{not}(\mathsf{0}) & \to & \mathsf{1} \\ (8) & \mathsf{not}(\mathsf{and}(x, y)) & \to & \mathsf{or}(\mathsf{not}(x), \mathsf{not}(y)) \end{array} \right\}.$$

Note that $\mathcal{R}_1$ is not terminating since $\mathsf{and}(\mathsf{1}, x) \to_{\mathcal{R}_1} \mathsf{not}(\mathsf{not}(\mathsf{and}(\mathsf{1}, x))) \to_{\mathcal{R}_1} \mathsf{not}(\mathsf{not}(\mathsf{not}(\mathsf{not}(\mathsf{and}(\mathsf{1}, x))))) \to_{\mathcal{R}_1} \cdots$. We show that $\mathcal{R}_1$ is sufficiently complete, using Theorem 12. For this, take a lexicographic path ordering $>_{lpo}$ induced by any precedence $>$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$. In Figure 2, we give derivations of $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow} \{\,\}$ for $h \in \mathcal{D}$. Also, we have $\{\mathsf{0}|\{\mathsf{0}\}\} \rightsquigarrow \{\,\}$ and $\{\mathsf{1}|\{\mathsf{1}\}\} \rightsquigarrow \{\,\}$ using *Decompose*. Thus by Theorem 12, $\mathcal{R}_1$ is sufficiently complete. ◀

The next example shows that there exists a TRS for which the method of [3, 4] does not work but our method works.

▶ **Example 14.** Let $\mathcal{R}_2$ be the TRS obtained from $\mathcal{R}_1$ of Example 13 by deleting the rule (1). Then $\mathcal{R}_2$ is still sufficiently complete. Indeed, derivations for defined symbols except and are the same as those of Figure 2. For and, we have a derivation of $\{\mathsf{and}(x_1, x_2)|\{\mathsf{and}(x_1, x_2)\}\} \overset{*}{\rightsquigarrow} \{\,\}$ as shown in Figure 3. Note that the *Simplify* step using the rule (8) (the fifth row in the figure) cannot be made by the abstract-narrow-based process of [3, 4] (cf. Appendix A). ◀

$$
\begin{array}{ll}
 & \{\ \mathsf{not}(x)|\{\mathsf{not}(x)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{not}(0)|\{\mathsf{not}(0)\},\ \mathsf{not}(1)|\{\mathsf{not}(1)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ 1|\{\mathsf{not}(0)\},\ 0|\{\mathsf{not}(1)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Decompose} & \{\ \ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \{\ \mathsf{or}(x_1,x_2)|\{\mathsf{or}(x_1,x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{or}(0,x_2)|\{\mathsf{or}(0,x_2)\},\ \mathsf{or}(1,x_2)|\{\mathsf{or}(1,x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ x_2|\{\mathsf{or}(0,x_2)\},\ 1|\{\mathsf{or}(1,x_2)\}\ \} \\
\rightsquigarrow_{Decompose} & \{\ x_2|\{\mathsf{or}(0,x_2)\}\ \} \\
\rightsquigarrow_{Delete} & \{\ \ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \{\ \mathsf{and}(x_1,x_2)|\{\mathsf{and}(x_1,x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{and}(0,x_2)|\{\mathsf{and}(0,x_2)\},\ \mathsf{and}(1,x_2)|\{\mathsf{and}(1,x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ 0|\{\mathsf{and}(0,x_2)\},\ x_2|\{\mathsf{and}(1,x_2)\}\ \} \\
\rightsquigarrow_{Decompose} & \{\ x_2|\{\mathsf{and}(1,x_2)\}\ \} \\
\rightsquigarrow_{Delete} & \{\ \ \}
\end{array}
$$

Figure 2 Derivations for proving sufficient completeness of $\mathcal{R}_1$ in Example 13.

$$
\begin{array}{ll}
 & \{\ \mathsf{and}(x_1,x_2)|\{\mathsf{and}(x_1,x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{and}(0,x_2)|\{\mathsf{and}(0,x_2)\},\ \mathsf{and}(1,x_2)|\{\mathsf{and}(1,x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ 0|\{\mathsf{and}(0,x_2)\},\ \mathsf{not}(\mathsf{not}(\mathsf{and}(1,x_2)))|\{\mathsf{and}(1,x_2)\}\ \} \\
\rightsquigarrow_{Decompose} & \{\ \mathsf{not}(\mathsf{not}(\mathsf{and}(1,x_2)))|\{\mathsf{and}(1,x_2)\}\ \} \\
\rightsquigarrow_{Simplify} & \{\ \mathsf{not}(\mathsf{or}(\mathsf{not}(1),\mathsf{not}(x_2)))|\{\mathsf{and}(1,x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ \mathsf{not}(\mathsf{not}(x_2))|\{\mathsf{and}(1,x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{not}(\mathsf{not}(0))|\{\mathsf{and}(1,0)\},\ \mathsf{not}(\mathsf{not}(1))|\{\mathsf{and}(1,1)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ 0|\{\mathsf{and}(1,0)\},\ 1|\{\mathsf{and}(1,1)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Decompose} & \{\ \ \}
\end{array}
$$

Figure 3 A derivation for proving sufficient completeness of $\mathcal{R}_2$ in Example 14.

