Termination of Context-Sensitive Rewriting with Built-In Numbers and Collection Data Structures*

Stephan Falke and Deepak Kapur

Department of Computer Science
University of New Mexico
Albuquerque, NM 87131, USA
{spf, kapur}@cs.unm.edu

Abstract. Context-sensitive rewriting is a restriction of rewriting that can be used to elegantly model declarative specification and programming languages such as `Maude`. Furthermore, it can be used to model lazy evaluation in functional languages such as `Haskell`. Building upon our previous work on an expressive and elegant class of rewrite systems (called CERSs) that contains built-in numbers and supports the use of collection data structures such as sets or multisets, we consider context-sensitive rewriting with CERSs in this paper. This integration results in a natural way for specifying algorithms in the rewriting framework. In order to prove termination of this kind of rewriting automatically, we develop a dependency pair framework for context-sensitive rewriting with CERSs, resulting in a flexible termination method that can be automated effectively. Several powerful termination techniques are developed within this framework. An implementation in the termination prover AProVE has been successfully evaluated on a large collection of examples.

1 Introduction

While ordinary term rewrite systems (TRSs) allow to model algorithms in a functional programming style, there still remain serious drawbacks. First, modeling data structures such as sets or multisets is not easily possible since these non-free data structures typically cause non-termination of the rewrite relation. Equally severe, domain-specific knowledge about primitive data types such as natural numbers or integers is not directly available. We have shown in [13] that constrained equational rewrite systems (CERSs) provide an expressive and convenient tool for modeling algorithms that solves both of these drawbacks. Since [13] considers only natural numbers as a primitive built-in data type, the first contribution of this paper is a reformulation of the ideas from [13] that allows for both built-in natural numbers or built-in integers in one convenient framework.

Even though CERSs are an expressive and elegant tool for modeling algorithms, they do not allow to incorporate reduction strategies that are commonly

* Partially supported by NSF grants CCF-0541315 and CNS-0831462.

used in declarative specification and programming languages such as `Maude` [8]. Context-sensitive rewriting [22, 23] has been introduced as an operational restriction of term rewriting that can be used to model such reduction strategies. Additionally, context-sensitive rewriting allows to model lazy (non-strict) evaluation as used in functional programming languages such as `Haskell`. The second contribution of this paper is to introduce context-sensitive rewriting for CERSs, thus combining the expressiveness of CERSs with increased flexibility on the reduction strategy provided by context-sensitive rewriting.

In context-sensitive rewriting, the arguments where an evaluation may take place are specified for each function symbol. Then a reduction is only allowed if it takes place at a position that is not forbidden by a function symbol occurring somewhere above it.

Example 1. Consider the following rewrite rules, where ins is used to construct a set by adding a further element to a set:

$$\begin{aligned}
\mathsf{from}(x) &\to \mathsf{ins}(x, \mathsf{from}(x + \mathsf{1})) \\
\mathsf{take}(\mathsf{0}, xs) &\to \mathsf{nil} \\
\mathsf{take}(x, \mathsf{ins}(y, ys)) &\to \mathsf{cons}(y, \mathsf{take}(x - \mathsf{1}, ys))[\![x > \mathsf{0}]\!] \\
\mathsf{head}(\mathsf{ins}(x, xs)) &\to x \\
\mathsf{tail}(\mathsf{ins}(x, xs)) &\to xs
\end{aligned}$$

Here, the function symbol from is used to generate the (infinite) subsets of integers that are greater than or equal to the argument of from. The meaning of "$[\![x > \mathsf{0}]\!]$" in the second take-rule will be made precise in Section 3. Intuitively, it allows application of that rule only if the instantiation of the variable x is a positive number. The term $\mathsf{take}(\mathsf{2}, \mathsf{from}(\mathsf{0}))$ admits an infinite reduction in which the from-rule is applied again and again. However, there also is a finite reduction of that term which results in the normal form $\mathsf{cons}(\mathsf{0}, \mathsf{cons}(\mathsf{1}, \mathsf{nil}))$. This reduction can be enforced using context-sensitive rewriting if evaluation of the second argument of ins is forbidden since the recursive call to from is then blocked. ◊

As for ordinary rewriting, termination is a fundamental property of context-sensitive rewriting. Since context-sensitive rewriting may result in a terminating rewrite relation where regular rewriting is diverging, proving termination of context-sensitive rewriting is quite challenging.

For ordinary TRSs, there are two approaches to proving termination of context-sensitive rewriting. The first approach is to apply a syntactic transformation in such a way that termination of context-sensitive rewriting with a TRS is implied by (regular) termination of the TRS obtained by the transformation. For details on this approach, see [15, 25]. While the application of these transformations allows the use of any method for proving termination of the transformed TRS, they often generate TRSs whose termination is hard to establish.

The second approach consists of the development of dedicated methods for proving termination of context-sensitive rewriting. Examples for adaptations of classical methods are context-sensitive recursive path orderings [7] and context-sensitive polynomial interpretations [24]. The main drawback of these adaptations is the limited power which is inherited from the classical methods. Adapting the more powerful dependency pair method [4] to context-sensitive TRSs has proven to be quite challenging. A first adaptation of the dependency pair method to context-sensitive TRSs has been presented in [2]. But this adaptation has severe disadvantages compared to the ordinary dependency pair method since dependency pairs may be collapsing, which requires strong restrictions on how the method can be applied.

An alternative adaptation of the dependency pair method to context-sensitive TRSs has recently been presented in [1]. This adaptation does not require collapsing dependency pairs and makes it much easier to adapt techniques developed within the ordinary dependency pair method to the context-sensitive case.

The third and main contribution of this paper the development of a dependency pair method for context-sensitive rewriting with CERSs that takes [1] as a starting point. This adaptation is non-trivial since [1] is concerned with ordinary (syntactic) rewriting, whereas rewriting with CERSs is based on normalized equational rewriting that uses constructor equations and constructor rules.

After fixing some terminology in Section 2, Section 3 recalls and extends the CERSs introduced in [13]. In contrast to [13], it is now possible to consider both built-in natural numbers or built-in integers. The basic terminology of context-sensitive rewriting with CERSs is introduced in Section 4, and Section 5 presents the main technical result of this paper. By a non-trivial extension of [1], termination of context-sensitive rewriting with a CERS is reduced to showing absence of infinite chains of dependency pairs. Section 6 introduces several powerful termination techniques that can be applied in combination with dependency pairs. These techniques lift the most commonly used termination techniques introduced for CERSs in [13] to context-sensitive CERSs. An implementation of these techniques in the termination prover AProVE [16] is discussed and evaluated in Section 7. This evaluation shows that our implementation succeeds in proving termination of a large class of context-sensitive CERSs.

2 Preliminaries

Familiarity with the notation and terminology of term rewriting is assumed, see [5] for an in-depth treatment. This paper uses many-sorted term rewriting over a set S of sorts. It is assumed in the following that all terms, substitutions, replacements, etc. are sort-correct.

For a signature $\mathcal{F}$ and a disjoint set $\mathcal{V}$ of variables, the set of all terms over $\mathcal{F}$ and $\mathcal{V}$ is denoted by $\mathcal{T}(\mathcal{F}, \mathcal{V})$. The set of positions of a term t is denoted by $\mathcal{P}os(t)$, where Λ denotes the root position. $|t|$ is the *size* of the term t, defined by $|t| = |\mathcal{P}os(t)|$. The set of variables occurring in a term t is denoted by $\mathcal{V}(t)$. Similarly, the set of function symbols occurring in t is denoted by $\mathcal{F}(t)$. This naturally extends to pairs of terms, sets of terms, etc. The root symbol of a term t is denoted by $\mathrm{root}(t)$. A term t is *linear* if each $v \in \mathcal{V}(t)$ occurs only once in t.

A *context* over $\mathcal{F}$ is a term $C \in \mathcal{T}(\mathcal{F} \cup \bigcup_{s \in S}\{\Box_s\}, \mathcal{V})$. Here, $\Box_s : \rightarrow s$ is a fresh constant symbol of sort s, called *hole*. If the sort of a hole can be derived or is not important, then $\Box$ will be used to stand for any of the $\Box_s$. If C is a context with n holes and $t_1, \ldots, t_n$ are terms of the appropriate sorts, then $C[t_1, \ldots, t_n]$ is the result of replacing the occurrences of holes by $t_1, \ldots, t_n$ "from left to right".

A *substitution* is a mapping from variables to terms, where the domain of the substitution may be infinite. The application of a substitution σ to a term t is written as $t\sigma$, using postfix notation.

A finite set $\mathcal{E} = \{u_1 \approx v_1, \ldots, u_n \approx v_n\}$ of equations induces a rewrite relation $\rightarrow_{\mathcal{E}}$ by letting $s \rightarrow_{\mathcal{E}} t$ iff there exist a position $p \in \mathcal{P}os(s)$ and a substitution σ such that $s|_p = u_i\sigma$ and $t = s[v_i\sigma]_p$ for some $u_i \approx v_i \in \mathcal{E}$. The symmetric and reflexive-transitive-symmetric closures of $\rightarrow_{\mathcal{E}}$ are denoted by $\vdash\dashv_{\mathcal{E}}$ and $\sim_{\mathcal{E}}$, respectively. Since $\sim_{\mathcal{E}}$ is a congruence relation, it defines a partition of $\mathcal{T}(\mathcal{F}, \mathcal{V})$ into equivalence classes. The members of $\mathcal{T}(\mathcal{F}, \mathcal{V})/_{\sim_{\mathcal{E}}}$ are called *$\mathcal{E}$-equivalence classes*.

The following syntactic properties of equations are relevant for this paper.

Definition 2 (Properties of Equations). *Let $u \approx v$ be an equation.*

1. *$u \approx v$ is* collapse-free *if neither u nor v is a variable.*
2. *$u \approx v$ is* regular *if $\mathcal{V}(u) = \mathcal{V}(v)$.*
3. *$u \approx v$ has* identical unique variables (is i.u.v.) *if it is regular and both u and v are linear.*
4. *$u \approx v$ is* size-preserving *if it is i.u.v. and $|u| = |v|$.*

A set $\mathcal{E}$ of equations is said to have one of these properties if all equations $u \approx v \in \mathcal{E}$ satisfy that property.

If equations are to be used in only one direction, they are called *rules*. A *term rewrite system (TRS)* is a finite set $\mathcal{R} = \{l_1 \rightarrow r_1, \ldots, l_m \rightarrow r_m\}$ of rules.

Equational rewriting uses both a set $\mathcal{E}$ of equations and a set $\mathcal{R}$ of rules. Intuitively, $\mathcal{E}$ is used to model "structural" properties, while $\mathcal{R}$ is used to model "simplifying" properties.

Definition 3 ($\mathcal{E}$-Extended Rewriting). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of equations. Then $s \rightarrow_{\mathcal{E}\backslash\mathcal{R}} t$ if there exist a rule $l \rightarrow r \in \mathcal{R}$, a position $p \in \mathcal{P}os(s)$, and a substitution σ such that*

1. *$s|_p \sim_{\mathcal{E}} l\sigma$, and*
2. *$t = s[r\sigma]_p$.*

Writing $\overset{>\Lambda}{\sim}_{\mathcal{E}}$ and $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{R}}$ denotes that all steps are applied below the root position.

Given $\mathcal{E}$ and $\mathcal{R}$, an investigation of the normal forms of $\rightarrow_{\mathcal{E}\backslash\mathcal{R}}$ becomes important. For this, strong results can be obtained if $\rightarrow_{\mathcal{E}\backslash\mathcal{R}}$ satisfies the following properties.

Definition 4 ($\mathcal{E}$-Confluence, $\mathcal{E}$-Convergence, Strong $\mathcal{E}$-Coherence). *Let $\mathcal{R}$ be a TRS and let $\mathcal{E}$ be a set of equations.*

1. $\to_{\mathcal{E}\backslash\mathcal{R}}$ *is* $\mathcal{E}$-confluent *iff* $\leftarrow^*_{\mathcal{E}\backslash\mathcal{R}} \circ \to^*_{\mathcal{E}\backslash\mathcal{R}} \subseteq \to^*_{\mathcal{E}\backslash\mathcal{R}} \circ \sim_{\mathcal{E}} \circ \leftarrow^*_{\mathcal{E}\backslash\mathcal{R}}$.
2. $\to_{\mathcal{E}\backslash\mathcal{R}}$ *is* $\mathcal{E}$-convergent *iff it is terminating and $\mathcal{E}$-confluent.*
3. $\to_{\mathcal{E}\backslash\mathcal{R}}$ *is* strongly $\mathcal{E}$-coherent *iff* $\sim_{\mathcal{E}} \circ \to_{\mathcal{E}\backslash\mathcal{R}} \subseteq \to_{\mathcal{E}\backslash\mathcal{R}} \circ \sim_{\mathcal{E}}$.

Notice that strong $\mathcal{E}$-coherence is a special case of coherence modulo $\mathcal{E}$ [20].

3 Constrained Equational Rewrite Systems

This section recalls and extends the class of rewrite systems with collection data structures and built-in numbers from [13]. Both built-in numbers and collection data structures are modeled using $\mathcal{E}$-extended rewriting as in Definition 3. It is first discussed how this is done for built-in numbers. The approach taken for them easily allows for collection data structures as well.

3.1 Built-in Numbers as Canonizable Theories

In order to model the set of integers using terms and equations, recall that $\mathbb{Z}$ is an Abelian group under addition that is generated using the element 1. A signature for modeling integers thus consists of the following function symbols:

$$\begin{aligned} \mathsf{0} &: \to \texttt{int} \\ \mathsf{1} &: \to \texttt{int} \\ - &: \texttt{int} \to \texttt{int} \\ + &: \texttt{int}, \texttt{int} \to \texttt{int} \end{aligned}$$

The defining properties of an Abelian group can easily be stated as equations:

$$\begin{array}{cc} x + y \approx y + x & x + \mathsf{0} \approx x \\ x + (y + z) \approx (x + y) + z & x + (-x) \approx \mathsf{0} \end{array}$$

This set of equations is in general unsuitable for modeling algorithms using extended rewriting since the last equation is not regular. To avoid this problem, the idea is to only keep associativity and commutativity as equations and to turn the remaining properties (idempotency and inverse elements) into rewrite rules. For this, it does not suffice to just orient the last two equations as rules. Instead, the well-known method of *equational completion* [20, 6] needs to be applied to obtain "equivalent" sets $\mathcal{S}$ of rewrite rules and $\mathcal{E}$ of equations. Here, the goal is that two terms are in the same equivalence class of the initial set of equations if and only if their normal forms w.r.t. $\to_{\mathcal{E}\backslash\mathcal{S}}$ are in the same $\mathcal{E}$-equivalence class. Applying equational completion to the above properties, the following rewrite rules and equations are obtained:

$$\begin{array}{lr} \mathcal{S}: & x + \mathsf{0} \to x \\ & - - x \to x \\ & -\mathsf{0} \to \mathsf{0} \end{array}$$

$$\begin{aligned} -(x+y) &\to (-x)+(-y) \\ x+(-x) &\to \mathsf{0} \\ (x+(-x))+y &\to \mathsf{0}+y \\ \mathcal{E}: \qquad x+y &\approx y+x \\ x+(y+z) &\approx (x+y)+z \end{aligned}$$

This idea can be generalized from the integers, and it can furthermore be made more useful by allowing predefined predicate symbols that can then be used as constraints in the rewrite rules used for modeling algorithms.

In order to model built-in numbers and other theories, an initial signature $\mathcal{F}_{\mathcal{T}h}$ over a sort `base` containing the function symbols of the built-in theory $\mathcal{T}h$ is used. It is assumed that $\mathcal{T}h$ is equipped with a set $\mathsf{P}_{\mathcal{T}h}$ of predicate symbols. In the following, $[P]_{\mathcal{T}h}$ denotes the interpretation of $P \in \mathsf{P}_{\mathcal{T}h}$ in the theory $\mathcal{T}h$, and $[s]_{\mathcal{T}h}$ denotes the interpretation of $s \in \mathcal{T}(\mathcal{F}_{\mathcal{T}h})$ in $\mathcal{T}h$. Furthermore, it is assumed that $\simeq \in \mathsf{P}_{\mathcal{T}h}$ is the equality predicate of $\mathcal{T}h$ and that $\simeq$ is specified using the set $\mathcal{E}_{\mathcal{T}h}$ of equations. Within this paper, attention is restricted to the case where $\mathcal{E}_{\mathcal{T}h}$ is *canonizable* in the following sense.

Definition 5 (Canonizable Sets of Equations). *A set $\mathcal{E}$ of equations is* canonizable *iff there exist a set $\overrightarrow{\mathcal{E}}$ of rewrite rules and a set $\widehat{\mathcal{E}}$ of equations such that*

1. *$\overrightarrow{\mathcal{E}} \cup \widehat{\mathcal{E}}$ is equivalent to $\mathcal{E}$, i.e., $\sim_{\mathcal{E}} = \sim_{\overrightarrow{\mathcal{E}} \cup \widehat{\mathcal{E}}}$,*
2. *$\widehat{\mathcal{E}}$ is i.u.v.,*
3. *$\to_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}}$ is $\widehat{\mathcal{E}}$-convergent, and*
4. *$\to_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}}$ is strongly $\widehat{\mathcal{E}}$-coherent.*

For a canonizable set $\mathcal{E}$ of equations, $s \sim_{\mathcal{E}} t$ if and only if $s \to^!_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}} \circ \sim_{\widehat{\mathcal{E}}} \circ \leftarrow^!_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}} t$ since *3.* and *4.* imply that $\to_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}}$ is Church-Rosser modulo $\widehat{\mathcal{E}}$ [20]. The sets $\overrightarrow{\mathcal{E}}$ and $\widehat{\mathcal{E}}$ can be obtained from $\mathcal{E}$ using *equational completion* [20, 6]. A theory $\mathcal{T}h$ is said to be canonizable if $\mathcal{E}_{\mathcal{T}h}$ is. Figure 1 lists two of the most important canonizable theories: $\mathcal{T}h_{\mathbb{N}}$ denotes Presburger arithmetic of the natural numbers, and $\mathcal{T}h_{\mathbb{Z}}$ denotes Presburger arithmetic of the integers. For both of these theories, $\mathsf{P}_{\mathcal{T}h} = \{\simeq, \geq, >\}$. For $\mathcal{T}h_{\mathbb{Z}}$, the rewrite rule $(x+(-x))+y \to \mathsf{0}+y$ is an *extension* of $x+(-x) \to \mathsf{0}$ that is needed to make $\to_{\widehat{\mathcal{E}} \backslash \overrightarrow{\mathcal{E}}}$ strongly $\widehat{\mathcal{E}}$-coherent. This paper takes some liberties in writing terms for $\mathcal{T}h_{\mathbb{N}}$ and $\mathcal{T}h_{\mathbb{Z}}$, e.g., $x-2$ is shorthand for any term that is equivalent to $x+((-1)+(-1))$.

The rewrite rules that are used in order to specify defined functions will have constraints from $\mathcal{T}h$ that guard when a rewrite step may be performed.

Definition 6 (Syntax of $\mathcal{T}h$-constraints). *An* atomic $\mathcal{T}h$-constraint *has the form $Pt_1 \ldots t_n$ for a predicate symbol $P \in \mathsf{P}_{\mathcal{T}h}$ and terms $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F}_{\mathcal{T}h}, \mathcal{V})$. The set of* $\mathcal{T}h$-constraints *is inductively defined as follows:*

	$\mathcal{F}_{\mathcal{T}h}$	$\overrightarrow{\mathcal{E}}_{\mathcal{T}h}$	$\widehat{\mathcal{E}}_{\mathcal{T}h}$
Natural numbers	$\mathsf{0}, \mathsf{1}, +$	$x + \mathsf{0} \to x$	$x + y \approx y + x$ $x + (y + z) \approx (x + y) + z$
Integers	$\mathsf{0}, \mathsf{1}, +, -$	$x + \mathsf{0} \to x$ $- - x \to x$ $-\mathsf{0} \to \mathsf{0}$ $-(x + y) \to (-x) + (-y)$ $x + (-x) \to \mathsf{0}$ $(x + (-x)) + y \to \mathsf{0} + y$	$x + y \approx y + x$ $x + (y + z) \approx (x + y) + z$

Fig. 1. Numbers as canonizable theories.

1. *$\top$ is a $\mathcal{T}h$-constraint.*
2. *Every atomic $\mathcal{T}h$-constraint is a $\mathcal{T}h$-constraint.*
3. *If φ is a $\mathcal{T}h$-constraint, then $\neg\varphi$ is a $\mathcal{T}h$-constraint.*
4. *If φ_1, φ_2 are $\mathcal{T}h$-constraints, then $\varphi_1 \wedge \varphi_2$ is a $\mathcal{T}h$-constraint.*

The Boolean connectives $\vee$, $\Rightarrow$, and $\Leftrightarrow$ can be defined as usual. Also, $\mathcal{T}h$-constraints have the expected semantics. The main interest is in *satisfiability* (i.e., the constraint is true for some instantiation of its variables) and *validity* (i.e., the constraint is true for all instantiations of its variables).

Definition 7 (Semantics of $\mathcal{T}h$-constraints). *A variable-free $\mathcal{T}h$-constraint φ is $\mathcal{T}h$-*valid *iff*

1. *C has the form $\top$, or*
2. *φ has the form $Pt_1 \ldots t_n$ and $[P]_{\mathcal{T}h}[t_1]_{\mathcal{T}h} \ldots [t_n]_{\mathcal{T}h}$ is true, or*
3. *φ has the form $\neg\varphi'$ and φ' is not $\mathcal{T}h$-valid, or*
4. *φ has the form $\varphi_1 \wedge \varphi_2$ and both φ_1 and φ_2 are $\mathcal{T}h$-valid.*

*A $\mathcal{T}h$-constraint φ with variables is $\mathcal{T}h$-*valid *iff $\varphi\sigma$ is $\mathcal{T}h$-valid for all ground substitutions $\sigma : \mathcal{V}(\varphi) \to \mathcal{T}(\mathcal{F}_{\mathcal{T}h})$. A $\mathcal{T}h$-constraint φ is $\mathcal{T}h$-*satisfiable *iff there exists a ground substitution $\sigma : \mathcal{V}(\varphi) \to \mathcal{T}(\mathcal{F}_{\mathcal{T}h})$ such that $\varphi\sigma$ is $\mathcal{T}h$-valid. Otherwise, φ is $\mathcal{T}h$-*unsatisfiable.

It is assumed in the following that $\mathcal{T}h$-validity is decidable. This is the case for both $\mathcal{T}h_{\mathbb{N}}$ and $\mathcal{T}h_{\mathbb{Z}}$ [27].

3.2 Canonizable Collection Data Structures

In order to extend $\mathcal{T}h$ by collection data structures and defined functions, $\mathcal{F}_{\mathcal{T}h}$ is extended by a finite signature $\mathcal{F}$ over the sort `base` and a new sort `univ`. The restriction to two sorts is not essential, but the techniques that are presented in the remainder of this paper only differentiate between terms of sort `base` and terms of any other sort.

Collection data structures can be handled similarly to the built-in theories, i.e., properties of collection data structures are modeled using a set $\mathcal{E}_\mathcal{C}$ of equations. As in Section 3.1, attention is restricted to the case where $\mathcal{E}_\mathcal{C}$ is canonizable. Collection data structures that satisfy this property are also called *canonizable collection data structures*. Some commonly used canonizable collection data structures are given in Figure 2. *Compact lists* are lists where the number of contiguous occurrences of the same element does not matter. This kind of lists has been shown to be useful in the constraint logic programming framework [10, 9]. Notice that there are typically two possibilities for modeling the "same" collection data structure:

1. Using a list-like representation with an empty collection and a constructor to add an element to a collection, where, according to the intended semantics, the order in which the elements are added might not matter and duplicates might be removed.
2. Using an empty collection, singleton collections ($\langle\cdot\rangle$ and $\{\cdot\}$, respectively), and a constructor to concatenate two collections. Again, the order of concatenation might not matter and duplicates might be removed.

For the second possibility to model sets, the rewrite rule $(x \cup x) \cup y \to x \cup y$ is an extension of $x \cup x \to x$ that is needed to make $\to_{\widehat{\mathcal{E}_\mathcal{C}} \setminus \overrightarrow{\mathcal{E}_\mathcal{C}}}$ strongly $\widehat{\mathcal{E}_\mathcal{C}}$-coherent.

	Constructors	$\overrightarrow{\mathcal{E}_\mathcal{C}}$ and $\widehat{\mathcal{E}_\mathcal{C}}$
Lists	$\mathsf{nil}, \mathsf{cons}$	n/a
Lists	$\mathsf{nil}, \langle\cdot\rangle, +\!\!+$	$x +\!\!+ \mathsf{nil} \to x$ $\mathsf{nil} +\!\!+ y \to y$ $x +\!\!+ (y +\!\!+ z) \approx (x +\!\!+ y) +\!\!+ z$
Compact Lists	$\mathsf{nil}, \mathsf{cons}$	$\mathsf{cons}(x, \mathsf{cons}(x, ys)) \to \mathsf{cons}(x, ys)$
Compact Lists	$\mathsf{nil}, \langle\cdot\rangle, +\!\!+$	$x +\!\!+ \mathsf{nil} \to x$ $\mathsf{nil} +\!\!+ y \to y$ $\langle x\rangle +\!\!+ \langle x\rangle \to \langle x\rangle$ $x +\!\!+ (y +\!\!+ z) \approx (x +\!\!+ y) +\!\!+ z$
Multisets	$\emptyset, \mathsf{ins}$	$\mathsf{ins}(x, \mathsf{ins}(y, zs)) \approx \mathsf{ins}(y, \mathsf{ins}(x, zs))$
Multisets	$\emptyset, \{\cdot\}, \cup$	$x \cup \emptyset \to x$ $x \cup (y \cup z) \approx (x \cup y) \cup z$ $x \cup y \approx y \cup x$
Sets	$\emptyset, \mathsf{ins}$	$\mathsf{ins}(x, \mathsf{ins}(x, ys)) \to \mathsf{ins}(x, ys)$ $\mathsf{ins}(x, \mathsf{ins}(y, zs)) \approx \mathsf{ins}(y, \mathsf{ins}(x, zs))$
Sets	$\emptyset, \{\cdot\}, \cup$	$x \cup \emptyset \to x$ $x \cup x \to x$ $(x \cup x) \cup y \to x \cup y$ $x \cup (y \cup z) \approx (x \cup y) \cup z$ $x \cup y \approx y \cup x$

Fig. 2. Commonly used canonizable collection data structures.

3.3 CERSs

In the following, a combination of $\mathcal{T}h$ with none or more (signature-disjoint) canonizable collection data structures $\mathcal{C}_1, \ldots, \mathcal{C}_n$ is considered. For this, let $\mathcal{S} = \overrightarrow{\mathcal{E}}_{\mathcal{T}h} \cup \bigcup_{i=1}^{n} \overrightarrow{\mathcal{E}}_{\mathcal{C}_i}$ and $\mathcal{E} = \widehat{\mathcal{E}}_{\mathcal{T}h} \cup \bigcup_{i=1}^{n} \widehat{\mathcal{E}}_{\mathcal{C}_i}$.

The defined functions are specified using constrained rewrite rules. As formalized below in Definition 11, the $\mathcal{T}h$-constraint guards when a rewrite step may be performed.

Definition 8 (Constrained Rewrite Rules). *A* constrained rewrite rule *has the form $l \to r[\![\varphi]\!]$ for terms $l, r \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ and a $\mathcal{T}h$-constraint φ such that* $\mathrm{root}(l) \in \mathcal{F} - \mathcal{F}(\mathcal{E} \cup \mathcal{S})$ *and $\mathcal{V}(r) \subseteq \mathcal{V}(l)$. In a rule $l \to r[\![\top]\!]$, the constraint $\top$ will usually be omitted.*

A finite set of constrained rewrite rules and the sets $\mathcal{S}$ and $\mathcal{E}$ for modeling $\mathcal{T}h$ and collection data structures are combined into a *constrained equational rewrite system.*

Definition 9 (Constrained Equational Rewrite Systems (CERSs)). *A* constrained equational rewrite system (CERS) *has the form $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ for a finite set $\mathcal{R}$ of constrained rewrite rules, a finite set $\mathcal{E}$ of equations, and a finite set $\mathcal{S}$ of rewrite rules such that*

1. *$\mathcal{S}$ is right-linear, i.e., each variable occurs at most once in r for all $l \to r \in \mathcal{S}$,*
2. *$\to_{\mathcal{E}\backslash\mathcal{S}}$ is $\mathcal{E}$-convergent, and*
3. *$\to_{\mathcal{E}\backslash\mathcal{S}}$ is strongly $\mathcal{E}$-coherent.*

The rewrite relation obtained from a CERS is based on the key idea of [26]. First, the subterm where a rule from $\mathcal{R}$ should be applied is normalized with $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$. Then, $\mathcal{E}$-matching is performed, where it is additionally required that the $\mathcal{T}h$-constraint of the constrained rewrite rule is $\mathcal{T}h$-valid after being instantiated by the matcher. If the matcher instantiates all variables of sort `base` by terms from $\mathcal{T}(\mathcal{F}_{\mathcal{T}h}, \mathcal{V})$, this can be answered by a decision procedure for $\mathcal{T}h$-validity.

