Unravelings and Ultra-properties

Massimo Marchiori
Department of Pure and Applied Mathematics
University of Padova
Via Belzoni 7, 35131 Padova, Italy
Phone: +39 49 8275972 Fax: +39 49 8758596
max@math.unipd.it

Abstract

Conditional rewriting is universally recognized as being much more complicated than unconditional rewriting. In this paper we study how much of conditional rewriting can be automatically inferred from the simpler theory of unconditional rewriting. We introduce a new tool, called *unraveling*, to automatically translate a conditional term rewriting system (CTRS) into a term rewriting system (TRS). An unraveling enables to infer properties of a CTRS by studying the corresponding *ultra-property* on the corresponding TRS. We show how to rediscover properties like decreasingness, and to give easy proofs of some existing results on CTRSs. Moreover, we show how unravelings provide a valuable tool to study *modularity* of CTRSs, automatically giving a multitude of new results.

Keywords and Phrases: Conditional Term Rewriting, Term Rewriting, Modularity.

1 Introduction

The growing interest towards conditional term rewriting systems (CTRSs for short) clashes with the fact that their analysis is by far much more complicated than for term rewriting systems (TRSs). In this paper, we try to see how much of the theory of CTRSs can be *automatically* recovered from the theory of TRSs. To this extent, we introduce the tool of *unravelings*, that are particular transformations from CTRSs to TRSs. Roughly speaking, an unraveling maps a CTRS into an approximation defined by a TRS. Then, to infer properties of the CTRS one can study the corresponding TRS, therefore employing all the machinery developed for term rewriting systems. This allows to automatically lift known results of TRSs to CTRSs. The idea of transforming CTRSs into TRSs dates back to the seminal work of Bergstra and Klop [1], where they say that using such a transformation can be very useful to get better intuition on the behaviour of a CTRS. Later works include [6] and [11]. However, all these works only cover partial cases, since their aims are different: in [6] Giovannetti and Moiso seek for a transformation that completely preserves the operational behaviour of a TRS (the goal is to extend the syntax of the logic-functional language K-LEAF), and consequently the obtained results are rather limited. In [11] Hintermeier aims to provide a constructive proof of the fact that the computable power of CTRSs equals that of TRSs: he gives a two-steps transformation (based on order-sorted rewriting) from the class of decreasing and ground-confluent CTRSs to TRSs.

In this paper, instead, we are interested in developing a general theory of transformations that allows to study every particular property $\mathcal{P}$ of a CTRS by studying the corresponding *ultra-property* on the unraveled CTRS. We develop three unravelings, whose definition is extremely simple and natural, one for the whole class of join CTRSs, and the others for normal CTRSs. Then we formally show that every major property of a CTRS can be analyzed by at least one of these unravelings.

Via unravelings, we show that the analysis of termination and confluence of CTRSs can be tackled using the results developed for TRSs.

We also show that unravelings are useful to study in a certain sense whether the extensions of properties that are specific to TRSs, like non-duplication and left-linearity, play a role for CTRSs. In particular, it is formally shown that left-linearity does play a relevant role for normal CTRSs, and that this class of CTRSs has a behaviour very similar to TRSs: For instance, the fundamental notion of *decreasingness* ([5, 4]) is provided with a much more meaningful justification, since it just corresponds to 'ultra-termination' (termination of the unraveled CTRS). Moreover, results specifically obtained for normal CTRSs via a certain effort, like the major result in [1] stating confluence of normal orthogonal CTRSs, can be automatically obtained from TRSs.

Then we turn to one of the major applications of unraveling, the study of *modularity* of CTRSs, and show that many new results can be obtained without efforts using the known results on modularity for TRSs. We analyze modular properties of (disjoint, constructor-sharing and composable) CTRSs and, for the first time, we develop results specifically for normal CTRSs. Afterwards, we study the even more complicated combinations of hierarchical CTRSs (a so far practically unexplored field, due to its intrinsic complexity), and again show how the results obtained for TRSs automatically provide powerful results for hierarchical combinations of join and left-linear normal CTRSs.

Finally, we perform a systematic study of the abstract power of the unraveling approach, and show that the results here presented are in a sense the best possible: for every major property it is found the corresponding 'maximal power' of analysis via unravelings, and it is shown this limit is just what obtained in this paper.

2 Preliminaries

We assume knowledge of the basic notions regarding conditional term rewriting systems and term rewriting systems (cf. [3, 15]).

