Towards an Incremental and Modular Termination Analysis of Context-Sensitive Rewriting Systems (Work in Progress)*

Raúl Gutiérrez
DSIC, Universitat Politècnica de València, Spain
rgutierrez@dsic.upv.es

Salvador Lucas
DSIC, Universitat Politècnica de València, Spain
slucas@dsic.upv.es

Modularity is essential in software development, where a piece of software is often designed and implemented as a composition of simpler modules. So, if we want to prove that a program satisfies a given property, a modular approach becomes natural. With the development and successful use of the Dependency Pair Framework, which rather focuses on the *decomposition* of termination problems, less attention has been payed to modularity issues (which rather require the opposite approach). But the modular analysis of termination is still paramount for software developers. In this paper, we analyze modularity of context-sensitive rewrite systems. A modularity analysis was carried out by Gramlich and Lucas in 2002, but a correct notion of context-sensitive dependency pair (CS-DP) was not obtained until 2006. In this paper, we analyze modularity using CS-DPs.

1 Introduction

Term rewriting systems (TRSs) with many rules are frequently specified following a modular and incremental pattern and using well-known constructions such as *if-then-else* or *while* statements or generic modules (mathematical operands, functions that operate with lists, etc.) that are combined and reused many times to obtain the final program. When we try to prove computational properties on these systems with many rules, it is helpful to get use of the modular decomposition given by the developer to check properties by decomposition.

Termination is a fundamental property in programming languages, which allows us to know if for every computation the system will return in a finite time. The main problem dealing with termination from a modular perspective is that termination is not modular, even the union of two terminating TRSs that share no function symbol can be a non-terminating TRS, as shown by Toyama in 1987 [16].

Example 1 ([16]) *Toyama's example:*

$$\mathcal{R}_1 = \{ \quad \mathsf{f}(0,1,x) \;\to\; \mathsf{f}(x,x,x)\} \qquad \mathcal{R}_2 = \left\{ \begin{array}{lcl} \mathsf{c}(x,y) & \to & x \\ \mathsf{c}(x,y) & \to & y \end{array}\right\}$$

The TRS $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ resulting from the union of these two terminating TRSs can generate the following infinite rewrite sequence:

$$\mathsf{f}(\mathsf{c}(0,1),\mathsf{c}(0,1),\mathsf{c}(0,1)) \to_{\mathcal{R}_2} \mathsf{f}(0,\mathsf{c}(0,1),\mathsf{c}(0,1)) \to_{\mathcal{R}_2} \mathsf{f}(0,1,\mathsf{c}(0,1)) \to_{\mathcal{R}_1} \mathsf{f}(\mathsf{c}(0,1),\mathsf{c}(0,1),\mathsf{c}(0,1)) \to_{\mathcal{R}_2} \cdots$$

*Partially supported by the EU (FEDER), MINECO projects TIN2010-21062-C02-02 and TIN 2013-45732-C4-1-P, and project PROMETEO/2011/052. Salvador Lucas' research was developed during a sabbatical year at the CS Dept. of the UIUC and was also partially supported by NSF grant CNS 13-19109, Spanish MECD grant PRX12/00214, and GV grant BEST/2014/026. Raúl Gutiérrez is also partially supported by a Juan de la Cierva Fellowship from the Spanish MINECO, ref. JCI-2012-13528.

Submitted to:
PROLE 2014

This problem appears when combining *duplicating* and *collapsing* rules. A rule is duplicating if the number of occurrences of a variable in the right-hand side is greater than in the left-hand side, and a rule is collapsing if its right-hand side is a variable.

To obtain a modular analysis of termination, a more restrictive notion of termination is imposed: $\mathscr{C}_\varepsilon$-termination [10]. A system is $\mathscr{C}_\varepsilon$-terminating if it is still terminating after adding the rules in $\mathscr{R}_2$, called $\mathscr{C}_\varepsilon$-rules. These rules are used to simulate the behavior of the problematic collapsing rules. $\mathscr{C}_\varepsilon$-termination is a modular property for constructor sharing TRSs.

Context-sensitive rewriting (CSR [14, 15]) extends the signature of a rewrite system with a *replacement map*. A *replacement map* is a mapping $\mu : \mathscr{F} \to \wp(\mathbb{N})$ satisfying $\mu(f) \subseteq \{1,\ldots,\mathsf{ar}(f)\}$ for every function symbol f in the signature $\mathscr{F}$ [14], where $\mathsf{ar}(f)$ means the arity of f. We use it to discriminate the argument positions in which the rewriting steps are *allowed*; rewriting at the topmost position is always possible. In this way, a restriction of the rewrite relation is obtained. CSR has shown useful to model evaluation strategies in programming languages and also to achieve a terminating behaviour by pruning (all) infinite rewrite sequences. In particular, it is an essential ingredient to analyze the *termination behavior* of programming languages like CafeOBJ, Maude, OBJ, etc. [8, 12].

Example 2 *Consider the following context-sensitive term rewriting system (CS-TRS) from [9] that computes the list of all prime numers using the sieve of Eratosthenes in a lazy way:*

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{if}(\mathsf{true},x,y) & \to & x & \mathsf{sieve}(x{:}y) & \to & x{:}\mathsf{sieve}(\mathsf{filter}(x,y))\\
\mathsf{if}(\mathsf{false},x,y) & \to & y & \mathsf{from}(x) & \to & x{:}\mathsf{from}(\mathsf{s}(x))\\
\mathsf{filter}(x,y{:}z) & \to & \mathsf{if}(\mathsf{divides}(x,y),\mathsf{filter}(x,z),y{:}\mathsf{filter}(x,z)) & \mathsf{primes} & \to & \mathsf{sieve}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0))))
\end{array}$$

together with $\mu(\mathsf{if}) = \mu(:) = \{1\}$ and $\mu(f) = \{1,\ldots,\mathsf{ar}(f)\}$ for all other symbols f. Function from *is used to generate an infinite list of natural numbers and function* sieve *filters those that are primes. The replacement restriction on the second argument of* $(:)$ *avoids an infinite computation. This system was proved μ-terminating in [2] showing that the dependency graph has no cycles. The reason is that* divides *rule was not defined and the term* $\mathsf{divides}(\mathsf{s}(\mathsf{s}(x)),y)$ *could not be rewritten to* true *or* false*. But, if we define* divides *in a standard way:*

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{zero}(0) & \to & \mathsf{true} & x-0 & \to & x\\
\mathsf{zero}(\mathsf{s}(x)) & \to & \mathsf{false} & \mathsf{s}(x)-\mathsf{s}(y) & \to & x-y\\
\mathsf{mod}(0,\mathsf{s}(x)) & \to & 0 & \mathsf{s}(x)\leq 0 & \to & \mathsf{false}\\
\mathsf{mod}(\mathsf{s}(x),\mathsf{s}(y)) & \to & \mathsf{if}(y\leq x,\mathsf{mod}(x-y,\mathsf{s}(y)),\mathsf{s}(x)) & 0\leq x & \to & \mathsf{true}\\
\mathsf{divides}(x,y) & \to & \mathsf{zero}(\mathsf{mod}(y,x)) & \mathsf{s}(x)\leq\mathsf{s}(y) & \to & x\leq y
\end{array}$$

There is no tool for proving termination that can prove this system μ-terminating although we know that these new rules are μ-terminating.

In [11], a modularity analysis of termination of CSR was carried out, but since there was no correct definition of CS-DP until [1], a modular analysis of termination based on CS-DPs was not possible. In this paper, we exploit the modular behaviour of CS-DPs to obtain modularity results from a different perspective of the obtained by Gramlich and Lucas.