Definition 10 ($\mathcal{T}h$-based Substitutions). *A substitution σ is $\mathcal{T}h$-*based *iff $\sigma(x) \in \mathcal{T}(\mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ for all variables x of sort* `base`.

The rewrite relation is now restricted to use a $\mathcal{T}h$-based substitution. This restriction could be slightly relaxed by requiring $\sigma(x) \in \mathcal{T}(\mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ for only those variables of sort `base` that occur in the constraint of the rewrite rule.

Definition 11 (Rewrite Relation of a CERS). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ be a CERS. Then $s \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R}} t$ iff there exist a constrained rewrite rule $l \to r[\![\varphi]\!] \in \mathcal{R}$, a position $p \in \mathcal{P}os(s)$, and a $\mathcal{T}h$-based substitution σ such that*

1. *$s|_p \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$,*
2. *$\varphi\sigma$ is $\mathcal{T}h$-valid, and*
3. *$t = s[r\sigma]_p$.*

Example 12. This example shows a quicksort algorithm that takes a set of integers and returns a sorted list of the elements of that set. For this, integers are modeled as in Figure 1 and sets are modeled using $\emptyset$, $\{\cdot\}$, and $\cup$ as in Figure 2, i.e., the sets $\mathcal{S}$ and $\mathcal{E}$ are as follows:

$$\begin{array}{lrcl}
\mathcal{S}: & x + \mathsf{0} & \to & x \\
 & - - x & \to & x \\
 & -\mathsf{0} & \to & \mathsf{0} \\
 & -(x+y) & \to & (-x) + (-y) \\
 & x + (-x) & \to & \mathsf{0} \\
 & (x + (-x)) + y & \to & \mathsf{0} + y \\
 & x \cup \emptyset & \to & x \\
 & x \cup x & \to & x \\
 & (x \cup x) \cup y & \to & x \cup y \\
\mathcal{E}: & x + y & \approx & y + x \\
 & x + (y + z) & \approx & (x + y) + z \\
 & x \cup y & \approx & y \cup x \\
 & x \cup (y \cup z) & \approx & (x \cup y) \cup z
\end{array}$$

The quicksort algorithm is specified by the following constrained rewrite rules:

$$\begin{array}{rcl}
\mathsf{app}(\mathsf{nil}, zs) & \to & zs \\
\mathsf{app}(\mathsf{cons}(x, ys), zs) & \to & \mathsf{cons}(x, \mathsf{app}(ys, zs)) \\
\mathsf{low}(x, \emptyset) & \to & \emptyset \\
\mathsf{low}(x, \{y\}) & \to & \{y\} \; [\![x > y]\!] \\
\mathsf{low}(x, \{y\}) & \to & \emptyset \; [\![x \not> y]\!] \\
\mathsf{low}(x, y \cup z) & \to & \mathsf{low}(x, y) \cup \mathsf{low}(x, z) \\
\mathsf{high}(x, \emptyset) & \to & \emptyset \\
\mathsf{high}(x, \{y\}) & \to & \{y\} \; [\![x \not> y]\!] \\
\mathsf{high}(x, \{y\}) & \to & \emptyset \; [\![x > y]\!] \\
\mathsf{high}(x, y \cup z) & \to & \mathsf{high}(x, y) \cup \mathsf{high}(x, z) \\
\mathsf{qsort}(\emptyset) & \to & \mathsf{nil} \\
\mathsf{qsort}(\{x\}) & \to & \mathsf{cons}(x, \mathsf{nil}) \\
\mathsf{qsort}(\{x\} \cup y) & \to & \mathsf{app}(\mathsf{qsort}(\mathsf{low}(x, y)), \mathsf{cons}(x, \mathsf{qsort}(\mathsf{high}(x, y))))
\end{array}$$

Assume that the term $t = \mathsf{qsort}(\{-\mathsf{1}\} \cup (\{\mathsf{1}\} \cup \{-\mathsf{1}\}))$ is to be reduced using $\xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R}}$. By considering the substitution $\sigma = \{x \mapsto \mathsf{1}, y \mapsto \{-\mathsf{1}\}\}$, the third qsort-rule can be applied since $t \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \mathsf{qsort}(\{-\mathsf{1}\} \cup \{\mathsf{1}\}) \overset{>\Lambda}{\sim}_{\mathcal{E}} \mathsf{qsort}(\{x\} \cup y)\sigma$, i.e., condition *1.* in Definition 11 is satisfied. Therefore,

$$t \xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R}} \mathsf{app}(\mathsf{qsort}(\mathsf{low}(\mathsf{1}, \{-\mathsf{1}\})), \mathsf{cons}(\mathsf{1}, \mathsf{qsort}(\mathsf{high}(\mathsf{1}, \{-\mathsf{1}\}))))$$

Next, $\mathsf{low}(\mathsf{1}, \{-\mathsf{1}\}) \xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R}} \{-\mathsf{1}\}$ by the second low-rule since the instantiated constraint $1 > -1$ is $\mathcal{Th}_{\mathbb{Z}}$-valid. Similarly, $\mathsf{high}(\mathsf{1}, \{-\mathsf{1}\}) \xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R}} \emptyset$. Continuing the reduction eventually yields $\mathsf{cons}(-\mathsf{1}, \mathsf{cons}(\mathsf{1}, \mathsf{nil}))$. ◊

The function symbols occurring at the root position of left-hand sides in $\mathcal{R}$ are of particular interest since they are the only ones that allow a reduction to take place. Therefore, the following notation is introduced for them.

Definition 13 (Defined Symbols). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ be a CERS. Then the* defined symbols *of $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ are $\mathcal{D}(\mathcal{R}) = \{f \mid f = \mathrm{root}(l)$ for some $l \to r[\![\varphi]\!] \in \mathcal{R}\}$.*

It is not immediately obvious whether the rewrite relation $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R}}$ is decidable. In the important case where $\mathcal{E}$ is size-preserving, the following positive answer is obtained. Notice that the canonizable theories of Figure 1 and the canonizable collection data structures of Figure 2 satisfy this requirement on $\mathcal{E}$.

Lemma 14. *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ be a CERS such that $\mathcal{E}$ is size-preserving.*

1. *If s, t are terms, then it is decidable whether $s \sim_{\mathcal{E}} t$. Furthermore, the $\mathcal{E}$-equivalence class of s can be computed effectively.*
2. *For any term s, it is decidable whether s is reducible by $\to_{\mathcal{E}\backslash\mathcal{S}}$, and if so, a term t with $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ is effectively computable.*
3. *For any term s, it is decidable whether s is reducible by $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R}}$, and if so, a term t with $s \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R}} t$ is effectively computable.*

4 Context-Sensitive Rewriting with CERSs

Context-sensitive rewriting strategies are obtained using *replacement maps* that define the context under which a reduction may take place. This is done by specifying the argument positions of a function symbol f where a reduction is allowed. Intuitively, if the replacement map restricts reductions in a certain argument position, then the whole subterm below that argument position may not be reduced.

Definition 15 (Replacement Maps). *A* replacement map *is a mapping μ with $\mu(f) \subseteq \{1, \ldots, \mathrm{arity}(f)\}$ for every function symbol $f \in \mathcal{F} \cup \mathcal{F}_{\mathit{Th}}$.*

Replacement maps are used to denote a subset of all positions in a term as *active*. A position is active if can be reached from the root of the term by only descending into argument positions that are not restricted by the replacement map.

Definition 16 (Active and Inactive Positions). *Let μ be a replacement map and let t be a term. Then the set of* active positions *of t, written $\mathcal{P}os^{\mu}(t)$, is defined by*

1. *$\mathcal{P}os^{\mu}(x) = \{\Lambda\}$ for $x \in \mathcal{V}$, and*
2. *$\mathcal{P}os^{\mu}(f(t_1, \ldots, t_n)) = \{\Lambda\} \cup \{i.p \mid i \in \mu(f)$ and $p \in \mathcal{P}os^{\mu}(t_i)\}$.*

The set of inactive positions *of t is defined as $\mathcal{P}os^{\not\mu}(t) = \mathcal{P}os(t) - \mathcal{P}os^{\mu}(t)$.*

The concept of active positions can also be used to define active (and inactive) subterms of a given term.

Definition 17 (Active and Inactive Subterms). *Let μ be a replacement map and let t be a term. If $t|_p = s$ for an active position $p \in \mathcal{P}os^{\mu}(t)$, then s is an* active subterm *of t, written $t \unrhd_{\mu} s$. If additionally $p \neq \Lambda$, then s is an* active strict subterm *of t, written $t \rhd_{\mu} s$. If $t|_p = s$ for an inactive position $p \in \mathcal{P}os^{\not\mu}(t)$, then s is an* inactive strict subterm *of t, written $t \rhd_{\not\mu} s$.*[1]

The classification of active and inactive subterms can easily be extended to other notions as well to obtain the sets $\mathcal{V}^{\mu}(t)$ of variables occurring in active positions in t, $\mathcal{V}^{\not\mu}(t)$ of variables occurring in inactive positions in t, $\mathcal{F}^{\mu}(t)$ of function symbols occurring in active positions in t, $\mathcal{F}^{\not\mu}(t)$ of function symbols occurring in inactive positions in t, etc.

Now a *context-sensitive constrained equational rewrite system* combines a CERS as in Definition 9 with a replacement map. Notice that the replacement map μ needs to satisfy several conditions on the variables in $\mathcal{S}$ and $\mathcal{E}$. As already noticed in [14], this is due to the permutative nature of equations in $\mathcal{E}$ that may otherwise bring subterms from inactive positions into active positions, and vice versa. Figure 3 lists the replacement maps that are allowed by this definition for the data structures from Section 3.

Definition 18 (Context-Sensitive CERSs). *A* context-sensitive CERS (CS-CERS) *has the form form $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ for a CERS $(\mathcal{R}, \mathcal{S}, \mathcal{E})$ and a replacement map μ such that:*

1. *$\mathcal{E}$ is collapse-free.*
2. *For all $u \approx v \in \mathcal{E}$,*
   (a) *$\mathcal{V}^{\mu}(u) = \mathcal{V}^{\mu}(v)$ and $\mathcal{V}^{\not\mu}(u) = \mathcal{V}^{\not\mu}(v)$,*
   (b) *for all inactive non-variable subterms u' of u, $u' \rhd_{\mu} x$ for a variable x implies $v' \rhd_{\mu} x$ for an inactive non-variable subterm v' of v, and*
   (c) *for all inactive non-variable subterms v' of v, $v' \rhd_{\mu} x$ for a variable x implies $u' \rhd_{\mu} x$ for an inactive non-variable subterm u' of u.*
3. *For all $l \to r \in \mathcal{S}$,*
   (a) *$\mathcal{V}^{\mu}(r) \cap \mathcal{V}^{\not\mu}(l) = \emptyset$, and*
   (b) *for all inactive non-variable subterms r' of r, $r' \rhd_{\mu} x$ for a variable x implies $l' \rhd_{\mu} x$ for an inactive non-variable subterm l' of l.*

The rewrite relation of a CS-CERS is obtained by a small modification of Definition 11 that restricts reductions to active positions.

Definition 19 (Rewrite Relation of a CS-CERS). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS. Then $s \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$ iff there are a constrained rewrite rule $l \to r[\![\varphi]\!] \in \mathcal{R}$, an active position $p \in \mathcal{P}os^{\mu}(s)$, and a $\mathcal{T}h$-based substitution σ such that*

1. *$s|_p \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$,*
2. *$\varphi\sigma$ is $\mathcal{T}h$-valid, and*
3. *$t = s[r\sigma]_p$.*

[1] Notice that $p \neq \Lambda$ is this case since Λ is an active position of any term.

	Constructors	Conditions on μ
Natural numbers	$\mathsf{0}, \mathsf{1}, +$	$\mu(+) = \{1, 2\}$
Integers	$\mathsf{0}, \mathsf{1}, +, -$	$\mu(+) = \{1, 2\}$ $\mu(-) = \{1\}$
Lists	$\mathsf{nil}, \mathsf{cons}$	none
Lists	$\mathsf{nil}, \langle\cdot\rangle, +\!\!+$	$\mu(+\!\!+) = \{1, 2\}$
Compact Lists	$\mathsf{nil}, \mathsf{cons}$	$\mu(\mathsf{cons}) = \emptyset$ or $\mu(\mathsf{cons}) = \{1, 2\}$
Compact Lists	$\mathsf{nil}, \langle\cdot\rangle, +\!\!+$	$\mu(+\!\!+) = \{1, 2\}$
Multisets	$\emptyset, \mathsf{ins}$	$\mu(\mathsf{ins}) = \emptyset$ or $\mu(\mathsf{ins}) = \{1, 2\}$
Multisets	$\emptyset, \{\cdot\}, \cup$	$\mu(\cup) = \{1, 2\}$
Sets	$\emptyset, \mathsf{ins}$	$\mu(\mathsf{ins}) = \emptyset$ or $\mu(\mathsf{ins}) = \{1, 2\}$
Sets	$\emptyset, \{\cdot\}, \cup$	$\mu(\cup) = \{1, 2\}$

Fig. 3. Conditions imposed on the replacement map by Definition 18.

Notice that the rewrite relation from Definition 11 is re-obtained if the replacement map μ is such that $\mu(f) = \{1, \ldots, \mathrm{arity}(f)\}$ for all function symbols.

Example 20. The CERS from Example 1 becomes a CS-CERS by considering the replacement map μ with $\mu(\mathsf{ins}) = \emptyset$ and $\mu(f) = \{1, \ldots, \mathrm{arity}(f)\}$ for all $f \neq \mathsf{ins}$. Then the reduction of the term $\mathsf{take}(2, \mathsf{from}(0))$ has the following form:

$$\begin{aligned}
\mathsf{take}(2, \mathsf{from}(0)) &\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \mathsf{take}(2, \mathsf{ins}(0, \mathsf{from}(1))) \\
&\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \mathsf{cons}(0, \mathsf{take}(2-1, \mathsf{from}(1))) \\
&\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \mathsf{cons}(0, \mathsf{cons}(1, \mathsf{take}(1-1, \mathsf{from}(2)))) \\
&\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \mathsf{cons}(0, \mathsf{cons}(1, \mathsf{nil}))
\end{aligned}$$

Notice that an infinite reduction of this term is not possible since the recursive call to from in the rule $\mathsf{from}(x) \to \mathsf{ins}(x, \mathsf{from}(x+1))$ occurs in an inactive position. ♢

The following properties of rewriting with CERSs are needed in the remainder of this paper. They are similar to analogous Lemmas from [12, 13]. Here, a context C is an *active context* iff □ occurs in an active position in it.

Lemma 21. *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS and let s, t be terms.*

1. *Let $s = C[f(s^*)]$ for an active context C where $f \notin \mathcal{F}(\mathcal{E})$. If $s \sim_{\mathcal{E}} t$, then $t = C'[f(t^*)]$ for an active context C' such that $C \sim_{\mathcal{E}} C'$ and $f(s^*) \overset{>\Lambda}{\sim_{\mathcal{E}}} f(t^*)$.*
2. *$\sim_{\mathcal{E}} \circ \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}}$, where the $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$ steps can be performed using the same constrained rewrite rule and Th-based substitution.*
3. *$\to_{\mathcal{E}\backslash\mathcal{S}} \circ \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \xrightarrow{\mathcal{S}}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^{=}_{\mathcal{E}\backslash\mathcal{S}}$*

Corollary 22. *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS and let s, t be terms.*

1. *If $s \sim_{\mathcal{E}} t$, then s starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction iff t starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction.*
2. *If $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ and t starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction, then s starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction.*

5 Dependency Pairs for Rewriting with CS-CERSs

Extending the dependency pair method for ordinary rewriting [4] to ordinary context-sensitive rewriting has proven to be quite challenging. Recall that dependency pairs are built from recursive calls to defined symbols occurring on the right-hand side of rewrite rules since only these recursive calls may cause non-termination. For this, a signature $\mathcal{F}^\sharp$ is introduced, containing the function symbol $f^\sharp$ for each function symbol $f \in \mathcal{F}$, where $f^\sharp$ has the sort declaration $s_1 \times \ldots \times s_n \to \texttt{top}$ if f has the sort declaration $s_1 \times \ldots \times s_n \to s$ with $s \in \{\texttt{base}, \texttt{univ}\}$. Here, $\texttt{top}$ is a new sort that is distinct from $\texttt{base}$ and $\texttt{univ}$. For the term $t = f(t_1, \ldots, t_n)$, the term $f^\sharp(t_1, \ldots, t_n)$ is denoted by $t^\sharp$. Then a dependency pair generated from a rule $l \to r$ has the shape $l^\sharp \to t^\sharp$, where t is a subterm of r with $\text{root}(t) \in \mathcal{D}(\mathcal{R})$. The main theorem of the dependency pair method for ordinary TRSs [4] and for CERSs [13] states that a TRS or CERS is terminating iff it is not possible to construct infinite chains from the dependency pairs. Here, a sequence $s_1 \to t_1, s_2 \to t_2, \ldots$ of dependency pairs is a *chain* if there exists a substitution σ such that $t_i\sigma$ reduces to $s_{i+1}\sigma$ for all $i \geq 1$.[2]

For context-sensitive rewriting, one might be tempted to restrict the generation of dependency pairs to recursive calls occurring in active positions since these are the only place where reductions may take place. The following example from [2] shows that this results in an unsound method.

Example 23. Consider the following ordinary TRS [2]:

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

Let $\mu(\mathsf{c}) = \emptyset$ and $\mu(\mathsf{f}) = \{1\}$. Since the recursive calls in the first rule are in inactive positions, no dependency pair would be generated if only recursive calls occurring in active positions are considered. Then, context-sensitive termination of the TRS could be concluded, even though it is not terminating: $\mathsf{f}(\mathsf{a}) \to_{\mathcal{R},\mu} \mathsf{f}(\mathsf{c}(\mathsf{f}(\mathsf{a}))) \to_{\mathcal{R},\mu} \mathsf{f}(\mathsf{a}) \to_{\mathcal{R},\mu} \ldots$. Here, $\to_{\mathcal{R},\mu}$ is the standard context-sensitive rewrite relation which is obtained from Definition 19 by disregarding $\mathcal{S}$, $\mathcal{E}$, and every mention of $\mathcal{Th}$. ♢

The problem of the naive approach outlined above is that recursive calls in inactive positions of right-hand sides may become active again after applying other rules. In Example 23, the recursive call to $\mathsf{f}(\mathsf{a})$ that occurs in an inactive position is migrated to an active position by applying the second rule. This is

[2] It is assumed that different (occurrences of) dependency pairs are variable-disjoint.

the reason that the method of [2] has to create a collapsing dependency for any *migrating variable*, where, for a rule $l \to r$, a variable x is migrating if $r \trianglerighteq_\mu x$ but $l \not\trianglerighteq_\mu x$. In Example 23, the collapsing dependency pair $\mathsf{f}^\sharp(\mathsf{c}(x)) \to x$ is created.

As noticed in [1], the need for collapsing dependency pairs causes severe disadvantages since it becomes quite hard to extend methods for proving termination from ordinary rewriting to context-sensitive rewriting. While progress has been made to some extent [2, 3, 18], the resulting methods are weak in practice.

An alternative to the collapsing dependency pairs needed in [2] has recently been presented in [1]. The main observation of [1] is that only certain instantiations of the migrating variables need to be considered. A first, naive approach for this would be to only consider instantiations by *hidden terms*, which are terms with a defined root symbol occurring inactively in right-hand sides. The reason for considering these terms is that they may become active if they are bound to a migrating variable of another rewrite rule.

Definition 24 (Hidden Terms). *Let* $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ *be a CS-CERS. A term* t *is* hidden *iff* $\mathrm{root}(t) \in \mathcal{D}(\mathcal{R})$ *and there exists a rule* $l \to r[\![\varphi]\!]$ *from* $\mathcal{R}$ *such that* $r \rhd_{\not\mu} t$.

Example 25. For the CS-CERS from Example 20, the term $\mathsf{from}(x+\mathsf{1})$ is hidden since $\mathsf{ins}(x, \mathsf{from}(x+\mathsf{1})) \rhd_{\not\mu} \mathsf{from}(x+\mathsf{1})$. ◊

Instantiating the migrating variable in Example 23 by the hidden term $\mathsf{f}(\mathsf{a})$ is sufficient, since the dependency pair $\mathsf{f}^\sharp(\mathsf{c}(\mathsf{f}(\mathsf{a}))) \to \mathsf{f}^\sharp(\mathsf{a})$ obtained by this instantiation gives rise to an infinite chain. In general, considering only instantiations by hidden terms results in an unsound method, as shown by the following example.

Example 26. Consider the following ordinary TRS [1]:

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

Let $\mu(\mathsf{g}) = \{1\}$ and $\mu(\mathsf{a}) = \mu(\mathsf{b}) = \mu(\mathsf{f}) = \mu(\mathsf{h}) = \emptyset$. The only hidden term is b obtained from the first rule. If migrating variables are only instantiated by hidden terms, then the following dependency pairs are obtained:

$$\begin{aligned}
\mathsf{a}^\sharp &\to \mathsf{f}^\sharp(\mathsf{g}(\mathsf{b})) \\
\mathsf{f}^\sharp(x) &\to \mathsf{h}^\sharp(x) \\
\mathsf{h}^\sharp(\mathsf{b}) &\to \mathsf{b}^\sharp \\
\mathsf{b}^\sharp &\to \mathsf{a}^\sharp
\end{aligned}$$

Since these dependency pairs do not give rise to an infinite chain, termination could be concluded, even though $\mathsf{a} \to_{\mathcal{R},\mu} \mathsf{f}(\mathsf{g}(\mathsf{b})) \to_{\mathcal{R},\mu} \mathsf{h}(\mathsf{g}(\mathsf{b})) \to_{\mathcal{R},\mu} \mathsf{g}(\mathsf{b}) \to_{\mathcal{R},\mu} \mathsf{g}(\mathsf{a}) \to_{\mathcal{R},\mu} \ldots$ is an infinite reduction. ◊

As motivated by this example, it becomes necessary to consider certain contexts that may be built above a hidden term using the rewrite rules. In Example

26, this context is $\mathsf{g}(\Box)$. Formally, this observation is captured using the notion of *hiding contexts*. The definition in this paper differs from the one given in [1] by also considering $\mathcal{S}$ and $\mathcal{E}$, which are the key component of CS-CERSs.

Definition 27 (Hiding Contexts). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS. Then $f \in \mathcal{F} \cup \mathcal{F}_{Th}$* hides position *$i$ iff $i \in \mu(f)$ and either*

1. *$f \in \mathcal{F}(\mathcal{E}) \cup \mathcal{F}(\mathcal{S})$, or*
2. *there exist a rule $l \to r[\![\varphi]\!]$ from $\mathcal{R}$ and a term $s = f(s_1, \ldots, s_i, \ldots, s_n)$ such that $r \rhd_{\not\mu} s$ and $s_i \unrhd_\mu x$ for a variable x or $s_i \unrhd_\mu g(\ldots)$ for a $g \in \mathcal{D}(\mathcal{R})$.*

A context C with one hole is hiding *iff either*

1. *$C = \Box$, or*
2. *$C = f(t_1, \ldots, t_{i-1}, C', t_{i+1}, \ldots, t_n)$ such that f hides position i and C' is a hiding context.*

Example 28. For the CS-CERS from Example 20, $+$ hides positions 1 and 2, $-$ hides position 1, and from hides position 1 due to the first rule. ◊

Notice that there are, in general, infinitely many hiding contexts. For example, the hiding contexts in Example 26 are $\Box, \mathsf{g}(\Box), \mathsf{g}(\mathsf{g}(\Box)), \ldots$. In order to represent these infinitely many hiding contexts using only finitely many dependency pairs, fresh function symbols $\mathsf{U}_{\mathtt{base}}$ and $\mathsf{U}_{\mathtt{univ}}$ are introduced that will be used to deconstruct a hiding context in order to obtain the hidden term contained in it. Given these definitions, the following definition of context-sensitive dependency pairs is obtained. Here, DP_u is used instead of the collapsing dependency pairs needed in [2].

Definition 29 (Context-Sensitive Dependency Pairs). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS. The set of* context-sensitive dependency pairs *of $\mathcal{R}$ is defined as $\mathsf{DP}(\mathcal{R}, \mu) = \mathsf{DP}_\mathsf{o}(\mathcal{R}, \mu) \cup \mathsf{DP}_\mathsf{u}(\mathcal{R}, \mu)$ where*

$$\begin{aligned}
\mathsf{DP}_\mathsf{o}(\mathcal{R}, \mu) &= \{l^\sharp \to t^\sharp[\![\varphi]\!] \mid l \to r[\![\varphi]\!] \in \mathcal{R}, r \unrhd_\mu t, \mathrm{root}(t) \in \mathcal{D}(\mathcal{R})\} \\
\mathsf{DP}_\mathsf{u}(\mathcal{R}, \mu) &= \{l^\sharp \to \mathsf{U}_s(x)[\![\varphi]\!] \mid l \to r[\![\varphi]\!] \in \mathcal{R}, r \unrhd_\mu x, l \not\unrhd_\mu x\} \\
&\cup \{\mathsf{U}_s(g(x_1, \ldots, x_i, \ldots, x_n)) \to \mathsf{U}_{s'}(x_i)[\![\top]\!] \mid g \textit{ hides position } i\} \\
&\cup \{\mathsf{U}_s(h) \to h^\sharp[\![\top]\!] \mid h \textit{ is a hidden term}\}
\end{aligned}$$

Here, s and s' are the sorts of x, $g(x_1, \ldots, x_i, \ldots, x_n)$, x_i, and h, respectively, and $\mathsf{U}_{\mathtt{base}}, \mathsf{U}_{\mathtt{univ}} \in \mathcal{F}^\sharp$ are fresh function symbols of arity 1 with sort declaration $\mathtt{base} \to \mathtt{top}$ and $\mathtt{univ} \to \mathtt{top}$, respectively. Furthermore, $\mu(\mathsf{U}_{\mathtt{base}}) = \mu(\mathsf{U}_{\mathtt{univ}}) = \emptyset$ and $\mu(f^\sharp) = \mu(f)$ for all $f \in \mathcal{F}$.

Example 30. For the CS-CERS from Example 20, the context-sensitive dependency pairs are as follows:

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{take}^\sharp(x - \mathsf{1}, ys)\ [\![x > \mathsf{0}]\!] \tag{1}$$

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{U}_{\mathtt{base}}(y)\ [\![x > \mathsf{0}]\!] \tag{2}$$

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{U}_{\mathtt{univ}}(ys)\ [\![x > \mathsf{0}]\!] \tag{3}$$

$$\begin{aligned}
\mathsf{head}^\sharp(\mathsf{ins}(x, xs)) &\to \mathsf{U}_{\mathtt{base}}(x) && (4)\\
\mathsf{tail}^\sharp(\mathsf{ins}(x, xs)) &\to \mathsf{U}_{\mathtt{univ}}(ys) && (5)\\
\mathsf{U}_{\mathtt{univ}}(\mathsf{from}(x+1)) &\to \mathsf{from}^\sharp(x+1) && (6)\\
\mathsf{U}_{\mathtt{base}}(x+y) &\to \mathsf{U}_{\mathtt{base}}(x) && (7)\\
\mathsf{U}_{\mathtt{base}}(x+y) &\to \mathsf{U}_{\mathtt{base}}(y) && (8)\\
\mathsf{U}_{\mathtt{base}}(-x) &\to \mathsf{U}_{\mathtt{base}}(x) && (9)\\
\mathsf{U}_{\mathtt{univ}}(\mathsf{from}(x)) &\to \mathsf{U}_{\mathtt{base}}(x) && (10)
\end{aligned}$$

For this, recall the hidden term $\mathsf{from}(x+1)$ from Example 25 and the hiding contexts from Example 28. $\Diamond$

As usual in methods based on dependency pairs, context-sensitive dependency pairs can be used in order to build chains, and the goal is to show that $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ is terminating if there are no infinite minimal chains.

Definition 31 ((Minimal) $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-Chains). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS and let $\mathcal{P}$ be a set of dependency pairs. A (possibly infinite) sequence of dependency pairs $s_1 \to t_1[\![\varphi_1]\!], s_2 \to t_2[\![\varphi_2]\!], \ldots$ from $\mathcal{P}$ is a $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-*chain *iff there exists a $\mathcal{T}h$-based substitution σ such that $t_i\sigma \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_{i+1}\sigma$, the instantiated $\mathcal{T}h$-constraint $\varphi_i\sigma$ is $\mathcal{T}h$-valid, and $s_i\sigma$ is a normal form w.r.t. $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ for all $i \geq 1$. The above $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain is* minimal *iff $t_i\sigma$ does not start an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction for all $i \geq 1$.*

Here, $\xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ corresponds to reductions occurring strictly below the root of $t_i\sigma$ (notice that $\mathrm{root}(t_i) \in \mathcal{F}^\sharp$), and $\xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}}$ corresponds to normalization and matching before applying $s_{i+1} \to t_{i+1}[\![\varphi_i]\!]$ at the root position.