In this paper we will mainly deal with *join* and *normal CTRSs*, that is in the first case rules are of the form $l \to r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$ (with $Var(r, s_1, t_1, \ldots, s_k, t_k) \subseteq Var(l)$, where $Var(s)$ denotes the variables of the term s), and in the second of the form $l \to r \Leftarrow s_1 \to^*_! n_1, \ldots, s_k \to^*_! n_k$ (with $Var(l) \supseteq Var(s_1, \ldots, s_k)$, and $n_1, \ldots, n_k$ ground normal forms).

As far as the major properties of (C)TRSs are concerned, we will employ the standard acronyms $\mathrm{UN}^{\to}$ (uniqueness of normal forms w.r.t. reduction: a term can have at most one normal form), UN (uniqueness of normal form: a term can have at most one normal form modulo convertibility), NF (normal form property: every term convertible to a normal form rewrites to it), $\mathrm{CON}^{\to}$ (consistency w.r.t. reduction: a term cannot rewrite to two different variables) and CON (consistency: two variables cannot be convertible).

If $s_1, \ldots, s_k$ and $t_1, \ldots, t_k$ are sequences of terms, we use $s_1, \ldots, s_k = t_1, \ldots, t_k$ as a shorthand for $s_1 = t_1, \ldots, s_k = t_k$. Sequences in formulae should be seen just as abbreviations: for example, if S is the sequence t_1, t_2, then $f(S)$ denotes the term $f(t_1, t_2)$. Finally, if $S = s_1, \ldots, s_k$ is a sequence then $|S|$ indicates its cardinality, i.e. k.

3 Unravelings

An unraveling is, roughly speaking, a map associating to every CTRS an approximating TRS. More formally:

Definition 3.1 An *unraveling* is a computable map $\mathbf{U}$ from CTRSs to TRSs such that

1. $\downarrow_R \subseteq \downarrow_{\mathbf{U}(R)}$
2. $\mathbf{U}(T \cup R) = T \cup \mathbf{U}(R)$ if T is a TRS □

These two conditions should appear rather natural: The first condition requires that the join relation of the unraveled CTRS is an extension of the original CTRS: roughly, it means that the TRS does not compute 'less' than its original CTRS; more formally, it states that the unraveled CTRS does not have less logical strength (cf. [5, 4]) than the original CTRS. The second condition says that if we are unraveling a CTRS, we can extract from it the part that is already a TRS, and then go on computing the unraveling.

Actually, all the unravelings that we will introduce in this paper enjoy a more regular structure, since they satisfy the following properties: 1) Compositionality: $\mathbf{U}(T_1 \cup T_2) = \mathbf{U}(T_1) \cup \mathbf{U}(T_2)$ 2) Finiteness: R finite $\Rightarrow$ $\mathbf{U}(R)$ finite 3) The unraveling of the empty TRS is the empty TRS. The first condition expresses the fact that the unraveling is compositional, i.e. that we can incrementally build up the unraveling of a CTRS by computing the unravelings of its parts. The second says that finite objects are mapped into finite objects, and the third condition implies that the unraveling of a TRS T is just T.

We call *tidy* an unraveling satisfying the above three properties. By compositionality, tidy unravelings only need to be defined on single rules. Moreover, they are the identity function when restricted to TRSs. Hence, from now on, when defining a (tidy) unraveling we will only define it for rules with a non-void conditional part.

We now turn our attention to properties of CTRSs:

Definition 3.2 Let $\mathcal{P}$ be a property and $\mathbf{U}$ be an unraveling. The property *ultra-*$\mathcal{P}$ (w.r.t. $\mathbf{U}$), briefly $\mathbf{U}(\mathcal{P})$, is defined as follows: $T \in$ *ultra-*$\mathcal{P} \Leftrightarrow \mathbf{U}(T) \in \mathcal{P}$. □

In the sequel, the unraveling will be in most cases considered understood, being clear from the context, and we will simply speak of ultraproperties. Also, the dash after the ultra prefix may be omitted as well.

Definition 3.3 An unraveling is said to be *sound* (resp. *complete*) for a property $\mathcal{P}$ if *ultra-*$\mathcal{P} \Rightarrow \mathcal{P}$ (resp. *ultra-*$\mathcal{P} \Leftrightarrow \mathcal{P}$). Moreover, we say that an unraveling *preserves* the property $\mathcal{P}$ if $\mathcal{P} \Rightarrow$ *ultra-*$\mathcal{P}$. □

Thus, a sound unraveling provides a *sufficient* criterion for the study of the property $\mathcal{P}$ on CTRS: just try to infer $\mathcal{P}$ for its unraveled CTRS (i.e. prove that the CTRS enjoys ultra-$\mathcal{P}$). If the unraveling is complete, this criterion is not only sufficient but even *necessary*.