The next example is a modification of [17, Example A.2].

▶ **Example 15.** Consider a signature with $\mathcal{S} = \{N\}$ and

$$
\mathcal{F} = \left\{ \begin{array}{lll} \mathsf{d} : N \to N, & \mathsf{if} : N \times N \times N \to N, & \\ - : N \times N \to N, & 0 : N, & \mathsf{s} : N \to N \end{array} \right\}
$$

where $\mathcal{C} = \{0 : N,\ \mathsf{s} : N \to N\}$. Let $\mathcal{R}_3$ be the following TRS where $\mathsf{d}(n)$ computes the double of a given natural number n:

$$
\mathcal{R}_3 = \left\{ \begin{array}{lll}
(1) & \mathsf{d}(x) & \to \ \mathsf{if}(x, 0, \mathsf{s}(\mathsf{s}(\mathsf{d}(-(x, \mathsf{s}(0)))))) \\
(2) & \mathsf{if}(0, y, z) & \to \ y \\
(3) & \mathsf{if}(\mathsf{s}(x), y, z) & \to \ z \\
(4) & -(0, y) & \to \ 0 \\
(5) & -(x, 0) & \to \ x \\
(6) & -(\mathsf{s}(x), \mathsf{s}(y)) & \to \ -(x, y)
\end{array} \right\}.
$$

Note that $\mathcal{R}_3$ is not terminating by repeated application of the rule (1). We show that $\mathcal{R}_3$ is sufficiently complete, using Theorem 12. For this, take a lexicographic path ordering $>_{lpo}$ induced by any precedence $>$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$. In Figure 4,

$$
\begin{array}{ll}
 & \{\ \mathsf{d}(x)|\{\mathsf{d}(x)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{d}(0)|\{\mathsf{d}(0)\},\ \mathsf{d}(\mathsf{s}(x_1))|\{\mathsf{d}(\mathsf{s}(x_1))\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ 0|\{\mathsf{d}(0)\},\ \mathsf{s}(\mathsf{s}(\mathsf{d}(-(\mathsf{s}(x_1),\mathsf{s}(0)))))|\{\mathsf{d}(\mathsf{s}(x_1))\}\ \} \\
\overset{*}{\rightsquigarrow}_{Decompose} & \{\ \mathsf{d}(-(\mathsf{s}(x_1),\mathsf{s}(0)))|\{\mathsf{d}(\mathsf{s}(x_1))\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ \mathsf{d}(x_1)|\{\mathsf{d}(\mathsf{s}(x_1))\}\ \} \\
\rightsquigarrow_{Delete} & \{\ \ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \{\ \mathsf{if}(x_1,x_2,x_3)|\{\mathsf{if}(x_1,x_2,x_3)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ \mathsf{if}(0,x_2,x_3)|\{\mathsf{if}(0,x_2,x_3)\},\ \mathsf{if}(\mathsf{s}(x_4),x_2,x_3)|\{\mathsf{if}(\mathsf{s}(x_4),x_2,x_3)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ x_2|\{\mathsf{if}(0,x_2,x_3)\},\ x_3|\{\mathsf{if}(\mathsf{s}(x_4),x_2,x_3)\}\ \} \\
\overset{*}{\rightsquigarrow}_{Delete} & \{\ \ \}
\end{array}
$$

$$
\begin{array}{ll}
 & \{\ -(x_1,x_2)|\{-(x_1,x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ -(0,x_2)|\{-(0,x_2)\},\ -(\mathsf{s}(x_3),x_2)|\{-(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{Simplify} & \{\ 0|\{-(0,x_2)\},\ -(\mathsf{s}(x_3),x_2)|\{-(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{Decompose} & \{\ -(\mathsf{s}(x_3),x_2)|\{-(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{Expand} & \{\ -(\mathsf{s}(x_3),0)|\{-(\mathsf{s}(x_3),0)\},\ -(\mathsf{s}(x_3),\mathsf{s}(x_4))|\{-(\mathsf{s}(x_3),\mathsf{s}(x_4))\}\ \} \\
\overset{*}{\rightsquigarrow}_{Simplify} & \{\ \mathsf{s}(x_3)|\{-(\mathsf{s}(x_3),0)\},\ -(x_3,x_4)|\{-(\mathsf{s}(x_3),\mathsf{s}(x_4))\}\ \} \\
\overset{*}{\rightsquigarrow}_{Delete} & \{\ \ \}
\end{array}
$$

Figure 4 Derivations for proving sufficient completeness of $\mathcal{R}_3$ in Example 15.

we give derivations of $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow} \{\,\}$ for $h \in \mathcal{D}$. Also, we have $\{0|\{0\}\} \rightsquigarrow \{\,\}$ and $\{\mathsf{s}(x)|\{\mathsf{s}(x)\}\} \rightsquigarrow \{\,\}$ using *Decompose* and *Delete*. Thus by Theorem 12, $\mathcal{R}_3$ is sufficiently complete. ◀

Without the rule (4), the above TRS $\mathcal{R}_3$ still has some kind of sufficient completeness, which we discuss in the next section.