While the definition of chains is essentially identical to the non-context-sensitive case in [13], proving the main result for CS-CERSs is more complicated than in the non-context-sensitive case and requires several technical definitions and lemmas. First, it is convenient to formally introduce *minimal non-terminating terms.* These terms start an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction, but their active strict subterms do not start infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reductions.

Definition 32 (Minimal Non-Terminating Terms). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS. A term t is* minimal non-terminating*, written $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, iff t starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction but no t' with $t \rhd_\mu t'$ starts an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction.*

The following properties are an easy consequence of Corollary 22.

Lemma 33. *Let $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.*

1. *If $t \sim_{\mathcal{E}} t'$, then $t' \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.*

2. *If $t \to^*_{\mathcal{E}\backslash\mathcal{S}} t'$ and t' is non-terminating, then $t' \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.*

The next lemma intuitively states that application of equations from $\mathcal{E}$ or rules from $\mathcal{S}$ transforms a hiding context into another hiding context. Notice that a corresponding lemma is not needed in [1] since that paper is only concerned with ordinary rewriting. This lemma is the reason why any $f \in \mathcal{F}(\mathcal{E}) \cup \mathcal{F}(\mathcal{S})$ is considered to hide any active argument position in Definition 27.

Lemma 34. *Let C be a hiding context and let $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.*

1. *If $C[t] \vdash\dashv_{\mathcal{E}} s$ and the $\vdash\dashv_{\mathcal{E}}$-step is applied at a position in C, then $s = C'[t]$ for a hiding context C'.*
2. *Let $C[t] \to_{\mathcal{S}} s$ at a position in C using a rule $l \to r \in \mathcal{S}$ and a substitution σ such that the variable $x \in \mathcal{V}(l)$ with $x\sigma \trianglerighteq_\mu t$ (if any) satisfies $r \trianglerighteq_\mu x$. Then $s = C'[t]$ for a hiding context C'.*

The proof of the main theorem, i.e., that $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ is terminating if there are no infinite minimal chains, is modularized by introducing the following abstract property. A term has the *hiding property* if its minimal non-terminating subterms in inactive positions are obtained from hidden terms and surrounded by hiding contexts. The definition in this paper differs from the original definition in [1] by making use of $\mathcal{E}$ and $\mathcal{S}$ in order to obtain the minimal non-terminating terms from instances of hidden terms.

Definition 35 (Hiding Property). *A term $u \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ has the* hiding property *iff whenever $u \rhd_{\not\mu} s \trianglerighteq_\mu t$ with $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, then $s = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^*_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$.*

The following key lemma states that the hiding property is preserved by $\sim_{\mathcal{E}}$, $\to_{\mathcal{E}\backslash\mathcal{S}}$, and $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$. This is needed for the proof of the main theorem and is a non-trivial (technical) extension of a similar statement in [1].

Lemma 36. *Let $u \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ have the hiding property.*

1. *If $u \vdash\dashv_{\mathcal{E}} v \trianglerighteq_\mu w$ with $w \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, then w has the hiding property.*
2. *If $u \to_{\mathcal{E}\backslash\mathcal{S}} v \trianglerighteq_\mu w$ with $w \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, then w has the hiding property.*
3. *If $u \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} v \trianglerighteq_\mu w$ with $w \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, then w has the hiding property.*

With this machinery at hand, the following main theorem can now be proved. As usual in methods based on dependency pairs, it states that rewriting is terminating if there are no infinite minimal chains.

Theorem 37. *Let $(\mathcal{R},\mathcal{S},\mathcal{E},\mu)$ be a CS-CERS. Then $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ is terminating if there are no infinite minimal $(\mathsf{DP}(\mathcal{R},\mu),\mathcal{R},\mathcal{S},\mathcal{E},\mu)$-chains.*

In the next section, a number of techniques for showing absence of infinite chains is presented. In order to show soundness of these techniques independently, and in order to obtain flexibility on the order in which these techniques are applied, these techniques are stated in a dependency pair framework in the spirit of [17]. Termination techniques in this framework operate on *CS-DP problems* $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, where $\mathcal{P}$ is a finite set of dependency pairs and $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ is a CS-CERS. Now CS-DP problems are transformed using *CS-DP processors*. Here, a CS-DP processor is a function that takes a CS-DP problem as input and returns a finite set of CS-DP problems as output. The CS-DP processor Proc is *sound* iff for all CS-DP problems $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ with an infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain there exists a CS-DP problem $(\mathcal{P}', \mathcal{R}', \mathcal{S}', \mathcal{E}', \mu') \in \mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ with an infinite minimal $(\mathcal{P}', \mathcal{R}', \mathcal{S}', \mathcal{E}', \mu')$-chain.

For a termination proof of the CS-CERS $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, sound CS-DP processors are applied recursively to the initial CS-DP problem $(\mathsf{DP}(\mathcal{R}, \mu), \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$. If all resulting CS-DP problems have been transformed into the empty set, then termination has been shown.

6 CS-DP Processors

After introducing the dependency pair framework for context-sensitive rewriting with CERSs in Section 5, the goal of this section is to introduce several sound CS-DP processors. Most of these processors are similar to corresponding processors in [13], and with the exception of Sections 6.4 and 6.5, this extension is relatively straightforward.

First, *context-sensitive dependency graphs* are introduced. This technique decomposes a CS-DP problem into several independent CS-DP problems by determining which dependency pairs may follow each other in chains. Next, the subterm criterion of [19, 13] is adapted to CS-CERSs. Thirdly, reduction pairs [21] are considered, and it is furthermore shown that polynomial interpretations with negative coefficients are applicable even though they do not give rise to reduction pairs. Finally, the technique based on reduction pairs (or polynomial interpretations with negative coefficients) is made more powerful by showing that it is possible to restrict attention to certain subsets of $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{E}$ when considering the CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$.

6.1 Dependency Graphs

Like the corresponding DP processor from [13], the CS-DP processor introduced in this section decomposes a CS-DP problem into several independent CS-DP problems by determining which dependency pairs from $\mathcal{P}$ may follow each other in a $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain. The processor relies on the notion of *dependency graphs*, which were initially introduced for ordinary TRSs [4].

Definition 38 (Context-Sensitive Dependency Graphs). *For a CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, the $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-*dependency graph $\mathsf{DG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$

has the dependency pairs in $\mathcal{P}$ as nodes and there is an arc from $s_1 \to t_1[\![\varphi_1]\!]$ to $s_2 \to t_2[\![\varphi_2]\!]$ iff $s_1 \to t_1[\![\varphi_1]\!]$, $s_2 \to t_2[\![\varphi_2]\!]$ is a $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain.

As usual for dependency graphs, $\mathsf{DG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ cannot be computed exactly in general and an estimation has to be used instead. The estimation used in this section is similar to the estimation of [13] but has been adapted to the context-sensitive case. This adaptation is similar to the estimated dependency graphs for ordinary context-sensitive rewriting used in [2, 1].

Definition 39 (Estimated Context-Sensitive Dependency Graphs). *For a CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, the nodes in the* estimated $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-dependency graph $\mathsf{EDG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ *are the dependency pairs in $\mathcal{P}$ and there is an arc from $s_1 \to t_1[\![\varphi_1]\!]$ to $s_2 \to t_2[\![\varphi_2]\!]$ iff there exists a substitution σ that is $\mathcal{Th}$-based for $\mathcal{V}(s_1) \cup \mathcal{V}(s_2)$*[3] *such that $\textsc{cap}_\mu(t_1)\sigma \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma$, the terms $s_1\sigma$ and $s_2\sigma$ are normal forms w.r.t. $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$, and $\varphi_1\sigma$ and $\varphi_2\sigma$ are $\mathcal{Th}$-valid. The function $\textsc{cap}_\mu$ is defined by*

1. $\textsc{cap}_\mu(x) = x$ for variables x of sort `base`*,*
2. $\textsc{cap}_\mu(x) = y$ for variables x of sort `univ`*,*
3. $\textsc{cap}_\mu(f(t_1, \ldots, t_n)) = f(t'_1, \ldots, t'_n)$ where

$$t'_i = \begin{cases} t_i & \textit{if } i \notin \mu(f) \\ \textsc{cap}_\mu(t_i) & \textit{otherwise} \end{cases}$$

if there does not exist an $l \to r[\![\varphi]\!] \in \mathcal{R}$ such that $f(t'_1, \ldots, t'_n)\sigma \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$ for a substitution σ that is $\mathcal{Th}$-based for $\mathcal{V}(f(t_1, \ldots, t_n)) \cup \mathcal{V}(l)$ where $\varphi\sigma$ is $\mathcal{Th}$-valid, and

4. $\textsc{cap}_\mu(f(t_1, \ldots, t_n)) = y$ otherwise.

Here, y is the next variable in an infinite list $y_1, y_2, \ldots$ of fresh variables.

It is also possible to omit the checks for irreducibility by $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ and $\mathcal{Th}$-validity, and it is possible to replace case *3.* in the definition of $\textsc{cap}_\mu$ by a simple check for $f \notin \mathcal{D}(\mathcal{R})$.

Next, it is shown that the estimated dependency graph is indeed an overapproximation of the dependency graph, i.e., $\mathsf{EDG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ is a supergraph of $\mathsf{DG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$.

Theorem 40 (Correctness of EDG). *For any DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, the estimated dependency graph $\mathsf{EDG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ is a supergraph of the dependency graph $\mathsf{DG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$.*

The following CS-DP processor uses the (estimated) dependency graph in order to decompose a CS-DP problem into several independent CS-DP problems.

[3] I.e., $\sigma(x) \in \mathcal{T}(\mathcal{F}_{\mathcal{Th}}, \mathcal{V})$ for all variables $x \in \mathcal{V}(s_1) \cup \mathcal{V}(s_2)$ of sort `base`.

For this, notice that every infinite $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain contains an infinite tail that stays within one *strongly connected component (SCC)* of $\mathsf{EDG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, and it is thus sufficient to prove the absence of infinite chains for all SCCs separately.

Theorem 41 (CS-DP Processor Based on Dependency Graphs). *Let* Proc *be a CS-DP processor such that* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) = \{(\mathcal{P}_1, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu), \ldots, (\mathcal{P}_n, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$*, where* $\mathcal{P}_1, \ldots, \mathcal{P}_n$ *are the SCCs of* $(\mathsf{E})\mathsf{DG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$*. Then* Proc *is sound.*

Example 42. Recall the following dependency pairs from Example 30:

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{take}^\sharp(x - \mathsf{1}, ys)\ [\![x > \mathsf{0}]\!] \tag{1}$$

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{U}_{\mathtt{base}}(y)\ [\![x > \mathsf{0}]\!] \tag{2}$$

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{U}_{\mathtt{univ}}(ys)\ [\![x > \mathsf{0}]\!] \tag{3}$$

$$\mathsf{head}^\sharp(\mathsf{ins}(x, xs)) \to \mathsf{U}_{\mathtt{base}}(x) \tag{4}$$

$$\mathsf{tail}^\sharp(\mathsf{ins}(x, xs)) \to \mathsf{U}_{\mathtt{univ}}(ys) \tag{5}$$

$$\mathsf{U}_{\mathtt{univ}}(\mathsf{from}(x + \mathsf{1})) \to \mathsf{from}^\sharp(x + \mathsf{1}) \tag{6}$$

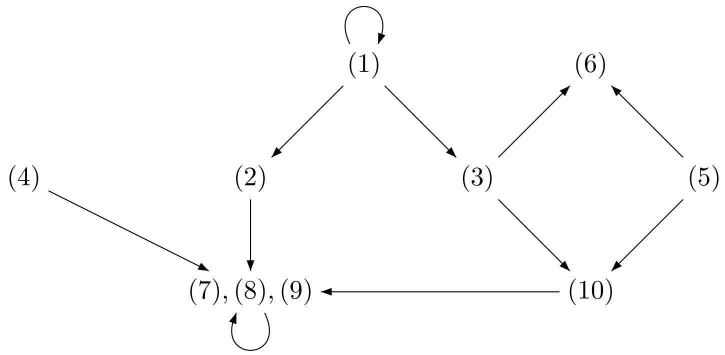
$$\mathsf{U}_{\mathtt{base}}(x + y) \to \mathsf{U}_{\mathtt{base}}(x) \tag{7}$$

$$\mathsf{U}_{\mathtt{base}}(x + y) \to \mathsf{U}_{\mathtt{base}}(y) \tag{8}$$

$$\mathsf{U}_{\mathtt{base}}(-x) \to \mathsf{U}_{\mathtt{base}}(x) \tag{9}$$

$$\mathsf{U}_{\mathtt{univ}}(\mathsf{from}(x)) \to \mathsf{U}_{\mathtt{base}}(x) \tag{10}$$

Then the following estimated dependency graph $\mathsf{EDG}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ is obtained:



Here, the nodes for (7), (8), and (9) have been combined since they have "identical" incoming and outgoing arcs. This estimated dependency graph contains two SCCs, and according to Theorem 41, the following CS-DP problems are obtained:

$$(\{(1)\}, \mathcal{Q}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) \tag{11}$$

$$(\{(7), (8), (9)\}, \mathcal{Q}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) \tag{12}$$

These CS-DP problem can now be handled independently of each other. ◊

6.2 Subterm Criterion

The subterm criterion for ordinary TRSs [19] is a relatively simple technique which is none the less surprisingly powerful. The technique works particularly well for functions that are defined using primitive recursion. The subterm criterion applies a *simple projection* which collapses a term $f^\sharp(t_1, \ldots, t_n)$ to one of its direct subterms. Given a set $\mathcal{P}$ of dependency pairs and a subset $\mathcal{P}' \subseteq \mathcal{P}$, the method consists of finding a simple projection such that the collapsed right-hand side is a subterm of the collapsed left-hand side for all dependency pairs in $\mathcal{P}$, where furthermore this subterm relation is strict for all dependency pairs from $\mathcal{P}'$. Then the dependency pairs in $\mathcal{P}'$ may be removed from the CS-DP problem.

Definition 43 (Simple Projections). *A* simple projection *is a mapping π that assigns to every $f^\sharp \in \mathcal{F}^\sharp$ with* $\text{arity}(f^\sharp) = n$ *an argument position i with $1 \leq i \leq n$. The mapping that assigns to every term $f^\sharp(t_1, \ldots, t_n)$ its argument $t_{\pi(f^\sharp)}$ is also denoted by π.*

In the context of CERSs, the subterm relation modulo $\mathcal{E}$ can be used [13]. For CS-CERSs, this relation needs to take the replacement map into account by only considering subterms in active positions. This is similar to [2].

Definition 44 ($\mathcal{E}$-μ-Subterms). *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS and let s, t be terms. Then t is a* strict $\mathcal{E}$-μ-subterm of *s, written $s \rhd_{\mathcal{E},\mu} t$, iff $s \sim_{\mathcal{E}} \circ \rhd_\mu \circ \sim_{\mathcal{E}} t$. The term t is an* $\mathcal{E}$-μ-subterm of *s, written $s \unrhd_{\mathcal{E},\mu} t$, iff $s \rhd_{\mathcal{E},\mu} t$ or $s \sim_{\mathcal{E}} t$.*

If $\mathcal{E}$ is size-preserving, the subterm relation thus defined has the following pleasant properties.

Lemma 45. *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CERS such that $\mathcal{E}$ is size-preserving.*

1. *Given terms s, t, it is decidable whether $s \rhd_{\mathcal{E},\mu} t$ or $s \unrhd_{\mathcal{E},\mu} t$.*
2. *$\rhd_{\mathcal{E},\mu}$ is well-founded.*
3. *$\rhd_{\mathcal{E},\mu}$ and $\unrhd_{\mathcal{E},\mu}$ are stable.*
4. *$\rhd_{\mathcal{E},\mu}$ and $\unrhd_{\mathcal{E},\mu}$ are compatible with $\sim_{\mathcal{E}}$.*

Now the subterm criterion as outlined above can easily be implemented using a CS-DP processor.

Theorem 46 (CS-DP Processor Based on the Subterm Criterion). *For a simple projection π, let* Proc *be a CS-DP processor with* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) =$

- *$\{(\mathcal{P} - \mathcal{P}', \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, if $\mathcal{E}$ is size-preserving and $\mathcal{P}' \subseteq \mathcal{P}$ such that*
  - *$\pi(s) \rhd_{\mathcal{E},\mu} \pi(t)$ for all $s \to t[\![\varphi]\!] \in \mathcal{P}'$, and*
  - *$\pi(s) \unrhd_{\mathcal{E},\mu} \pi(t)$ for all $s \to t[\![\varphi]\!] \in \mathcal{P} - \mathcal{P}'$.*
- *$(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, otherwise.*

Then Proc *is sound.*

Example 47. Recall the CS-DP problem (12) from Example 42, containing the following dependency pairs:

$$\mathsf{U}_{\mathsf{base}}(x+y) \to \mathsf{U}_{\mathsf{base}}(x) \tag{7}$$

$$\mathsf{U}_{\mathsf{base}}(x+y) \to \mathsf{U}_{\mathsf{base}}(y) \tag{8}$$

$$\mathsf{U}_{\mathsf{base}}(-x) \to \mathsf{U}_{\mathsf{base}}(x) \tag{9}$$

Using the simple projection with $\pi(\mathsf{U}_{\mathsf{base}}) = 1$, this CS-DP problem can easily be handled. $\Diamond$

6.3 Reduction Pairs

The dependency pair framework for ordinary rewriting makes heavy use of reduction pairs $(\gtrsim, \succ)$ [21] in order to remove dependency pairs from DP problems.

Definition 48 (Reduction Pairs). *Let $\gtrsim$ be reflexive, transitive, monotonic, and stable. Let $\succ$ be well-founded and stable. Then $(\gtrsim, \succ)$ is a* reduction pair *iff $\succ$ is compatible with $\gtrsim$, i.e., iff $\gtrsim \circ \succ \circ \gtrsim \subseteq \succ$. The relation $\gtrsim \cap \gtrsim^{-1}$ is denoted by $\sim$.*

The idea for using reduction pairs is simple: If all dependency pairs from a CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ are decreasing w.r.t. $\gtrsim$ or $\succ$, then all dependency pairs that are decreasing w.r.t. $\succ$ cannot occur infinitely often in infinite chains and may thus be deleted. In order to capture the reductions that take place between the instantiated dependency pairs, it becomes necessary to also require that all rules in $\mathcal{R}$ are decreasing w.r.t. $\gtrsim$. Similar conditions need to be imposed on $\mathcal{S}$ and $\mathcal{E}$ as well.

Theorem 49 (CS-DP Processor Based on Reduction Pairs). *For a reduction pair $(\gtrsim, \succ)$, let* Proc *be a CS-DP processor with* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) =$

- $\{(\mathcal{P} - \mathcal{P}', \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, *if $\mathcal{P}' \subseteq \mathcal{P}$ and*
  - *$s \succ t$ for all $s \to t[\![\varphi]\!] \in \mathcal{P}'$,*
  - *$s \gtrsim t$ for all $s \to t[\![\varphi]\!] \in \mathcal{P} - \mathcal{P}'$,*
  - *$l \gtrsim r$ for all $l \to r[\![\varphi]\!] \in \mathcal{R}$,*
  - *$l \gtrsim r$ for all $l \to r \in \mathcal{S}$, and*
  - *$u \sim v$ for all $u \approx v \in \mathcal{E}$.*
- $\{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, *otherwise.*

Then Proc *is sound.*

Full monotonicity of $\gtrsim$ is a quite strong requirement, and as in [13] it is desirable to relax this requirement. If natural numbers in the form of $\mathcal{T}h_{\mathbb{N}}$ are built-in, then the $\mathcal{PA}$-reduction pairs of [13] can be applied for CS-CERSs as well. In the following, it is shown how a similar relaxation is also possible for $\mathcal{T}h_{\mathbb{Z}}$. For sake of concreteness, this is done using polynomial interpretations, but it is also possible to develop an abstract framework of $\mathcal{T}h_{\mathbb{Z}}$-reduction pairs [11].

A *$\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation $\mathcal{P}ol$* fixes a constant $c_{\mathcal{P}ol} \in \mathbb{Z}$ and maps

1. the symbols in $\mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}}$ to polynomials over $\mathbb{Z}$ in the natural way, i.e., $\mathcal{P}ol(\mathsf{0}) = 0$, $\mathcal{P}ol(\mathsf{1}) = 1$, $\mathcal{P}ol(-) = -x_1$ and $\mathcal{P}ol(+) = x_1 + x_2$,
2. the symbols in $\mathcal{F}$ to polynomials over $\mathbb{N}$ such that $\mathcal{P}ol(f) \in \mathbb{N}[x_1, \ldots, x_n]$ if $\text{arity}(f) = n$, and
3. the symbols in $\mathcal{F}^\sharp$ to polynomials over $\mathbb{Z}$ such that $\mathcal{P}ol(f^\sharp) \in \mathbb{Z}[x_1, \ldots, x_n]$ if $\text{arity}(f^\sharp) = n$.

Terms are mapped to polynomials by defining $[x]_{\mathcal{P}ol} = x$ for variables $x \in \mathcal{V}$ and $[f(t_1, \ldots, t_n)]_{\mathcal{P}ol} = \mathcal{P}ol(f)([t_1]_{\mathcal{P}ol}, \ldots, [t_n]_{\mathcal{P}ol})$ for $f \in \mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}} \cup \mathcal{F} \cup \mathcal{F}^\sharp$, i.e., the polynomials obtained from the direct subterms $t_1, \ldots, t_n$ are combined using the polynomial associated to the root symbol.

The reason for fixing the polynomials for the symbols from $\mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}}$ in the natural way is that a term from $\mathcal{T}(\mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}})$ is then mapped to the integer it represents. This will allow to directly use the $\mathcal{T}h_{\mathbb{Z}}$-constraint when comparing two constrained terms. A $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation operates on (unconstrained) terms as usual.

Definition 50 ($\succ_{\mathcal{P}ol}$ and $\gtrsim_{\mathcal{P}ol}$ for $\mathcal{T}h_{\mathbb{Z}}$-Polynomial Interpretations). *Let $\mathcal{P}ol$ be a $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation. Then $\succ_{\mathcal{P}ol}$ is defined by $s \succ_{\mathcal{P}ol} t$ iff $[s\sigma]_{\mathcal{P}ol} \geq c_{\mathcal{P}ol}$ and $[s\sigma]_{\mathcal{P}ol} > [t\sigma]_{\mathcal{P}ol}$ for all ground substitutions $\sigma : \mathcal{V}(s) \cup \mathcal{V}(t) \to \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}})$. Similarly, $\gtrsim_{\mathcal{P}ol}$ is defined by $s \gtrsim_{\mathcal{P}ol} t$ iff $[s\sigma]_{\mathcal{P}ol} \geq [t\sigma]_{\mathcal{P}ol}$ for all ground substitutions $\sigma : \mathcal{V}(s) \cup \mathcal{V}(t) \to \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}})$. Thus, $s \sim_{\mathcal{P}ol} t$ iff $[s\sigma]_{\mathcal{P}ol} = [t\sigma]_{\mathcal{P}ol}$ for all ground substitutions $\sigma : \mathcal{V}(s) \cup \mathcal{V}(t) \to \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}})$.*

For constrained terms, it suffices if $s\sigma \succ_{\mathcal{P}ol} t\sigma$ (or $s\sigma \gtrsim_{\mathcal{P}ol} t\sigma$) for all substitutions σ that make the constraint $\mathcal{T}h_{\mathbb{Z}}$-valid. This is similar to [13].

Definition 51 ($\succ_{\mathcal{P}ol}$ and $\gtrsim_{\mathcal{P}ol}$ on Constrained Terms). *Let $(\gtrsim, \succ)$ be a $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation, let s, t be terms and let φ be a $\mathcal{T}h_{\mathbb{Z}}$-constraint. Then $s[\![\varphi]\!] \gtrsim_{\mathcal{P}ol} t[\![\varphi]\!]$ iff $s\sigma \gtrsim_{\mathcal{P}ol} t\sigma$ for all $\mathcal{T}h_{\mathbb{Z}}$-based substitutions σ such that $\varphi\sigma$ is $\mathcal{T}h_{\mathbb{Z}}$-valid. Similarly, $s[\![\varphi]\!] \succ_{\mathcal{P}ol} t[\![\varphi]\!]$ iff $s\sigma \succ_{\mathcal{P}ol} t\sigma$ for all $\mathcal{T}h_{\mathbb{Z}}$-based substitutions σ such that $\varphi\sigma$ is $\mathcal{T}h_{\mathbb{Z}}$-valid.*

Checking $s[\![\varphi]\!] \succ_{\mathcal{P}ol} t[\![\varphi]\!]$ is done by showing that the following formulas are true in the integers:

$$\forall x_1, \ldots, x_n.\ \varphi \Rightarrow [s]_{\mathcal{P}ol} \geq c_{\mathcal{P}ol}$$
$$\forall x_1, \ldots, x_n.\ \varphi \Rightarrow [s]_{\mathcal{P}ol} > [t]_{\mathcal{P}ol}$$

Here, $x_1, \ldots, x_n$ are the variables occurring in s and t. This requirement might be impossible to show if one of the x_i has sort `univ` since then the possible values it can take are not restricted by the $\mathcal{T}h_{\mathbb{Z}}$-constraint φ. By restricting the $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation $\mathcal{P}ol$ such that for each $f \in \mathcal{F}$ with resulting sort `univ`, the polynomial $\mathcal{P}ol(f)$ may only depend on a variable x_i if the i^{th} argument of f has sort `univ`, an easier requirement is obtained since then ground terms of sort `univ` evaluate to non-negative integers. Using this restriction, showing

$s[\![\varphi]\!] \succ_{\mathcal{P}ol} t[\![\varphi]\!]$ can be achieved by showing that the following formulas are true in the integers:

$$\forall x_1, \ldots, x_k.\ \forall y_1 \geq 0, \ldots, y_l \geq 0.\ \varphi \Rightarrow [s]_{\mathcal{P}ol} \geq c_{\mathcal{P}ol}$$
$$\forall x_1, \ldots, x_k.\ \forall y_1 \geq 0, \ldots, y_l \geq 0.\ \varphi \Rightarrow [s]_{\mathcal{P}ol} > [t]_{\mathcal{P}ol}$$

Here, $x_1, \ldots, x_k$ are the variables of sort `base` in s and t and $y_1, \ldots, y_l$ are the variables of sort `univ` in s and t.

Since $\mathcal{P}ol(-)$ is clearly not monotonic in its argument, it becomes necessary to impose restrictions under which $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretations may be applied. More precisely, it has to be ensured that no reduction with $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ may take place below an occurrence of the function symbol $-$. There are two simple ways to ensure this:

1. All arguments of right-hand sides of $\mathcal{P}$ are terms from $\mathcal{T}(\mathcal{F}_{\mathcal{T}h_{\mathbb{Z}}}, \mathcal{V})$. Then no reductions w.r.t. $\xrightarrow{\mathcal{S},\mathcal{Q}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R}}$ can take place between instantiated dependency pairs in a chain since chains are built using $\mathcal{T}h_{\mathbb{Z}}$-based substitutions.
2. $\mathcal{F}$ does not contain a function symbol with resulting sort `base`.

Theorem 52 (CS-DP Processor Based on $\mathcal{T}h_{\mathbb{Z}}$-Polynomial Interpretations). *For a $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretations $\mathcal{P}ol$, let* Proc *be a CS-DP processor with* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) =$

- $\{(\mathcal{P} - \mathcal{P}', \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, *if $\mathcal{P}ol$ is applicable, $\mathcal{P}' \subseteq \mathcal{P}$, and*
  - $s[\![\varphi]\!] \succ_{\mathcal{P}ol} t[\![\varphi]\!]$ *for all* $s \to t[\![\varphi]\!] \in \mathcal{P}'$,
  - $s[\![\varphi]\!] \gtrsim_{\mathcal{P}ol} t[\![\varphi]\!]$ *for all* $s \to t[\![\varphi]\!] \in \mathcal{P} - \mathcal{P}'$,
  - $l[\![\varphi]\!] \gtrsim_{\mathcal{P}ol} r[\![\varphi]\!]$ *for all* $l \to r[\![\varphi]\!] \in \mathcal{R}$,
  - $l\sigma \gtrsim_{\mathcal{P}ol} r\sigma$ *for all* $l \to r \in \mathcal{S}$ *and all substitutions* σ, *and*
  - $u\sigma \sim_{\mathcal{P}ol} v\sigma$ *for all* $u \approx v \in \mathcal{E}$ *and all substitutions* σ.
- $\{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, *otherwise.*

Then Proc *is sound.*

If $\mathcal{P}ol$ is not applicable, it might be possible to use a *non-collapsing argument filtering* [21] for the function symbols $f^\sharp \in \mathcal{F}^\sharp$ that ensures that condition *1.* from above is true after application of the argument filtering. This idea is illustrated in the following example and formally stated in [11].