2 Preliminaries

See [7] and [14] for basics on term rewriting and CSR, respectively. Throughout the paper, $\mathscr{X}$ denotes a countable set of variables and $\mathscr{F}$ denotes a signature, i.e., a set of function symbols each having a fixed arity given by a mapping $\mathsf{ar} : \mathscr{F} \to \mathbb{N}$. The set of terms built from $\mathscr{F}$ and $\mathscr{X}$ is $\mathscr{T}(\mathscr{F},\mathscr{X})$. Terms are viewed as labelled trees in the usual way. The symbol labeling the root of the term s is denoted as $root(s)$. Positions $p,q,\ldots$ are represented by chains of positive natural numbers used to address subterms of s. Given positions p,q, we denote its concatenation as $p.q$. We denote the empty chain by Λ. The set of positions of a term s is $\mathscr{P}os(s)$. For a replacement map μ, the set of *active positions* $\mathscr{P}os^{\mu}(s)$ of $s \in \mathscr{T}(\mathscr{F},\mathscr{X})$ is: $\mathscr{P}os^{\mu}(s) = \{\Lambda\}$, if $s \in \mathscr{X}$ and $\mathscr{P}os^{\mu}(s) = \{\Lambda\} \cup \bigcup_{i\in\mu(root(s))} i.\mathscr{P}os^{\mu}(s|_i)$, if $t \notin \mathscr{X}$. Let $\mathscr{V}ar(s) = \{x \in \mathscr{X} \mid \exists p \in \mathscr{P}os(s), s|_p = x\}$, $\mathscr{V}ar^{\mu}(s) = \{x \in \mathscr{V}ar(s) \mid \exists p \in \mathscr{P}os^{\mu}(s), s|_p = x\}$ and $\mathscr{V}ar^{\not\mu}(s) = \{x \in \mathscr{V}ar(s) \mid s \rhd_{\not\mu} x\}$. We say that $s \trianglerighteq_{\mu} t$ if there is $p \in \mathscr{P}os^{\mu}(s)$ such that $t = s|_p$. We write $s \rhd_{\mu} t$ if $s \trianglerighteq_{\mu} t$ and $s \neq t$. Moreover, $s \rhd_{\not\mu} t$ if there is a *frozen position* p, i.e. $p \in \mathscr{P}os(s) - Pos^{\mu}(s)$, such that $t = s|_p$. A rewrite rule is an ordered pair (ℓ,r), written $\ell \to r$, with $\ell, r \in \mathscr{T}(\mathscr{F},\mathscr{X})$, $\ell \notin \mathscr{X}$ and $\mathscr{V}ar(r) \subseteq \mathscr{V}ar(\ell)$. A TRS is a pair $\mathscr{R} = (\mathscr{F},R)$ where R is a set of rewrite rules. Given $\mathscr{R} = (\mathscr{F},R)$, we consider $\mathscr{F}$ as the disjoint union $\mathscr{F} = \mathscr{C} \uplus \mathscr{D}$ of symbols $\mathsf{c} \in \mathscr{C}$, called *constructors* and symbols $\mathsf{f} \in \mathscr{D}$, called *defined functions*, where $\mathscr{D} = \{\mathsf{root}(\ell) \mid \ell \to r \in R\}$ and $\mathscr{C} = \mathscr{F} - \mathscr{D}$. We say that $t \in \mathscr{T}(\mathscr{F},\mathscr{X}) - \mathscr{X}$ is a *hidden term* of s, $t \in \mathsf{HT}(s,\mu)$, if $s \rhd_{\not\mu} t$ and $\mathsf{root}(t)$ is defined. Given a CS-TRS $(\mathscr{R},\mu)$, we have $s \hookrightarrow_{\mathscr{R},\mu} t$ (alternative $s \overset{p}{\hookrightarrow}_{\mathscr{R},\mu} t$ if we want to make the position explicit) if there are $\ell \to r \in \mathscr{R}$, $p \in \mathscr{P}os^{\mu}(s)$ and a substitution σ with $s|_p = \sigma(\ell)$ and $t = t[\sigma(r)]_p$. We write $s \overset{>q}{\hookrightarrow}_{\mathscr{R},\mu} t$ if $s \overset{p}{\hookrightarrow}_{\mathscr{R},\mu} t$ and $q > p$. A rule $\ell \to r$ is *conservative* if $\mathscr{V}ar^{\mu}(r) \subseteq \mathscr{V}ar^{\mu}(\ell)$. A rule $\ell \to r$ is *strongly conservative* if it is conservative and $\mathscr{V}ar^{\mu}(\ell) \cap \mathscr{V}ar^{\not\mu}(\ell) = \mathscr{V}ar^{\mu}(r) \cap \mathscr{V}ar^{\not\mu}(r) = \varnothing$. $\mathscr{R}$ is conservative (resp. strongly conservative) if all rules in $\mathscr{R}$ are. A CS-TRS $(\mathscr{R},\mu)$ is *terminating* if $\hookrightarrow_{\mathscr{R},\mu}$ is well-founded. A CS-TRS $(\mathscr{R},\mu)$ is $\mathscr{C}_{\varepsilon}$-terminating [11] iff $(\mathscr{R} \uplus \mathscr{C}_{\varepsilon}, \mu \uplus \mu_{\mathscr{C}_{\varepsilon}})$ is terminating, where $\mathscr{C}_{\varepsilon} = (\{\mathsf{c}\}, \{\mathsf{c}(x,y) \to x, \mathsf{c}(x,y) \to y\})$ (with c being a fresh symbol) and $\mu_{\mathscr{C}_{\varepsilon}}(\mathsf{c}) = \{1,2\}$.

2.1 Context Sensitive Dependency Pairs

Dependency pairs [6] describe the *propagation* of *function calls* in rewrite sequences. In CSR, we have two kind of (potential) function calls: *direct* calls, i.e., calls at *active* (*replacing*) positions and *delayed* calls, i.e., calls at *frozen* (*non-replacing*) positions that can be activated in forthcoming reduction steps. These function calls are captured in two different ways. For rules $\ell \to r$ such that r contains some *defined* symbol g at an active position, the function call to g is represented as a new rule $u \to v$ (called *dependency pair*) where $u = \mathsf{f}^{\sharp}(\ell_1,\ldots,\ell_k)$ if $\ell = \mathsf{f}(\ell_1,\ldots,\ell_k)$ and $v = \mathsf{g}^{\sharp}(s_1,\ldots,s_m)$ if $\mathsf{g}(s_1,\ldots,s_m)$ is an active subterm of r and g is defined. The notation $\mathsf{f}^{\sharp}$ for a given symbol f means that f is *marked*. In practice, we often capitalize f and use F instead of $\mathsf{f}^{\sharp}$ in our examples. Function calls to g which are at frozen positions of r cannot be issued 'immediately', but could be activated 'in the future'. This situation is carried out by the *migrating variables* and modeled by *collapsing* DPs. Given a rule $\ell \to r$, x is a *migrating variable* if x is at an active position in r but not in ℓ [1]. For rules $\ell \to r$, *collapsing* DPs are pairs of the form $u \to x$ where $u = \mathsf{f}^{\sharp}(\ell_1,\ldots,\ell_k)$ if $\ell = \mathsf{f}(\ell_1,\ldots,\ell_k)$ and x is a *migrating variable*. The idea is that calls which can eventually be activated are subterms of $\sigma(x)$ for σ being the matching substitution of the rewriting step involving the rule $\ell \to r$. Formally, $\mathsf{DP}(\mathscr{R},\mu) = \mathsf{DP}_{\mathscr{F}}(\mathscr{R},\mu) \cup \mathsf{DP}_{\mathscr{X}}(\mathscr{R},\mu)$ where $\mathsf{DP}_{\mathscr{F}}(\mathscr{R},\mu) = \{\ell^{\sharp} \to s^{\sharp} \mid \ell \to r \in R, r \trianglerighteq_{\mu} s, root(s) \in \mathscr{D}\}$, $\mathsf{DP}_{\mathscr{X}}(\mathscr{R},\mu) = \{\ell^{\sharp} \to x \mid \ell \to r \in R, x \in \mathscr{V}ar^{\mu}(r) - \mathscr{V}ar^{\mu}(l)\}$ and $\mu^{\sharp}(f) = \mu(f)$ if $f \in \mathscr{F}$, and $\mu^{\sharp}(f^{\sharp}) = \mu(f)$ if $f \in \mathscr{D}$.