4 A Simple Derivation System for Local Sufficient Completeness with Signature Restriction

In the remainder of the paper, we are concerned with local sufficient completeness [10], which is a generalised notion of sufficient completeness. In this section, we consider local sufficient completeness on the set of ground terms consisting of particular function symbols.

▶ **Definition 16** (Local sufficient completeness with signature restriction)**.** Let $\mathcal{R}$ be a TRS, and let $\mathcal{F}' \subseteq \mathcal{F}$. Then $\mathcal{R}$ is *locally sufficiently complete* on $T(\mathcal{F}')$ if $SC(t_g)$ for every $t_g \in T(\mathcal{F}')$.

The standard notion of sufficient completeness, as given in Definition 1, is the case of local sufficient completeness on $T(\mathcal{F}')$ where $\mathcal{F}' = \mathcal{F}$. By the restriction to $\mathcal{F}'$, we can talk about sufficient completeness locally, e.g. on $T(\{\mathsf{d},\mathsf{s},0\})$ in Example 15 (cf. Example 25).

The notion of a derivation of the system in this section is defined as follows.

▶ **Definition 17** (Derivation)**.** Let $\mathcal{R}$ be a TRS, and let $\mathcal{F}' \subseteq \mathcal{F}$. Suppose that $>_{lpo}$ is a lexicographic path order induced by some precedence $>$ on $\mathcal{F}$ such that $f > f' > g$ for every $f \in \mathcal{D} \setminus \mathcal{F}'$, $f' \in \mathcal{F}' \setminus \mathcal{C}$ and $g \in \mathcal{C}$. The derivation rules of the system are the same as those listed in Figure 1. We write $\Gamma \rightsquigarrow_\ell \Gamma'$ if Γ' is derived from Γ by one of the derivation rules.

▶ **Lemma 18.** *Let $>_{lpo}$ be a lexicographic path order induced by a precedence $>$ as above. If $s_g \in T(\mathcal{F}' \cup \mathcal{C})$ and $s_g >_{lpo} t_g$ then $t_g \in T(\mathcal{F}' \cup \mathcal{C})$.*

As before, we have a lemma on preservation of variables occurring in guarded terms.

▶ **Lemma 19.** *If* $\Gamma \rightsquigarrow_\ell \Gamma'$ *and* $VP(\Gamma)$ *then* $VP(\Gamma')$.

In addition, we have a lemma on preservation of function symbols of guarded terms.

▶ **Definition 20** (*SigP*). *For a set* Γ *of guarded terms,* $SigP(\Gamma)$ *means that for every* $t|H \in \Gamma$ *and every* $u \in H$, $u \in T(\mathcal{F}' \cup \mathcal{C}, \mathcal{V})$.

▶ **Lemma 21.** *If* $\Gamma \rightsquigarrow_\ell \Gamma'$ *and* $SigP(\Gamma)$ *then* $SigP(\Gamma')$.

The predicate about the well-founded induction schema is defined as follows.

▶ **Definition 22** (WIS_ℓ). *For a set* Γ *of guarded terms,* $WIS_\ell(\Gamma)$ *means that for every* $t|H \in \Gamma$ *and every ground substitution* $\sigma_g : \mathcal{V} \to T(\mathcal{F}' \cup \mathcal{C})$, *the following holds:*

$$(\forall u \in H.\ \forall w_g <_{lpo} u\sigma_g.\ SC(w_g)) \Rightarrow SC(t\sigma_g). \tag{WIS 3}$$

▶ **Lemma 23.** *Let* $\Gamma \rightsquigarrow_\ell \Gamma'$, $VP(\Gamma)$ *and* $SigP(\Gamma)$. *Then,* $WIS_\ell(\Gamma')$ *implies* $WIS_\ell(\Gamma)$.

Proof. The proof proceeds by case analysis in the same way as that of Lemma 11, except that $\sigma : \mathcal{V} \to T(\mathcal{F}' \cup \mathcal{C})$ is enforced on every ground substitution σ appearing in the proof. ◀

We are now ready to show the theorem on local sufficient completeness on $T(\mathcal{F}')$.

▶ **Theorem 24.** *Let* $\mathcal{R}$ *be a TRS,* $\mathcal{F}' \subseteq \mathcal{F}$ *and* $>_{lpo}$ *be a lexicographic path order induced by some precedence* $>$ *on* $\mathcal{F}$ *such that* $f > f' > g$ *for every* $f \in \mathcal{D} \setminus \mathcal{F}'$, $f' \in \mathcal{F}' \setminus \mathcal{C}$ *and* $g \in \mathcal{C}$. *If* $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow}_\ell \{\,\}$ *for every* $h \in \mathcal{F}'$, *then* $\mathcal{R}$ *is locally sufficiently complete on* $T(\mathcal{F}')$.