Another use of unravelings is to lift properties which are typical of TRSs to CTRSs (indeed, observe by Definition 3.2 that, in order to define an ultra property for CTRSs, we need only a property $\mathcal{P}$ of TRSs). For instance, some syntactical properties like being non-duplicating, left-linear, non-collapsing etc. have been introduced for TRSs. Later, they have been extended to CTRSs just saying that the TRS obtained dropping the conditional parts of each rule must be non-duplicating, left-linear and so on. That is to say, one considers for every CTRS R the so-called *unconditional CTRS* (cf. [15]) R_u which from every rule $l \rightarrow r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$ takes the rule $l \rightarrow r$, and requires it to be non-collapsing, left-linear etc.

Now, it is easy to see that *the map* $u : R \mapsto R_u$ *is an unraveling*: hence, what done above is just to lift a property $\mathcal{P}$ of TRSs to CTRSs by taking the ultraproperty *ultra-*$\mathcal{P}$ w.r.t. the unraveling u.

Hence, it makes sense to see what happens if, instead of the 'trivial' unraveling u we consider a more sophisticated unraveling: this is what we will do in this paper.

4 The Transformation

We consider CTRSs composed of terms from a certain set Terms, built up from variables $\mathcal{V}$ and function symbols $\mathcal{F}$ (we assume this universal set of symbols to be fully expressive, in the sense that it contains countable symbols for every arity). In addition, when unraveling a CTRS into a TRS we will need some extra symbols: for every conditional rule ρ we take a new fresh symbol $\mathcal{U}_\rho$; we so define a set Terms$^+$ of 'extended terms' to be the terms obtained from the variables $\mathcal{V}$ and the terms $\mathcal{F}$ plus these new symbols $\mathcal{U}_\rho$.

We also need an operator VAR that once applied to a term gives the sequence of its variables in some fixed order, for instance left-to-right writing order: e.g. $\mathrm{VAR}(f(X, g(Y, X), Z))$ gives the sequence X, Y, X, Z.

Definition 4.1 (Unraveling $\mathbb{U}$)
Take a rule $\rho : l \rightarrow r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$. Its unraveling $\mathbb{U}(\rho)$ is equal to

$$l \rightarrow \mathcal{U}_\rho(s_1, t_1, \ldots, s_k, t_k, \mathrm{VAR}(r))$$
$$\mathcal{U}_\rho(X_1, X_1, \ldots, X_k, X_k, \mathrm{VAR}(r)) \rightarrow r$$

where $X_1, \ldots, X_k$ are fresh variables. □

The explanation of the unraveling should be rather obvious: in order to apply the rule ρ we have first to check its conditional part. So, the first rule performs the rewrite from l to this 'unraveled' test, which is made explicit: the terms in the condition $(s_1, t_1, \ldots, s_k, t_k)$ are gathered by the 'unraveling symbol' $\mathcal{U}$: if they are joinable $(s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k)$ the second rule is applied, completing the original rewrite step of the CTRS. Note that the variables of r $(\mathrm{VAR}(r))$ are passed to the unraveled test (first rule), so that if the test succeeds they can be passed to r (second rule).

As expected, the following result holds:

Theorem 4.2 *$\mathbb{U}$ is a (tidy) unraveling.*

Indeed, one could ask whether the converse hold, that is if given two terms s and t in TERMS, $s{\downarrow_{\mathbb{U}(R)}}t \Rightarrow s{\downarrow_R}t$. It is rather tricky to prove that it is *not* the case (see the appendix and the full version of this paper).

4.1 Soundness and Completeness

We now analyze what properties are complete or sound w.r.t. the introduced unraveling. First we start with the positive results:

Theorem 4.3 *$\mathbb{U}$ is sound for* $\mathrm{CON}^{\rightarrow}$ *and* CON.

Theorem 4.4 *$\mathbb{U}$ is sound for termination.*

Theorem 4.5 *$\mathbb{U}$ is sound for innermost termination.*

The above results are the best we can obtain for $\mathbb{U}$: via counterexamples it can be proved that $\mathbb{U}$ is not complete for all the above properties, and it is not sound for all the other major properties of CTRSs (cf. the full version of this paper). We will come back on this situation later in Section 9, when we will perform an abstract study showing that this situation is not due to a weakness of the unravelings here introduced, but it depends on *general* limitations of unravelings.

5 Some Applications

Many properties of CTRSs like being non-collapsing or non-duplicating have been derived from the corresponding notions for TRSs and applied to CTRSs, in practise, by considering the ultra-properties w.r.t. the trivial unraveling u (i.e. dropping off the conditional parts).

So, first we analyze whether with the more sophisticated unravelings we have introduced something changes.