Example 3 *In Example 2, we obtain the following set of CS-DPs:*

$$
\begin{array}{rcllrcl}
\mathsf{PRIMES} & \to & \mathsf{SIEVE}(\mathsf{from}(\mathsf{s}(\mathsf{s}(0)))) & & \mathsf{IF}(\mathsf{true},x,y) & \to & x \\
\mathsf{PRIMES} & \to & \mathsf{FROM}(\mathsf{s}(\mathsf{s}(0))) & & \mathsf{IF}(\mathsf{false},x,y) & \to & y \\
\mathsf{FILTER}(x,y{:}z) & \to & \mathsf{IF}(\mathsf{divides}(x,y),\mathsf{filter}(x,z),y{:}\mathsf{filter}(x,z)) & & \mathsf{DIVIDES}(x,y) & \to & \mathsf{ZERO}(\mathsf{mod}(y,x)) \\
\mathsf{FILTER}(x,y{:}z) & \to & \mathsf{DIVIDES}(x,y) & & \mathsf{DIVIDES}(x,y) & \to & \mathsf{MOD}(y,x) \\
\mathsf{MOD}(\mathsf{s}(x),\mathsf{s}(y)) & \to & y \leq^{\sharp} x & & \mathsf{s}(x) -^{\sharp} \mathsf{s}(y) & \to & x -^{\sharp} y \\
\mathsf{MOD}(\mathsf{s}(x),\mathsf{s}(y)) & \to & \mathsf{IF}(y \leq x, \mathsf{mod}(x-y,\mathsf{s}(y)),\mathsf{s}(x)) & & \mathsf{s}(x) \leq^{\sharp} \mathsf{s}(y) & \to & x \leq^{\sharp} y
\end{array}
$$

together with $\mu(\mathsf{IF}) = \mu(:) = \{1\}$ and $\mu(f) = \{1,\ldots,\mathsf{ar}(f)\}$ for all other symbols f.

To prove termination, we have to show that there is no infinite *chain* of CS-DPs. A sequence $u_1 \to v_1$, $u_2 \to v_2$, ... of CS-DPs is a chain if there is a substitution such that for all $i \geq 1$, (1) if $u_i \to v_i$ is not collapsing, then $\sigma(v_i) \hookrightarrow^*_{\mathscr{R},\mu} \sigma(u_{i+1})$ or (2) if $u_i \to v_i$ is collapsing, then there is a term w_i such that $(2a)$ $\sigma(v_i) \trianglerighteq_\mu w_i$ and $(2b)$ $w_i^\sharp \hookrightarrow^*_{\mathscr{R},\mu} \sigma(u_{i+1})$. From now on, we assume that all CS-TRSs are finite.

3 Rewriting Modules

In programming, the idea of module comes in a natural way. A programmer groups in a module definitions of functions having something in common (not necessarily *among* them; often as a set of services provided to *external* users –i.e., other modules–). Then new modules which use these functions are written. In term rewriting, modules arise in a natural way, when rules defining a given function symbol f (by means of rules $f(\ell_1,\ldots,\ell_k) \to r$) are collected together, and they are used by other rules from other modules. This modular and hierarchical approach is exploited in [17] to prove termination in a modular and incremental way. Although termination is not modular (in general), the author succeeded thanks to imposing a harder termination condition for modules: the $\mathscr{C}_\varepsilon$-termination. The notion of module is introduced by Urbain using the following notation.

Definition 1 **[17]** *Let $\mathscr{R}_1 = (\mathscr{F}_1, R_1)$ be a TRS. A module extending $\mathscr{R}_1$ is a pair $[\mathscr{F}_2 \mid \mathscr{R}_2]$ such that:*

1. *$\mathscr{F}_1 \cap \mathscr{F}_2 = \varnothing$;*
2. *$\mathscr{R}_2$ is a TRS over $\mathscr{F}_1 \cup \mathscr{F}_2$;*
3. *For all $\ell \to r \in R_2$, $\mathsf{root}(\ell) \in \mathscr{F}_2$.*

Then, $\mathscr{R}_1 \cup \mathscr{R}_2$ over $\mathscr{F}_1 \cup \mathscr{F}_2$ is a hierarchical extension *of $\mathscr{R}_1$ with module $[\mathscr{F}_2 \mid \mathscr{R}_2]$, written:*

$$\mathscr{R}_1 \longleftarrow [\mathscr{F}_2 \mid \mathscr{R}_2]$$

Note that $\mathscr{D}_2 \subseteq \mathscr{F}_2$. Roughly speaking, the notation $[\mathscr{F} \mid \mathscr{R}]$ behaves as an interface of $\mathscr{R}$ where $\mathscr{F}$ represents the symbols that can be imported by other modules. Context-sensitive rewriting extends the signature of TRSs with a replacement map. Then, if we want to extend the previous modular approach to CSR, we impose an agreement among the replacement maps of the shared symbols between modules. For that reason we require the replacement maps for the modules to be compatible in the following sense.

Definition 2 (Compatibility [11]) *A replacement map μ_1 on $\mathscr{F}_1$ is compatible with a replacement map μ_2 on $\mathscr{F}_2$ if they have the same replacement restrictions for shared function symbols, i.e., if $\mu_1(f) = \mu_2(f)$ for every $f \in \mathscr{F}_1 \cap \mathscr{F}_2$.*

Now, we are going to extend Definition 1 for taking into account the replacement restrictions.

Definition 3 *Let $\mathscr{R}_1 = (\mathscr{F}_1, R_1)$ be a TRS and μ_1 a replacement map on $\mathscr{F}_1$. A module extending $(\mathscr{R}_1, \mu_1)$ is a pair $[\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$ such that:*

1. *$\mathscr{F}_1 \cap \mathscr{F}_2 = \varnothing$;*
2. *$\mathscr{R}_2$ is a TRS over $\mathscr{F}_1 \cup \mathscr{F}_2$ and μ_2 is a replacement map on $\mathscr{F}_1 \cup \mathscr{F}_2$;*
3. *μ_1 are μ_2 are compatible;*
4. *for all $\ell \to r \in R_2$, $\mathsf{root}(\ell) \in \mathscr{F}_2$.*

System $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ over $\mathscr{F}_1 \cup \mathscr{F}_2$ is a hierarchical extension of $(\mathscr{R}_1, \mu_1)$ with module $[\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$ and we write it like:

$$(\mathscr{R}_1, \mu_1) \longleftarrow [\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$$

Note that symbols from $\mathscr{F}_1$ can appear in rules from $\mathscr{R}_2$, but not as root symbols on the left-hand side of the rules. With this notation, we can also describe the union of composable systems. For the sake of readability, we denote $[\mathscr{F}_0 \mid (\mathscr{R}_0, \mu_0)] \longleftarrow [\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)]$ the hierarchical extension with $[\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)]$ of the whole hierarchy extended with $[\mathscr{F}_0 \mid (\mathscr{R}_0, \mu_0)]$.