Proof. Let $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow}_\ell \{\,\}$ for every $h \in \mathcal{F}'$. (Using *Decompose* and *Delete*, we can automatically have $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow}_\ell \{\,\}$ for every $h \in \mathcal{C}$.) It suffices to prove $SC(h(\vec{x})\theta_g)$ for every $h \in \mathcal{F}' \cup \mathcal{C}$ and every ground instance $h(\vec{x})\theta_g$ with $\theta_g : \mathcal{V} \to T(\mathcal{F}' \cup \mathcal{C})$ by induction on $T(\mathcal{F}' \cup \mathcal{C})$ with respect to $>_{lpo}$. This is shown by a similar argument to that of Theorem 12, and we have that $\mathcal{R}$ is locally sufficiently complete on $T(\mathcal{F}' \cup \mathcal{C})$ and thus on $T(\mathcal{F}')$. ◀

Now we consider the example mentioned at the end of the previous section.

▶ **Example 25.** Let $\mathcal{R}_4$ be the TRS obtained from $\mathcal{R}_3$ of Example 15 by deleting the rule (4). Then $\mathcal{R}_4$ is not globally sufficiently complete any more, since $-(\mathsf{0}, \mathsf{s}(\mathsf{0})) \in NF(\mathcal{R}_4)$ but $-(\mathsf{0}, \mathsf{s}(\mathsf{0})) \notin T(\mathcal{C})$. However, it can be shown that $\mathcal{R}_4$ is locally sufficiently complete on $T(\mathcal{F}')$ where $\mathcal{F}' = \{\mathsf{d}, \mathsf{s}, \mathsf{0}\}$. Indeed, the derivation for the function symbol $\mathsf{d} \in \mathcal{D}$ shown in Figure 4 works where the rule (4) is not used in the *Simplify* steps. The precedence can be given as $\mathsf{if}, - > \mathsf{d} > \mathsf{s}, \mathsf{0}$. Hence by Theorem 24, $\mathcal{R}_4$ is locally sufficiently complete on $T(\mathcal{F}')$. ◀

5 A Simple Derivation System for Local Sufficient Completeness with Sort Partition

In this section, we treat another type of local sufficient completeness than that discussed in the previous section. Specifically, we consider local sufficient completeness on the set of ground terms of particular sorts.

▶ **Definition 26** (Conditions on the signature). We assume the following conditions S1–S4 on the signature of $\mathcal{R}$.

S1. $\mathcal{S} = \mathcal{S}_0 \uplus \mathcal{S}_1$. ($\uplus$ stands for the disjoint union.)

S2. The sets $\mathcal{F}_i$ $(i = 0, 1)$ of function symbols are defined by $\mathcal{F}_i = \{f \in \mathcal{F} \mid f : \alpha_1 \times \cdots \times \alpha_n \to \alpha,\ \alpha \in \mathcal{S}_i\}$.
S3. The sets T_i $(i = 0, 1)$ of ground terms are defined by $T_i = \{t_g \in T(\mathcal{F}) \mid sort(t_g) \in \mathcal{S}_i\}$.
S4. For every $g \in \mathcal{F}_0 \cap \mathcal{C}$, if $g : \alpha_1 \times \cdots \times \alpha_n \to \alpha$ then $\alpha_1, \ldots, \alpha_n \in \mathcal{S}_0$.

Our aim in this section is to show that the following holds under certain conditions.

▶ **Definition 27** (Local sufficient completeness with sort partition)**.** Let $\mathcal{R}$ be a TRS with a signature satisfying the conditions S1–S4 of Definition 26. Then $\mathcal{R}$ is said to be *locally sufficiently complete* on T_0 if $SC(t_g)$ for every $t_g \in T_0$.

The next example illustrates the difference between global sufficient completeness and local sufficient completeness on T_0.

▶ **Example 28** ([15, Example 8])**.** Consider a signature with $\mathcal{S}_0 = \{N\}$, $\mathcal{S}_1 = \{L\}$ and

$$\mathcal{F} = \left\{ \begin{array}{llll} \mathsf{sum} : N \times L \to N, & + : N \times N \to N, & \mathsf{from} : N \to L, \\ \mathsf{0} : N, & \mathsf{s} : N \to N, & [] : L, & :: : N \times L \to L \end{array} \right\}$$

where $\mathcal{C} = \{\mathsf{0} : N,\ \mathsf{s} : N \to N,\ [] : L,\ :: : N \times L \to L\}$. Let $\mathcal{R}_5$ be the following TRS where $\mathsf{sum}(n, ts)$ computes the summation of the first n elements of a (possibly infinite) list ts of natural numbers:

$$\mathcal{R}_5 = \left\{ \begin{array}{lllll} (1) & \mathsf{sum}(\mathsf{0}, xs) & \to & \mathsf{0} \\ (2) & \mathsf{sum}(\mathsf{s}(x), []) & \to & \mathsf{0} \\ (3) & \mathsf{sum}(\mathsf{s}(x), y :: ys) & \to & +(y, \mathsf{sum}(x, ys)) \\ (4) & +(\mathsf{0}, y) & \to & y \\ (5) & +(\mathsf{s}(x), y) & \to & \mathsf{s}(+(x, y)) \\ (6) & \mathsf{from}(x) & \to & x :: \mathsf{from}(\mathsf{s}(x)) \end{array} \right\}.$$