Lemma 5.1 *$\mathbb{U}$ is complete for the following properties: being non-collapsing, non-overlapping, constructor system, overlay system.*

Therefore, there is no change for the definitions of the above properties even using $\mathbb{U}$ (this is somehow an indication that the above properties for CTRSs have been 'well defined').

However, note that for two main properties the unraveling is not complete: *non-duplication* and *left-linearity*. The lack of the latter is total: indeed, it is easy to check that every CTRS unraveled via $\mathbb{U}$ does not preserve left-linearity unless it is already a TRS; this can be regarded as a sign that left-linearity does not play a role for CTRSs: indeed, it is well known that every CTRS can be simulated via a left-linear CTRS (cf. [5, 4]).

The study of non-duplication, instead, raises some interesting points. Indeed, *ultra-duplication* can, interestingly, be defined syntactically in a nice way. Let $|t|_X$ denote the number of occurrences of the variable X in t. Then we have the following characterization of ultra-non-duplication:

Lemma 5.2 *A CTRS is ultra-non-duplicating if and only if each of its rules $l \rightarrow r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$*

$$\forall X \in \mathrm{Var}(l).\ |l|_X \geq |r, s_1, t_1, \ldots, s_k, t_k|_X$$

That is to say, the original 'naïve' notion of non-duplication ($\forall X \in Var(l).\ |l|_X \geq |r|_X$) is stressed by taking into account not only the right hand side of a rule but also its conditional part.

This is quite useful to see why certain properties that hold for non-duplicating TRSs do not hold for non-duplicating CTRSs. Indeed, what seen could lead to think the usual notion of 'non-duplication' is *not* the correct one (it is defined using the trivial unraveling u). Our notion, if not the right one, seems to be 'more correct' (at least from a modularity viewpoint) than the standard one, as we will see in Section 7.

5.1 Effective Termination

One of the major problems with CTRSs is that the rewrite relation is defined recursively. A bad side-effect of this greater flexibility is that termination of a CTRS does not imply any more the *decidability* of the rewrite relation (cf. [13]). So, while for TRSs termination implies effective computability, for CTRSs basic questions like 'is a term a normal form' or 'does s reduce to t', 'are s and t joinable' and so on can be undecidable in the presence of termination.

To avoid this bad situation, some criteria that ensure, so to say, 'effective termination' have been developed. First of all, it should be clarified what 'effective termination' means. The intuition says that, just like in terminating finite TRSs every "reasonable question" like the ones seen above is decidable, in an effectively terminating finite CTRSs they should be decidable as well.

To cope with such intuition, it has been proposed a set of properties like representative of such 'effective termination' ([14, 12, 5, 4]): a CTRS R is 'effectively terminating' if $\underset{R}{\rightarrow}$ is terminating and it is decidable whether $s\underset{R}{\rightarrow}t$, $s\underset{R}{\rightarrow}^{*}t$, $s{\downarrow_R}t$, and if a term is in normal form.

Three major criteria are known that ensure a (finite) CTRS is effectively computable: *simplifyingness* ([14]), *reductivity* ([12]) and *decreasingness* ([5]). It is well-known that simplifyingness $\Rightarrow$ reductivity $\Rightarrow$ decreasingness (all the implications are strict); thus, decreasingness is the best known condition to ensure feasibility of a CTRS. Its definition is the following:

Definition 5.3 (Decreasingness)
A CTRS R is *decreasing* if there is a partial order $>$ such that

1. $\underset{R}{\rightarrow} \subseteq >$
2. $>$ is well-founded
3. if the conditional rule $l \rightarrow r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$ belongs to R, then for every substitution σ we have that $l\sigma > s_i\sigma$ and $l\sigma > t_i\sigma$ for every $i \in [1, k]$.
4. $>$ has the subterm property (i.e. $C[t] > t$ for every context C and term t). □

Now, we face the problem of effective termination using unravelings: the obvious approach is to use ultra-termination. Comparing ultra-termination and decreasingness, we have that

Lemma 5.4 *Ultra-termination* $\Rightarrow$ *Decreasingness*

Hence, ultra-termination guarantees effective termination:

Corollary 5.5 *Ultra-terminating finite CTRSs are 'effectively terminating'.*

Proof Straightforward by the above Lemma and the result proved in [5, 4]. □

Note that while with decreasingness we are endowed with a theoretical result, with ultra-termination we can utilize all the existing (and future) techniques to prove termination of TRSs.

For instance, one of the more successful techniques developed to prove termination of TRSs is that of simplification orderings (e.g. rpo's), that are widely available in almost every rewrite rule laboratory.