Example 53. Recall the CS-DP problem (11) from Example 42, containing the following dependency pair:

$$\mathsf{take}^\sharp(x, \mathsf{ins}(y, ys)) \to \mathsf{take}^\sharp(x - 1, ys)\ [\![x > 0]\!] \tag{1}$$

Using a non-collapsing argument filtering that only retains the first argument of $\mathsf{take}^\sharp$, this dependency pair is transformed into the following dependency pair:

$$\mathsf{take}^\sharp(x) \to \mathsf{take}^\sharp(x - 1)\ [\![x > 0]\!] \tag{13}$$

Now condition *1.* from above is satisfied, i.e., $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretations are applicable. ♢

6.4 Function Dependencies

The DP processors of Theorems 49 and 52 have to consider all of $\mathcal{R}$. As shown in [13], this requirement can be relaxed substantially for CERSs. This section shows that this requirement can also be weakened for CS-CERSs. In contrast to the relatively straightforward adaptations of the dependency graph and the subterm criterion, this adaptation is non-trivial. For ordinary context-sensitive rewriting, corresponding results were obtained only very recently [18, 1].

The intuitive idea for the technique based on function dependencies is that every $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain can be transformed into a sequence that only uses subsets $\mathcal{R}' \subseteq \mathcal{R}$, $\mathcal{S}' \subseteq \mathcal{S}$, and $\mathcal{E}' \subseteq \mathcal{E}$. Here, the subsets $\mathcal{R}'$, $\mathcal{S}'$, and $\mathcal{E}'$ are based on the dependencies between function symbols. As the following example from [18] shows, the method is unsound if the definition of function dependencies from [13] is used, where a function symbol f depends on all function symbols g occurring in the right-hand side of a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ or $l \to r \in \mathcal{S}$ with $\mathrm{root}(l) = f$ or occurring in an equation $u \approx v \in \mathcal{E}$ with $\mathrm{root}(u) = f$ or $\mathrm{root}(v) = f$.

Example 54. Consider the following ordinary TRS [18]:

$$\begin{aligned} \mathsf{b} &\to \mathsf{c}(\mathsf{b}) \\ \mathsf{f}(\mathsf{c}(x), x) &\to \mathsf{f}(x, x) \end{aligned}$$

Let $\mu(\mathsf{f}) = \{1, 2\}$ and $\mu(\mathsf{c}) = \emptyset$. The following dependency pairs are obtained:

$$\mathsf{f}^\sharp(\mathsf{c}(x), x) \to \mathsf{f}^\sharp(x, x) \tag{14}$$

$$\mathsf{f}^\sharp(\mathsf{c}(x), x) \to \mathsf{U}_{\mathtt{univ}}(x) \tag{15}$$

$$\mathsf{U}_{\mathtt{univ}}(\mathsf{b}) \to \mathsf{b}^\sharp \tag{16}$$

$$\mathsf{U}_{\mathsf{base}}(x + y) \to \mathsf{U}_{\mathsf{base}}(x) \tag{17}$$

$$\mathsf{U}_{\mathsf{base}}(x + y) \to \mathsf{U}_{\mathsf{base}}(y) \tag{18}$$

$$\mathsf{U}_{\mathsf{base}}(-x) \to \mathsf{U}_{\mathsf{base}}(x) \tag{19}$$

The CS-DP problem consisting of (17)–(19) can easily be handled using the subterm criterion of Section 6.2. Using the definition of function dependencies from [13], $\mathcal{R}$ would not need to be considered for the CS-DP problem consisting of (14). Then termination could falsely be concluded using a reduction pair based on a polynomial interpretation with $\mathcal{P}ol(\mathsf{f}^\sharp) = x_1$ and $\mathcal{P}ol(\mathsf{c}) = x_1 + 1$, although $\mathsf{f}(\mathsf{c}(\mathsf{b}), \mathsf{b}) \to_{\mathcal{R},\mu} \mathsf{f}(\mathsf{b}, \mathsf{b}) \to_{\mathcal{R},\mu} \mathsf{f}(\mathsf{c}(\mathsf{b}), \mathsf{b}) \to_{\mathcal{R},\mu} \ldots$ is an infinite reduction. ♢

The function dependencies as defined in [13] thus need to be adapted as follows, similarly to [18, 1]. Notice that function symbols occurring in inactive positions in left-hand sides sides of dependency pairs and rules need to be considered.

Definition 55 (Context-Sensitive Function Dependencies). *For a CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ where $\mathcal{E}$ is size-preserving and two symbols $f, h \in \mathcal{F}$, let $f \blacktriangleright^1_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} h$ iff $f = h$ or there exists a symbol g with $g \blacktriangleright^1_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} h$ and a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ such that $\mathrm{root}(l) = f$ and $g \in \mathcal{F}^{\not\mu}(l) \cup \mathcal{F}(r)$. Let*

$$\begin{aligned}\varDelta^1(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu) = {} & \mathcal{F}_{\mathit{Th}} \cup \mathcal{F}(\mathcal{S}) \cup \mathcal{F}(\mathcal{E}) \\ & \cup \textstyle\bigcup_{s\to t[\![\varphi]\!]\in\mathcal{P}} \{g \mid f \blacktriangleright^1_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} g \text{ for an } f \in \mathcal{F}^{\not\mu}(s) \cup \mathcal{F}(t)\} \\ & \cup \textstyle\bigcup_{l\to r[\![\varphi]\!]\in\mathcal{R}} \{g \mid f \blacktriangleright^1_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} g \text{ for an } f \in \mathcal{F}^{\not\mu}(r)\}\end{aligned}$$

Within this section it is assumed that $(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)$ is a CS-DP problem such that $\mathcal{E}$ is size-preserving. Furthermore, let $\varDelta^1 = \varDelta^1(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)$. The mapping $\mathcal{I}$ used in [13] needs to be modified as well. In particular, it needs to be possible to apply the mapping to non-terminating terms as well since terminating terms may contain non-terminating subterms in inactive positions.

Definition 56 ($\mathcal{I}^1$). *For any term $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathit{Th}}, \mathcal{V})$ define $\mathcal{I}^1(t)$ by*

- *$\mathcal{I}^1(x) = x$ if $x \in \mathcal{V}$*
- *$\mathcal{I}^1(f(t_1,\ldots,t_n)) = f(\mathcal{I}^1(t_1),\ldots,\mathcal{I}^1(t_n))$ if $f \in \varDelta^1$ or $f(t_1,\ldots,t_n)$ starts an infinite $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction*
- *$\mathcal{I}^1(t) = \mathcal{C}omp_{\mathrm{sort}(t)}(\mathcal{R}ed^1_{\mathcal{S}}(t) \cup \mathcal{R}ed^1_{\mathcal{R}}(t) \cup \mathcal{E}q^1_{\mathcal{E}}(t))$ if t is terminating and $t = f(t_1,\ldots,t_n)$ with $f \notin \varDelta^1$.*

Here, the sets $\mathcal{R}ed^1_{\mathcal{S}}(t)$, $\mathcal{R}ed^1_{\mathcal{R}}(t)$, and $\mathcal{E}q^1_{\mathcal{E}}(t)$ are defined as

$$\begin{aligned}\mathcal{R}ed^1_{\mathcal{S}}(t) &= \{\mathcal{I}^1(t') \mid t \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t'\} \\ \mathcal{R}ed^1_{\mathcal{R}}(t) &= \{\mathcal{I}^1(t') \mid t \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}} t'\} \\ \mathcal{E}q^1_{\mathcal{E}}(t) &= \{g(\mathcal{I}^1(t_1),\ldots,\mathcal{I}^1(t_m)) \mid t \sim_{\mathcal{E}} g(t_1,\ldots,t_m)\}\end{aligned}$$

For $s \in \{\texttt{univ},\texttt{base}\}$, let $\mathcal{C}omp_s(\{t\} \uplus M) = \Pi_s(t, \mathcal{C}omp_s(M))$ and $\mathcal{C}omp_s(\emptyset) = \bot_s$ where Π_s is a fresh function symbols with sort declaration $s \times s \to s$ and $\bot_s$ is a fresh variable of sort s. In order to make this definition unambiguous it is assumed that t is the minimal element of $\{t\} \uplus M$ w.r.t. some total well-founded order $>_{\mathcal{T}}$ on terms. For any substitution σ, define the substitution $\mathcal{I}^1(\sigma)$ by $\mathcal{I}^1(\sigma)(x) = \mathcal{I}^1(\sigma(x))$.

It is not obvious that $\mathcal{I}^1$ is indeed well-defined, i.e., that $\mathcal{I}^1(t)$ is a finite term for any term t. This turns out to be the case, however.

Lemma 57. *For any term t, the term $\mathcal{I}^1(t)$ is finite.*

The idea for the result in this section is now the same as in [13]: reductions using $\mathcal{R}$ are "simulated" by reductions that only use $\mathcal{R}(\varDelta^1) = \{l \to r[\![\varphi]\!] \in \mathcal{R} \mid \mathrm{root}(l) \in \varDelta^1\}$. For this it becomes necessary to introduce rewrite rules for the fresh function symbols $\Pi_{\texttt{univ}}$ and $\Pi_{\texttt{base}}$.

Definition 58 ($\mathcal{R}_\Pi$). *For the fresh function symbols $\Pi_{\texttt{univ}}$ and $\Pi_{\texttt{base}}$ from Definition 56, let $\mathcal{R}_\Pi = \{\Pi_s(x,y) \to x,\ \Pi_s(x,y) \to y \mid s \in \{\texttt{univ},\texttt{base}\}\}$.*

The following is a well-known simple property of $\mathcal{R}_\Pi$.

Lemma 59. *Let M be a set of terms such that all elements of M have the same sort. If $t \in M$ then $\mathcal{C}omp_{\mathrm{sort}(t)}(M) \to^+_{\mathcal{R}_\Pi} t$.*

Next, several properties of the mapping $\mathcal{I}^1$ are needed.

Lemma 60. *Let $s, t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ and let σ be a $\mathcal{T}h$-based substitution.*

1. *If $s \in \mathcal{T}(\Delta^1, \mathcal{V})$ then $\mathcal{I}^1(s\sigma) = s\mathcal{I}^1(\sigma)$.*
2. *If $s \rhd_{\not\mu} s'$ implies $s' \in \mathcal{T}(\Delta^1, \mathcal{V})$, then $\mathcal{I}^1(s\sigma) \to^*_{\mathcal{R}_\Pi} s\mathcal{I}^1(\sigma)$.*
3. *If $s \sim_{\mathcal{E}} t$ then $\mathcal{I}^1(s) \sim_{\mathcal{E}} \mathcal{I}^1(t)$.*
4. *If $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$ then $\mathcal{I}^1(s) \rightsquigarrow^*_1 \mathcal{I}^1(t)$,*
   where $\rightsquigarrow_1 \; = \; \to_{\mathcal{E}\backslash\mathcal{S}} \cup \to^+_{\mathcal{R}_\Pi}$.
5. *If s is terminating and $s \overset{\mathcal{S}}{\to}{}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$ then $\mathcal{I}^1(s) \rightsquigarrow^*_2 \mathcal{I}^1(t)$,*
   where $\rightsquigarrow_2 \; = \; \rightsquigarrow^*_1 \circ \sim_{\mathcal{E}} \circ \to^*_{\mathcal{R}_\Pi} \circ \to_{\mathcal{R}(\Delta^1),\mu} \cup \to^+_{\mathcal{R}_\Pi}$ such that the $\to_{\mathcal{R}(\Delta^1),\mu}$ step uses a $\mathcal{T}h$-based substitution that makes the instantiated constraint of the used rule $\mathcal{T}h$-valid.
6. *Let $s \in \mathcal{T}(\Delta^1, \mathcal{V})$ be terminating and let $t \rhd_{\not\mu} t'$ imply $t' \in \mathcal{T}(\Delta^1, \mathcal{V})$. If $s\sigma \overset{\mathcal{S}}{\to}{}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^!_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t\sigma$, then $s\mathcal{I}^1(\sigma) \rightsquigarrow^*_2 \circ \rightsquigarrow^*_1 \circ \sim_{\mathcal{E}} \circ \to^*_{\mathcal{R}_\Pi} t\mathcal{I}^1(\sigma)$.*

Using Lemma 60.6, soundness of the following CS-DP processor can be shown. This processor adapts the corresponding DP processor from [13] to the context-sensitive case.

Theorem 61 (CS-DP Processor Based on Function Dependencies). *Let* Proc *be the DP processor with* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) =$

- *$\{(\mathcal{P} - \mathcal{P}', \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, if $\mathcal{E}$ is size-preserving, $\Delta^1 = \Delta^1(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, $\mathcal{P}' \subseteq \mathcal{P}$, and either*
  - *there exists a reduction pair $(\gtrsim, \succ)$ that satisfies the conditions from Theorem 49 where $\mathcal{R}$ is replaced by $\mathcal{R}(\Delta^1) \cup \mathcal{R}_\Pi$, or*
  - *there exists a $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation that is applicable and satisfies the conditions from Theorem 52 where $\mathcal{R}$ is replaced by $\mathcal{R}(\Delta^1) \cup \mathcal{R}_\Pi$.*
- *$\{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, otherwise.*

Then Proc *is sound.*

Example 62. Using Theorem 61, it is no longer possible to falsely prove termination of the TRS from Example 54. Recall the following:

$$\begin{aligned} \mathsf{b} &\to \mathsf{c}(\mathsf{b}) \\ \mathsf{f}(\mathsf{c}(x), x) &\to \mathsf{f}(x, x) \\ \mathsf{f}^\sharp(\mathsf{c}(x), x) &\to \mathsf{f}^\sharp(x, x) \qquad (14) \end{aligned}$$

Then $\mathsf{b} \in \Delta^1(\{(14), \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, i.e., the rule $\mathsf{b} \to \mathsf{c}(\mathsf{b})$ needs to be considered when handling the CS-DP consisting of (14). But then no reduction pair satisfies the requirements. ◊

6.5 Function Dependencies for Strongly Conservative Systems

The CS-DP processor based on function dependencies introduced in the previous section differs from the corresponding DP processor for non-context-sensitive rewriting introduced in [13] by needing to consider the left-hand sides of the dependency pairs and rules. Additionally, all of $\mathcal{S}$ and $\mathcal{E}$ needs to be considered.

In this section, it is shown that the left-hand sides of $\mathcal{P}$ and $\mathcal{R}$ do not need to be considered if (certain subsets of) $\mathcal{P}$, $\mathcal{R}$ and $\mathcal{S}$ are *strongly conservative* [18].

Definition 63 (Strongly Conservative). *Given terms s, t and a CS-DP problem $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, the pair (s, t) is* strongly conservative *iff $\mathcal{V}^{\mu}(t) \subseteq \mathcal{V}^{\mu}(s)$ and $\mathcal{V}^{\mu}(s) \cap \mathcal{V}^{\not\mu}(s) = \mathcal{V}^{\mu}(t) \cap \mathcal{V}^{\not\mu}(t) = \emptyset$.*

A set of pairs of terms (i.e., rewrite rules or equations) is strongly conservative if all of its members are. Notice that $\mathcal{E}$ is always strongly conservative by Definition 18. For strongly conservative systems, the following definition of function dependencies can be used. The reason why *two* different relations $\overline{\blacktriangleright}^2$ and $\blacktriangleright^2$ are used is that reductions with $\mathcal{S}$ and $\mathcal{E}$ may also take place in inactive positions. Introducing two different relations then allows to only consider subsets of $\mathcal{S}$ and $\mathcal{E}$ as well, whereas the CS-DP processor introduced in the previous section needs to consider all of $\mathcal{S}$ and $\mathcal{E}$.

Definition 64 (Context-Sensitive Function Dependencies for Strongly Conservative Systems). *Let $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a DP problem where $\mathcal{E}$ is size-preserving. For any term t, let $\mathcal{F}^{\mu}_{\mathcal{S},\mathcal{E}}(t) = \mathcal{F}^{\mu}(t) \cup [(\mathcal{F}(\mathcal{S}) \cup \mathcal{F}(\mathcal{E})) \cap \mathcal{F}(t)]$.*

1. *For two symbols $f, h \in \mathcal{F}$ let $f \; \overline{\blacktriangleright}^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} \; h$ iff $f = h$ or there exists a symbol g with $g \; \overline{\blacktriangleright}^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} \; h$ and either*
   (a) a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ with $\mathrm{root}(l) = f$ and $g \in \mathcal{F}(r)$, or
   (b) a rule $l \to r \in \mathcal{S}$ with $\mathrm{root}(l) = f$ and $g \in \mathcal{F}(r)$, or
   (c) an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ with $\mathrm{root}(u) = f$ and $g \in \mathcal{F}(u \approx v)$.
   In the following, let
   $$\overline{\Delta}^2(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) = \mathcal{F}_{\mathit{Th}} \cup \bigcup_{s \to t[\![\varphi]\!] \in \mathcal{P}} \{g \mid f \; \overline{\blacktriangleright}^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} \; g \text{ for an } f \in \mathcal{F}(t)\}$$
2. *For two symbols $f, h \in \mathcal{F}$ let $f \blacktriangleright^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} h$ iff $f = h$ or there exists a symbol g with $g \blacktriangleright^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} h$ and either*
   (a) a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ with $\mathrm{root}(l) = f$ and $g \in \mathcal{F}^{\mu}_{\mathcal{S},\mathcal{E}}(r)$, or
   (b) a rule $l \to r \in \mathcal{S}$ with $\mathrm{root}(l) = f$ and $g \in \mathcal{F}(r)$, or
   (c) an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ with $\mathrm{root}(u) = f$ and $g \in \mathcal{F}(u \approx v)$.
   In the following, let
   $$\Delta^2(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) = \mathcal{F}_{\mathit{Th}} \cup \bigcup_{s \to t[\![\varphi]\!] \in \mathcal{P}} \{g \mid f \blacktriangleright^2_{(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)} g \text{ for an } f \in \mathcal{F}^{\mu}_{\mathcal{S},\mathcal{E}}(t)\}$$

Within this section it is assumed that $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ is a DP problem such that $\mathcal{E}$ is size-preserving. Furthermore, let $\Delta^2 = \Delta^2(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ and $\overline{\Delta}^2 = \overline{\Delta}^2(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$. Notice that $\Delta^2 \subseteq \overline{\Delta}^2$, but that these two sets agree on symbols from $\mathcal{F}(\mathcal{S}) \cup \mathcal{F}(\mathcal{E})$. Finally, it is assumed in the following that $\mathcal{R}(\Delta^2) = \{l \to r[\![\varphi]\!] \in \mathcal{R} \mid \text{root}(l) \in \Delta^2\}$ and $\mathcal{S}(\Delta^2) = \{l \to r[\![\in \mathcal{S} \mid \text{root}(l) \in \Delta^2\}$ are strongly conservative. Also, $\mathcal{E}(\Delta^2) = \{u \approx v \in \mathcal{E} \mid \text{root}(u) \in \Delta^2 \text{ or } \text{root}(v) \in \Delta^2\}$.

The mapping $\mathcal{I}^1$ from Definition 56 needs to be modified for the strongly conservative case. Since two relations $\overline{\blacktriangleright}^2$ and $\blacktriangleright^2$ are considered, *two* mappings $\overline{\mathcal{I}}^2$ and $\mathcal{I}^2$ are needed as well. Notice that $\overline{\mathcal{I}}^2$ is only used in order to simulate reductions with $\mathcal{S}$ and $\mathcal{E}$, but not reductions with $\mathcal{R}$.

Definition 65 ($\overline{\mathcal{I}}^2$ and $\mathcal{I}^2$).

1. *For any term $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ define $\overline{\mathcal{I}}^2(t)$ by*
   - $\overline{\mathcal{I}}^2(x) = x$ *if* $x \in \mathcal{V}$
   - $\overline{\mathcal{I}}^2(f(t_1, \ldots, t_n)) = f(\overline{\mathcal{I}}^2(t_1), \ldots, \overline{\mathcal{I}}^2(t_n))$ *if* $f \in \overline{\Delta}^2$
   - $\overline{\mathcal{I}}^2(t) = \mathcal{C}omp_{\text{sort}(t)}(\overline{\mathcal{R}ed}^2_{\mathcal{S}}(t) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(t))$ *if* $t = f(t_1, \ldots, t_n)$ *with* $f \notin \overline{\Delta}^2$.

   Here, the sets $\overline{\mathcal{R}ed}^2_{\mathcal{S}}(t)$ and $\overline{\mathcal{E}q}^2_{\mathcal{E}}(t)$ are defined as

$$\begin{aligned}\overline{\mathcal{R}ed}^2_{\mathcal{S}}(t) &= \{\overline{\mathcal{I}}^2(t') \mid t \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t'\} \\ \overline{\mathcal{E}q}^2_{\mathcal{E}}(t) &= \{g(\overline{\mathcal{I}}^2(t_1), \ldots, \overline{\mathcal{I}}^2(t_m)) \mid t \sim_{\mathcal{E}} g(t_1, \ldots, t_m)\}\end{aligned}$$

   For any substitution σ, define the substitution $\overline{\mathcal{I}}^2(\sigma)$ by letting $\overline{\mathcal{I}}^2(\sigma)(x) = \overline{\mathcal{I}}^2(\sigma(x))$.
2. *For any terminating term $t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ define $\mathcal{I}^2(t)$ by*
   - $\mathcal{I}^2(x) = x$ *if* $x \in \mathcal{V}$
   - $\mathcal{I}^2(f(t_1, \ldots, t_n)) = f(\overline{t_1}, \ldots, \overline{t_n})$ *if* $f \in \Delta^2$
   - $\mathcal{I}^2(t) = \mathcal{C}omp_{\text{sort}(t)}(\mathcal{R}ed^2_{\mathcal{S}}(t) \cup \mathcal{R}ed^2_{\mathcal{R}}(t) \cup \mathcal{E}q^2_{\mathcal{E}}(t) \cup \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t))$ *if it is the case that* $t = f(t_1, \ldots, t_n)$ *with* $f \notin \Delta^2$.

   Here, $\overline{t_i} = \mathcal{I}^2(t_i)$ if $i \in \mu(f)$ and $\overline{t_i} = \overline{\mathcal{I}}^2(t_i)$ otherwise. Moreover, the sets $\mathcal{R}ed^2_{\mathcal{S}}(t)$, $\mathcal{R}ed^2_{\mathcal{R}}(t)$, $\mathcal{E}q^2_{\mathcal{E}}(t)$, and $\overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t)$ are defined as

$$\begin{aligned}\mathcal{R}ed^2_{\mathcal{S}}(t) &= \{\mathcal{I}^2(t') \mid t \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t'\} \\ \mathcal{R}ed^2_{\mathcal{R}}(t) &= \{\mathcal{I}^2(t') \mid t \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}} t'\} \\ \mathcal{E}q^2_{\mathcal{E}}(t) &= \{g(\overline{t_1}, \ldots, \overline{t_m}) \mid t \sim_{\mathcal{E}} g(t_1, \ldots, t_m)\} \\ \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t) &= \{\overline{\mathcal{I}}^2(s) \mid t \sim_{\mathcal{E}} s\}\end{aligned}$$

   Again, $\overline{t_i} = \mathcal{I}^2(t_i)$ if $i \in \mu(g)$ and $\overline{t_i} = \overline{\mathcal{I}}^2(t_i)$ otherwise. For a terminating substitution σ, let $[t, \sigma]$ be the term that results from t by replacing all occurrences of $x \in \mathcal{V}(t)$ in active positions of t by $\mathcal{I}^2(\sigma(x))$ and all occurrences of $x \in \mathcal{V}(t)$ in inactive positions of t by $\overline{\mathcal{I}}^2(\sigma(x))$.

Again, it first needs to be shown that these mappings are well defined.

Lemma 66. *For any term t, the term $\overline{\mathcal{I}}^2(t)$ is finite. If t is terminating, then $\mathcal{I}^2(t)$ is finite.*

Next, several properties of the mappings $\overline{\mathcal{I}}^2$ and $\mathcal{I}^2$ are shown. This is in analogy to Lemma 60.

Lemma 67. *Let $s, t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ and let σ be a $\mathcal{T}h$-based substitution such that $s, t, s\sigma$ are terminating.*

1. *If $s \in \mathcal{T}(\overline{\Delta}^2, \mathcal{V})$ then $\overline{\mathcal{I}}^2(s\sigma) = s\overline{\mathcal{I}}^2(\sigma)$.*
2. *If $s \in \mathcal{T}(\overline{\Delta}^2, \mathcal{V})$ such that $\mathcal{F}^\mu(s) \subseteq \Delta^2$, then $\mathcal{I}^2(s\sigma) = [s, \sigma]$.*
3. *$\overline{\mathcal{I}}^2(s\sigma) \to^*_{\mathcal{R}_\Pi} s\mathcal{I}^2(\sigma)$.*
4. *$\mathcal{I}^2(s\sigma) \to^*_{\mathcal{R}_\Pi} [s, \sigma]$.*
5. *$\mathcal{I}^2(s) \to^*_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(s)$.*
6. *If $s \sim_{\mathcal{E}} t$ then $\overline{\mathcal{I}}^2(s) \sim_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(t)$.*
7. *If $s \sim_{\mathcal{E}} t$ then $\mathcal{I}^2(s) \sim_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t)$.*
8. *If $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$ then $\overline{\mathcal{I}}^2(s) \leadsto^*_1 \overline{\mathcal{I}}^2(t)$,*
   where $\leadsto_1 \,=\, \sim_{\mathcal{E}(\Delta^2)} \circ \to^*_{\mathcal{R}_\Pi} \circ \to_{\mathcal{S}(\Delta^2)} \cup \to^+_{\mathcal{R}_\Pi}$.
9. *If $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$ then $\mathcal{I}^2(s) \leadsto^*_1 \mathcal{I}^2(t)$.*
10. *If $s \overset{\mathcal{S}}{\to}{}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$ then $\mathcal{I}^2(s) \leadsto^*_2 \mathcal{I}^2(t)$,*
    where $\leadsto_2 \,=\, \leadsto^*_1 \circ \sim_{\mathcal{E}(\Delta^2)} \circ \to^*_{\mathcal{R}_\Pi} \circ \to_{\mathcal{R}(\Delta^2),\mu} \cup \to^+_{\mathcal{R}_\Pi}$ such that the $\to_{\mathcal{R}(\Delta^2),\mu}$ step uses a $\mathcal{T}h$-based substitution that makes the instantiated constraint of the used rule $\mathcal{T}h$-valid.
11. *If $s \overset{\mathcal{S}}{\to}{}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^!_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t\sigma$, then $\mathcal{I}^2(s) \leadsto^*_2 \circ \leadsto^*_1 \circ \sim_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t)$.*

As before, Lemma 67.11 allows to show soundness of the following CS-DP processor. In contrast to the CS-DP processor introduced in the previous section, it is only applicable if $\mathcal{R}(\Delta^2)$, $\mathcal{S}(\Delta^2)$, and $\mathcal{P}$ are strongly conservative.

Theorem 68 (CS-DP Processor Based on Function Dependencies for Strongly Conservative Systems). *Let* Proc *be the CS-DP processor with* $\mathsf{Proc}(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu) =$

- *$\{(\mathcal{P} - \mathcal{P}', \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, if $\mathcal{E}$ is size-preserving, $\Delta^2 = \Delta^2(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$, all of $\mathcal{R}(\Delta^2)$, $\mathcal{S}(\Delta^2)$, and $\mathcal{P}$ are strongly conservative, $\mathcal{P}' \subseteq \mathcal{P}$, and either*
  - *there exists a reduction pair $(\gtrsim, \succ)$ that satisfies the conditions from Theorem 49 where $\mathcal{R}$ is replaced by $\mathcal{R}(\Delta^2) \cup \mathcal{R}_\Pi$, $\mathcal{S}$ is replaced by $\mathcal{S}(\Delta^2)$, and $\mathcal{E}$ is replaced by $\mathcal{E}(\Delta^2)$, or*
  - *there exists a $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation that is applicable and satisfies the conditions from Theorem 52 where $\mathcal{R}$ is replaced by $\mathcal{R}(\Delta^2) \cup \mathcal{R}_\Pi$, $\mathcal{S}$ is replaced by $\mathcal{S}(\Delta^2)$, and $\mathcal{E}$ is replaced by $\mathcal{E}(\Delta^2)$.*
- *$\{(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)\}$, otherwise.*

Then Proc *is sound.*

Example 69. The CS-DP processor of Theorem 68 allows to finish the termination proof of the running example. Recall the following dependency pair from Example 53:

$$\mathsf{take}^\sharp(x) \to \mathsf{take}^\sharp(x - 1) \; [\![x > \mathsf{0}]\!] \tag{13}$$

Then this dependency pair is strongly conservative, and the same holds true for $\mathcal{R}(\Delta^2) = \emptyset$ and $\mathcal{S}(\Delta^2) = \mathcal{S}_{\mathsf{base}}$. It is thus possible to apply the $\mathcal{T}h_{\mathbb{Z}}$-polynomial interpretation with $c_{\mathcal{P}ol} = 0$ and $\mathcal{P}ol(\mathsf{take}^\sharp) = x_1$. ◊

7 Evaluation and Conclusions

This paper has presented a generalization of the constrained equational rewrite systems (CERSs) introduced in [13]. Then, context-sensitive rewriting strategies for these generalized CERSs have be investigated. The main interest has been in the automated termination analysis for such context-sensitive CERSs. To this extent, a dependency pair framework for CS-CERSs has been developed, taking the recent method of [1] for ordinary context-sensitive TRSs as a starting point. Similar to [1], this approach allows for relatively easy adaptations of methods that were originally developed for non-context-sensitive rewriting to the context-sensitive case. This has been demonstrated in Section 6 for several of the most important techniques from [13].