Definition 4 *We say that a module $[\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$ extends a hierarchy headed by $[\mathscr{F}_0 \mid (\mathscr{R}_0, \mu_0)]$* independently *of a module $[\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)]$ if:*

1. *$\mathscr{F}_1 \cap \mathscr{F}_2 = \varnothing$;*
2. *$[\mathscr{F}_0 \mid (\mathscr{R}_0, \mu_0)] \longleftarrow [\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)]$ and*
3. *$[\mathscr{F}_0 \mid (\mathscr{R}_0, \mu_0)] \longleftarrow [\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$.*

3.1 Modular Decomposition

As for TRSs, we show how to decompose a CS-TRS into a 'canonical' modular hierarchy, a hierarchy of minimal modules which cannot be split up further. In order to do that, we follow the graph of purely syntactical dependency relation between symbols given in [17].

Definition 5 (Dependency) *Given a TRS $\mathscr{R} = (\mathscr{F}, R)$, we say that $f \in \mathscr{F}$ directly depends on $g \in \mathscr{F}$, written $f \rhd g$, if and there is a rule $\ell \to r \in R$ with*

- *$f = \mathsf{root}(\ell)$ and*
- *g occurs in ℓ or r.*

Besides the original dependency relation in [17], our dependency relation also considers the symbols in the left-hand side of the rule. Using this relation, the decomposition is done in two steps:

Definition 6 (Modular Decomposition of a TRS) *For a TRS $\mathscr{R} = (\mathscr{F}, R)$:*

1. *we build a graph $\mathscr{G}$ the nodes of which are symbols of $\mathscr{F}$ and such that there is an arc from a node x to a node y if and only if $x \rhd y$*
2. *we pack together symbols of strongly connected components of $\mathscr{G}$, i.e., symbols f and g such that:*

$$f \rhd^* g \text{ and } g \rhd^* f$$

In the obtained hierarchy there is no cycle because symbols of mutually recursive functions appear in the same module. Thus, they belong to the same modules. In the dependency pair framework, a similar graph is constructed for decomposing a DP problem into smaller DP problems, but the dependency relation is more involved because the idea is to capture possible infinite chains between pairs. Dealing with CSR, the replacement restrictions do not change the natural decomposition of the modules.

Example 4 *Figure 1 shows the modular decomposition of Example 2.*

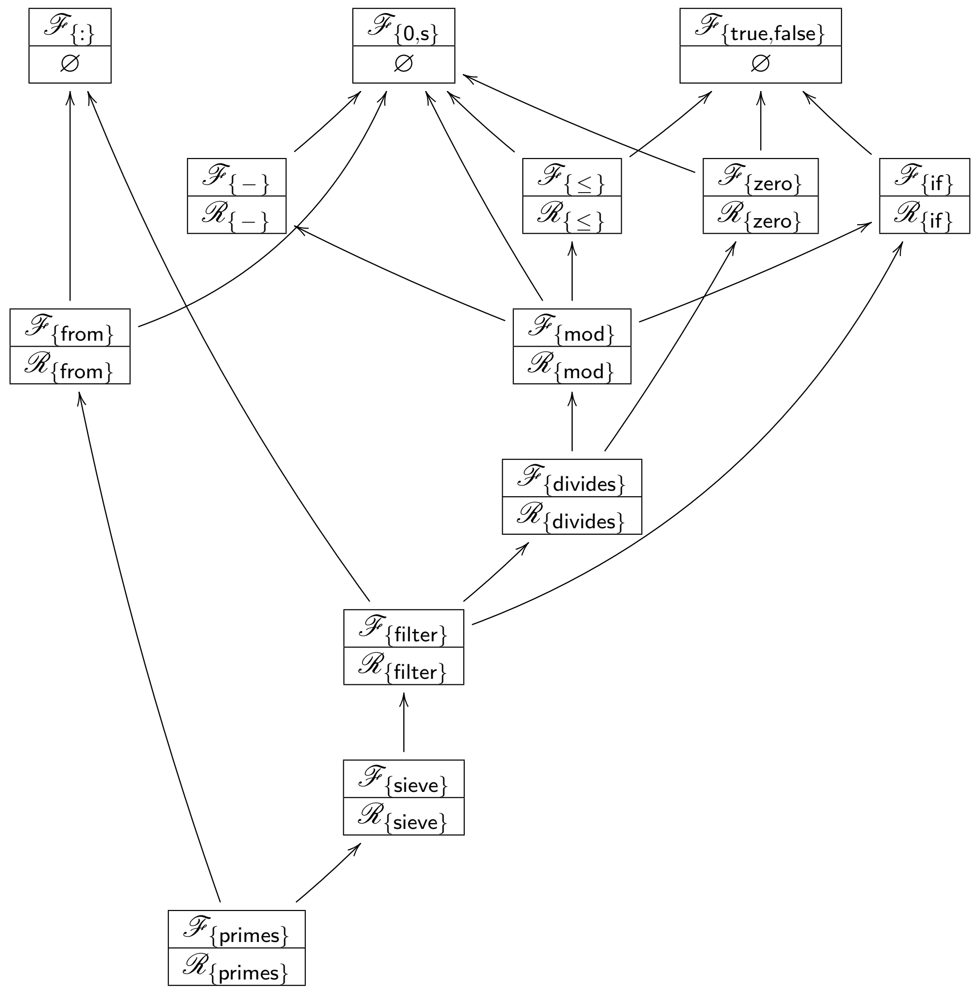

Figure 1: Modular decomposition of Example 2

4 Incremental and Modular Termination

Modular decomposition is quite natural, but from a CS-DP and CS-DP chain point of view, dependencies between modules differ. In this section, we define the notions of CS-DP of a module and relative CS-DP chain.

4.1 CS-DPs of Modules

In CSR, we have to consider two kinds of dependency pairs, the DPs that represent direct calls and the DPs that represent activations of function calls. When dealing with modules, the notion of collapsing DP is still important.

Example 5 [18] (Example 5 modified) *Let us consider the following example:*

$$\mathscr{R}_1 = \begin{array}{rcl} \{\mathsf{if}(\mathsf{true},x,y) & \to & x \\ \mathsf{if}(\mathsf{false},x,y) & \to & y\} \end{array} \qquad \mathscr{R}_2 = \{\mathsf{f}(x) \to \mathsf{if}(x,\mathsf{c},\mathsf{f}(\mathsf{false}))\}$$

where $\mu_2(\mathsf{f}) = \{1\}$, $\mu_1(\mathsf{if}) = \mu_2(\mathsf{if}) = \{1,2\}$, *and* $\mu_2(\mathsf{c}) = \mu_1(\mathsf{true}) = \mu_1(\mathsf{false}) = \mu_2(\mathsf{false}) = \varnothing$*. We can see* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *over* $\mathcal{F}_1 \cup \mathcal{F}_2$ *as a hierarchichal extension of* $(\mathcal{R}_1, \mu_1)$ *with module* $[\mathcal{F}_2 \mid (\mathcal{R}_2, \mu_2)]$ *(i.e.,* $(\mathcal{R}_1, \mu_1) \leftarrow [\mathcal{F}_2 \mid (\mathcal{R}_2, \mu_2)]$*). The set of CS-DPs of* $(\mathcal{R}_1, \mu_1)$ *is* $\mathsf{DP}_1 = \{\mathsf{IF}(\mathsf{false}, x, y) \to y\}$ *and the set of CS-DPs of* $(\mathcal{R}_2, \mu_2)$ *is* $\mathsf{DP}_2 = \varnothing$ *(*if *is not a defined function in* $\mathcal{R}_2$*). Both CS-TRSs are* $\mathcal{C}_\varepsilon$*-terminating independently, but if we consider the union of these two CS-TRSs* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$*, the set of CS-DPs of* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *is* $DP_3 = \{\mathsf{IF}(\mathsf{false}, x, y) \to y, \mathsf{F}(x) \to \mathsf{IF}(x, \mathsf{c}, \mathsf{f}(\mathsf{false}))\}$ *(a new CS-DP is considered) and an infinite CS-DP chain exists:*

$$\mathsf{F}(\mathsf{false}) \to_{\mathsf{DP}_3} \mathsf{IF}(\mathsf{false}, \mathsf{c}, \mathsf{f}(\mathsf{false})) \to_{\mathsf{DP}_3} \mathsf{F}(\mathsf{false}) \to_{\mathsf{DP}_3} \cdots$$

where the new CS-DP appeared by the union is relevant to capture the infinite computation.