$\mathcal{R}_5$ is not terminating since $\mathsf{from}(\mathsf{0}) \to_{\mathcal{R}} \mathsf{0} :: \mathsf{from}(\mathsf{s}(\mathsf{0})) \to_{\mathcal{R}} \mathsf{0} :: \mathsf{s}(\mathsf{0}) :: \mathsf{from}(\mathsf{s}(\mathsf{s}(\mathsf{0}))) \to_{\mathcal{R}} \cdots$. $\mathcal{R}_5$ is not globally sufficiently complete either since $u \notin T(\mathcal{C})$ for any u with $\mathsf{from}(\mathsf{0}) \xrightarrow{*}_{\mathcal{R}} u$. However, it can be shown that $\mathcal{R}_5$ is locally sufficiently complete on T_0 (cf. Example 35). ◀

The notion of a derivation of the system in this section is defined similarly to Definition 7 except that the *Expand* rule is replaced by

Expand

$$\frac{\Gamma \cup \{t|H\}}{\Gamma \cup \{t\sigma_i|H\sigma_i\}_i} \quad \begin{array}{l} \{\sigma_i\}_i = \{\{x \mapsto f(\vec{x})\} \mid f \in \mathcal{C} \cup \mathcal{F}_1,\ sort(x) = sort(f(\vec{x}))\} \\ \text{where } x \in \mathcal{V}(t) \text{ and } \vec{x} \text{ is a sequence of fresh variables} \end{array}$$

We write $\Gamma \rightsquigarrow_S \Gamma'$ if Γ' is derived from Γ by one of the derivation rules.

▶ **Lemma 29.** *If* $\Gamma \rightsquigarrow_S \Gamma'$ *and* $VP(\Gamma)$ *then* $VP(\Gamma')$.

In addition, we have a lemma on preservation of sorts of guarded terms.

▶ **Definition 30** (*SrtP*)**.** *For a set* Γ *of guarded terms,* $SrtP(\Gamma)$ *means that for every* $t|H \in \Gamma$, $sort(t) \in \mathcal{S}_0$ *and* $sort(u) \in \mathcal{S}_0$ *for every* $u \in H$.

▶ **Lemma 31.** *If* $\Gamma \rightsquigarrow_S \Gamma'$ *and* $SrtP(\Gamma)$ *then* $SrtP(\Gamma')$.

Proof. By case analysis depending on the rule used in the derivation step $\Gamma \rightsquigarrow_S \Gamma'$. In the case of the *Decompose* rule, we use the condition S4 of Definition 26. ◀

The predicate about the well-founded induction schema is defined as follows.

▶ **Definition 32** (WIS_S)**.** *For a set Γ of guarded terms, $WIS_S(\Gamma)$ means that for every $t|H \in \Gamma$ and every ground substitution σ_g, the following holds:*

$$(\forall u \in H.\ \forall w_g \in T_0.\ w_g <_{lpo} u\sigma_g \Rightarrow SC(w_g)) \Rightarrow SC(t\sigma_g). \tag{WIS 4}$$

▶ **Lemma 33.** *Let $\Gamma \rightsquigarrow_S \Gamma'$, $VP(\Gamma)$ and $SrtP(\Gamma)$. Then, $WIS_S(\Gamma')$ implies $WIS_S(\Gamma)$.*

Proof. We prove that if

$$(\forall u \in H'.\ \forall w_g \in T_0.\ w_g <_{lpo} u\sigma_g \Rightarrow SC(w_g)) \Rightarrow SC(t'\sigma_g)$$

for every $t'|H' \in \Gamma'$ and every ground substitution σ_g, then

$$(\forall u \in H.\ \forall w_g \in T_0.\ w_g <_{lpo} u\sigma_g \Rightarrow SC(w_g)) \Rightarrow SC(t\sigma_g)$$

for every $t|H \in \Gamma$ and every ground substitution σ_g. The proof proceeds by case analysis depending on the rule used in the derivation step $\Gamma \rightsquigarrow_S \Gamma'$. Here we only consider the cases of *Expand* and *Delete*. The other cases are proved in the same way as those of Lemma 11.

(*Expand*) Then $\Gamma = \Sigma \cup \{t'|H'\}$ and $\Gamma' = \Sigma \cup \{t'\sigma_i|H'\sigma_i\}_i$, where $x \in \mathcal{V}(t')$, $\vec{x}$ is a sequence of fresh variables, and $\{\sigma_i\}_i = \{\{x \mapsto f(\vec{x})\} \mid f \in \mathcal{C} \cup \mathcal{F}_1,\ sort(x) = sort(f(\vec{x}))\}$. The case $t|H \in \Sigma$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = t'|H'$. Let σ_g be a ground substitution such that (β): $\forall u \in H.\ \forall w_g \in T_0.\ w_g <_{lpo} u\sigma_g \Rightarrow SC(w_g)$. Our aim is to show $SC(t\sigma_g)$. For this, we distinguish two cases.

1. Suppose that there exists an index i such that $t\sigma_g = (t\sigma_i)\sigma'_g$ for some σ'_g. This case is proved similarly to the same case of the proof of Lemma 11.
2. Otherwise. Then we have $t\sigma_g = (t\theta)\sigma'_g$ for some $\theta = \{x \mapsto f(\vec{y})\}$ with $f \in \mathcal{F}_0 \cap \mathcal{D}$. This case is proved similarly to the case 2 of the proof of Lemma 11.