The techniques presented in this paper have been fully implemented in the termination prover APRoVE [16] for built-in integers. While most of the implementation is relatively straightforward, the automatic generation of suitable $\mathcal{Th}_{\mathbb{Z}}$-polynomial interpretations is non-trivial. Details on this can be found in [11]. In order to evaluate the effectiveness of the approach on "typical" algorithms, the implementation has been evaluated on a large collection of (both context-sensitive and non-context-sensitive) examples. Most of these examples stem from the *Termination Problem Data Base* [28], suitably adapted to make use of built-in integers and/or collection data structures. The collection furthermore contains more than 40 examples that were obtained by encoding programs from the literature on termination proving of imperative programs into CERSs. Within a time limit of 60 seconds for each example, our implementation succeeds on 93.1% of the examples. A detailed empirical evaluation, including all termination proofs generated by AProVE, is available at `http://www.cs.unm.edu/~spf/tdps/`.

A Proofs

Proof (Lemma 14).

1. Since $\mathcal{E}$ is size-preserving, the $\mathcal{E}$-equivalence classes are finite. Furthermore, the $\mathcal{E}$-equivalence class of a given term s can be computed effectively using the equations from $\mathcal{E}$. In order to decide whether $s \sim_{\mathcal{E}} t$ it then suffices to check whether t is in the $\mathcal{E}$-equivalence class of s.
2. It needs to be decided whether there exist a rewrite rule $l \to r \in \mathcal{S}$, a position $p \in \mathcal{P}os(s)$, and a substitution σ such that $s|_p \sim_{\mathcal{E}} l\sigma$. Since $\mathcal{S}$ and $\mathcal{P}os(s)$ are finite, it suffices to consider a single rule and a single position, without loss of generality consider $p = \Lambda$. Thus, it needs to be decided whether there exists a substitution σ such that $s \sim_{\mathcal{E}} l\sigma$. In order to check this, it suffices to check whether there exists a term s' in the $\mathcal{E}$-equivalence class of s such that $s' = l\sigma$ for some substitution σ. But this is just syntactic matching, which is well-known to be decidable. Once such a substitution σ has been found it is easily possible to compute a term t with $s \to_{\mathcal{E}\backslash\mathcal{S}} t$.
3. It needs to decided whether there exist a constrained rewrite rule $l \to r[\![\varphi]\!] \in \mathcal{R}$, a position $p \in \mathcal{P}os(s)$, and a Th-based substitution σ such that $s|_p \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$ and $\varphi\sigma$ is Th-valid. Since $\mathcal{R}$ and $\mathcal{P}os(s)$ are finite, it suffices to consider a single rule and a single position, without loss of generality consider $p = \Lambda$. Thus, it needs to decided whether there exists a substitution σ such that $s \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$. First, notice that a term s' with $s \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} s'$ can be computed by Lemma 14.2. It then suffices to check whether there exists a term s'' in the $\mathcal{E}$-equivalence class of s' such that $s'' = l\sigma$ and $\varphi\sigma$ is Th-valid for some Th-based substitution σ. Candidate substitutions σ can be computed using syntactic matching, and it is easy to check whether such a substitution σ is Th-based. Th-validity of $\varphi\sigma$ can then be decided due to the assumption on Th. Once such a substitution σ has been found it is easily possible to compute a term t with $s \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R}} t$. □

Proof (Lemma 21).

1. Let $C[f(s^*)] \sim_{\mathcal{E}} t$, i.e., there exist terms $t_0, \ldots, t_n$ with $n \geq 0$ such that $C[f(s^*)] = t_0 \vdash\!\dashv_{\mathcal{E}} t_1 \vdash\!\dashv_{\mathcal{E}} \ldots \vdash\!\dashv_{\mathcal{E}} t_n = t$. The claim is proved by induction on n. If $n = 0$ then $C[f(s^*)] = t$ and the claim is obvious.
   If $n > 0$, the inductive hypothesis implies $t_{n-1} = C''[f(s''^*)]$ with $C'' \sim_{\mathcal{E}} C$ and $s''^* \sim_{\mathcal{E}} s^*$, where additionally C'' is active. Since $t_{n-1} \vdash\!\dashv_{\mathcal{E}} t_n$, there exists an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ such that $t_{n-1}|_p = u\sigma$ and $t_n = t_{n-1}[v\sigma]_p$ for some position p and some substitution σ. Let q be the position with $t_{n-1}|_q = f(s''^*)$, i.e., $C''|_q = \Box$. Now perform a case analysis on the relationship between the positions p and q.
   Case 1: $p \perp q$. In this case, obtain $t_n = t_{n-1}[v\sigma]_p = (C''[f(s''^*)])[v\sigma]_p = (C''[v\sigma]_p)[f(s''^*)]$ with $C''[v\sigma]_p \sim_{\mathcal{E}} C''[u\sigma]_p = C''$ and $C''[v\sigma]_p$ is active.
   Case 2: $p = q.q'$ for some position $q' \neq \Lambda$. In this case, $t_n = t_{n-1}[v\sigma]_p = (C''[f(s''^*)])[v\sigma]_{q.q'} = C'[f(s''^*)[v\sigma]_{q'}]$. Since $q' \neq \Lambda$, the position q' can be

written as $q' = i.q''$ for some $1 \leq i \leq \text{arity}(f)$ and some position q''. Then $s'_j = s''_j$ if $i \neq j$ and $s'_i = s''_i[v\sigma]_{q''} \sim_{\mathcal{E}} s''_i[u\sigma]_{q''} = s''_i$, i.e., $s'^* \sim_{\mathcal{E}} s''^*$.
Case 3: $q = p.p'$ for some position p' (possibly $p' = \Lambda$). Since $f \notin \mathcal{F}(\mathcal{E})$, the position p' can be written as $p' = p'_1.p'_2$ such that $u|_{p'_1}$ is a variable x and $x\sigma|_{p'_2} = f(s''^*)$. Since the equation $u \approx v$ (or $v \approx u$) is i.u.v., there exists a unique position p''_1 in v such that $v|_{p''_1} = x$. This implies $v\sigma|_{p''_1.p'_2} = x\sigma|_{p'_2} = f(s''^*)$. Define the substitution σ' by $\sigma'(y) = \sigma(y)$ for $y \neq x$ and $\sigma'(x) = x\sigma[\Box]_{p'_2}$. Let $C' = (t_{n-1}[v\sigma]_p)[\Box]_{p.p''_1.p'_2} = t_{n-1}[v\sigma[\Box]_{p''_1.p'_2}]_p = t_{n-1}[v\sigma']_p \sim_{\mathcal{E}} t_{n-1}[u\sigma']_p = C''$. Since q is active in C'', the position p'_1 is active in u and Definition 18.2a implies that p''_1 is active in v. Therefore, C' is active as well. Thus, $t_n = t_{n-1}[v\sigma]_p = C'[f(s''^*)]$ and the claim follows.

2. Assume $s' \sim_{\mathcal{E}} s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t$. This means that $s = C[f(u^*)]$ for some active context C with $f(u^*) \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$ for some constrained rewrite rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ and some Th-based substitution σ such that $\varphi\sigma$ is Th-valid and $t = C[r\sigma]$. Since $s \sim_{\mathcal{E}} s'$, Lemma 21.1 implies $s' = C'[f(u'^*)]$ for some active context C' with $C' \sim_{\mathcal{E}} C$ and $u'^* \sim_{\mathcal{E}} u^*$. Therefore, $f(u'^*) \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma$ since $\to_{\mathcal{E}\backslash\mathcal{S}}$ is $\mathcal{E}$-convergent and strongly $\mathcal{E}$-coherent and the substitution σ can be used to rewrite $s' = C'[f(u'^*)]$ to $t' = C'[r\sigma] \sim_{\mathcal{E}} C[r\sigma] = t$.
3. Let $s \to_{\mathcal{E}\backslash\mathcal{S}} t \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} u$, i.e., there exist positions $p_1 \in \mathcal{P}os(s)$ and $p_2 \in \mathcal{P}os^{\mu}(t)$, rules $l_1 \to r_1 \in \mathcal{S}$ and $l_2 \to r_2[\![\varphi_2]\!] \in \mathcal{R}$, a substitution σ_1, and a Th-based substitution σ_2 such that
   (a) $s|_{p_1} \sim_{\mathcal{E}} l_1\sigma_1$ and $t = s[r_1\sigma_1]_{p_1}$, and
   (b) $t|_{p_2} \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_2\sigma_2$, the instantiated Th-constraint $\varphi_2\sigma_2$ is Th-valid, and $u = t[r_2\sigma_2]_{p_2}$.

   Perform a case distinction on the relationship between p_1 and p_2.
   Case 1: $p_1 \perp p_2$. In this case $s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} u$ is immediate.
   Case 2: $p_1 = p_2.q$ for some position $q \neq \Lambda$. In this case $s|_{p_2} = f(s^*)$, $t|_{p_2} = f(t^*)$, and $s^* \to_{\mathcal{E}\backslash\mathcal{S}} t^*$. Therefore, $f(s^*) \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_2\sigma_2$ since $\to_{\mathcal{E}\backslash\mathcal{S}}$ is $\mathcal{E}$-convergent and strongly $\mathcal{E}$-coherent and $f(t^*) \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_2\sigma_2$. Thus, $s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s[r_2\sigma_2]_{p_2} = t[r_2\sigma_2]_{p_2} = u$.
   Case 3: $p_2 = p_1.q$ for some position q, possibly $q = \Lambda$. Since r_1 does not contain symbols from $\mathcal{D}(\mathcal{R})$, there exists a position $q_1 \in \mathcal{P}os^{\mu}(r_1)$ such that $r_1|_{q_1} = x$ is a variable and $q = q_1.q_2$ for some position q_2. Define the substitution σ'_1 to behave like σ_1, with the exception that $\sigma'_1(x) = \sigma_1(x)[r_2\sigma_2]_{q_2}$. Then $l_1\sigma_1 \overset{\mathcal{S}}{\to}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} l_1\sigma'_1$ by Definition 18.3a and thus $s|_{p_1} \overset{\mathcal{S}}{\to}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}} l_1\sigma'_1$ by Lemma 21.2 since $s|_{p_1} \sim_{\mathcal{E}} l_1\sigma_1$. Thus, $s|_{p_1} \overset{\mathcal{S}}{\to}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} r_1\sigma'$. Since r_1 is linear, $s[r_1\sigma'_1]_{p_1} = t[r_1\sigma'_1]_{p_1} = u$, i.e., $s \overset{\mathcal{S}}{\to}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} u$. □

Proof (Corollary 22).

1. Assume s starts an infinite $\overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction

$$s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_1 \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_2 \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_3 \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \cdots$$

Using Lemma 21.2 this implies

$$t \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_1 \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_2 \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_3 \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \cdots$$

where $s_i \sim_{\mathcal{E}} t_i$, i.e., t starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction as well. The other direction is shown the same way.

2. Let $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ and assume that t starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction. Using the inclusion $\to_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\to}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^{=}_{\mathcal{E}\backslash\mathcal{S}}$ from Lemma 21.3 repeatedly yields an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction starting with s. □

Proof (Lemma 33).

1. If $t \sim_{\mathcal{E}} t'$ and $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, then t' starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction by Corollary 22.1. Notice that $\mathrm{root}(t) \in \mathcal{D}(\mathcal{R})$, which implies $t \overset{>\Lambda}{\sim}_{\mathcal{E}} t'$. Therefore, $t = f(t_1, \ldots, t_n)$ and $t' = f(t'_1, \ldots, t'_n)$ where $t_i \sim_{\mathcal{E}} t'_i$ for all $1 \leq i \leq n$. Assume that $t' \rhd_{\mu} s$ where s starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction. Then $t'_i \unrhd_{\mu} s$ for some $1 \leq i \leq n$ and t'_i starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction as well. But then t_i starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction by Corollary 22.1, contradicting $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.
2. By the assumption, t' starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction. Notice that $\mathrm{root}(t) \in \mathcal{D}(\mathcal{R})$ and thus $t \xrightarrow{>\Lambda}^{*}_{\mathcal{E}\backslash\mathcal{S}} t'$. Therefore, $t = f(t_1, \ldots, t_n)$ and $t' = f(t'_1, \ldots, t'_n)$ where $t_i \to^{*}_{\mathcal{E}\backslash\mathcal{S}} t'_i$ for all $1 \leq i \leq n$. Assume that $t' \rhd_{\mu} s$ where s starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction. Then $t'_i \unrhd_{\mu} s$ for some $1 \leq i \leq n$ and t'_i starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction as well. But then t_i starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction by Corollary 22.2, contradicting $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. □

Proof (Lemma 34).

1. If $C[t] \vdash\!\dashv_{\mathcal{E}} s$ at a position $p \in \mathcal{P}os(C)$, then there exists a substitution σ and an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ such that $C[t]|_p = u\sigma$ and $s = C[t][v\sigma]_p$. Proceed by induction on p.
   If $p = \Lambda$, then, since $\mathrm{root}(t) \in \mathcal{D}(\mathcal{R})$, there is a variable $x \in \mathcal{V}(u)$ with $u \unrhd_{\mu} x$ and $x\sigma \unrhd_{\mu} t$. Now Definition 18.2a implies $v \unrhd_{\mu} x$. Let q be the position of x in v and let q' be the position of t in $x\sigma$. Then $s = v\sigma = C'[t]$ with $C' = r\sigma[\Box]_{q.q'}$, where C' is a hiding context.
   If $p \neq \Lambda$, then $C[t] = f(t_1, \ldots, t_{i-1}, C_1[t], t_{i+1}, \ldots, t_n)$ for a hiding context C_1. If p is inside one of the t_j, then $s = f(t'_1, \ldots, t'_{i-1}, C_1[t], t'_{i+1}, \ldots, t'_n) = C'[t]$ where C' is clearly a hiding context. Otherwise, $C_1[t] \vdash\!\dashv_{\mathcal{E}} s'$ and $s = f(t_1, \ldots, t_{i-1}, s', t_{i+1}, \ldots, t_n)$. The inductive hypothesis implies $s' = C'_1[t]$ for a hiding context C'_1, which implies $s = C'[t]$ for a hiding context C'.
2. If $C[t] \to_{\mathcal{S}} s$ at a position $p \in \mathcal{P}os(C)$, then there exists a substitution σ and a rule $l \to r \in \mathcal{S}$ such that $C[t]|_p = l\sigma$ and $s = C[t][r\sigma]_p$. Proceed by induction on p.

If $p = \Lambda$, then, since $\mathrm{root}(t) \in \mathcal{D}(\mathcal{R})$, there is a variable $x \in \mathcal{V}(l)$ with $l \trianglerighteq_\mu x$ and $x\sigma \trianglerighteq_\mu t$. By the assumption, $r \trianglerighteq_\mu x$. Let q be the position of x in r and let q' be the position of t in $x\sigma$. Then $s = r\sigma = C'[t]$ with $C' = r\sigma[\Box]_{q.q'}$, where C' is a hiding context.
If $p \neq \Lambda$, then $C[t] = f(t_1, \ldots, t_{i-1}, C_1[t], t_{i+1}, \ldots, t_n)$ for a hiding context C_1. If p is inside one of the t_j, then $s = f(t'_1, \ldots, t'_{i-1}, C_1[t], t'_{i+1}, \ldots, t'_n) = C'[t]$ where C' is clearly a hiding context. Otherwise, $C_1[t] \rightarrow_{\mathcal{S}} s'$ and $s = f(t_1, \ldots, t_{i-1}, s', t_{i+1}, \ldots, t_n)$. The inductive hypothesis implies $s' = C'_1[t]$ for a hiding context C'_1, which implies $s = C'[t]$ for a hiding context C'. □

Proof (Lemma 36.1). Let $w \rhd_{\not\mu} s \trianglerighteq_\mu t$ with $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. Thus $v \rhd_{\not\mu} s \trianglerighteq_\mu t$ since $v \trianglerighteq_\mu w \rhd_{\not\mu} s$. The claim is shown by performing a case distinction on whether s is a subterm of u.

1. $u \rhd s$.
   (a) If $u \rhd_\mu s$, then $u \rhd_\mu s \trianglerighteq_\mu t$ contradicts $u \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ since $u \rhd_\mu t$ with $t \in \mathcal{M}^\infty_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.
   (b) Otherwise, $u \rhd_{\not\mu} s$. Then $u \rhd_{\not\mu} s \trianglerighteq_\mu t$ and the claim follows from the hiding property of u.
2. $u \not\rhd s$.
   Then $u|_p = l\sigma$ and $v = u[r\sigma]_p$ for a position $p \in \mathcal{P}os(u)$, a substitution σ, and an equation $l \approx r$ (or $r \approx l$) in $\mathcal{E}$. Now $l\sigma \not\rhd s$ since $u \not\rhd s$ and, in particular, $\sigma(x) \not\rhd s$ for all $x \in \mathcal{V}(l)$. Also, u does not contain s as a subterm at a position $\widehat{p}$ with $p \perp \widehat{p}$.
   (a) First, assume that $r\sigma \trianglerighteq s$. Since $\mathrm{root}(t) \notin \mathcal{F}(\mathcal{E})$, $\sigma(x) \trianglerighteq_\mu t$ for some variable $x \in \mathcal{V}(r)$. Let q be the outermost position in v above s that satisfies $v|_q \trianglerighteq_\mu s$.
      (i) First, assume that q is above p or $q = p$. Since $r\sigma \trianglerighteq_\mu s \trianglerighteq_\mu t$, Definition 18.2a implies $l\sigma \trianglerighteq_\mu t$, i.e., $u \rhd_{\not\mu} l\sigma \trianglerighteq_\mu t$. From the hiding property of u, $l\sigma = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^*_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Then $r\sigma = C'[t]$ where C' is a hiding context by Lemma 34.1. Now $r\sigma \trianglerighteq_\mu s$ implies $s = C''[t]$, where C'' is a hiding context.
      (ii) Otherwise, $q = p.p'$ for a position $p' \neq \Lambda$ such that $r|_{p'} \notin \mathcal{V}$. Then $r|_{p'} \rhd_\mu x$, and Definition 18.2b (or Definition 18.2c) implies $l \rhd_{\not\mu} l' \trianglerighteq_\mu x$ for some non-variable subterm l' of l. Therefore, $u \rhd_{\not\mu} l'\sigma \trianglerighteq_\mu t$ and the hiding property of u implies $l'\sigma = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^*_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Then, $r|_{p'}\sigma = C'[t]$ for some hiding context C' and $r\sigma|_{p'} \trianglerighteq_\mu s$ implies $s = C''[t]$, where C'' is a hiding context.
   (b) If $r\sigma \not\trianglerighteq s$, then $s \rhd r\sigma$, i.e., $s|_{q_1} = r\sigma$ for some position q_1. In this case, $u \rhd_{\not\mu} s[l\sigma]_{q_1}$.
      (i) If t occurs above q_1, then $s|_{q_2} = t$ and $q_1 = q_2.q_3$ for some position $q_3 \neq \Lambda$. Thus, $u \rhd_{\not\mu} s[l\sigma]_{q_1} \rhd_\mu t[l\sigma]_{q_3}$ and the hiding property of u gives $s[l\sigma]_{q_1} = C[t[l\sigma]_{q_3}]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^*_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t[l\sigma]_{q_3}$.

Thus $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$ since $t[l\sigma]_{q_3} \overset{>\Lambda}{\sim}_{\mathcal{E}} t[r\sigma]_{q_3} = t$. Furthermore, $s = s[r\sigma]_{q_1} = C[t]$.

(ii) Otherwise, $\sigma(x) \rhd_{\mu} t$ for some variable $x \in \mathcal{V}(r)$. In this case, $u \rhd_{\not\mu} s[l\sigma]_{q_1} \unrhd_{\mu} t$ and the hiding property of u implies $s[l\sigma]_{q_1} = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Thus, $s = s[r\sigma]_{q_1} = C'[t]$ for some context C'. The context C' is hiding by Lemma 34.1. □

Proof (Lemma 36.2). If $u \to_{\mathcal{E}\backslash\mathcal{S}} v \unrhd_{\mu} w$, then $u \sim_{\mathcal{E}} u' \to_{\mathcal{S}} v \unrhd_{\mu} w$. By Lemma 33, $u' \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, and Lemma 36.1 implies that u' has the hiding property. Let $w \rhd_{\not\mu} s \unrhd_{\mu} t$ with $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. Thus $v \rhd_{\not\mu} s \unrhd_{\mu} t$ since $v \unrhd_{\mu} w \rhd_{\not\mu} s$. Perform a case distinction on whether s is a subterm of u'.

1. $u' \rhd s$.

(a) If $u' \rhd_{\mu} s$, then $u' \rhd_{\mu} s \unrhd_{\mu} t$ contradicts $u' \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ since $u' \rhd_{\mu} t$ with $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.

(b) Otherwise, $u' \rhd_{\not\mu} s$. Then $u' \rhd_{\not\mu} s \unrhd_{\mu} t$ and the claim follows from the hiding property of u'.

2. $u' \not\rhd s$.

Then $u'|_p = l\sigma$ and $v = u'[r\sigma]_p$ for a position $p \in \mathcal{P}os(u')$, a substitution σ, and a rule $l \to r \in \mathcal{S}$. Now $l\sigma \not\rhd s$ since $u' \not\rhd s$ and, in particular, $\sigma(x) \not\rhd s$ for all $x \in \mathcal{V}(l)$. Also, u' does not contain s as a subterm at a position $\widehat{p}$ with $p \perp \widehat{p}$.

(a) First, assume that $r\sigma \unrhd s$. Since $\mathrm{root}(t) \notin \mathcal{F}(\mathcal{E})$, $\sigma(x) \unrhd_{\mu} t$ for some variable $x \in \mathcal{V}(r)$. Let q be the outermost position in v above s that satisfies $v|_q \unrhd_{\mu} s$.

(i) First, assume that q is above p or $q = p$. Since $r\sigma \unrhd_{\mu} s \unrhd_{\mu} t$, Definition 18.3a implies $l\sigma \unrhd_{\mu} t$, i.e., $u \rhd_{\not\mu} l\sigma \unrhd_{\mu} t$. From the hiding property of u', $l\sigma = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Then $r\sigma = C'[t]$ where C' is a hiding context by Lemma 34.2. Now $r\sigma \unrhd_{\mu} s$ implies $s = C''[t]$, where C'' is a hiding context.

(ii) Otherwise, $q = p.p'$ for a position $p' \neq \Lambda$ such that $r|_{p'} \notin \mathcal{V}$. Then $r|_{p'} \rhd_{\mu} x$, and Definition 18.3b implies $l \rhd_{\not\mu} l' \unrhd_{\mu} x$ for some non-variable subterm l' of l. Therefore, $u' \rhd_{\not\mu} l'\sigma \unrhd_{\mu} t$ and the hiding property of u' implies $l'\sigma = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Then, $r|_{p'}\sigma = C'[t]$ for some hiding context C' and $r\sigma|_{p'} \unrhd_{\mu} s$ implies $s = C''[t]$, where C'' is a hiding context.

(b) If $r\sigma \ntrianglerighteq s$, then $s \rhd r\sigma$, i.e., $s|_{q_1} = r\sigma$ for some position q_1. In this case, $u' \rhd_{\not\mu} s[l\sigma]_{q_1}$.

(i) If t occurs above q_1, then $s|_{q_2} = t$ and $q_1 = q_2.q_3$ for some position $q_3 \neq \Lambda$. Thus, $u' \rhd_{\not\mu} s[l\sigma]_{q_1} \rhd_{\mu} t[l\sigma]_{q_3}$. Now the hiding property of u' gives $s[l\sigma]_{q_1} = C[t[l\sigma]_{q_3}]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t[l\sigma]_{q_3}$. Thus $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$ since $t[l\sigma]_{q_3} \to^{>\Lambda}_{\mathcal{S}} t[r\sigma]_{q_3} = t$ and $\to_{\mathcal{E}\backslash\mathcal{S}}$ is strongly $\mathcal{E}$-coherent by Definition 9. Furthermore, $s = s[r\sigma]_{q_1} = C[t]$.

(ii) Otherwise, $\sigma(x) \rhd_{\mu} t$ for some variable $x \in \mathcal{V}(r)$. In this case, $u' \rhd_{\not\mu} s[l\sigma]_{q_1} \unrhd_{\mu} t$ and the hiding property of u' implies $s[l\sigma]_{q_1} = C[t]$ for a hiding context C and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Thus, $s = s[r\sigma]_{q_1} = C'[t]$ for some context C'. The context C' is hiding by Lemma 34.2. □

Proof (Lemma 36.3). If $u \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} v \unrhd_{\mu} w$, then $u = C[u']$ for an active context C such that $u' \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma \to_{\mathcal{R}} r\sigma$ for a constrained rewrite rule $l \to r\llbracket\varphi\rrbracket \in \mathcal{R}$ and a Th-based substitution σ. Since $C[l\sigma]$ starts an infinite $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction, Lemma 33 implies $C[l\sigma] \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. Also, Lemmas 36.1 and 36.2 imply that $C[l\sigma]$ has the hiding property. Let $w \rhd_{\not\mu} s \unrhd_{\mu} t$ with $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. Then $v \rhd_{\not\mu} s \unrhd_{\mu} t$ since $v \unrhd_{\mu} w \rhd_{\not\mu} s$. Perform a case distinction on whether s is a subterm of $C[l\sigma]$.

1. $C[l\sigma] \rhd s$.
   (a) If $C[l\sigma] \rhd_{\mu} s$, then $C[l\sigma] \rhd_{\mu} s \unrhd_{\mu} t$ contradicts $C[l\sigma] \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ since $C[l\sigma] \rhd_{\mu} t$ with $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.
   (b) Otherwise, $C[l\sigma] \rhd_{\not\mu} s$. Then $C[l\sigma] \rhd_{\not\mu} s \unrhd_{\mu} t$ and the claim follows from the hiding property of $C[l\sigma]$.
2. $C[l\sigma] \not\rhd s$.
   First, notice that $v = C[r\sigma]$. Since $C[l\sigma] \not\rhd s$, $C \not\rhd s$, $l\sigma \not\rhd s$ and, in particular, $\sigma(x) \not\rhd s$ for all $x \in \mathcal{V}(l)$. Finally, the root of s in v cannot be above □ in C since these positions are active. Hence, $v \rhd_{\not\mu} s$ implies $r\sigma \rhd_{\not\mu} s$ such that s is an instance of a non-variable subterm of r occurring at a position from $\mathcal{P}os^{\not\mu}(r)$, i.e., $r \rhd_{\not\mu} r'$ with $r' \notin \mathcal{V}$ such that $r'\sigma = s$ and $s|_p = t$ for an active position $p \in \mathcal{P}os^{\mu}(s)$.
   (a) First, assume $p \in \mathcal{P}os^{\mu}(r')$ and $r'|_p \notin \mathcal{V}$. Then $r'|_p$ is a hidden term and $t = s|_p = r'\sigma|_p = r'|_p\sigma$ is an instance of a hidden term. Furthermore, $C' = r'[\Box]_p$ is a hiding context, which implies that $C'\sigma$ is a hiding context as well. Thus, $s = r'\sigma = C'[r'|_p]\sigma = C'\sigma[r'|_p\sigma]$ and the claim follows.
   (b) Otherwise, $p = p_1.p_2$ such that $r'|_{p_1} = x$ for a variable $x \in \mathcal{V}$. In this case, $C' = r'[\Box]_{p_1}$ is a hiding context. Now $l\sigma \rhd t$ since $x\sigma \unrhd_{\mu} t$. If $l\sigma \rhd_{\mu} t$, then $C[l\sigma] \unrhd_{\mu} l\sigma \rhd_{\mu} x\sigma \unrhd_{\mu} t$ with $t \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, contradicting $C[l\sigma] \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$. Thus, $l\sigma \rhd_{\not\mu} x\sigma$ and $C[l\sigma] \rhd_{\not\mu} x\sigma \unrhd_{\mu} t$ and the hiding property of $C[l\sigma]$ implies $x\sigma = \widehat{C}[t]$ for a hiding context $\widehat{C}$ and there exists an instance t' of a hidden term such that $t' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t$. Let $C'' = C'\sigma[\widehat{C}]$. Then the context C'' is hiding and $s = r'\sigma = r'\sigma[x\sigma[t]_{p_2}]_{p_1} = C'\sigma[\widehat{C}[t]] = C''[t]$, thus establishing the claim. □

Proof (Theorem 37). If $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$ is not terminating, then there exists a term t which starts an infinite $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-reduction such that every proper subterm of t (even at inactive positions) is terminating w.r.t. $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$. Thus, t trivially has the hiding property. Furthermore, there are terms $t_i, s_i, \widehat{t_i}$ such that

$$t = \widehat{t_1} \xrightarrow{\mathcal{S}}{}^{*}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t_1 \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_1 \trianglerighteq_\mu \widehat{t_2} \xrightarrow{\mathcal{S}}{}^{*}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t_2 \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_2 \trianglerighteq_\mu \widehat{t_3} \ldots$$

where the $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps in $\widehat{t_i} \xrightarrow{\mathcal{S}}{}^{*}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t_i$ are applied strictly below the root position and the $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-step $t_i \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_i$ is applied at the root position. Furthermore, $\widehat{t_i}, t_i \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ for all i.