It is well known that Kaplan's criterion of simplifyingness is an attempt to use simplification orderings for CTRSs. It is so interesting to see what happen when we use the 'real' simplification orderings for TRSs in combination with unravelings, in order to prove ultra-termination: i.e., using *ultra-simplifyingness*. Note that ultra-simplifyingness is of particular practical importance, since the transformation is automatic and one only needs to check the simplifyingness of the obtained TRS. Another interesting fact about this criterion is that it is more powerful than Kaplan's:

Lemma 5.6 *Ultra-simplifyingness is strictly more general than simplifyingness.*

The above lemma can be rephrased saying that $\mathbb{U}$ preserves simplifyingness but it is not sound.

In fact, the proof of the above lemma gives the stronger result that ultra-simplifyingness is strictly more powerful than simplifyingness even when restricting to *ground CTRSs* only.

6 Normal Unravelings

In this section we develop the corresponding unravelings for normal CTRSs.

We define a *normal unraveling* to be like an unraveling, but for the domain which is restricted to normal CTRSs. The adjective normal will be often omitted in the sequel.

The definition of the corresponding unraveling for normal CTRSs should look quite obvious:

Definition 6.1 (Unraveling $\mathbb{U}_n$)
Take a rule $\rho : l \to r \Leftarrow s_1 \to_!^* n_1, \ldots, s_k \to_!^* n_k$. Its unraveling $\mathbb{U}_n(\rho)$ is equal to

$$l \to \mathcal{U}_\rho(s_1, \ldots, s_k, \mathrm{VAR}(r))$$
$$\mathcal{U}_\rho(n_1, \ldots, n_k, \mathrm{VAR}(r)) \to r$$

□

Note the $\mathcal{U}_\rho$ symbols here have different arity than in the join CTRSs case.

This map enjoys the same properties as $\mathbb{U}$ (also the proofs are almost the same):

Theorem 6.2 *$\mathbb{U}_n$ is a (tidy) unraveling.*

Theorem 6.3 *$\mathbb{U}_n$ is sound for* $\mathrm{CON}^{\to}$ *and* CON.

Theorem 6.4 *$\mathbb{U}_n$ is sound for termination.*

Theorem 6.5 *$\mathbb{U}_n$ is sound for innermost termination.*

Also, it is a routine matter to see that all the counterexamples developed in the previous sections for join CTRSs carry over to normal CTRSs (indeed, the involved CTRSs have been chosen to be normal CTRSs).

6.1 Left-linear Normal CTRSs

One might wonder why we bother about normal CTRSs, since they are encompassed by join CTRSs and, moreover, it has been shown in [5, 4] that every join CTRS can be simulated by a normal CTRS. The fact is that there is a *fundamental* difference w.r.t. join CTRSs as far as the properties that can be lifted from TRSs to CTRSs are concerned:

Lemma 6.6 *$\mathbb{U}_n$ is complete for the following properties: being left-linear, non-collapsing, non-erasing, non-overlapping, constructor system, overlay system.*

That is, this time *left-linearity is preserved*. This will have striking consequences on modularity, as we will later see in Subsection 7.2 and Section 8. Besides modularity, the preservation of left-linearity is itself a sign that this property, unlike in the join case, plays a role for normal CTRSs (indeed, the simulation of [5, 4] employs *non left-linear* normal CTRSs).

The following theorem shows that, when restricting to the original terms, a left-linear normal CTRS and its unraveling compute the same things:

Theorem 6.7 *For every left-linear normal CTRS R, $\forall s, t \in$ Terms. $s \xrightarrow[R]{}^* t \Leftrightarrow s \xrightarrow[\mathbb{U}_n(R)]{}^* t$*

Next we address soundness and completeness of $\mathbb{U}_n$. The key property of confluence, that was not sound for $\mathbb{U}$ and $\mathbb{U}_n$, can now be recovered:

Theorem 6.8 *$\mathbb{U}_n$ is sound for confluence of left-linear normal CTRSs.*

As a remarkable application, we can easily obtain from TRSs the following major result of Bergstra and Klop ([1]):

Theorem 6.9 *Every orthogonal normal CTRS is confluent.*

Proof Since orthogonal TRSs are confluent, the result follows immediately from Lemma 6.6 and Theorems 6.8, 6.4. □

There is even a property for which we can infer completeness, namely consistency w.r.t. reduction:

Theorem 6.10 *$\mathbb{U}_n$ is complete for* CON$^{\rightarrow}$ *of left-linear normal CTRSs.*

Finally, we obtain right away another soundness property:

Corollary 6.11 *$\mathbb{U}_n$ is sound for completeness of left-linear normal CTRSs.*