(*Delete*) Then $\Gamma = \Gamma' \cup \{t'|H'\}$, where $t' <_{lpo} u$ for some $u \in H'$. The case $t|H \in \Gamma'$ follows directly from the hypothesis for Γ'. Thus, it remains to show the case $t|H = t'|H'$. Let σ_g be a ground substitution such that $\forall u \in H.\ \forall w_g \in T_0.\ w_g <_{lpo} u\sigma_g \Rightarrow SC(w_g)$. Since $t = t' <_{lpo} u$ for some $u \in H' = H$, we have $t\sigma_g <_{lpo} u\sigma_g$ for some $u \in H$, and by $SrtP(\Gamma)$, we have $t\sigma_g \in T_0$. Hence $SC(t\sigma_g)$ holds. ◀

We are now ready to show the theorem on local sufficient completeness on T_0.

▶ **Theorem 34.** *Let $\mathcal{R}$ be a TRS with a signature satisfying the conditions S1–S4 of Definition 26, and let $>_{lpo}$ be a lexicographic path order induced by some precedence $>$ on $\mathcal{F}$ such that $f > g$ for every $f \in \mathcal{D}$ and $g \in \mathcal{C}$. If $\{h(\vec{x})|\{h(\vec{x})\}\} \rightsquigarrow_S^* \{\,\}$ for every $h \in \mathcal{F}_0$, then $\mathcal{R}$ is locally sufficiently complete on T_0.*

Proof. Let $\{h(\vec{x})|\{h(\vec{x})\}\} \rightsquigarrow_S^* \{\,\}$ for every $h \in \mathcal{F}_0$. We prove $SC(h(\vec{x})\theta_g)$ for every $h \in \mathcal{F}_0$ and every ground instance $h(\vec{x})\theta_g$ by induction on $T(\mathcal{F})$ with respect to the order $>_{lpo}$. Let $h \in \mathcal{F}_0$ and consider the derivation $\{h(\vec{x})|\{h(\vec{x})\}\} \rightsquigarrow_S^* \{\,\}$. Then by Lemmas 9 and 31, $VP(\Gamma)$ and $SrtP(\Gamma)$ hold for every Γ appearing in the derivation. Since $WIS_S(\{\,\})$ vacuously holds, we have by Lemma 33 $WIS_S(\{h(\vec{x})|\{h(\vec{x})\}\})$, i.e.,

$$(\forall w_g \in T_0.\ w_g <_{lpo} h(\vec{x})\theta_g \Rightarrow SC(w_g)) \Rightarrow SC(h(\vec{x})\theta_g) \tag{WIS 5}$$

for each ground instance $h(\vec{x})\theta_g$. For every $w_g \in T_0$, we have $w_g = f(\vec{y})\rho_g$ for some $f \in \mathcal{F}_0$ and some ground substitution ρ_g. Hence, for every $w_g \in T_0$, $w_g <_{lpo} h(\vec{x})\theta_g$ implies $SC(w_g)$ by the induction hypothesis. Thus, by (WIS 5), we obtain $SC(h(\vec{x})\theta_g)$. ◀

$$
\begin{array}{ll}
 & \{\ \mathsf{sum}(x_1,x_2)|\{\mathsf{sum}(x_1,x_2)\}\ \} \\
\rightsquigarrow_{s\,Expand} & \{\ \mathsf{sum}(0,x_2)|\{\mathsf{sum}(0,x_2)\},\ \mathsf{sum}(\mathsf{s}(x_3),x_2)|\{\mathsf{sum}(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{s\,Simplify} & \{\ 0|\{\mathsf{sum}(0,x_2)\},\ \mathsf{sum}(\mathsf{s}(x_3),x_2)|\{\mathsf{sum}(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{s\,Decompose} & \{\ \mathsf{sum}(\mathsf{s}(x_3),x_2)|\{\mathsf{sum}(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{s\,Expand} & \left\{\begin{array}{l}\mathsf{sum}(\mathsf{s}(x_3),[])|\{\mathsf{sum}(\mathsf{s}(x_3),[])\},\\ \mathsf{sum}(\mathsf{s}(x_3),x_4::x_5)|\{\mathsf{sum}(\mathsf{s}(x_3),x_4::x_5)\},\\ \mathsf{sum}(\mathsf{s}(x_3),\mathsf{from}(x_6))|\{\mathsf{sum}(\mathsf{s}(x_3),\mathsf{from}(x_6))\}\end{array}\right\} \\
\overset{*}{\rightsquigarrow}_{s\,Simplify} & \left\{\begin{array}{l}0|\{\mathsf{sum}(\mathsf{s}(x_3),[])\},\\ +(x_4,\mathsf{sum}(x_3,x_5))|\{\mathsf{sum}(\mathsf{s}(x_3),x_4::x_5)\},\\ +(x_6,\mathsf{sum}(x_3,\mathsf{from}(\mathsf{s}(x_6))))|\{\mathsf{sum}(\mathsf{s}(x_3),\mathsf{from}(x_6))\}\end{array}\right\} \\
\rightsquigarrow_{s\,Decompose} & \left\{\begin{array}{l}+(x_4,\mathsf{sum}(x_3,x_5))|\{\mathsf{sum}(\mathsf{s}(x_3),x_4::x_5)\},\\ +(x_6,\mathsf{sum}(x_3,\mathsf{from}(\mathsf{s}(x_6))))|\{\mathsf{sum}(\mathsf{s}(x_3),\mathsf{from}(x_6))\}\end{array}\right\} \\
\overset{*}{\rightsquigarrow}_{s\,Delete} & \{\ \ \} \\
 & \\
 & \{\ +(x_1,x_2)|\{+(x_1,x_2)\}\ \} \\
\rightsquigarrow_{s\,Expand} & \{\ +(0,x_2)|\{+(0,x_2)\},\ +(\mathsf{s}(x_3),x_2)|\{+(\mathsf{s}(x_3),x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{s\,Simplify} & \{\ x_2|\{+(0,x_2)\},\ \mathsf{s}(+(x_3,x_2))|\{+(\mathsf{s}(x_3),x_2)\}\ \} \\
\rightsquigarrow_{s\,Decompose} & \{\ x_2|\{+(0,x_2)\},\ +(x_3,x_2)|\{+(\mathsf{s}(x_3),x_2)\}\ \} \\
\overset{*}{\rightsquigarrow}_{s\,Delete} & \{\ \ \}
\end{array}
$$