First, it is shown that every $\widehat{t_i}$ and every t_i has the hiding property. As shown above, $\widehat{t_1} = t$ has the hiding property by assumption. Also, if $\widehat{t_i}$ has the hiding property, then t_i has the hiding property by Lemma 36.3, and Lemma 36.3 furthermore implies that $\widehat{t_{i+1}}$ has the hiding property if t_i has the hiding property.

Next, it is shown that $t_i^\sharp \xrightarrow{\mathcal{S}}{}^{+}_{\mathit{Th}\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} \widehat{t'}_{i+1} \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}^\sharp_{i+1}$ for some $\widehat{t'}_{i+1}$ and that every term in this sequence is terminating w.r.t. $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$. Since $t_i \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} s_i \trianglerighteq_\mu \widehat{t_{i+1}}$ where the $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-step is applied at the root position, there exist a rule $l_i \to r_i[\![\varphi_i]\!] \in \mathcal{R}$, a Th-based substitution σ, and a position $p_i \in \mathcal{P}os^\mu(s_i)$ such that $t_i \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_i\sigma$, $\varphi_i\sigma$ is Th-valid, $s_i = r_i\sigma$, and $s_i|_{p_i} = \widehat{t_{i+1}}$. Perform a case distinction on the position p_i.

If $p_i \in \mathcal{P}os^\mu(r_i)$ and $r_i|_{p_i} \notin \mathcal{V}$, then there exists a dependency pair $l^\sharp \to (r_i|_{p_i})^\sharp[\![\varphi_i]\!] \in \mathsf{DP_o}(\mathcal{R},\mu)$ since $\mathrm{root}(r_i|_{p_i}) = \mathrm{root}(\widehat{t_{i+1}}) \in \mathcal{D}(\mathcal{R})$. But this clearly implies $t_i^\sharp \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_i^\sharp\sigma$ and thus $t_i^\sharp \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} (r_i|_{p_i})^\sharp\sigma = \widehat{t}^\sharp_{i+1}$. Furthermore, $t_i^\sharp$ and $\widehat{t}^\sharp_{i+1}$ are terminating w.r.t. $\xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$ since $t_i, \widehat{t_{i+1}} \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.

Otherwise, p_i is at or below the position of an active variable x_i in r_i. The variable x_i only occurs in inactive positions in l_i since $l_i\sigma \triangleright_\mu x_i\sigma \trianglerighteq_\mu \widehat{t_{i+1}}$ and $\widehat{t_{i+1}} \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ implies that $l_i\sigma \notin \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$, which, by Lemma 33 contradicts $t_i \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$ because $t_i \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_i\sigma$. Since $l_i\sigma$ has the hiding property by Lemma 36 (because t_i has the hiding property) and $l_i\sigma \ntriangleright_\mu x\sigma \trianglerighteq_\mu \widehat{t_{i+1}}$, Definition 35 implies $x\sigma = C[\widehat{t_{i+1}}]$ for a hiding context C and there exists an instance $\widehat{t'}_{i+1}$ of a hidden term such that $\widehat{t'}_{i+1} \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}_{i+1}$. Notice that $C[\widehat{t_{i+1}}]$ is irreducible by $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ since $l_i\sigma$ is irreducible by $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$. By letting s and s' be the appropriate sorts,

$$\begin{array}{llll}
t_i^\sharp & \xrightarrow{\mathcal{S}}_{\mathit{Th}\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} & \mathsf{U}_s(x_i\sigma) & \text{since } l_i^\sharp \to \mathsf{U}_s(x_i)[\![\varphi_i]\!] \in \mathsf{DP_u}(\mathcal{R},\mu) \\
 & = & \mathsf{U}_s(C[\widehat{t_{i+1}}]) & \text{where } C \text{ is hiding} \\
 & \xrightarrow{\mathcal{S}}{}^{*}_{\mathit{Th}\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} & \mathsf{U}_{s'}(\widehat{t_{i+1}}) & \text{since } \mathsf{U}_s(C[\widehat{t_{i+1}}]) \xrightarrow{\mathcal{S}}{}^{*}_{\mathit{Th}\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} \mathsf{U}_{s'}(\widehat{t_{i+1}}) \\
 & & & \text{for any hiding context } C
\end{array}$$

$$\begin{aligned} &\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} \widehat{t'}^{\sharp}_{i+1} && \text{using } \mathsf{U}(h) \to h^{\sharp}[\![\top]\!] \text{ since } \widehat{t'}_{i+1} \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}_{i+1} \\ & && \text{and } \widehat{t'}_{i+1} \text{ is an instance of a hidden term } h \\ &\xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}^{\sharp}_{i+1} && \end{aligned}$$

Notice that terms of the form $\mathsf{U}_s(\ldots)$ are terminating w.r.t. $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ since $\mu(\mathsf{U}_s) = \emptyset$. Also, $t^{\sharp}_i$ and $\widehat{t}^{\sharp}_{i+1}$ are terminating w.r.t. $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ since $t_i, \widehat{t}_{i+1} \in \mathcal{M}^{\infty}_{(\mathcal{R},\mathcal{S},\mathcal{E},\mu)}$.

Using $t^{\sharp}_i \xrightarrow{\mathcal{S}}{}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} \widehat{t'}^{\sharp}_{i+1} \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}^{\sharp}_{i+1}$, the sequence

$$t = \widehat{t}_1 \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_1 \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} s_1 \trianglerighteq_{\mu} \widehat{t}_2 \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_2 \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} s_2 \trianglerighteq_{\mu} \widehat{t}_3 \ldots$$

is transformed into

$$t^{\sharp} = \widehat{t}^{\sharp}_1 \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t^{\sharp}_1 \xrightarrow{\mathcal{S}}{}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathsf{DP}(\mathcal{R},\mu),\mu} \widehat{t'}^{\sharp}_2 \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} \widehat{t}^{\sharp}_2 \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t^{\sharp}_2 \ldots$$

Using Lemma 21 and the strong $\mathcal{E}$-coherence of $\to_{\mathcal{E}\backslash\mathcal{S}}$, this gives rise to an infinite minimal $(\mathsf{DP}(\mathcal{R},\mu), \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain. □

Proof (Theorem 40). It needs to be shown that there exists a substitution σ that is $\mathcal{T}h$-based for $\mathcal{V}(s_1) \cup \mathcal{V}(s_2)$ such that $\textsc{cap}(t_1)\sigma \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma$, the terms $s_1\sigma$ and $s_2\sigma$ are irreducible by $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$, and $\varphi_1\sigma$ and $\varphi_2\sigma$ are $\mathcal{T}h$-valid whenever $s_1 \to t_1[\![\varphi_1]\!]$, $s_2 \to t_2[\![\varphi_2]\!]$ is a $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain, i.e.,

(†) $t_1\sigma \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} u \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma$ for a $\mathcal{T}h$-based substitution σ such that $\varphi_1\sigma$ and $\varphi_2\sigma$ are $\mathcal{T}h$-valid and $s_1\sigma$ and $s_2\sigma$ are normal forms w.r.t. $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ implies $\textsc{cap}_{\mu}(t_1)\eta \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\eta$ for some substitution η that is $\mathcal{T}h$-based for $\mathcal{V}(s_1) \cup \mathcal{V}(s_2)$ such that $s_1\eta$ and $s_2\eta$ are normal forms w.r.t. $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ and $\varphi_1\eta$ and $\varphi_2\eta$ are $\mathcal{T}h$-valid.

In order to show (†), it is first shown that for all terms t and all substitutions η that are $\mathcal{T}h$-based for $\mathcal{V}(t)$,

(‡) $\textsc{cap}_{\mu}(t)\eta \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} u$ implies that there exists a substitution τ that is $\mathcal{T}h$-based for $\mathcal{V}(t)$ such that $u = \textsc{cap}_{\mu}(t)\tau$, where η and τ differ at most for the fresh variables introduced by $\textsc{cap}_{\mu}(t)$.

The property (‡) is shown by induction on t. If $\textsc{cap}_{\mu}(t) \in \mathcal{V}$, then it is a fresh variable y of sort `univ` since η is $\mathcal{T}h$-based for $\mathcal{V}(t)$. Letting $\tau = \{y \mapsto u\}$ establishes (‡). Otherwise, $t = f(t_1, \ldots, t_n)$, $\textsc{cap}_{\mu}(t) = f(t'_1, \ldots, t'_n)$ and there is no rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ such that $f(t'_1, \ldots, t'_n)\vartheta \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\vartheta$ for a substitution ϑ that is $\mathcal{T}h$-based for $\mathcal{V}(f(t_1, \ldots, t_n)) \cup \mathcal{V}(l)$ where $\varphi\vartheta$ is $\mathcal{T}h$-valid. First, it is shown that the $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-step cannot take place at the root position. If the reduction takes place at the root position, then there exist a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ and a $\mathcal{T}h$-based substitution ρ such that $f(t'_1, \ldots, t'_n)\eta \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\rho$ and $\varphi\rho$ is $\mathcal{T}h$-valid.

Since it can be assumed that l is variable-disjoint from $f(t'_1, \ldots, t'_n)$, it is possible to define the substitution $\vartheta = \sigma \cup \rho$ where ϑ is $\mathcal{T}h$-based for $\mathcal{V}(f(t'_1, \ldots, t'_n)) \cup \mathcal{V}(l)$. Since $\varphi\vartheta$ is $\mathcal{T}h$-valid this is a contradiction to the assumption. Hence, the $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-step takes place below he root position, i.e., there exists an $1 \leq i \leq n$ such that $t'_i\eta \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} u_i$ and $u = f(t'_1\eta, \ldots, u_i, \ldots, t'_n\eta)$. Furthermore, $i \in \mu(f)$, which implies $t'_i = \textsc{cap}_\mu(t_i)$. By the inductive assumption, this yields a substitution δ that is $\mathcal{T}h$-based for $\mathcal{V}(t_i)$ such that $u_i = \textsc{cap}_\mu(t_i)\delta = t'_i\delta$. Since the fresh variables introduced by $\textsc{cap}_\mu(t_i)$ are disjoint from the fresh variables introduced by $\textsc{cap}_\mu(t_j)$ for $1 \leq j \neq i \leq n$ and since η and δ differ at most for the fresh variables introduced by $\textsc{cap}_\mu(t_i)$, define the substitution τ with $\tau(x) = \delta(x)$ if x is a fresh variable introduced by $\textsc{cap}_\mu(t_i)$ and $\tau(x) = \eta(x)$ otherwise. Then τ is $\mathcal{T}h$-based for $\mathcal{V}(t)$ and

$$\begin{aligned} u &= f(t'_1\eta, \ldots, t'_i\delta, \ldots, t'_n\eta) \\ &= f(t'_1\tau, \ldots, t'_i\tau, \ldots, t'_n\tau) \\ &= \textsc{cap}_\mu(t)\tau \end{aligned}$$

By (‡) and an induction on the number of $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps, it is easily obtained that $\textsc{cap}_\mu(t)\eta \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} u$ for a substitution η which is $\mathcal{T}h$-based for $\mathcal{V}(t)$ implies that $u = \textsc{cap}_\mu(t)\delta$ for some substitution δ that is $\mathcal{T}h$-based for $\mathcal{V}(t)$ such that η and δ differ at most for fresh variables introduced by $\textsc{cap}_\mu(t)$. Since $t_1 = \textsc{cap}_\mu(t_1)\sigma'$ for some substitution σ' that only instantiates fresh variables introduced by $\textsc{cap}_\mu(t_1)$, in particular $t_1\sigma = \textsc{cap}_\mu(t_1)\sigma'\sigma \xrightarrow{\mathcal{S}}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} u$ implies that $u = \textsc{cap}_\mu(t_1)\delta$ for some substitution δ that is $\mathcal{T}h$-based for $\mathcal{V}(t_1)$ such that $\sigma'\sigma$ and δ differ at most for the fresh variables introduced by $\textsc{cap}_\mu(t_1)$. Thus $\textsc{cap}_\mu(t_1)\delta = u \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma$ since $u \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma$. Now define the substitution ξ by $\xi(x) = \delta(x)$ if x is a fresh variable introduced by $\textsc{cap}_\mu(t_1)$ and $\xi(x) = \sigma(x)$ otherwise. Notice that ξ is $\mathcal{T}h$-based for $\mathcal{V}(s_1) \cup \mathcal{V}(s_2)$. Then $\textsc{cap}_\mu(t_1)\xi = u \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_2\sigma = s_2\xi$. Since $s_1\xi = s_1\sigma$ and $s_2\xi = s_2\sigma$, the terms $s_1\xi$ and $s_2\xi$ are normal forms w.r.t. $\xrightarrow{>\Lambda}_{\mathcal{E}\backslash\mathcal{S}}$ by Definition 31. Also, $\varphi_1\xi = \varphi_1\sigma$ and $\varphi_2\xi = \varphi_2\sigma$ are $\mathcal{T}h$-valid. □

Proof (Theorem 41). After a finite number of dependency pairs in the beginning, any infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain only contains pairs from some SCC. Hence, every infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain gives rise to an infinite minimal $(\mathcal{P}_i, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain for some $1 \leq i \leq n$. □

Proof (Lemma 45).

1. Since $\mathcal{E}$ is size-preserving, the $\mathcal{E}$-equivalence classes are finite. Furthermore, the $\mathcal{E}$-equivalence class of a given term s can be effectively computed using the equations. In order to decide whether $s \rhd_{\mathcal{E},\mu} t$ it then suffices to check whether there exist a term s' in the $\mathcal{E}$-equivalence class of s and a term t' in the $\mathcal{E}$-equivalence class of t such that $s' \rhd_\mu t'$. Notice that $s \unrhd_{\mathcal{E},\mu} t$ can be decided in the same way.

2. Well-foundedness of $\rhd_{\mathcal{E},\mu}$ is immediate once it has been shown that $s \rhd_{\mathcal{E},\mu} t$ implies $|s| > |t|$. Thus, let $s \rhd_{\mathcal{E},\mu} t$, i.e., $s \sim_{\mathcal{E}} s' \rhd_{\mu} t' \sim_{\mathcal{E}} t$ for some terms s', t'.
   First, it is shown that $s \sim_{\mathcal{E}} s'$ implies $|s| = |s'|$. For this, induction on n in $s \vdash\!\dashv_{\mathcal{E}}^{n} s'$ is performed. If $n = 0$, the claim is obvious. Otherwise, $s \vdash\!\dashv_{\mathcal{E}}^{n-1} s'' \vdash\!\dashv_{\mathcal{E}} s'$, and the inductive hypothesis implies $|s| = |s''|$. Now $s'' \vdash\!\dashv_{\mathcal{E}} s'$ implies that there exists an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ such that $s'' = C[u\sigma]$ and $s' = C[v\sigma]$ for some context C and some substitution σ and it thus suffices to show $|u\sigma| = |v\sigma|$. But $|u\sigma| = |v\sigma|$ is an immediate consequence of the assumption that the equation $u \approx v$ (or $v \approx u$) is size-preserving and i.u.v.
   Now $|s| > |t|$ is obtained since $s' \rhd_{\mu} t'$ implies $|s'| > |t'|$.
3. For $\rhd_{\mathcal{E},\mu}$, let $s \rhd_{\mathcal{E},\mu} t$, i.e., $s \sim_{\mathcal{E}} s' \rhd_{\mu} t' \sim_{\mathcal{E}} t$ for some terms s', t'. Then $s\sigma \sim_{\mathcal{E}} s'\sigma \rhd_{\mu} t'\sigma \sim_{\mathcal{E}} t\sigma$ since both $\sim_{\mathcal{E}}$ and $\rhd_{\mu}$ are stable, i.e., $s\sigma \rhd_{\mathcal{E},\mu} t\sigma$. Now stability of $\unrhd_{\mathcal{E},\mu}$ is obvious since both $\rhd_{\mathcal{E},\mu}$ and $\sim_{\mathcal{E}}$ are stable.
4. Let $s \rhd_{\mathcal{E},\mu} t$, i.e., $s \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} t$, and let $s' \sim_{\mathcal{E}} s$ and $t' \sim_{\mathcal{E}} t$. Thus, $s' \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} t'$, i.e., $s' \rhd_{\mathcal{E},\mu} t'$. Since $\unrhd_{\mathcal{E},\mu} = \rhd_{\mathcal{E},\mu} \cup \sim_{\mathcal{E}}$ the claim for $\unrhd_{\mathcal{E},\mu}$ is now immediate. □

Proof (Theorem 46). In the second case soundness is obvious. Otherwise, it needs to be shown that every infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain contains only finitely many dependency pairs from $\mathcal{P}'$. Thus, let $s_1 \to t_1[\![\varphi_1]\!], s_2 \to t_2[\![\varphi_2]\!], \ldots$ by an infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain using the $\mathcal{T}h$-based substitution σ and apply the simple projection π to it.

First, consider the instantiation $s_i\sigma \to t_i\sigma[\![\varphi_i\sigma]\!]$ of the i^{th} dependency pair. Clearly, $\pi(s_i\sigma) = \pi(s_i)\sigma$ and $\pi(t_i\sigma) = \pi(t_i)\sigma$. Since $\pi(s_i) \unrhd_{\mathcal{E},\mu} \pi(t_i)$ by assumption, $\pi(s_i\sigma) \unrhd_{\mathcal{E},\mu} \pi(t_i\sigma)$ by Lemma 45.3. From $t_i\sigma \overset{\mathcal{S}}{\to}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}}$ $s_{i+i}\sigma$, the (possibly shorter) sequence $\pi(t_i\sigma) \overset{\mathcal{S}}{\to}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} \pi(s_{i+i}\sigma)$ is obtained since all reductions take place below the root. Therefore, having $t_i\sigma \overset{\mathcal{S}}{\to}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_{i+1}\sigma \to t_{i+1}\sigma$ implies $\pi(t_i\sigma) \overset{\mathcal{S}}{\to}{}^{*}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^{*}_{\mathcal{E}\backslash\mathcal{S}}$ $\circ \sim_{\mathcal{E}} \pi(s_{i+1}\sigma) \unrhd_{\mathcal{E},\mu} \pi(t_{i+1}\sigma)$.

Thus, the infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain is transformed into an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \cup \to_{\mathcal{E}\backslash\mathcal{S}} \cup \rhd_{\mathcal{E},\mu} \cup \sim_{\mathcal{E}}$ sequence starting with $\pi(t_1\sigma)$. Now perform a case distinction.

Case 1: The infinite sequence contains only finitely many $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps and only finitely many $\to_{\mathcal{E}\backslash\mathcal{S}}$-steps. Then, there exists an infinite $\rhd_{\mathcal{E},\mu} \cup \sim_{\mathcal{E}}$ sequence. This sequence cannot contain infinitely many $\rhd_{\mathcal{E},\mu}$-steps stemming from dependency pairs in $\mathcal{P}'$ since otherwise Lemma 45.4 yields an infinite $\rhd_{\mathcal{E},\mu}$ sequence, contradicting the well-foundedness of $\rhd_{\mathcal{E},\mu}$ (Lemma 45.2).

Case 2: The infinite sequence contains only finitely many $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps but infinitely many $\to_{\mathcal{E}\backslash\mathcal{S}}$-steps. Recall that $\sim_{\mathcal{E}} \circ \to_{\mathcal{E}\backslash\mathcal{S}} \subseteq \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}}$ since $\to_{\mathcal{E}\backslash\mathcal{S}}$ is strongly $\mathcal{E}$-coherent. Using this and the easily seen inclusion $\rhd_{\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} \subseteq \to_{\mathcal{E}\backslash\mathcal{S}} \circ \rhd_{\mu}$, it is furthermore straightforward to show that

$$\rhd_{\mathcal{E},\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} = \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \circ \to_{\mathcal{E}\backslash\mathcal{S}}$$

$$\begin{aligned} &\subseteq \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} \\ &\subseteq \sim_{\mathcal{E}} \circ \to_{\mathcal{E}\backslash\mathcal{S}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \\ &\subseteq \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \\ &= \to_{\mathcal{E}\backslash\mathcal{S}} \circ \rhd_{\mathcal{E},\mu} \end{aligned}$$

By making repeated use of these inclusions, an infinite $\to_{\mathcal{E}\backslash\mathcal{S}}$-sequence is obtained, contradiction the assumption that $\to_{\mathcal{E}\backslash\mathcal{S}}$ is terminating.

Case 3: The infinite sequence contains infinitely many $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps. Recall Lemma 21 and the inclusions $\sim_{\mathcal{E}} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}}$ and $\to_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\to}{}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^{=}_{\mathcal{E}\backslash\mathcal{S}}$. Using the first of these inclusions and the easily seen inclusion $\rhd_{\mu} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rhd_{\mu}$, it is furthermore straightforward to show that

$$\begin{aligned} \rhd_{\mathcal{E},\mu} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} &= \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \\ &\subseteq \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}} \\ &\subseteq \sim_{\mathcal{E}} \circ \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \\ &\subseteq \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}} \circ \rhd_{\mu} \circ \sim_{\mathcal{E}} \\ &= \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rhd_{\mathcal{E},\mu} \end{aligned}$$

By making repeated use of these inclusions, an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-sequence starting with $\pi(t_1\sigma)$ is obtained. But this contradicts the minimality of the infinite minimal $(\mathcal{P},\mathcal{R},\mathcal{S},\mathcal{E},\mu)$-chain since then $t_1\sigma$ starts an infinite $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ reduction. □

Proof (Theorem 49). Standard. □

Proof (Theorem 52). Standard. □

Before proving Lemma 57, several auxiliary results need to be obtained. The first result in similar to Lemma 21.1 but considers $\mathcal{S}$ and is not restricted to active contexts.

Lemma 70. *If $s \to^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} C[f(t^*)]$ for a context C and an $f \notin \mathcal{F}(\mathcal{E}) \cup \mathcal{F}(\mathcal{S})$, then $s = D[f(s^*)]$ for a context D such that $f(s^*) \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} f(t^*)$.*

Proof. By [12, Lemma 5], it suffices to consider $s \to_{\mathcal{S}} C[f(t^*)]$. If $s \to_{\mathcal{S}} t = C[f(t^*)]$, then there exists a rule $l \to r \in \mathcal{S}$ such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$ for some position p and some substitution σ. Let q be the position with $t|_q = f(t^*)$, i.e., $C|_q = \Box$. Perform a case distinction on the relationship between the positions p and q.

Case 1: $p \perp q$. Then, $s = t[l\sigma]_p = (C[f(t^*)])[l\sigma]_p = (C[l\sigma]_p)[f(t^*)]$.

Case 2: $p = q.q'$ for some position $q' \neq \varepsilon$. Then, $s = t[l\sigma]_p = (C[f(t^*)])[l\sigma]_{q.q'} = C[f(t^*)[l\sigma]_{q'}]$. Since $q' \neq \varepsilon$, the position q' can we written as $q' = i.q''$ for some $1 \leq i \leq \mathrm{arity}(f)$ and some position q''. Then $s_j = t_j$ if $i \neq j$ and $s_i = t_i[l\sigma]_{q''} \to_{\mathcal{S}} t_i[r\sigma]_{q''} = t_i$, i.e., $s^* \to_{\mathcal{S}} t^*$.
Case 3: $q = p.p'$ for some position p' (possibly $p' = \Lambda$). Since $f \notin \mathcal{F}(\mathcal{S})$, the position p' can be written as $p' = p'_1.p'_2$ such that $r|_{p'_1}$ is a variable x and $x\sigma|_{p'_2} = f(t^*)$. There exists a position p''_1 in l such that $l|_{p''_1} = x$. This implies $l\sigma|_{p''_1.p'_2} = x\sigma|_{p'_2} = f(t^*)$. Thus, $s = t[l\sigma]_p = D[f(t^*)]$ where $D = C[l\sigma[\Box]_{p''_1.p'_2}]$ and the claim follows. □

Next, it can be shown that $\to_{\mathcal{E}\backslash\mathcal{S}}$ and $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ are finitely branching if $\mathcal{E}$ is size preserving.

Lemma 71. *Let $(\mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$ be a CS-CES such that $\mathcal{E}$ is size preserving. Then $\to_{\mathcal{E}\backslash\mathcal{S}}$ and $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ are finitely branching.*

Proof. First, the property is shown for $\to_{\mathcal{E}\backslash\mathcal{S}}$. Since a term has only finitely many positions and since $\mathcal{S}$ is a finite set of rules, it suffices to show finite branching for one position p and one rule $l \to r \in \mathcal{S}$. Without loss of generality assume $p = \Lambda$, i.e., $s \sim_{\mathcal{E}} l\sigma$. Since $\mathcal{E}$ is size-preserving, the $\mathcal{E}$-equivalence classes are finite and there are thus only finitely many substitutions σ with $\mathcal{D}om(\sigma) = \mathcal{V}(l)$ for $\mathcal{E}$-matching. Therefore, $\to_{\mathcal{E}\backslash\mathcal{S}}$ is finitely branching.

The proof for $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ is similar, where it again suffices to consider only the root position and a single rule. Thus, $s \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} s' \sim^{>\Lambda}_{\mathcal{E}} l\sigma$, where σ is $\mathcal{T}h$-based. Since $\to_{\mathcal{E}\backslash\mathcal{S}}$ is $\mathcal{E}$-convergent and since $\mathcal{E}$ is size-preserving there are only finitely many possible terms s'. As above, there are only finitely many substitutions σ for each s', which implies that $\xrightarrow{\mathcal{S},\mathcal{Q}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R}}$ is finitely branching. □

Now it can be shown that $\mathcal{I}^1(t)$ is indeed well-defined.

Proof (Lemma 57). According to Definition 56, in order to get an infinite term as a result of $\mathcal{I}^1(t)$, it is necessary to perform an infinite number of applications of $\mathcal{R}ed^1_{\mathcal{R}}$ since $\to_{\mathcal{E}\backslash\mathcal{S}}$ and $\rhd_{\mathcal{E}}$ are well-founded, $\to_{\mathcal{E}\backslash\mathcal{S}}$ and $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ are finitely branching by Lemma 71, and the $\mathcal{E}$-equivalence classes are finite. This means that t is terminating and there exists an infinite sequence $t \bowtie_1 s_1 \bowtie_2 s_2 \ldots$ where $\bowtie_i \in \{\unrhd,\ \to_{\mathcal{E}\backslash\mathcal{S}},\ \sim_{\mathcal{E}},\ \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}\}$ and $\bowtie_i = \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ for infinitely many i. By considering two consecutive occurrences of $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ in this sequence and by using the inclusions $\sim_{\mathcal{E}} \circ \to_{\mathcal{E}\backslash\mathcal{S}} \subseteq \to_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}}$ obtained from the strong $\mathcal{E}$-coherence of $\to_{\mathcal{E}\backslash\mathcal{S}}$ and the easily seen inclusions $\unrhd \circ \to_{\mathcal{E}\backslash\mathcal{S}} \subseteq \to_{\mathcal{E}\backslash\mathcal{S}} \circ \unrhd$ and $\unrhd \circ \sim_{\mathcal{E}} \subseteq \sim_{\mathcal{E}} \circ \unrhd$, a sequence $t \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_1 \unrhd t_1 \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_2 \unrhd t_2 \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_3 \ldots$ where $\mathrm{root}(t_i) \notin \Delta$ and t_i is terminating for all $i \geq 1$ is obtained. It can be assumed without loss of generality that it is not the case that $t_i \rhd \circ \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_{i+1}$ (since otherwise the modified sequence obtained from $\ldots \unrhd t_{i-1} \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_i \rhd \circ \to^*_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_{i+1} \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \ldots$

by using the inclusions $\sim_{\mathcal{E}} \circ \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \subseteq \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}}$, $\trianglerighteq \circ \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \subseteq \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \trianglerighteq$, and $\trianglerighteq \circ \sim_{\mathcal{E}} \subseteq \sim_{\mathcal{E}} \circ \trianglerighteq$ mentioned above can be considered instead).