6.2 Effective Termination

We face again the problem of effective termination, this time for left-linear normal CTRSs. We have seen that ultra-termination for join CTRSs provide a powerful criterion for effective termination. Yet, it is not powerful as decreasingness. Quite surprisingly, in the left-linear normal CTRSs case we manage to reach decreasingness, since the following result holds:

Theorem 6.12 *For left-linear normal CTRS, Ultra-termination = Decreasingness.*

Hence, the somehow ad-hoc notion of decreasingness is provided with a much more meaningful justification, being just the termination of the unraveled CTRS. We so naturally obtain the result of [5, 4]:

Theorem 6.13 *For left-linear normal CTRSs, decreasing finite CTRSs are 'effectively terminating'.*

Proof Straightforward by the above Theorem and Theorem 6.7. □

Note that all the difficulties and possible objections in expressing what the 'right notion' of effective termination is, are nicely coped with using ultra-termination: the leading intuition was that things for effectively terminating CTRSs should be like for terminating TRSs; and ultratermination is just the concept that the TRS corresponding to a CTRS terminates.

The following is a result similar to that proved by Dershowitz, Okada and Sivakumar ([5]):

Corollary 6.14 *A decreasing left-linear normal CTRS R is confluent if every critical pair of $\mathbb{U}_n(R)$ is convergent.*

Proof We know that for terminating TRSs convergence of critical pairs implies confluence. Since R is decreasing then by Theorem 6.12 $\mathbb{U}_n(R)$ is terminating. So, if every critical pair of $\mathbb{U}_n(R)$ is convergent $\mathbb{U}_n(R)$ is confluent, which implies by Theorem 6.8 that R is confluent as well. □

6.3 Normal Forms Properties

It is well-known that in a CTRS basic problems like checking one-step reductions and so on can be undecidable, as said in the discussion of Subsection 5.1. As said there, in order to gain the basic properties common to every finite terminating TRS, many criteria for 'effective termination' have been given. However, the same problems occur for the normal forms properties UN$^{\rightarrow}$, UN and NF, since in general the problem of being a normal form in a CTRS is undecidable (as shown in [13]). This phenomenon is independent on termination, since being a normal form is decidable for every finite TRS, even nonterminating. To this extent, Bergstra and Klop defined in [1] a criterion for decidability of normal forms that does not require termination: They proved that the subterm property implies decidability of normal forms (recall that a CTRS is said to have the *subterm property*, SP for short, if in every its rule $l \to r \Leftarrow s_1 \to_!^* n_1, \ldots, s_k \to_!^* n_k$ the terms $s_1, \ldots, s_k$ are proper subterms of l).

Interestingly, imposing the SP property recovers soundness of $\mathbb{U}_n$ for all the normal forms properties. We have the following results:

Theorem 6.15 *$\mathbb{U}_n$ is sound for* UN$^{\rightarrow}$ *of left-linear SP normal CTRSs.*

Theorem 6.16 *$\mathbb{U}_n$ is sound for* UN *of left-linear SP normal CTRSs.*

Theorem 6.17 *$\mathbb{U}_n$ is sound for* NF *of left-linear SP normal CTRSs.*

As seen, imposing decidability of normal forms (via SP) allows to recover soundness for all the normal forms properties UN$^{\rightarrow}$, UN and NF. It is therefore natural to ask whether the same holds when imposing the alternative property of decreasingness. Indeed, it is the case (cf. the full version of this paper, where it is proven an even more general result).

6.4 Normalization Properties

As seen, we managed to recover soundness of $\mathbb{U}_n$ for every main property of CTRSs, but for the so-called '(weak) normalization properties', i.e. normalization, innermost normalization and semicompleteness. The fact is that the normalization properties are somehow dual to the normal forms properties: the first ones give a lower bound on the number of normal forms, while the second ones give an upper bound. Indeed, one case see that the same reasonment that allowed to recover soundness for the normal forms properties (Subsection 6.3) gives not soundness, but the dual property of *preservation* for normalization.

There is, however, another way we can follow. As seen, the concept of normal form is in general undecidable even for finite CTRSs, hence it is not fully satisfactory (unless some property like SP or decreasingness is introduced). Using unravelings provide a natural solution to the problem of defining normal forms in CTRSs: just like non-duplication, one can lift the concept of normal forms of TRSs to 'ultra normal forms' for CTRSs. Indeed, call a term $n \in$ TERMS an *ultra normal form* for the CTRS R if it is a normal form in $\mathbb{U}(R)$. It is readily seen that this definition coincides for every other unraveling here mentioned, i.e. $\mathbb{U}_n$ and u. Using the latter, we can rephrase saying that a term n is an ultra normal form in a CTRS iff it is a normal form for its unconditional CTRS (note that the same problem occurred when defining normal CTRSs, and indeed was analogously solved in [1]). Thus, we can define:

Definition 6.18

- A CTRS is *strictly normalizing* (briefly SWN) if in it every term has an ultra normal form
- A CTRS is *strictly innermost normalizing* if in it every term rewrites by innermost rewriting to an ultra normal form.
- A CTRS is *strictly semicomplete* if it is confluent and strictly normalizing. □

Note that strict normalization implies normalization, and that, for TRSs, these properties coincide.