The original notion of DP of module only considers DPs appeared in the module, but as we have seen in the example this is not the case when dealing with CS-DPs. To obtiain a similar notion of DP of module, we have to work with *conservative* CS-TRSs, i.e., CS-TRSs without function call activations (collapsing CS-DPs).

Definition 7 *Let* $\mathcal{M} = [\mathcal{F}_{\mathcal{M}} \mid (\mathcal{R}, \mu)]$ *be a module where* $\mathcal{R} = (\mathcal{F}, R) = (\mathcal{C} \uplus \mathcal{D}, R)$ *is a TRS,* $\mathcal{F}_{\mathcal{M}} \subseteq \mathcal{F}$*,* μ *a replacement map on* $\mathcal{F}$ *and* $(\mathcal{R}, \mu)$ *is conservative. We define* $\mathsf{MDP}(\mathcal{M}) = \mathsf{MDP}_{\mathcal{F}}(\mathcal{M})$ *to be the set of* conservative context-sensitive dependency pairs[1] *of module* $\mathcal{M}$ *where:*

$$\mathsf{MDP}_{\mathcal{F}}(\mathcal{M}) \quad = \quad \{\ell^\sharp \to s^\sharp \mid \quad \ell \to r \in R, r \trianglerighteq_\mu s, \mathsf{root}(s) \in \mathcal{D} \cap \mathcal{F}_{\mathcal{M}}\}$$

We extend μ *into* $\mu^\sharp$ *by* $\mu^\sharp(f) = \mu(f)$ *if* $f \in \mathcal{F}$ *and* $\mu^\sharp(f^\sharp) = \mu(f)$ *if* $f \in \mathcal{D} \cap \mathcal{F}_{\mathcal{M}}$*.*

4.2 Relative CS-DP Chains

From the definition of CS-DPs of a module, we define CS-DP chains *relative* to some CS-TRS.

Definition 8 *Let* $\mathcal{M} = [\mathcal{F}_1 \mid (\mathcal{R}_1, \mu_1)]$ *be a module where* $(\mathcal{R}_1, \mu_1)$ *is conservative and* $(\mathcal{R}_2, \mu_2)$ *an arbitrary CS-TRS where* μ_1 *and* μ_2 *are compatible. A* CS-DP chain *of* $\mathsf{MDP}(\mathcal{M})$ *relative to* $(\mathcal{R}_2, \mu_2)$ *is a sequence of pairs* $u_i \to v_i \in \mathsf{MDP}(\mathcal{M})$ *together with a substitution* σ *such that for all* $i \geq 1$*, We assume that different occurrences of pairs do not share any variable. A CS-DP chain is* minimal *iff* $\sigma(v_i)$ *is* $(\mathcal{R}_2, \mu_2)$*-terminating.*

Most recent notion of chain, the $(\mathcal{P}, \mathcal{R}, \mathcal{S}, \mu)$-chain, contains three TRSs: $\mathcal{P}$ models the behaviour of CS-DPs; $\mathcal{R}$ models the behaviour of the rules; and $\mathcal{S}$ models the subterm and marking in CS-DP chains. But the given definition is enough for the purpose of the paper because we are dealing with conservative CS-TRSs.

Proposition 1 *A conservative CS-TRS* $(\mathcal{R}, \mu)$ *where* $\mathcal{R} = (\mathcal{F}, R)$ *is* $\mathcal{C}_\varepsilon$*-terminating if and only if there is no infinite minimal chain of* $\mathsf{MDP}([\mathcal{F} \mid (\mathcal{R}, \mu)])$ *relative to* $(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon})$*.*

Proof 1 *Since* c *does not belong to* $\mathcal{F}$*, and since* $\mu_{\mathcal{C}_\varepsilon}(\mathsf{c}) = \{1,2\}$ *then* $\mathsf{DP}(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon}) = \mathsf{DP}(\mathcal{R}, \mu)$*, where* DP *is a function that obtains the CS-DPs of a CS-TRS. So,* $\mathsf{MDP}([\mathcal{F} \mid (\mathcal{R}, \mu)]) = \mathsf{DP}(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon})$*. Therefore, proving the termination of* $\mathsf{MDP}([\mathcal{F} \mid (\mathcal{R}, \mu)])$ *relative to* $(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon})$ *is the same as proving termination of* $\mathsf{DP}(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon})$ *relative to* $(\mathcal{R} \cup \mathcal{C}_\varepsilon, \mu \cup \mu_{\mathcal{C}_\varepsilon})$*, that is proving the* $\mathcal{C}_\varepsilon$*-termination of* $(\mathcal{R}, \mu)$*.* ■

[1] In [3], the notion of CS-DP includes some extra conditions to discard CS-DPs that, by construction, are not involved in infinite chains (narrowability and subterm conditions). To ease readability, we do not include these extra conditions, but the results obtained in the paper are still applicable adding these conditions.

4.3 Termination with modules

In contrast to the unrestricted approach (pure term rewriting), in CSR a chain is possible in a hierarchical extension even being impossible for the specific component (Example 5). The following modularity result can be extracted when the pairs are not collapsing.

Lemma 1 *Let $(\mathcal{R}_1,\mu_1)$ be a CS-TRS where $\mathcal{R}_1 = (\mathcal{F}_1, R_1)$ and $[\mathcal{F}_2 \mid (\mathcal{R}_2,\mu_2)]$ be a module such that $[\mathcal{F}_1 \mid (\mathcal{R}_1,\mu_1)] \longleftarrow [\mathcal{F}_2 \mid (\mathcal{R}_2,\mu_2)]$. Then, for any two pairs $u_1 \to v_1 \in \mathsf{MDP}_{\mathcal{F}_1}([\mathcal{F}_1 \mid (\mathcal{R}_1,\mu_1)])$ and $u_2 \to v_2 \in \mathsf{MDP}([\mathcal{F}_2 \mid (\mathcal{R}_2,\mu_2)])$, there is no substitution σ such that:*

$$\sigma(v_1) \overset{>\Lambda}{\underset{\mathcal{R}_1\cup\mathcal{R}_2}{\hookrightarrow}}{}^{*} \sigma(u_2)$$

Proof 2 *Since* $\mathsf{root}(\sigma(u_2)) = \mathsf{root}(u_2) \in \mathcal{D}_2^\sharp \subseteq \mathcal{F}_2^\sharp$, $\mathsf{root}(\sigma(v_1)) = \mathsf{root}(v_1) \in \mathcal{D}_1^\sharp \subseteq \mathcal{F}_1^\sharp$ *and* $\mathcal{D}_1^\sharp \cap \mathcal{D}_2^\sharp = \varnothing$, *we obtain that* $\mathsf{root}(v_1) \neq \mathsf{root}(u_2)$. *Hence,* $\sigma(v_1)$ *cannot be rewritten* below the root *to* $\sigma(u_2)$. ■

In [17] the termination of modules is based on the following two theorems:

Theorem 1 [17, Theorem 1] *Let $[\mathcal{F}_1 \mid \mathcal{R}_1] \longleftarrow [\mathcal{F}_2 \mid \mathcal{R}_2]$ be a hierarchical extension of $\mathcal{R}_1 = (\mathcal{F}_1, R_1)$; if*

- *$\mathcal{R}_1$ is $\mathcal{C}_\varepsilon$-terminating, and*
- *there is no infinite dependency chain of $[\mathcal{F}_2 \mid \mathcal{R}_2]$ relative to $\mathcal{R}_1 \cup \mathcal{R}_2$,*

then $\mathcal{R}_1 \cup \mathcal{R}_2$ is terminating.