For all $i \geq 1$, there exist a rule $l_i \to r_i[\![\varphi_i]\!] \in \mathcal{R}$, a $\mathcal{T}h$-based substitution σ_i, and an active position p_i in t_i such that $t_i = C_i[l_i']_{p_i}$, $l_i' \xrightarrow{>\Lambda}{}^{!}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l_i\sigma_i$, $C_i[r_i\sigma_i]_{p_i} \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_{i+1}$, and $u_{i+1} \trianglerighteq t_{i+1}$. Now $u_{i+1} \trianglerighteq t_{i+1}$ and $\mathrm{root}(t_{i+1}) \notin \mathcal{F}(\mathcal{E}) \cup \mathcal{F}(\mathcal{S})$ (since $\mathrm{root}(t_{i+1}) \notin \Delta^1$ and $\mathcal{F}(\mathcal{E}) \cup \mathcal{F}(\mathcal{S}) \subseteq \Delta^1$) imply by Lemma 70 that $C_i[r_i\sigma_i] \trianglerighteq t_{i+1}'$ for some t_{i+1}' with $\mathrm{root}(t_{i+1}') = \mathrm{root}(t_{i+1})$ and $t_{i+1}' \xrightarrow{>\Lambda}{}^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} t_{i+1}$. There are three possibilities:

1. t_{i+1}' is a subterm of $C_i[r_i\sigma_i]_{p_i}$ at a position above p_i, i.e., $t_{i+1}' \trianglerighteq r_i\sigma_i$. Then $C_i[r_i\sigma_i]_{p_i} \trianglerighteq_{\mu} t_{i+1}'$ since p_i is an active position.
2. t_{i+1}' is a subterm of $C_i[r_i\sigma_i]_{p_i}$ at a position that is independent of p_i. Then $t_i \vartriangleright t_{i+1}' \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_{i+1}$, contradicting the assumption.
3. t_{i+1}' is a subterm of $C_i[r_i\sigma_i]_{p_i}$ at a position strictly below p_i, i.e., $r_i\sigma_i \vartriangleright t_{i+1}'$. Note that there is no variable $x \in \mathcal{V}(r_i)$ such that $\sigma_i(x) \vartriangleright t_{i+1}'$ since that would imply $l_i\sigma_i \vartriangleright t_{i+1}'$. Then $l_i' \vartriangleright t_{i+1}''$ for some t_{i+1}'' such that $t_{i+1}'' \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_{i+1}'$ by Lemma 70, which implies $t_i \vartriangleright t_{i+1}'' \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_{i+1}$, contradicting the assumption. Therefore, there exists a term r_i' with $r_i' \notin \mathcal{V}$ and $r_i \trianglerighteq r_i'$ such that $r_i'\sigma_i = t_{i+1}'$. Since $\mathrm{root}(r_i') = \mathrm{root}(t_{i+1}') \notin \Delta^1$ and $\mathcal{F}^{\not\mu}(r_i) \subseteq \Delta^1$, the relation $r_i \trianglerighteq_{\mu} r_i'$ is obtained and thus $C_i[r_i\sigma_i]_{p_i} \trianglerighteq_{\mu} t_{i+1}'$.

The resulting sequence is thus $t \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} u_1 \trianglerighteq_{\mu} t_1 \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} C_1[r_1\sigma_1]_{p_1} \trianglerighteq_{\mu} t_2' \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_2 \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} C_2[r_2\sigma_2]_{p_2} \trianglerighteq_{\mu} t_3' \rightarrow^{*}_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t_3 \ldots$.

By using $\sim_{\mathcal{E}} \circ \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \sim_{\mathcal{E}}$ and $\rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\rightarrow}{}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rightarrow^{=}_{\mathcal{E}\backslash\mathcal{S}}$ from Lemma 21 and $\vartriangleright_{\mathcal{E},\mu} \circ \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \vartriangleright_{\mathcal{E},\mu}$ from the proof of Theorem 46, an infinite $\overset{\mathcal{S}}{\rightarrow}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ sequence starting with t is obtained, contradicting the assumption that t is terminating. □

Proof (Lemma 60). Let $s, t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ and let σ be a $\mathcal{T}h$-based substitution.

1. $\mathcal{I}^1(s\sigma) = s\mathcal{I}^1(\sigma)$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $\mathcal{I}^1(\sigma)$. Otherwise, $s = f(s_1, \ldots, s_n)$ with $f \in \Delta^1$. But then
$$\begin{aligned}\mathcal{I}^1(s\sigma) &= \mathcal{I}^1(f(s_1\sigma, \ldots, s_n\sigma))\\ &= f(\mathcal{I}^1(s_1\sigma), \ldots, \mathcal{I}^1(s_n\sigma))\\ &= f(s_1\mathcal{I}^1(\sigma), \ldots, s_n\mathcal{I}^1(\sigma))\\ &= s\mathcal{I}^1(\sigma)\end{aligned}$$
by the inductive assumption.
2. $\mathcal{I}^1(s\sigma) \rightarrow^{*}_{\mathcal{R}_{\Pi}} s\mathcal{I}^1(\sigma)$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $\mathcal{I}^1(\sigma)$. Otherwise, $s = f(s_1, \ldots, s_n)$. By the

assumption, if $i \notin \mu(f)$, then $s_i \in \mathcal{T}(\Delta^1, \mathcal{V})$ and hence $\mathcal{I}^1(s_i\sigma) = s_i\mathcal{I}^1(\sigma)$ by *1.* If $i \in \mu(f)$, then $\mathcal{I}^1(s_i\sigma) \to^*_{\mathcal{R}_\Pi} s_i\mathcal{I}^1(\sigma)$ by the inductive hypothesis. If $f \in \Delta^1$ or s is not terminating, then $\mathcal{I}^1(s\sigma) = \mathcal{I}^1(f(s_1\sigma, \ldots, s_n\sigma)) = f(\mathcal{I}^1(s_1\sigma), \ldots, \mathcal{I}^1(s_n\sigma)) \to^*_{\mathcal{R}_\Pi} f(s_1\mathcal{I}^1(\sigma), \ldots, s_n\mathcal{I}^1(\sigma)) = s\mathcal{I}^1(\sigma)$. If $f \notin \Delta^1$ and s is terminating, then $\mathcal{I}^1(s\sigma) = \mathcal{C}omp_{\mathrm{sort}(s\sigma)}(\mathcal{R}ed^1_{\mathcal{S}}(s\sigma) \cup \mathcal{R}ed^1_{\mathcal{R}}(s\sigma) \cup \mathcal{E}q^1_{\mathcal{E}}(s\sigma))$. The definition of $\mathcal{E}q^1_{\mathcal{E}}(s\sigma)$ gives $f(\mathcal{I}^1(s_1\sigma), \ldots, \mathcal{I}^1(s_n\sigma)) \in \mathcal{E}q^1_{\mathcal{E}}(s\sigma)$. Thus, $\mathcal{I}^1(s\sigma) \to^+_{\mathcal{R}_\Pi} f(\mathcal{I}^1(s_1\sigma), \ldots, \mathcal{I}^1(s_n\sigma)) \to^*_{\mathcal{R}_\Pi} f(s_1\mathcal{I}^1(\sigma), \ldots, s_n\mathcal{I}^1(\sigma))$ by Lemma 59.

3. It suffices to show that $s \vdash\dashv_{\mathcal{E}} t$ implies $\mathcal{I}^1(s) \vdash\dashv^{=}_{\mathcal{E}} \mathcal{I}^1(t)$ since the statement then follows by induction on the number of $\vdash\dashv_{\mathcal{E}}$-steps in $s \sim_{\mathcal{E}} t$.
   Thus, let $s \vdash\dashv_{\mathcal{E}} t$ and perform an induction on the position p where the step takes places. If $\mathrm{root}(s) \notin \Delta^1$ and s is terminating, then $\mathrm{root}(t) \notin \Delta^1$ as well by the definition of Δ^1 and t is terminating by Corollary 22.1. Since $s \sim_{\mathcal{E}} t$, Definition 9 implies that whenever $s \to_{\mathcal{E}\backslash\mathcal{S}} s'$, then $t \to_{\mathcal{E}\backslash\mathcal{S}} t'$ for some $t' \sim_{\mathcal{E}} s'$. Thus, $\mathcal{R}ed^1_{\mathcal{S}}(s) \subseteq \mathcal{R}ed^1_{\mathcal{S}}(t)$. Similarly, Lemma 21.2 implies $\mathcal{R}ed^1_{\mathcal{R}}(s) \subseteq \mathcal{R}ed^1_{\mathcal{R}}(t)$. Finally, if $s \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, then $t \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, which immediately implies $\mathcal{E}q^1_{\mathcal{E}}(s) = \mathcal{E}q^1_{\mathcal{E}}(t)$.
   Using these properties, $\mathcal{R}ed^1_{\mathcal{S}}(s) \cup \mathcal{R}ed^1_{\mathcal{R}}(s) \cup \mathcal{E}q^1_{\mathcal{E}}(s) \subseteq \mathcal{R}ed^1_{\mathcal{S}}(t) \cup \mathcal{R}ed^1_{\mathcal{R}}(t) \cup \mathcal{E}q^1_{\mathcal{E}}(t)$. Since the same reasoning can be applied with s and t interchanged,
   $$\begin{aligned} &\mathcal{C}omp_{\mathrm{sort}(s)}(\mathcal{R}ed^1_{\mathcal{S}}(s) \cup \mathcal{R}ed^1_{\mathcal{R}}(s) \cup \mathcal{E}q^1_{\mathcal{E}}(s)) \\ = \ &\mathcal{C}omp_{\mathrm{sort}(s)}(\mathcal{R}ed^1_{\mathcal{S}}(t) \cup \mathcal{R}ed^1_{\mathcal{R}}(t) \cup \mathcal{E}q^1_{\mathcal{E}}(t)) \end{aligned}$$
   and thus $\mathcal{I}^1(s) = \mathcal{I}^1(t)$.
   Otherwise, $\mathrm{root}(s) \in \Delta^1$ or s is not terminating. If $p = \Lambda$, then there exist an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ and a substitution σ such that $s = u\sigma$ and $t = v\sigma$. By the definition of Δ^1, $u, v \in \mathcal{T}(\Delta^1, \mathcal{V})$. Hence, $\mathcal{I}^1(s) = \mathcal{I}^1(u\sigma) = u\mathcal{I}^1(\sigma) \vdash\dashv_{\mathcal{E}} v\mathcal{I}^1(\sigma) = \mathcal{I}^1(v\sigma) = \mathcal{I}^1(t)$ by *1.* If $p \neq \Lambda$ then $s = f(s_1, \ldots, s_i, \ldots, s_n)$, $t = f(s_1, \ldots, t_i, \ldots, s_n)$, and $s_i \vdash\dashv_{\mathcal{E}} t_i$. Now $\mathcal{I}(s_i) \vdash\dashv^{=}_{\mathcal{E}} \mathcal{I}(t_i)$ by the inductive assumption, which implies
   $$\begin{aligned} \mathcal{I}(s) &= f(\mathcal{I}^1(s_1), \ldots, \mathcal{I}^1(s_i), \ldots, \mathcal{I}^1(s_n)) \\ &\vdash\dashv^{=}_{\mathcal{E}} f(\mathcal{I}^1(s_1), \ldots, \mathcal{I}^1(s'_i), \ldots, \mathcal{I}^1(s_n)) \\ &= \mathcal{I}^1(t) \end{aligned}$$
4. It suffices to show that $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ implies $\mathcal{I}^1(s) \leadsto_1 \mathcal{I}^1(t)$ since the statement then follows by induction on the number of $\to_{\mathcal{E}\backslash\mathcal{S}}$-steps in $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$.
   Thus, let $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ and perform an induction on the position p where the reduction takes places. If $\mathrm{root}(s) \notin \Delta^1$ and s is terminating, then $\mathcal{I}^1(t) \in \mathcal{R}ed^1_{\mathcal{S}}(s)$, which implies $\mathcal{I}^1(s) \to^+_{\mathcal{R}_\Pi} \mathcal{I}^1(t)$ by Lemma 59.
   If $\mathrm{root}(s) \in \Delta^1$ or s is not terminating, first consider the case $p = \Lambda$. Then, there exist a rule $l \to r \in \mathcal{S}$ and a substitution σ such that $s \sim_{\mathcal{E}} l\sigma \to_{\mathcal{S}} r\sigma = t$. Using *1.* and *3.*, $\mathcal{I}^1(s) \sim_{\mathcal{E}} \mathcal{I}^1(l\sigma) = l\mathcal{I}^1(\sigma) \to_{\mathcal{S}} r\mathcal{I}^1(\sigma) = \mathcal{I}^1(r\sigma) = \mathcal{I}^1(t)$, and thus $\mathcal{I}^1(s) \to_{\mathcal{E}\backslash\mathcal{S}} \mathcal{I}^1(t)$. If $\mathrm{root}(s) \in \Delta^1$ and $p \neq \Lambda$, then $s = f(s_1, \ldots, s_i, \ldots, s_n)$ and $t = f(s_1, \ldots, t_i, \ldots, s_n)$, where $s_i \to_{\mathcal{E}\backslash\mathcal{S}} t_i$. This implies $\mathcal{I}^1(s_i) \leadsto_1 \mathcal{I}^1(t_i)$ by the inductive hypothesis and therefore $\mathcal{I}^1(s) = f(\ldots, \mathcal{I}^1(s_i), \ldots) \leadsto_1 f(\ldots, \mathcal{I}^1(t_i), \ldots) = \mathcal{I}^1(t)$.

5. It suffices to show that $s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t$ implies $\mathcal{I}^1(s) \rightsquigarrow_2 \mathcal{I}^1(t)$ since the statement then follows by induction on the number of $\overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps in $s \overset{\mathcal{S}}{\to}^*_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t$.
   Thus, let $s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t$ and perform an induction on the position p where the reduction takes places. If $\mathrm{root}(s) \notin \Delta^1$, then $\mathcal{I}^1(t) \in \mathcal{R}ed^1_{\mathcal{R}}(s)$, which implies $\mathcal{I}^1(s) \to^+_{\mathcal{R}_\Pi} \mathcal{I}^1(t)$ by Lemma 59.
   If $\mathrm{root}(s) \in \Delta^1$, first consider the case $p = \Lambda$. Then, there exist a rule $l \to r[\![\varphi]\!] \in \mathcal{R}$ and a Th-based substitution σ with $s = f(s^*) \xrightarrow{>\Lambda}{}^!_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma \to_{\mathcal{R}} r\sigma = t$ such that $\varphi\sigma$ is Th-valid. Since $\mathrm{root}(l) = \mathrm{root}(s) = f$ and $f \in \Delta^1$, the definition of Δ^1 implies that $l \to r[\![\varphi]\!] \in \mathcal{R}(\Delta^1)$, $r \in \mathcal{T}(\Delta^1, \mathcal{V})$, and $l' \in \mathcal{T}(\Delta^1, \mathcal{V})$ whenever $l \rhd_{\not\mu} l'$. Using *1.*, *2.*, *3.*, and *4.*, $\mathcal{I}^1(s) \rightsquigarrow^*_1 \circ \sim_{\mathcal{E}} \mathcal{I}^1(l\sigma) \to^*_{\mathcal{R}_\Pi} l\mathcal{I}^1(\sigma) \to_{\mathcal{R}(\Delta^1),\mu} r\mathcal{I}^1(\sigma) = \mathcal{I}^1(r\sigma) = \mathcal{I}^1(t)$ where $\varphi\mathcal{I}^1(\sigma) = \varphi\sigma$ is Th-valid, and thus $\mathcal{I}^1(s) \rightsquigarrow_2 \mathcal{I}^1(t)$. If $\mathrm{root}(s) \in \Delta^1$ and $p \neq \Lambda$, then $s = f(s_1, \ldots, s_i, \ldots, s_n)$ and $t = f(s_1, \ldots, t_i, \ldots, s_n)$, where $s_i \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t_i$ for an $i \in \mu(f)$. Now $\mathcal{I}^1(s_i) \rightsquigarrow_2 \mathcal{I}^1(t_i)$ by the inductive hypothesis since s_i is terminating and therefore $\mathcal{I}^1(s) = f(\mathcal{I}^1(s_1), \ldots, \mathcal{I}^1(s_i), \ldots, \mathcal{I}^1(s_n)) \rightsquigarrow_2 f(\mathcal{I}^1(s_1), \ldots, \mathcal{I}^1(t_i), \ldots, \mathcal{I}^1(s_n)) = \mathcal{I}^1(t)$.
6. Let $s\sigma \overset{\mathcal{S}}{\to}^*_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^!_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t\sigma$. Using *3.*, *4.*, and *5.*, $\mathcal{I}^1(s\sigma) \rightsquigarrow^*_2 \circ \rightsquigarrow^*_1 \circ \sim_{\mathcal{E}} \mathcal{I}^1(t\sigma)$. Using *1.* and *2.* this implies $s\mathcal{I}^1(\sigma) \rightsquigarrow^*_2 \circ \rightsquigarrow^*_1 \circ \sim_{\mathcal{E}} \circ \to^*_{\mathcal{R}_\Pi} t\mathcal{I}^1(\sigma)$. □

Proof (Theorem 61). Standard by making use of Lemma 60.6. □

Before proving Lemma 66, some auxiliary results need to be obtained. Lemma 66 will be shown by well-founded induction using the following relations. This is similar to [13].

Definition 72 ($\overline{\hookrightarrow}^2, \hookrightarrow^2$). *Let s, t be terms and. Then $s \overline{\hookrightarrow}^2 t$ iff $s \to_{\mathcal{E}\backslash\mathcal{S}} \cup \rhd_{\mathcal{E}} t$ and $s \hookrightarrow^2 t$ iff $s \to_{\mathcal{E}\backslash\mathcal{S}} \cup \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \cup \rhd_{\mathcal{E},\mu} t$.*

In order to use $\overline{\hookrightarrow}^2$ and $\hookrightarrow^2$ for inductive proofs, they need to be well-founded when restricted to the relevant terms.

Lemma 73.

1. *If s is terminating and $s \hookrightarrow^2 t$, then t is terminating.*
2. *$\overline{\hookrightarrow}^2$ is well-founded.*
3. *$\hookrightarrow^2$ is well-founded on terminating terms.*

Proof.

1. If $s \to_{\mathcal{E}\backslash\mathcal{S}} t$, then t is terminating by Corollary 22.2. If $s \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} t$ then t is clearly terminating if s is terminating. If $s \rhd_{\mathcal{E},\mu} t$ then t is terminating due to the inclusion $\rhd_{\mathcal{E},\mu} \circ \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \overset{\mathcal{S}}{\to}_{\mathit{Th}\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rhd_{\mathcal{E},\mu}$ from the proof of Theorem 46.

2. Assume that $\overline{\looparrowright}^2$ is not well-founded. Then, there exists an infinite $\rightarrow_{\mathcal{E}\backslash\mathcal{S}} \cup \rhd_{\mathcal{E}}$-sequence. Since $\rhd_{\mathcal{E}}$ is well-founded by Lemma 45.2, this sequence contains infinitely many $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$-steps. Using the inclusion $\rhd_{\mathcal{E}} \circ \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \subseteq \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \rhd_{\mathcal{E}}$ from the proof of Theorem 46, an infinite $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ sequence is obtained, contradicting the assumption that $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ is terminating.
3. Assume that $\looparrowright^2$ is not well-founded on terminating terms. Then, there exists an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \cup \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \cup \rhd_{\mathcal{E},\mu}$-sequence containing only terminating terms.
   If this sequence contains only finitely many $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps, then an infinite $\rightarrow_{\mathcal{E}\backslash\mathcal{S}} \cup \rhd_{\mathcal{E},\mu}$-sequence is obtained. Since $\rhd_{\mathcal{E},\mu}$ is well-founded by Lemma 45.2, this sequence contains infinitely many $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$-steps. Using the inclusion $\rhd_{\mathcal{E},\mu} \circ \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \subseteq \rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \rhd_{\mathcal{E},\mu}$ from the proof of Theorem 46, an infinite $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$-sequence is obtained, contradicting the assumption that $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ is terminating.
   Otherwise, the sequence contains infinitely many $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps. Using the inclusions $\rightarrow_{\mathcal{E}\backslash\mathcal{S}} \circ \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \xrightarrow{\mathcal{S}}{}^{+}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rightarrow^{=}_{\mathcal{E}\backslash\mathcal{S}}$ from Lemma 21.3 and $\rhd_{\mathcal{E},\mu} \circ \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \subseteq \xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \rhd_{\mathcal{E},\mu}$ from the proof of Theorem 46, an infinite $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-sequence starting with a terminating term is obtained, which is clearly impossible. □

The first property will be used freely in the following.

Proof (Lemma 66). The first claim is proved by induction on $\overline{\looparrowright}^2$, which, by Lemma 73.2, is well-founded.

If $t \in \mathcal{V}$ then $\overline{\mathcal{I}}^2(t) = t$ and nothing needs to be shown. If $t = f(t_1, \ldots, t_n)$ with $f \in \overline{\Delta}^2$ then $\overline{\mathcal{I}}^2(t) = f(\overline{\mathcal{I}}^2(t_1), \ldots, \overline{\mathcal{I}}^2(t_n))$ and the inductive assumption implies that $\overline{\mathcal{I}}^2(t_i)$ is a finite term for all $1 \leq i \leq n$. This implies that $\overline{\mathcal{I}}^2(t)$ is a finite term. Finally, assume $\text{root}(t) \notin \overline{\Delta}^2$. Then the sets $\overline{\mathcal{R}ed}^2_{\mathcal{S}}(t)$ and $\overline{\mathcal{E}q}^2_{\mathcal{E}}(t)$ are finite since $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ is finitely branching by Lemma 71 and $\mathcal{E}$ is size preserving, which implies that the $\mathcal{E}$-equivalence classes are finite. By the inductive assumption, $\overline{\mathcal{I}}^2(t')$ is a finite term for any $\overline{\mathcal{I}}^2(t') \in \overline{\mathcal{R}ed}^2_{\mathcal{S}}(t)$ and $\overline{\mathcal{I}}^2(t_i)$ is a finite term for any $g(\overline{\mathcal{I}}^2(t_1), \ldots, \overline{\mathcal{I}}^2(t_m)) \in \overline{\mathcal{E}q}^2_{\mathcal{E}}(t)$ and all $1 \leq i \leq m$. But then $\overline{\mathcal{I}}^2(t)$ is a finite term as well.

Similarly, the second claim is proved by induction on $\looparrowright^2$, which, by Lemma 73.3, is well-founded on terminating terms.

If $t \in \mathcal{V}$ then $\mathcal{I}^2(t) = t$ and nothing needs to be shown. If $t = f(t_1, \ldots, t_n)$ with $f \in \Delta^2$ then $\mathcal{I}^2(t) = f(\overline{t_1}, \ldots, \overline{t_n})$ and the inductive assumption implies that $\overline{t_i} = \mathcal{I}^2(t_i)$ is a finite term for all $1 \leq i \leq n$ with $i \in \mu(f)$. If $1 \leq i \leq n$ with $i \notin \mu(f)$, then $\overline{t_i} = \overline{\mathcal{I}}^2(t_i)$, which is a finite term by the first claim. Together, this implies that $\mathcal{I}^2(t)$ is a finite term. Finally, assume $\text{root}(t) \notin \Delta^2$. Then the sets $\mathcal{R}ed^2_{\mathcal{S}}(t)$, $\mathcal{R}ed^2_{\mathcal{R}}(t)$, $\mathcal{E}q^2_{\mathcal{E}}(t)$, and $\overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t)$ are finite since $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ and $\xrightarrow{\mathcal{S}}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$ are finitely branching by Lemma 71 and $\mathcal{E}$ is size preserving, which implies that the $\mathcal{E}$-equivalence classes are finite. By the inductive assumption, $\mathcal{I}^2(t')$ is a finite term for any $\mathcal{I}^2(t') \in \mathcal{R}ed^2_{\mathcal{S}}(t) \cup \mathcal{R}ed^2_{\mathcal{R}}(t)$ and $\overline{t_i}$ is a finite term for any $g(\overline{t_1}, \ldots, \overline{t_m}) \in \mathcal{E}q^2_{\mathcal{E}}(t)$ and all $1 \leq i \leq m$ as above. Furthermore, $\overline{\mathcal{I}}^2(s)$ is a finite term for all $\overline{\mathcal{I}}^2(s) \in \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t)$ by the first claim. But then $\mathcal{I}^2(t)$ is a finite term as well. □

Next, some simple notation is introduced.

Definition 74. *For any term t and any terminating substitution σ define the substitution σ_t as $\sigma_t(x) = \mathcal{I}^2(\sigma(x))$ if $x \in \mathcal{V}^\mu(t)$ and $\sigma_t(x) = \overline{\mathcal{I}}^2(\sigma(x))$ otherwise.*

Lemma 75. *If $s\sigma$ is terminating and $\mathcal{V}^\mu(s) \cap \mathcal{V}^{\not\mu}(s) = \emptyset$, then $s\sigma_s = [s, \sigma]$.*

Proof. Trivial. □

Proof (Lemma 67). Let $s, t \in \mathcal{T}(\mathcal{F} \cup \mathcal{F}_{\mathcal{T}h}, \mathcal{V})$ and let σ be a $\mathcal{T}h$-based substitution such that $s, t, s\sigma$ are terminating.

1. $\overline{\mathcal{I}}^2(s\sigma) = s\overline{\mathcal{I}}^2(\sigma)$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $\overline{\mathcal{I}}^2(\sigma)$. Otherwise, $s = f(s_1, \ldots, s_n)$ with $f \in \overline{\Delta}^2$. But then $\overline{\mathcal{I}}^2(s\sigma) = \overline{\mathcal{I}}^2(f(s_1\sigma, \ldots, s_n\sigma)) = f(\overline{\mathcal{I}}^2(s_1\sigma), \ldots, \overline{\mathcal{I}}^2(s_n\sigma)) = f(s_1\overline{\mathcal{I}}^2(\sigma), \ldots, s_n\overline{\mathcal{I}}^2(\sigma)) = s\overline{\mathcal{I}}^2(\sigma)$ by the inductive assumption.
2. $\mathcal{I}^2(s\sigma) = [s, \sigma]$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $[s, \sigma]$. Otherwise, $s = f(s_1, \ldots, s_n)$ with $f \in \Delta^2$ and thus $\mathcal{I}^2(s\sigma) = \mathcal{I}^2(f(s_1\sigma, \ldots, s_n\sigma)) = f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma})$. For $i \in \mu(f)$, the inductive assumption gives $\overline{s_i\sigma} = \mathcal{I}^2(s_i\sigma) = [s_i, \sigma]$. For $i \notin \mu(f)$, *1.* implies $\overline{s_i\sigma} = \overline{\mathcal{I}}^2(s_i\sigma) = s_i\overline{\mathcal{I}}^2(\sigma)$. Together this implies $\mathcal{I}^2(s\sigma) = f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma}) = [s, \sigma]$.
3. $\overline{\mathcal{I}}^2(s\sigma) \to^*_{\mathcal{R}_\Pi} s\overline{\mathcal{I}}^2(\sigma)$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $\overline{\mathcal{I}}^2(\sigma)$. Otherwise, $s = f(s_1, \ldots, s_n)$. If $f \in \overline{\Delta}^2$ then $\overline{\mathcal{I}}^2(s\sigma) = f(\overline{\mathcal{I}}^2(s_1\sigma), \ldots, \overline{\mathcal{I}}^2(s_n\sigma)) \to^*_{\mathcal{R}_\Pi} f(s_1\overline{\mathcal{I}}^2(\sigma), \ldots, s_n\overline{\mathcal{I}}^2(\sigma)) = s\overline{\mathcal{I}}^2(\sigma)$ by the inductive assumption. Otherwise, $\overline{\mathcal{I}}^2(s\sigma) = \mathcal{C}omp_{\mathrm{sort}(s\sigma)}(\overline{\mathcal{R}ed}^2_{\mathcal{S}}(s\sigma) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(s\sigma)))$. Notice that $f(\overline{\mathcal{I}}^2(s_1\sigma), \ldots, \overline{\mathcal{I}}^2(s_n\sigma)) \in \overline{\mathcal{E}q}^2_{\mathcal{E}}(s\sigma)$, which gives the desired $\overline{\mathcal{I}}^2(s\sigma) \to^+_{\mathcal{R}_\Pi} f(\overline{\mathcal{I}}^2(s_1\sigma), \ldots, \overline{\mathcal{I}}^2(s_n\sigma)) \to^*_{\mathcal{R}_\Pi} f(s_1\overline{\mathcal{I}}^2(\sigma), \ldots, s_n\overline{\mathcal{I}}^2(\sigma))$ using Lemma 59 and the inductive assumption.
4. $\mathcal{I}^2(s\sigma) \to^*_{\mathcal{R}_\Pi} [s, \sigma]$ is proved by induction on s. If $s \in \mathcal{V}$ then this is immediate by the definition of $[s, \sigma]$. Otherwise, $s = f(s_1, \ldots, s_n)$. If $f \in \Delta^2$, then $\mathcal{I}^2(s\sigma) = \mathcal{I}^2(f(s_1\sigma, \ldots, s_n\sigma)) = f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma})$. If $f \notin \Delta^2$, then $f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma}) \in \mathcal{E}q^2_{\mathcal{E}}(s\sigma)$ and thus $\mathcal{I}^2(s\sigma) \to^+_{\mathcal{R}_\Pi} f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma})$ by Lemma 59. For $i \in \mu(f)$, the inductive assumption implies $\overline{s_i\sigma} = \mathcal{I}^2(s_i\sigma) \to^*_{\mathcal{R}_\Pi} [s_i, \sigma]$. For $i \notin \mu(f)$, *3.* implies $\overline{s_i\sigma} = \overline{\mathcal{I}}^2(s_i\sigma) \to^*_{\mathcal{R}_\Pi} s_i\overline{\mathcal{I}}^2(\sigma)$. Together this implies $\mathcal{I}^2(s\sigma) \to^*_{\mathcal{R}_\Pi} f(\overline{s_1\sigma}, \ldots, \overline{s_n\sigma}) \to^*_{\mathcal{R}_\Pi} [s, \sigma]$.
5. The statement is proved by induction on s. If $s \in \mathcal{V}$ then $\mathcal{I}^2(s) = s = \overline{\mathcal{I}}^2(s)$. Otherwise, $s = f(s_1, \ldots, s_n)$. If $f \in \Delta^2 \subseteq \overline{\Delta}^2$, then by definition $\mathcal{I}^2(s) = \mathcal{I}^2(f(s_1, \ldots, s_n)) = f(\overline{s_1}, \ldots, \overline{s_n})$. For $i \in \mu(f)$, the inductive assumption implies $\overline{s_i} = \mathcal{I}^2(s_i) \to^*_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(s_i)$. If $i \notin \mu(f)$, then $\overline{s_i} = \overline{\mathcal{I}}^2(s_i)$ is immediate. Therefore, $\mathcal{I}^2(s) \to^*_{\mathcal{R}_\Pi} f(\overline{\mathcal{I}}^2(s_1), \ldots, \overline{\mathcal{I}}^2(s_n)) = \overline{\mathcal{I}}^2(s)$. If $f \notin \Delta^2$, then $\overline{\mathcal{I}}^2(s) \in \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(s)$ and thus $\mathcal{I}^2(s) \to^+_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(s)$ by Lemma 59.
6. It suffices to show that $s \vdash\!\dashv_{\mathcal{E}} t$ implies $\overline{\mathcal{I}}^2(s) \vdash\!\dashv^{=}_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(t)$ since the statement then follows by induction on the number of $\vdash\!\dashv_{\mathcal{E}}$-steps in $s \sim_{\mathcal{E}} t$.