Now, we modify the definitions of the unraveling $\mathbb{U}_n$, introducing a new unraveling specifically suited for the normalization properties:

Definition 6.19 (Unraveling $\widetilde{\mathbb{U}}_n$)
Take a rule $\rho : l \to r \Leftarrow s_1 \to_!^* n_1, \ldots, s_k \to_!^* n_k$, with $|\mathrm{VAR}(r)| = m$. Its unraveling $\widetilde{\mathbb{U}}_n(\rho)$ is equal to $\mathbb{U}_n(\rho)$ plus the rule

$$\mathcal{U}_\rho(X_1, \ldots, X_k, Y_1, \ldots, Y_m) \to \mathcal{U}_\rho(X_1, \ldots, X_k, Y_1, \ldots, Y_m)$$ □

The intuition is that the original transformation $\mathbb{U}_n$ could produce 'spurious' normal forms containing unresolved $\mathcal{U}$ symbols. Adding the above rule, we require every normal form does not contain an $\mathcal{U}$ symbol, and so, intuitively, that the unraveling cannot add normal forms to the original CTRS.

Note that since $\xrightarrow[\mathbb{U}_n(R)]{}^* = \xrightarrow[\widetilde{\mathbb{U}}_n(R)]{}^*$, all the nice properties of $\mathbb{U}_n$ (Theorems 6.2, 6.7, 6.3, 6.8) are still satisfied. The drawback is that we lose a satisfactory treatment of termination (ultra-termination w.r.t. $\widetilde{\mathbb{U}}_n$ is just the class of TRSs).

We can now see why this modification is useful:

Theorem 6.20 *$\widetilde{\mathbb{U}}_n$ is sound for strict normalization.*

Theorem 6.21 *$\widetilde{\mathbb{U}}_n$ is sound for strict innermost normalization.*

Theorem 6.22 *$\widetilde{\mathbb{U}}_n$ is sound for strict semicompleteness.*

7 Modularity

We will now apply the developed unravelings to the study of modularity properties for CTRSs. As stated in every work on the modularity for CTRSs, modularity for CTRSs is much more complicated and subtle than for TRSs. Unravelings provide a tool to automatically transfer results obtained in the TRSs field to CTRSs, not only for the modularity of disjoint combinations, but even for the unexplored field of more involved combinations like *composable* or even *hierarchical*.

Recall that a map μ is said *compositional* w.r.t. an operator $\odot$ ($\odot$-compositional for short) if $\forall R, S. \mu(R \odot S) = \mu(R) \odot \mu(S)$. The following easy but fundamental result explains why unravelings are so useful for the study of modularity:

Theorem 7.1 *Suppose an unraveling is $\odot$-compositional. Then $\mathcal{P}$ is $\odot$-modular for TRSs $\Rightarrow$ ultra-$\mathcal{P}$ is $\odot$-modular for CTRSs.*

Proof Take an unraveling $\mathbf{U}$ which is sound for $\mathcal{P}$. If $\mathcal{P}$ is $\odot$-modular for TRSs, then from $R_1 \in$ *ultra-*$\mathcal{P} \Rightarrow \mathbf{U}(R_1) \in \mathcal{P}$ and $R_2 \in$ *ultra-*$\mathcal{P} \Rightarrow \mathbf{U}(R_2) \in \mathcal{P}$ we get $\mathbf{U}(R_1) \odot \mathbf{U}(R_2) \in \mathcal{P}$, which implies $\mathbf{U}(R_1 \odot R_2) \in \mathcal{P}$ (by $\odot$-compositionality), which is equivalent to $R_1 \odot R_2 \in$ *ultra-*$\mathcal{P}$. □

When the unraveling is complete, we get an even stronger result:

Corollary 7.2 *Suppose an unraveling is complete for $\mathcal{P}$ and $\odot$-compositional. Then $\mathcal{P}$ is $\odot$-modular for TRSs $\Leftrightarrow$ $\mathcal{P}$ is $\odot$-modular for CTRSs.*

Proof One direction is obvious. The other follows right away from the above theorem since *ultra-*$\mathcal{P} = \mathcal{P}$ by completeness. □