Figure 5 Derivations for proving local sufficient completeness of $\mathcal{R}_5$ in Example 28.

Now we apply the theorem to the TRS of Example 28.

▶ **Example 35.** Consider the TRS $\mathcal{R}_5$ of Example 28. We show that $\mathcal{R}_5$ is locally sufficiently complete on T_0, using Theorem 34. It is easily seen that the signature of $\mathcal{R}_5$ satisfies the conditions S1–S4 of Definition 26. Let $>_{lpo}$ be the lexicographic path order induced by the precedence $>$ such that $\mathsf{sum} > + > \mathsf{from} > 0, \mathsf{s}, ::, []$. In Figure 5, we give derivations of $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow}_S \{\ \}$ for $h \in \mathcal{F}_0 \cap \mathcal{D}$. Also, using *Decompose* and *Delete*, we have derivations $\{h(\vec{x})|\{h(\vec{x})\}\} \overset{*}{\rightsquigarrow}_S \{\ \}$ for $h \in \mathcal{F}_0 \cap \mathcal{C}$. Thus, by Theorem 34, we conclude that $\mathcal{R}_5$ is locally sufficiently complete on T_0. ◀

6 Conclusion

We have presented simple derivation systems for proving sufficient completeness and two types of local sufficient completeness. We have given transparent correctness proofs of the derivation systems by introducing some suitable notions like well-founded induction schema. This is in contrast to the approach and correctness proofs of [3, 4] which are involved. The transparency allows us to provide derivation systems that deal with global and (two types of) local sufficient completeness in a uniform manner. Our proof methods using the derivation systems have been illustrated by applying them to some non-terminating TRSs.

The methods of [10] and our new methods are orthogonal in the following sense. In our methods, a group of function symbols are simultaneously tested, while an individual term pattern is tested in [10]. Then local sufficient completeness with signature restriction discussed in Section 4 cannot be inspected by the methods of [10], since one cannot substitute for a variable each term on a restricted signature (one can only substitute each term of the same sort as the variable; for problems in Section 5, it might be possible that the proof abilities of the two approaches are equivalent). On the other hand, any linear term pattern that is not necessarily of the form $f(x_1, \ldots, x_n)$ can be tested in [10]. This is not possible by the methods in the present paper.

The methods of [15] and our methods are also orthogonal. Example 28 of the present paper (i.e. Example 8 of [15]) is one of the examples that cannot be handled by the methods of [15]. On the other hand, the running example of [15] is difficult to handle by the current derivation systems (in [10] and the present paper), since they cannot help inducing a failing derivation with a divergent sequence as seen in Figure 4 of [15].

Găină et al. [5] have recently discussed a kind of local sufficient completeness with sort partition, where the set $\mathcal{S}_0$ in our terminology (Definition 26) contains every sort that is the codomain of some constructor. Earlier works [8, 13] discussed sufficient completeness relative to a set of constructors, where non-terminating systems are transformed into terminating ones using replacement restrictions of context-sensitive rewriting [12]. Detailed comparisons between these approaches and ours are left as future work.

In the present paper, we employed lexicographic path orders to define the derivation systems, but other simplification orders can be used if necessary. More generally, there would be a way to give abstract conditions on orders and generate constraints for derivations to be successful.

As a direction of further work, it is interesting to integrate the methods proposed in this paper and those in the previous work [10, 15] into a unified framework for proving various kinds of local sufficient completeness. Implementation of the methods and experiments to examine to what extent they work are also left as future work.

References

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

2 H. Comon and F. Jacquemard. Ground reducibility is EXPTIME-complete. *Inf. Comput.*, 187(1):123–153, 2003. doi:10.1016/S0890-5401(03)00134-2.