Thus, let $s \vdash\dashv_{\mathcal{E}} t$ and perform an induction on the position p where the step takes places. If $\operatorname{root}(s) \notin \overline{\Delta}^2$, then $\operatorname{root}(t) \notin \overline{\Delta}^2$ as well by the definition of $\overline{\Delta}^2$. Since $s \sim_{\mathcal{E}} t$, the strong $\mathcal{E}$-coherence of $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ implies that whenever $s \rightarrow_{\mathcal{E}\backslash\mathcal{S}} s'$, then $t \rightarrow_{\mathcal{E}\backslash\mathcal{S}} t'$ for some $t' \sim_{\mathcal{E}} s'$. Thus, $\overline{\mathcal{R}ed}^2_{\mathcal{S}}(s) \subseteq \overline{\mathcal{R}ed}^2_{\mathcal{S}}(t)$. Similarly, if $s \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, then $t \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, which immediately implies $\overline{\mathcal{E}q}^2_{\mathcal{E}}(s) = \overline{\mathcal{E}q}^2_{\mathcal{E}}(t)$. Using these properties, $\overline{\mathcal{R}ed}^2_{\mathcal{S}}(s) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(s) \subseteq \overline{\mathcal{R}ed}^2_{\mathcal{S}}(t) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(t)$. Since the same reasoning can be applied with s and t interchanged, $\mathcal{C}omp_{\operatorname{sort}(s)}(\overline{\mathcal{R}ed}^2_{\mathcal{S}}(s) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(s)) = \mathcal{C}omp_{\operatorname{sort}(t)}(\overline{\mathcal{R}ed}^2_{\mathcal{S}}(t) \cup \overline{\mathcal{E}q}^2_{\mathcal{E}}(t))$ and thus $\overline{\mathcal{I}}^2(s) = \overline{\mathcal{I}}^2(t)$.
Otherwise, $\operatorname{root}(s) \in \overline{\Delta}^2$. If $p = \Lambda$, then there exist an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ and a substitution σ such that $s = u\sigma$ and $t = v\sigma$. By the definition of $\overline{\Delta}^2$, $u, v \in \mathcal{T}(\overline{\Delta}^2, \mathcal{V})$ and hence $u \approx v$ (or $v \approx u$) is in $\mathcal{E}(\overline{\Delta}^2) = \mathcal{E}(\Delta^2)$. Thus, $\overline{\mathcal{I}}^2(s) = \overline{\mathcal{I}}^2(u\sigma) = u\overline{\mathcal{I}}^2(\sigma) \vdash\dashv_{\mathcal{E}(\Delta^2)} v\overline{\mathcal{I}}^2(\sigma) = \overline{\mathcal{I}}^2(v\sigma) = \overline{\mathcal{I}}^2(t)$ by *1.* If $p \neq \Lambda$ then $s = f(s_1, \ldots, s_i, \ldots, s_n)$, $t = f(s_1, \ldots, t_i, \ldots, s_n)$, and $s_i \vdash\dashv_{\mathcal{E}} t_i$. Furthermore, $\overline{\mathcal{I}}^2(s) = f(\overline{\mathcal{I}}^2(s_1), \ldots, \overline{\mathcal{I}}^2(s_i), \ldots, \overline{\mathcal{I}}^2(s_n))$ and $\overline{\mathcal{I}}^2(t) = f(\overline{\mathcal{I}}^2(s_1), \ldots, \overline{\mathcal{I}}^2(t_i), \ldots, \overline{\mathcal{I}}^2(s_n))$. Now $\overline{\mathcal{I}}^2(s_i) \vdash\dashv^{=}_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(t_i)$ by the inductive assumption and thus $\overline{\mathcal{I}}^2(s) \vdash\dashv^{=}_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(t)$.

7. It suffices to show that $s \vdash\dashv_{\mathcal{E}} t$ implies $\mathcal{I}^2(s) \vdash\dashv^{=}_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t)$ since the statement then follows by induction on the number of $\vdash\dashv_{\mathcal{E}}$-steps in $s \sim_{\mathcal{E}} t$.
Thus, let $s \vdash\dashv_{\mathcal{E}} t$ and perform an induction on the position p where the step takes places. If $\operatorname{root}(s) \notin \Delta^2$, then $\operatorname{root}(t) \notin \Delta^2$ as well by the definition of Δ^2. Since $s \sim_{\mathcal{E}} t$, the strong $\mathcal{E}$-coherence of $\rightarrow_{\mathcal{E}\backslash\mathcal{S}}$ implies that whenever $s \rightarrow_{\mathcal{E}\backslash\mathcal{S}} s'$, then $t \rightarrow_{\mathcal{E}\backslash\mathcal{S}} t'$ for some $t' \sim_{\mathcal{E}} s'$. Thus, $\mathcal{R}ed^2_{\mathcal{S}}(s) \subseteq \mathcal{R}ed^2_{\mathcal{S}}(t)$. Similarly, Lemma 21.2 implies $\mathcal{R}ed^2_{\mathcal{R}}(s) \subseteq \mathcal{R}ed^2_{\mathcal{R}}(t)$. Finally, if $s \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, then $t \sim_{\mathcal{E}} g(s_1, \ldots, s_n)$, which immediately implies $\mathcal{E}q^2_{\mathcal{E}}(s) = \mathcal{E}q^2_{\mathcal{E}}(t)$. Also, $\overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(s) = \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t)$ since $s \sim_{\mathcal{E}} t$. Using these properties, $\mathcal{R}ed^2_{\mathcal{S}}(s) \cup \mathcal{R}ed^2_{\mathcal{R}}(s) \cup \mathcal{E}q^2_{\mathcal{E}}(s) \cup \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(s) \subseteq \mathcal{R}ed^2_{\mathcal{S}}(t) \cup \mathcal{R}ed^2_{\mathcal{R}}(t) \cup \mathcal{E}q^2_{\mathcal{E}}(t) \cup \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t)$. Since the same reasoning can be applied with s and t interchanged, $\mathcal{C}omp_{\operatorname{sort}(s)}(\mathcal{R}ed^2_{\mathcal{S}}(s) \cup \mathcal{R}ed^2_{\mathcal{R}}(s) \cup \mathcal{E}q^2_{\mathcal{E}}(s) \cup \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(s)) = \mathcal{C}omp_{\operatorname{sort}(t)}(\mathcal{R}ed^2_{\mathcal{S}}(t) \cup \mathcal{R}ed^2_{\mathcal{R}}(t) \cup \mathcal{E}q^2_{\mathcal{E}}(t) \cup \overline{\overline{\mathcal{E}q}}^2_{\mathcal{E}}(t))$ and thus $\mathcal{I}^2(s) = \mathcal{I}^2(t)$.
Otherwise, $\operatorname{root}(s) \in \Delta^2$. If $p = \Lambda$, then there exist an equation $u \approx v$ (or $v \approx u$) in $\mathcal{E}$ and a substitution σ such that $s = u\sigma$ and $t = v\sigma$. By the definition of Δ^2, $u, v \in \mathcal{T}(\Delta^2, \mathcal{V})$. Since $\mathcal{E}$ is strongly conservative, $u\sigma_u = [u, \sigma]$ and $v\sigma_v = [v, \sigma]$ by Lemma 75. Moreover, for all variables x, Definition 18.2a implies $\sigma_u(x) = \sigma_v(x)$, i.e., $\sigma_u = \sigma_v$. Hence,

$$\begin{array}{rll}
\mathcal{I}^2(s) & = & \mathcal{I}^2(u\sigma) \\
& = & [u, \sigma] \qquad \text{by } \mathit{2.} \\
& = & u\sigma_u \\
& = & u\sigma_v \\
& \vdash\dashv_{\mathcal{E}(\Delta^2)} & v\sigma_v \\
& = & [v, \sigma] \\
& = & \mathcal{I}^2(v\sigma) \qquad \text{by } \mathit{2.}
\end{array}$$

$$= \mathcal{I}^2(t)$$

If $p \neq \Lambda$ then $s = f(s_1, \ldots, s_i, \ldots, s_n)$, $t = f(s_1, \ldots, t_i, \ldots, s_n)$, and $s_i \mathrel{\vdash\!\dashv}_{\mathcal{E}} t_i$. Notice that $\mathcal{I}^2(s) = f(\overline{s_1}, \ldots, \overline{s_i}, \ldots, \overline{s_n})$ and $\mathcal{I}^2(t) = f(\overline{s_1}, \ldots, \overline{t_i}, \ldots, \overline{s_n})$. If $i \in \mu(f)$, then $\overline{s_i} = \mathcal{I}^2(s_i) \mathrel{\vdash\!\dashv}^{=}_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t_i) = \overline{t_i}$ by the inductive assumption. If $i \notin \mu(f)$, then $\overline{s_i} = \overline{\mathcal{I}}^2(s_i) \mathrel{\vdash\!\dashv}^{=}_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(t_i) = \overline{t_i}$ by *6*. Thus, $\mathcal{I}^2(s) \mathrel{\vdash\!\dashv}^{=}_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t)$ in either case.

8. It suffices to show that $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ implies $\overline{\mathcal{I}}^2(s) \leadsto_1 \overline{\mathcal{I}}^2(t)$ since the statement then follows by induction on the number of $\to_{\mathcal{E}\backslash\mathcal{S}}$-steps in $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$.
   Thus, let $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ and perform an induction on the position p where the reduction takes places. If $\mathrm{root}(s) \notin \overline{\Delta}^2$, then $\overline{\mathcal{I}}^2(t) \in \overline{\mathcal{R}ed}^2_{\mathcal{S}}(s)$, which implies $\overline{\mathcal{I}}^2(s) \to^+_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(t)$ by Lemma 59.
   If $\mathrm{root}(s) \in \overline{\Delta}^2$, first consider the case $p = \Lambda$. Then, there exist a rule $l \to r \in \mathcal{S}$ and a substitution σ such that $s \sim_{\mathcal{E}} l\sigma \to_{\mathcal{S}} r\sigma = t$. Since $\mathrm{root}(s) \in \overline{\Delta}^2$, the definition of $\overline{\Delta}^2$ implies that $\mathrm{root}(l) \in \overline{\Delta}^2$, $l \to r \in \mathcal{S}(\overline{\Delta}^2) = \mathcal{S}(\Delta^2)$, and $r \in \mathcal{T}(\overline{\Delta}^2, \mathcal{V})$. Using *1.*, *3.*, and *6.*, $\overline{\mathcal{I}}^2(s) \sim_{\mathcal{E}(\Delta^2)} \overline{\mathcal{I}}^2(l\sigma) \to^*_{\mathcal{R}_\Pi} l\overline{\mathcal{I}}^2(\sigma) \to_{\mathcal{S}(\Delta^2)} r\overline{\mathcal{I}}^2(\sigma) = \overline{\mathcal{I}}^2(r\sigma) = \overline{\mathcal{I}}^2(t)$, and thus $\overline{\mathcal{I}}^2(s) \leadsto_1 \overline{\mathcal{I}}^2(t)$. If $\mathrm{root}(s) \in \overline{\Delta}^2$ and $p \neq \Lambda$, then $s = f(s_1, \ldots, s_i, \ldots, s_n)$ and $t = f(s_1, \ldots, t_i, \ldots, s_n)$, where $s_i \to_{\mathcal{E}\backslash\mathcal{S}} t_i$. By the inductive assumption, $\overline{\mathcal{I}}^2(s_i) \leadsto_1 \overline{\mathcal{I}}^2(t_i)$ and therefore $\overline{\mathcal{I}}^2(s) = f(\ldots, \overline{\mathcal{I}}^2(s_i), \ldots) \leadsto_1 f(\ldots, \overline{\mathcal{I}}^2(t_i), \ldots) = \overline{\mathcal{I}}^2(t)$.
9. It suffices to show that $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ implies $\mathcal{I}^2(s) \leadsto_1 \mathcal{I}^2(t)$ since the statement then follows by induction on the number of $\to_{\mathcal{E}\backslash\mathcal{S}}$-steps in $s \to^*_{\mathcal{E}\backslash\mathcal{S}} t$.
   Thus, let $s \to_{\mathcal{E}\backslash\mathcal{S}} t$ and perform an induction on the position p where the reduction takes places. If $\mathrm{root}(s) \notin \Delta^2$, then $\mathcal{I}^2(t) \in \mathcal{R}ed^2_{\mathcal{S}}(s)$, which implies $\mathcal{I}^2(s) \to^+_{\mathcal{R}_\Pi} \mathcal{I}^2(t)$ by Lemma 59.
   If $\mathrm{root}(s) \in \Delta^2$, first consider the case $p = \Lambda$. Then, there exist a rule $l \to r \in \mathcal{S}$ and a substitution σ such that $s \sim_{\mathcal{E}} l\sigma \to_{\mathcal{S}} r\sigma = t$. By the definition of Δ^2, $\mathrm{root}(l) \in \Delta^2$, $l \to r \in \mathcal{S}(\Delta^2)$, and $r \in \mathcal{T}(\Delta^2, \mathcal{V})$. Since $\mathcal{S}(\Delta^2)$ is strongly conservative, $l\sigma_l = [l, \sigma]$ and $r\sigma_r = [r, \sigma]$ by Lemma 75. Moreover, $\sigma_l(x) \to^*_{\mathcal{R}_\Pi} \sigma_r(x)$ for all variables x. To see this, notice that by strong conservativeness of $\mathcal{S}(\Delta^2)$, the substitutions σ_l and σ_r differ at most on variables $x \in \mathcal{V}^\mu(l) - \mathcal{V}^\mu(r)$. For these variables $\sigma_l(x) = \mathcal{I}^2(\sigma(x)) \to^*_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(\sigma(x)) = \sigma_r(x)$ by *5*. Hence,

$$\begin{array}{rll}
\mathcal{I}^2(s) \sim_{\mathcal{E}(\Delta^2)} & \mathcal{I}^2(l\sigma) & \text{by } \textit{7.} \\
\to^*_{\mathcal{R}_\Pi} & [l, \sigma] & \text{by } \textit{4.} \\
= & l\sigma_l & \\
\to^*_{\mathcal{R}_\Pi} & l\sigma_r & \\
\to_{\mathcal{S}(\Delta^2)} & r\sigma_r & \\
= & [r, \sigma] & \\
= & \mathcal{I}^2(r\sigma) & \text{by } \textit{2.} \\
= & \mathcal{I}^2(t) &
\end{array}$$

   If $p \neq \Lambda$, then $s = f(s_1, \ldots, s_i, \ldots, s_n)$, $t = f(s_1, \ldots, t_i, \ldots, s_n)$, and $s_i \to_{\mathcal{E}\backslash\mathcal{S}} t_i$. Also, $\mathcal{I}^2(s) = f(\overline{s_1}, \ldots, \overline{s_i}, \ldots, \overline{s_n})$ and $\mathcal{I}^2(t) = f(\overline{s_1}, \ldots, \overline{t_i}, \ldots, \overline{s_n})$. If $i \in \mu(f)$, then $\overline{s_i} = \mathcal{I}^2(s_i) \leadsto_1 \mathcal{I}^2(t_i) = \overline{t_i}$ by the inductive assumption. If

$i \notin \mu(f)$, then $\overline{s_i} = \overline{\mathcal{I}}^2(s_i) \rightsquigarrow_1 \overline{\mathcal{I}}^2(t_i) = \overline{t_i}$ by *8.* Thus, $\mathcal{I}^2(s) \rightsquigarrow_1 \mathcal{I}^2(t)$ in either case.

10. It suffices to show that $s \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$ implies $\mathcal{I}^2(s) \rightsquigarrow_2 \mathcal{I}^2(t)$ since the statement then follows by induction on the number of $\overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu}$-steps in $s \overset{\mathcal{S}}{\to}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$.

    Thus, let $s \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t$ and perform an induction on the position p where the reduction takes places. If $\text{root}(s) \notin \Delta^2$, then $\mathcal{I}^2(t) \in \mathcal{R}ed^1_{\mathcal{R}}(s)$, which implies $\mathcal{I}^2(s) \to^+_{\mathcal{R}_\Pi} \mathcal{I}^2(t)$ by Lemma 59.

    If $\text{root}(s) \in \Delta^2$, first consider the case $p = \Lambda$. Then, there exist a rule $l \to r\llbracket\varphi\rrbracket \in \mathcal{R}$ and a $\mathcal{T}h$-based substitution σ with $s = f(s^*) \xrightarrow{>\Lambda}{}^!_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} l\sigma \to_{\mathcal{R}} r\sigma = t$ such that $\varphi\sigma$ is $\mathcal{T}h$-valid. Since $\text{root}(l) = \text{root}(s) = f$ and $f \in \Delta^2$, the definitions of $\overline{\Delta}^2$ and Δ^2 implies that $l \to r\llbracket\varphi\rrbracket \in \mathcal{R}(\Delta^2)$, $r \in \mathcal{F}(\overline{\Delta}^2, \mathcal{V})$, and $\mathcal{F}^\mu(r) \subseteq \Delta^2$. Since $\mathcal{R}(\Delta^2)$ is strongly conservative, $l\sigma_l = [l, \sigma]$ and $r\sigma_r = [r, \sigma]$ by Lemma 75. Moreover, $\sigma_l(x) \to^*_{\mathcal{R}_\Pi} \sigma_r(x)$ for all variables x. To see this, notice that by strong conservativeness of $\mathcal{R}(\Delta^2)$, the substitutions σ_l and σ_r differ at most on variables $x \in \mathcal{V}^\mu(l) - \mathcal{V}^\mu(r)$. For these variables, $\sigma_l(x) = \mathcal{I}^2(\sigma(x)) \to^*_{\mathcal{R}_\Pi} \overline{\mathcal{I}}^2(\sigma(x)) = \sigma_r(x)$ by *5.* Also, notice that $\varphi\sigma_r$ is $\mathcal{T}h$-valid since $\sigma_r(x) = \sigma(x)$ for variables of sort `base` since σ is $\mathcal{T}h$-based. Hence,

$$\begin{array}{rll}
\mathcal{I}^2(s) \rightsquigarrow_1^* \circ \sim_{\mathcal{E}(\Delta^2)} & \mathcal{I}^2(l\sigma) & \text{by } \textit{7.} \text{ and } \textit{9.} \\
\to^*_{\mathcal{R}_\Pi} & [l, \sigma] & \text{by } \textit{4.} \\
= & l\sigma_l & \\
\to^*_{\mathcal{R}_\Pi} & l\sigma_r & \\
\to_{\mathcal{R}(\Delta^2),\mu} & r\sigma_r & \\
= & [r, \sigma] & \\
= & \mathcal{I}^2(r\sigma) & \text{by } \textit{2.} \\
= & \mathcal{I}^2(t) &
\end{array}$$

    If $p \neq \Lambda$, then $s = f(s_1, \ldots, s_i, \ldots, s_n)$, $t = f(s_1, \ldots, t_i, \ldots, s_n)$, with $i \in \mu(f)$ such that $s_i \overset{\mathcal{S}}{\to}_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} t_i$. Also, $\mathcal{I}^2(s) = f(\overline{s_1}, \ldots, \overline{s_i}, \ldots, \overline{s_n})$ and $\mathcal{I}^2(t) = f(\overline{s_1}, \ldots, \overline{t_i}, \ldots, \overline{s_n})$ where $\overline{s_i} = \mathcal{I}^2(s_i) \rightsquigarrow_2 \mathcal{I}^2(t_i) = \overline{t_i}$ by the inductive assumption.

11. If $s \overset{\mathcal{S}}{\to}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \to^!_{\mathcal{E}\backslash\mathcal{S}} \circ \sim_{\mathcal{E}} t$, then $\mathcal{I}^2(s) \rightsquigarrow_2^* \circ \rightsquigarrow_1^* \circ \sim_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(t)$ using *7.*, *9.*, and *10.* □

Proof (Theorem 68). In the second case soundness is obvious. Otherwise, it needs to be shown that every infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain only contains finitely many dependency pairs from $\mathcal{P}'$. Thus, assume $s_1 \to t_1\llbracket\varphi_1\rrbracket, s_2 \to t_2\llbracket\varphi_2\rrbracket, \ldots$ is an infinite minimal $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mathcal{E}, \mu)$-chain with the $\mathcal{T}h$-based substitution σ. This means that $t_i\sigma \overset{\mathcal{S}}{\to}^*_{\mathcal{T}h\|\mathcal{E}\backslash\mathcal{R},\mu} \circ \xrightarrow{>\Lambda}{}^!_{\mathcal{E}\backslash\mathcal{S}} \circ \overset{>\Lambda}{\sim}_{\mathcal{E}} s_{i+1}\sigma$ and $\varphi_i\sigma$ is $\mathcal{T}h$-valid for all $i \geq 1$. By the definition of Δ^2, $\text{root}(t'_i) \in \Delta^2$ for all $t'_i \notin \mathcal{V}$ such that $t_i \rhd_\mu t'_i$. Similarly to the proof of Lemma 67.11 and using Lemma 67, $t_i\sigma_{t_i} = [t_i, \sigma] = \mathcal{I}^2(t_i\sigma) \rightsquigarrow_2^* \circ \rightsquigarrow_1^* \circ \sim_{\mathcal{E}(\Delta^2)} \mathcal{I}^2(s_{i+1}\sigma) \to^*_{\mathcal{R}_\Pi} [s_{i+1}, \sigma] = s_{i+1}\sigma_{s_{i+1}} \to^*_{\mathcal{R}_\Pi} s_{i+1}\sigma_{t_{i+1}}$ since $\mathcal{P}$ is strongly conservative. The remaining proof is now standard. □

References

1. Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, and René Thiemann. Improving context-sensitive dependency pairs. In Iliano Cervesato, Helmut Veith, and Andrei Voronkov, editors, *Proceedings of the 15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR '08)*, volume 5330 of *Lecture Notes in Artificial Intelligence*, pages 636–651. Springer-Verlag, 2008.
2. Beatriz Alarcón, Raúl Gutiérrez, and Salvador Lucas. Context-sensitive dependency pairs. In S. Arun-Kumar and Naveen Garg, editors, *Proceedings of the 26th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '06)*, volume 4337 of *Lecture Notes in Computer Science*, pages 297–308. Springer-Verlag, 2006.
3. Beatriz Alarcón, Raúl Gutiérrez, and Salvador Lucas. Improving the context-sensitive dependency graph. *Electronic Notes in Theoretical Computer Science*, 188:91–103, 2007.
4. Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1–2):133–178, 2000.
5. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.
6. Leo Bachmair and Nachum Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2–3):173–201, 1989.
7. Cristina Borralleras, Salvador Lucas, and Albert Rubio. Recursive path orderings can be context-sensitive. In Andrei Voronkov, editor, *Proceedings of the 18th International Conference on Automated Deduction (CADE '02)*, volume 2392 of *Lecture Notes in Artificial Intelligence*, pages 314–331. Springer-Verlag, 2002.
8. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All About Maude–A High-Performance Logical Framework*, volume 4350 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.
9. Agostino Dovier, Carla Piazza, and Gianfranco Rossi. A uniform approach to constraint-solving for lists, multisets, compact lists, and sets. *ACM Transactions on Computational Logic*, 9(3), 2008.
10. Agostino Dovier, Alberto Policriti, and Gianfranco Rossi. A uniform axiomatic view of lists, multisets, and sets, and the relevant unification algorithms. *Fundamenta Informaticae*, 36(2–3):201–234, 1998.
11. Stephan Falke. *Term Rewriting with Built-In Numbers and Collection Data Structures*. PhD thesis, University of New Mexico, Albuquerque, NM, USA, 2009. To appear.
12. Stephan Falke and Deepak Kapur. Dependency pairs for rewriting with non-free constructors. In Frank Pfenning, editor, *Proceedings of the 21st International Conference on Automated Deduction (CADE '07)*, volume 4603 of *Lecture Notes in Artificial Intelligence*, pages 426–442. Springer-Verlag, 2007.
13. Stephan Falke and Deepak Kapur. Dependency pairs for rewriting with built-in numbers and semantic data structures. In Voronkov [29], pages 94–109.
14. Maria C. F. Ferreira and A. L. Ribeiro. Context-sensitive AC-rewriting. In Paliath Narendran and Michaël Rusinowitch, editors, *Proceedings of the 10th International Conference on Rewriting Techniques and Applications (RTA '99)*, volume 1631 of *Lecture Notes in Computer Science*, pages 286–300. Springer-Verlag, 1999.

15. Jürgen Giesl and Aart Middeldorp. Transformation techniques for context-sensitive rewrite systems. *Journal of Functional Programming*, 14(4):379–427, 2004.
16. Jürgen Giesl, Peter Schneider-Kamp, and René Thiemann. AProVE 1.2: Automatic termination proofs in the dependency pair framework. In Ulrich Furbach and Natarajan Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR '06)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 281–286. Springer-Verlag, 2006.
17. Jürgen Giesl, René Thiemann, and Peter Schneider-Kamp. The dependency pair framework: Combining techniques for automated termination proofs. In Franz Baader and Andrei Voronkov, editors, *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR '04)*, volume 3452 of *Lecture Notes in Artificial Intelligence*, pages 301–331. Springer-Verlag, 2005.
18. Raúl Gutiérrez, Salvador Lucas, and Xavier Urbain. Usable rules for context-sensitive rewrite systems. In Voronkov [29], pages 126–141.
19. Nao Hirokawa and Aart Middeldorp. Tyrolean termination tool: Techniques and features. *Information and Computation*, 205(4):474–511, 2007.
20. Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, 1986.
21. Keiichirou Kusakari, Masaki Nakamura, and Yoshihito Toyama. Argument filtering transformation. In Gopalan Nadathur, editor, *Proceedings of the 1st International Conference on Principles and Practice of Declarative Programming (PPDP '99)*, volume 1702 of *Lecture Notes in Computer Science*, pages 47–61. Springer-Verlag, 1999.
22. Salvador Lucas. Context-sensitive computations in functional and functional logic programs. *Journal of Functional and Logic Programming*, 1998(1), 1998.
23. Salvador Lucas. Context-sensitive rewriting strategies. *Information and Computation*, 178(1):294–343, 2002.
24. Salvador Lucas. Polynomials for proving termination of context-sensitive rewriting. In Igor Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS '04)*, volume 2987 of *Lecture Notes in Computer Science*, pages 318–332. Springer-Verlag, 2004.
25. Salvador Lucas. Proving termination of context-sensitive rewriting by transformation. *Information and Computation*, 204(12):1782–1846, 2006.
26. Claude Marché. Normalized rewriting: An alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation*, 21(3):253–288, 1996.
27. Mojzesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du Premier Congrès de Mathématiciens des Pays Slaves*, pages 92–101, 1929.
28. TPDB. Termination problem data base 5.0.2, 2008. Available from `http://dev.aspsimon.org/projects/termcomp/downloads/`.
29. Andrei Voronkov, editor. *Proceedings of the 19th International Conference on Rewriting Techniques and Applications (RTA '08)*, volume 5117 of *Lecture Notes in Computer Science*. Springer-Verlag, 2008.