Theorem 2 [17, Theorem 2] *Let $[\mathcal{F}_1 \mid \mathcal{R}_1] \longleftarrow [\mathcal{F}_2 \mid \mathcal{R}_2]$ be a hierarchical extension of $\mathcal{R}_1 = (\mathcal{F}_1, R_1)$, and $[\mathcal{F}_3 \mid \mathcal{R}_3]$ be a module extending $\mathcal{R}_1$ independently of $\mathcal{R}_2$. If*

- *$\mathcal{R}_1 \cup \mathcal{R}_2$ is $\mathcal{C}_\varepsilon$-terminating, and*
- *there is no infinite dependency chain of $[\mathcal{F}_3 \mid \mathcal{R}_3]$ relative to $\mathcal{R}_1 \cup \mathcal{R}_3 \cup \mathcal{C}_\varepsilon$,*

then $\mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$ is $\mathcal{C}_\varepsilon$-terminating.

But as we have seen before in the previous examples, the adaptation of these theorems to CSR needs to consider more conditions to safely extend hierarchical extensions to CSR. The key idea behind the results on hierarchical extensions is the possibility of building an infinite CS-DP chain of $\mathsf{MDP}([\mathcal{F}_1 \mid (\mathcal{R}_1,\mu_1)])$ relative to $(\mathcal{R}_1 \cup \mathcal{C}_\varepsilon, \mu_1 \cup \mu_{\mathcal{C}_\varepsilon})$ from an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathcal{F}_1 \mid (\mathcal{R}_1,\mu_1)])$ relative to $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$. In [13], a couple of interpretations are presented to simulate rewriting steps on $(\mathcal{R}_2,\mu_2)$ by rewriting steps on $(\mathcal{C}_\varepsilon, \mu_{\mathcal{C}_\varepsilon})$ when $(\mathcal{R}_2,\mu_2)$ is terminating, but none of them are suitable for just conservative CS-TRSs. In order to obtain a result similar to the previous theorems we have to impose a stronger statement (strongly conservative) to get use of the *basic μ-interpretation* in [13]. Basic μ-interpretation simulates rewriting steps on a terminating CS-TRS $(\mathcal{R}_2,\mu_2)$ as rewriting steps using $\mathcal{C}_\varepsilon$-rules.

Definition 9 [13] (Basic μ-interpretation) *Let $(\mathcal{R},\mu)$ be a CS-TRS over $\mathcal{F}$ and $\Delta \subseteq \mathcal{F}$. Let $>$ be an arbitrary total ordering over $\mathcal{T}(\mathcal{F}^\sharp \cup \{\bot, \mathsf{c}\}, \mathcal{X})$ where $\bot$ is a new constant symbol and c is a new binary symbol. The interpretation $\Phi_{\Delta,\mu}$ is a mapping from μ-terminating terms in $\mathcal{T}(\mathcal{F}^\sharp, \mathcal{X})$ to terms in $\mathcal{T}(\mathcal{F}^\sharp \cup \{\bot, \mathsf{c}\}, \mathcal{X})$ defined as follows:*

$$\Phi_{\Delta,\mu}(t) = \begin{cases} t & \text{if} \quad t \in \mathcal{X} \\ f(\Phi_{\Delta,\mu,f,1}(t_1), \ldots, \Phi_{\Delta,\mu,f,n}(t_n)) & \text{if} \quad t = f(t_1 \ldots t_n) \text{ and } f \in \Delta \\ \mathsf{c}(f(\Phi_{\Delta,\mu,f,1}(t_1), \ldots, \Phi_{\Delta,\mu,f,n}(t_n)), t') & \text{if} \quad t = f(t_1 \ldots t_n) \text{ and } f \notin \Delta \end{cases}$$

where $\Phi_{\Delta,\mu,f,i}(t) = \begin{cases} \Phi_{\Delta,\mu}(t) & \text{if } i \in \mu(f) \\ t & \text{if } i \notin \mu(f) \end{cases}$

$t' = order\left(\{\Phi_{\Delta,\mu}(u) \mid t \hookrightarrow_{\mathscr{R},\mu} u\}\right)$

$order(T) = \begin{cases} \perp, & \text{if } T = \varnothing \\ \mathsf{c}(t, order(T - \{t\})) & \text{if } t \text{ is minimal in } T \text{ w.r.t. } > \end{cases}$

Termination is crucial to obtain a correct approach.

Lemma 2 [13] *For each μ-terminating term s, the term $\Phi_{\Delta,\mu}(s)$ is finite.*

Imposing that all the rules are strongly conservative we ensure that a variable appearing at a frozen position in the left-hand side of the rule never appears at an active position in the right-hand side of the rule (conservative property is not enough to ensure this statement).

Theorem 3 (Strongly Conservative Hierarchical Extension) *Let $[\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)] \longleftarrow [\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$ be a hierarchical extension of $(\mathscr{R}_1, \mu_1)$ where $\mathscr{R}_1 = (\mathscr{F}_1, R_1)$; if*

- *$(\mathscr{R}_1, \mu_1)$ is strongly conservative and $\mathscr{C}_\varepsilon$-terminating, and*
- *$(\mathscr{R}_2, \mu_2)$ is strongly conservative, and*
- *there is no infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$,*

then $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ is terminating.

Proof 3 (Sketch) *By contradiction. Let us suppose that there is an infinite minimal CS-DP chain of $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$, then:*

- *there is an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$.*
- *$(\mathscr{R}_1, \mu_1)$ is not $\mathscr{C}_\varepsilon$-terminating, contradicting the hypothesis.*

We suppose that $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ is non-terminating, so there is an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$. CS-DPs consist of:

1. *CS-DPs from $\mathsf{MDP}([\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)])$;*
2. *CS-DPs from $\mathsf{MDP}([\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)])$;*
3. *CS-DPs $\ell^\sharp \to s^\sharp$ such that $\ell \to r \in \mathscr{R}_2$, $r \trianglerighteq_\mu s$ and $\mathsf{root}(s) \in \mathscr{F}_1$.*

Using Lemma 1, we get an infinite minimal CS-DP chain where CS-DPs are:

i. *from (2) only,*

ii. *from (1) only,*

iii. *from (2) in a finite number, then one pair from (3) and infinitely many pairs from (1).*

- *Case (i): an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ exists, contradicting the hypothesis.*
- *Cases (ii)-(iii): an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$. In a similar way to [13], using Definition 9, we can construct an infinite CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)])$ relative to $(\mathscr{R}_1 \cup \mathscr{C}_\varepsilon, \mu_1 \cup \mu_{\mathscr{C}_\varepsilon})$ from an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ where only $\mathscr{F}_2$ symbols are not in Δ.*

$\blacksquare$

Independent hierarchical extensions allow us to represent the union of composable CS-TRSs.