3 I. Gnaedig and H. Kirchner. Computing constructor forms with non terminating rewrite programs. In *Proceedings of the 8th PPDP*, pages 121–132. ACM, 2006. doi:10.1145/1140335.1140351.

4 I. Gnaedig and H. Kirchner. Proving weak properties of rewriting. *Theor. Comput. Sci.*, 412(34):4405–4438, 2011. doi:10.1016/j.tcs.2011.04.028.

5 D. Găină, M. Nakamura, K. Ogata, and K. Futatsugi. Stability of termination and sufficient-completeness under pushouts via amalgamation. *Theor. Comput. Sci.*, 848:82–105, 2020. doi:10.1016/j.tcs.2020.09.024.

6 J. V. Guttag. *The Specification and Application to Programming of Abstract Data Types*. PhD thesis, University of Toronto, 1975.

7 J. V. Guttag and J. J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10(1):27–52, 1978. doi:10.1007/BF00260922.

8 J. Hendrix and J. Meseguer. On the completeness of context-sensitive order-sorted specifications. In *Proceedings of the 18th RTA*, volume 4533 of *Lecture Notes in Computer Science*, pages 229–245. Springer, 2007. doi:10.1007/978-3-540-73449-9_18.

9 D. Kapur, P. Narendran, and H. Zhang. On sufficient-completeness and related properties of term rewriting systems. *Acta Informatica*, 24(4):395–415, 1987. doi:10.1007/BF00292110.

10 K. Kikuchi, T. Aoto, and I. Sasano. Inductive theorem proving in non-terminating rewriting systems and its application to program transformation. In *Proceedings of the 21st PPDP*, pages 13:1–13:14. ACM, 2019. doi:10.1145/3354166.3354178.

11 A. Lazrek, P. Lescanne, and J. J. Thiel. Tools for proving inductive equalities, relative completeness, and ω-completeness. *Inf. Comput.*, 84(1):47–70, 1990. doi:10.1016/0890-5401(90)90033-E.

12 S. Lucas. Context-sensitive computations in functional and functional logic programs. *J. Funct. Log. Program.*, 1998(1), 1998. URL: http://danae.uni-muenster.de/lehre/kuchen/JFLP/articles/1998/A98-01/A98-01.html.

13 S. Lucas. Completeness of context-sensitive rewriting. *Inf. Process. Lett.*, 115(2):87–92, 2015. doi:10.1016/j.ipl.2014.07.004.

14 E. Ohlebusch. *Advanced Topics in Term Rewriting*. Springer, 2002.

15 T. Shiraishi, K. Kikuchi, and T. Aoto. A proof method for local sufficient completeness of term rewriting systems. In *Proceedings of the 18th ICTAC*, volume 12819 of *Lecture Notes in Computer Science*, pages 386–404. Springer, 2021. doi:10.1007/978-3-030-85315-0_22.

16 Terese. *Term Rewriting Systems*. Cambridge University Press, 2003.

17 Y. Toyama. How to prove equivalence of term rewriting systems without induction. *Theor. Comput. Sci.*, 90(2):369–390, 1991.

A Proof Ability of the Method in Section 3

In this section, we compare the proof abilities of the method in [3, 4] and our method by the derivation system in Section 3.

The next example is a modification of a TRS described in [3, page 125 (the right column)]. ([3] discusses TRSs with possibly non-free constructors whereas we discuss TRSs with free constructors in this paper.)

▶ **Example 36.** Consider a signature with $\mathcal{S} = \{A\}$ and

$$\mathcal{F} = \{\ \mathsf{f} : A \to A,\quad \mathsf{a} : A,\quad \mathsf{b} : A,\quad \mathsf{c} : A\ \}$$

where $\mathcal{C} = \{\mathsf{c} : A\}$. Let $\mathcal{R}_6$ be the following TRS:

$$\mathcal{R}_6 = \left\{ \begin{array}{lllll} (1) & \mathsf{a} & \to & \mathsf{b} \\ (2) & \mathsf{f}(\mathsf{f}(\mathsf{b})) & \to & \mathsf{c} \\ (3) & \mathsf{b} & \to & \mathsf{f}(\mathsf{b}) \\ (4) & \mathsf{f}(\mathsf{c}) & \to & \mathsf{c} \end{array} \right\}.$$

Then, using Theorem 12, we can show that $\mathcal{R}_6$ is sufficiently complete. Required derivations are, e.g., $\{\mathsf{b}|\{\mathsf{b}\}\} \rightsquigarrow_{\mathit{Simplify}} \{\mathsf{f}(\mathsf{b})|\{\mathsf{b}\}\} \rightsquigarrow_{\mathit{Simplify}} \{\mathsf{f}(\mathsf{f}(\mathsf{b}))|\{\mathsf{b}\}\} \rightsquigarrow_{\mathit{Simplify}} \{\mathsf{c}|\{\mathsf{b}\}\} \rightsquigarrow_{\mathit{Decompose}} \{\,\}$. However, as remarked in [3, page 125 (the right column)], these *Simplify* steps are not represented by the abstract-narrow-based process of [3, 4]. ◀