Theorem 4 (Independent Strongly Conservative Hierarchical Extension) *Let $[\mathscr{F}_1 \mid (\mathscr{R}_1, \mu_1)] \longleftarrow [\mathscr{F}_2 \mid (\mathscr{R}_2, \mu_2)]$ be a hierarchical extension of $(\mathscr{R}_1, \mu_1)$ where $\mathscr{R}_1 = (\mathscr{F}_1, R_1)$, and $[\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)]$ be a module extending $(\mathscr{R}_1, \mu_1)$ independently of $(\mathscr{R}_2, \mu_2)$. If*

- *$(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ is strongly conservative and $\mathscr{C}_\varepsilon$-terminating,*
- *$(\mathscr{R}_1 \cup \mathscr{R}_3, \mu_1 \cup \mu_3)$ is strongly conservative, and*
- *there is no infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_3 \cup \mathscr{C}_\varepsilon, \mu_1 \cup \mu_3 \cup \mu_{\mathscr{C}_\varepsilon})$,*

then $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$ is $\mathscr{C}_\varepsilon$-terminating.

Proof 4 (Sketch) *By contradiction. Let us suppose that $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$ is non-terminating, so there is an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \cup \mathscr{F}_3 \mid (\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$. Using Lemma 1, we know that CS-DP chains are:*

1. *CS-DP chains of $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$;*
2. *CS-DPs chains of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$;*
3. *CS-DPs chains relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$ consisting of a finite number of CS-DPs in $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$, a CS-DP $\ell^\sharp \to s^\sharp$ such that $\ell \to r \in \mathscr{R}_3$, $r \trianglerighteq_\mu s$ and $\mathsf{root}(s) \in \mathscr{F}_1 \cup \mathscr{F}_2$ and infinitely many CS-DPs from $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$.*

Using Lemma 1, we get an infinite minimal CS-DP chain where CS-DPs are:

- *Case (1): an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$. In a similar way to [13], using Definition 9, we can construct an infinite CS-DP chain of $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_3 \cup \mathscr{C}_\varepsilon, \mu_1 \cup \mu_3 \cup \mu_{\mathscr{C}_\varepsilon})$ from an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_3 \mid (\mathscr{R}_3, \mu_3)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$ where only $\mathscr{F}_2$ symbols are not in Δ.*
- *Cases (2)-(3): an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$. In a similar way to [13], using Definition 9, we can construct an infinite CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{C}_\varepsilon, \mu_1 \cup \mu_2 \cup \mu_{\mathscr{C}_\varepsilon})$ from an infinite minimal CS-DP chain of $\mathsf{MDP}([\mathscr{F}_1 \cup \mathscr{F}_2 \mid (\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)])$ relative to $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \mathscr{R}_3, \mu_1 \cup \mu_2 \cup \mu_3)$ where only $\mathscr{F}_3$ symbols are not in Δ.*

■

Without the strongly conservative restriction (even considering only conservative rules) we cannot ensure the previous results.

Example 6 [4] *Consider the following conservative CS-TRS:*

$$\mathscr{R}_1 = \{\mathsf{f}(\mathsf{c}(x), x) \to \mathsf{f}(x, x)\} \qquad \mathscr{R}_2 = \{\mathsf{b} \to \mathsf{c}(\mathsf{b})\}$$

where $\mu_1(\mathsf{f}) = \{1,2\}$, $\mu_1(\mathsf{c}) = \mu_2(\mathsf{c}) = \mu_2(\mathsf{b}) = \varnothing$. We can see $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ as a hierarchichal extension of $(\mathscr{R}_0, \mu_0) = ((\{\mathsf{c}\}, \varnothing), \mu_0)$ with module $[\mathscr{F}_1 \mid \mathscr{R}_1]$ and $[\mathscr{F}_2 \mid \mathscr{R}_2]$ is a module extending $(\mathscr{R}_0, \mu_0)$ independently of $(\mathscr{R}_1, \mu_1)$. The set of CS-DPs of $(\mathscr{R}_1, \mu_1)$ is $\mathsf{DP}_1 = \{\mathsf{F}(\mathsf{c}(x), x) \to \mathsf{F}(x, x)\}$ and the set of CS-DPs of $(\mathscr{R}_2, \mu_2)$ is $\mathsf{DP}_2 = \varnothing$. Both conservative CS-TRSs are $\mathscr{C}_\varepsilon$-terminating independently, but the union of these two CS-TRSs $(\mathscr{R}_1 \cup \mathscr{R}_2, \mu_1 \cup \mu_2)$ generates an infinite CS-DP chain:

$$\mathsf{F}(\mathsf{c}(\mathsf{b}), \mathsf{b}) \to_{\mathsf{DP}_1} \mathsf{F}(\mathsf{b}, \mathsf{b}) \hookrightarrow_{\mathscr{R}_2, \mu_2} \mathsf{F}(\mathsf{c}(\mathsf{b}), \mathsf{b}) \to_{\mathsf{DP}_1} \cdots$$

The following example shows an application of our result.

Example 7 [5] *Consider the following example:*

$$\mathcal{R}_1 = \begin{array}{rcl}\{\mathsf{length}(\mathsf{nil}) & \to & 0\\ \mathsf{length}(x{:}y) & \to & \mathsf{s}(\mathsf{length1}(y))\\ \mathsf{length1}(x) & \to & \mathsf{length}(x)\}\end{array} \qquad \mathcal{R}_2 = \{\mathsf{from}(x) \to x{:}\mathsf{from}(\mathsf{s}(x))\}$$

where $\mu_2(\mathsf{from}) = \mu_1(:) = \mu_2(:) = \mu_1(\mathsf{s}) = \mu_2(\mathsf{s}) = \{1\}$ *and* $\mu_1(\mathsf{length}) = \mu_1(\mathsf{length1}) = \mu_1(\mathsf{nil}) = \mu_1(0) = \varnothing$*. We can see* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *as a hierarchichal extension of* $(\mathcal{R}_0, \mu_0) = ((\{:,\mathsf{s}\},\varnothing), \mu_0)$ *with module* $[\mathcal{F}_1 \mid \mathcal{R}_1]$ *and* $[\mathcal{F}_2 \mid \mathcal{R}_2]$ *is a module extending* $(\mathcal{R}_0, \mu_0)$ *independently of* $(\mathcal{R}_1, \mu_1)$*. The set of CS-DPs of* $(\mathcal{R}_1, \mu_1)$ *is* $\mathsf{DP}_1 = \{\mathsf{LENGTH}(x{:}y) \to \mathsf{LENGTH1}(y), \mathsf{LENGTH1}(x) \to \mathsf{LENGTH}(x)\}$ *and the set of CS-DPs of* $(\mathcal{R}_2, \mu_2)$ *is* $\mathsf{DP}_2 = \varnothing$*. Both CS-TRSs are* $\mathcal{C}_\varepsilon$*-terminating independently and we can use the results of the paper to conclude that the union* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *is terminating.*

However, we still cannot deal with the leading example of the paper and we must overcome problems as the one showed in the following example.

Example 8 *Let us consider the following example:*

$$\mathcal{R}_1 = \{\mathsf{take}(x{:}y) \to \mathsf{take}(y)\} \qquad \mathcal{R}_2 = \{\mathsf{from}(x) \to x{:}\mathsf{from}(\mathsf{s}(x))\}$$

where $\mu_1(\mathsf{take}) = \mu_1(:) = \mu_2(:) = \mu_2(\mathsf{from}) = \mu_2(\mathsf{s}) = \{1\}$*. We can see* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *as a hierarchichal extension of* $(\mathcal{R}_0, \mu_0) = ((\{:\},\varnothing), \mu_0)$ *with module* $[\mathcal{F}_1 \mid \mathcal{R}_1]$ *and* $[\mathcal{F}_2 \mid \mathcal{R}_2]$ *is a module extending* $(\mathcal{R}_0, \mu_0)$ *independently of* $(\mathcal{R}_1, \mu_1)$*. The set of CS-DPs of* $(\mathcal{R}_1, \mu_1)$ *is* $\mathsf{DP}_1 = \{\mathsf{TAKE}(x{:}y) \to \mathsf{TAKE}(y)\}$ *and the set of CS-DPs of* $(\mathcal{R}_2, \mu_2)$ *is* $\mathsf{DP}_2 = \varnothing$*. Both CS-TRSs are* $\mathcal{C}_\varepsilon$*-terminating independently, but the union of these two CS-TRSs* $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *generates an infinite CS-DP chain:*

$$\mathsf{TAKE}(x{:}\mathsf{from}(\mathsf{s}(x))) \to_{\mathsf{DP}_1} \mathsf{TAKE}(\mathsf{from}(\mathsf{s}(x))) \hookrightarrow_{\mathcal{R}_2,\mu_2} \mathsf{TAKE}(\mathsf{s}(x){:}\mathsf{from}(\mathsf{s}(\mathsf{s}(x)))) \to_{\mathsf{DP}_1} \cdots$$

where rules from $(\mathcal{R}_2, \mu_2)$ *are an important actor in the non-termination of the union of CS-TRSs.*

In a hierarchical extension, when we consider a terminating module which is nonterminating without the replacement map, the nonterminating behaviour can be due to new rules in the extended modules. These new rules take an important role in the adaptation of hierarchical extensions to arbitrary CS-TRSs.

5 Related Work

In [11], two results about modularity of CS-TRSs were given. One for the union of CS-TRSs with disjoint signatures and one for the union of CS-TRSs with shared constructors. In our work disjoint signature unions are not considered. For constructor sharing unions, they obtained the following result:

Theorem 5 [11] *Let* $(\mathcal{R}_1, \mu_1)$*,* $(\mathcal{R}_2, \mu_2)$ *be two constructor sharing, compatible, terminating CS-TRSs:*

1. $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *terminates if* $(\mathcal{R}_1, \mu_1)$ *and* $(\mathcal{R}_2, \mu_2)$ *are layer-preserving.*
2. $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *terminates if* $(\mathcal{R}_1, \mu_1)$ *and* $(\mathcal{R}_2, \mu_2)$ *are non-duplicating.*
3. $(\mathcal{R}_1 \cup \mathcal{R}_2, \mu_1 \cup \mu_2)$ *terminates if* $(\mathcal{R}_1, \mu_1)$ *or* $(\mathcal{R}_2, \mu_2)$ *is both, layer-preserving and non-duplicating.*

Layer preserving means that there is no rule $\ell \to r$ such that r is a variable or rooted by a shared constructor. This condition excludes rules like $\mathcal{R}_1$-rules in Example 8. A rule $\ell \to r$ is non-duplicating if for every $x \in \mathcal{V}ar(\ell)$ the multiset of replacing occurrencies of x in r is contained in the multiset of replacing occurrences of x in ℓ. A rule like $\mathsf{f}(x) \to \mathsf{g}(x,x)$ where $\mu(\mathsf{f}) = \{1\}$ and $\mu(\mathsf{g}) = \{1,2\}$ is strongly conservative but duplicating, hence, our results on strongly conservative modules are complementary to the ones obtained in [11].

6 Conclusions

In this paper we analyze modularity of termination for combinations of context-sensitive rewrite modules from the perspective of CS-DPs. The analysis shows that only in a very restrictive case (strong conservative hierarchical extension), the modularity results for term rewriting extends to CSR. When trying to generalize modularity results to arbitrary CS-TRSs, we find some counterexamples that force us to consider new restrictions in order to obtain a correct result. The main problem comes from modules that are nonterminating when removing the replacement map (those modules contain potential nonterminating μ-rewrite sequences that can appear by means of module hierarchical extensions). These modules with potential nonterminating rules must be considered to obtain a complete incremental and modular termination framework for CSR because these rules cannot be simulated by $\mathscr{C}_\varepsilon$-rules. Future work aims to analyze those problems and obtain a correct hierarchical extension results on arbitrary modules.

References

[1] B. Alarcón, R. Gutiérrez & S. Lucas (2006): *Context-Sensitive Dependency Pairs*. In S. Arun-Kumar & N. Garg, editors: *Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science, FST&TCS'06*, *LNCS* 4337, Springer-Verlag, pp. 297–308.

[2] B. Alarcón, R. Gutiérrez & S. Lucas (2007): *Improving the Context-Sensitive Dependency Graph*. *Electronic Notes in Theoretical Computer Science* 188, pp. 91–103. Proc. of the 6th Spanish Conference on Programming and Languages, PROLE'06.

[3] B. Alarcón, R. Gutiérrez & S. Lucas (2010): *Context-Sensitive Dependency Pairs*. *Information and Computation* 208, pp. 922–968.

[4] B. Alarcón & S. Lucas (2007): *Termination of Innermost Context-Sensitive Rewriting Using Dependency Pairs*. In F. Wolter, editor: *Proc. of the 6th International Symposium on Frontiers of Combining Systems, FroCoS'07*, *LNAI* 4720, Springer-Verlag, pp. 73–87.

[5] M. Alpuente, S. Escobar, B. Gramlich & S. Lucas (2010): *On-Demand Strategy Annotations Revisited: An Improved On-Demand Evaluation Strategy*. *Theoretical Computer Science* 411(2), pp. 504–541.

[6] T. Arts & J. Giesl (2000): *Termination of Term Rewriting Using Dependency Pairs*. *Theoretical Computer Science* 236(1–2), pp. 133–178.

[7] F. Baader & T. Nipkow (1998): *Term Rewriting and All That*. Cambridge University Press.

[8] F. Durán, S. Lucas, C. Marché, J. Meseguer & X. Urbain (2008): *Proving Operational Termination of Membership Equational Programs*. *Higher Order Symbolic Computation* 21(1-2), pp. 59–88.

[9] J. Giesl & A. Middeldorp (1999): *Transforming Context-Sensitive Rewrite Systems*. In P. Narendran & M. Rusinowitch, editors: *Proc. of the 10th International Conference on Rewriting Techniques and Applications, RTA'99*, *LNCS* 1631, Springer-Verlag, pp. 271–285.

[10] B. Gramlich (1994): *Generalized Sufficient Conditions for Modular Termination of Rewriting*. *Applicable Algebra in Engineering, Comunication and Computing* 5, pp. 131–151.

[11] B. Gramlich & S. Lucas (2002): *Modular Termination of Context-Sensitive Rewriting*. In: *Proc. of the 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming, PPDP'02*, ACM Press, pp. 50–61.

[12] R. Gutiérrez (2010): *Automatic Proofs of Termination of Context-Sensitive Rewriting*. Ph.D. thesis, Dept. de Sistemes Informàtics i Computació, Universitat Politècnica de València, Valencia, Spain.

[13] R. Gutiérrez, S. Lucas & X. Urbain (2008): *Usable Rules for Context-Sensitive Rewrite Systems*. In A. Voronkov, editor: *Proc. of the 19th International Conference on Rewriting Techniques and Applications, RTA'08*, *LNCS* 5117, Springer-Verlag, pp. 126–141.

[14] S. Lucas (1998): *Context-Sensitive Computations in Functional and Functional Logic Programs*. *Journal of Functional and Logic Programming* 1998(1), pp. 1–61.

[15] S. Lucas (2002): *Context-Sensitive Rewriting Strategies*. *Information and Computation* 178(1), pp. 293–343.

[16] Y. Toyama (1987): *Counterexamples to Termination for the Direct Sum of Term Rewriting Systems*. *Information Processing Letters* 25, pp. 141–143.

[17] X. Urbain (2004): *Modular & Incremental Automated Termination Proofs*. *Journal of Automated Reasoning* 32(4), pp. 315–355.

[18] H. Zantema (1997): *Termination of Context-Sensitive Rewriting*. In H. Comon, editor: *Proc. of the 7th International Conference on Rewriting Techniques and Applications, RTA'97*, *LNCS* 1232, Springer-Verlag, pp. 172–186.