Therefore, to apply an unraveling to the modularity of combinations w.r.t. some operator $\odot$, it must be the case it is $\odot$-compositional. The unravelings here developed are compositional w.r.t. all the modularity operators so far introduced (cf. e.g. [25, 15]), that is to say, in order of increasing power, the *disjoint union* $\oplus$ (disjoint signatures), *constructor-sharing union* $\underset{cs}{\oplus}$ (sharing only of constructor symbols), and *composable union* $\underset{comp}{\oplus}$ (rules defining shared defined symbols are shared):

Theorem 7.3 (Compositionality)
The unravelings $\mathbb{U}$, $\mathbb{U}_n$ and $\widetilde{\mathbb{U}}_n$ are compositional w.r.t. the operators $\oplus$, $\underset{cs}{\oplus}$ and $\underset{comp}{\oplus}$.

The even more involved case of *hierarchical combinations* will be treated separately in Section 8.

When talking about the property *ultra-*$\mathcal{P}$ without mentioning the corresponding unraveling, we consider understood it is $\widetilde{\mathbb{U}}_n$ if we mention one of the normalization properties, $\mathbb{U}_n$ for all the other properties of left-linear normal CTRSs, and $\mathbb{U}$ otherwise.

7.1 Join CTRSs

Termination

No results on the modularity of termination for composable CTRSs are so far known, but for the result of [24] showing the modularity of simplifyingness for composable CTRSs.

We will now show what we can automatically obtain using unravelings.

Theorem 7.4 *Ultra-termination is modular for non-collapsing composable CTRSs.*

Proof By Lemmata 5.1, 5.4, Theorem 7.1 and the modularity of termination for non-collapsing composable TRSs ([25]). □

Theorem 7.5 *Ultra-termination is modular for non-overlapping composable CTRSs.*

Proof By Lemmata 5.1, 5.4, Theorem 7.1 and the modularity of termination for non-overlapping composable TRSs ([2, 25]). □

As far as non-duplication is concerned, it is known that termination is modular for non-duplicating TRSs ([32]). Middeldorp observed (cf. [22]) that this result does not carry over to CTRSs (even for decreasing CTRSs). So, he tried to give stronger conditions restoring modularity, and managed to prove modularity imposing confluence. This can be seen as an example of what said in Section 5 on the fact that the notion of non-duplication, as has been lifted to CTRSs via the trivial unraveling u, is likely not to be the right extension to CTRSs of this concept. Indeed, employing non-ultraduplication we get:

Theorem 7.6 *Ultra-termination is modular for non-ultraduplicating composable CTRSs.*

Proof By Lemmata 5.1, 5.4, Theorem 7.1 and the modularity of termination for non-duplicating composable TRSs ([9, 25]). □

One of the most powerful results is the modularity of $\mathcal{C}_\mathcal{E}$-termination (cf. [28, 9]), that has been extended to finitely branching disjoint CTRSs ([8]). It is an open problem whether this result holds for disjoint CTRSs as well. Recall that a TRS T is $\mathcal{C}_\mathcal{E}$-terminating if $T \oplus \{or(X,Y) \to X, or(X,Y) \to Y\}$ is terminating. Interestingly, the lifted ultraproperty can also be given a similar nice formulation: It can be proved that T *is ultra-$\mathcal{C}_\mathcal{E}$-terminating iff* $T \oplus \{or(X,Y) \to X, or(X,Y) \to Y\}$ *is ultra-terminating.*

Theorem 7.7 *Ultra-$\mathcal{C}_\mathcal{E}$-termination is modular for CTRSs.*

Proof By Theorem 7.1, Lemma 5.4, and the modularity of $\mathcal{C}_\mathcal{E}$-termination for TRSs ([9, 28]). □

Corollary 7.8 *Ultra-termination is modular for ultra-$\mathcal{C}_\mathcal{E}$-terminating CTRSs.*

Proof By the above theorem, since ultra-$\mathcal{C}_\mathcal{E}$-termination implies ultra-termination. □

We now consider simplifyingness: as said, its modularity for composable CTRSs ([24]) is the only result so far known on the modularity of decreasing CTRSs. The interest of the simplifying property for TRSs, as well known, lies in the fact that is practically automatizable (e.g. using rpo's or other similar orderings). Hence this result is of particular practical importance:

Theorem 7.9 *Ultra-simplifyingness is modular for composable CTRSs.*

Proof By Theorem 7.1, Lemma 5.4, and the modularity of simplifyingness for composable TRSs ([16]). □

When comparing the above result with the aforementioned result [24], by Lemma 5.6 we obtain that it is *strictly more powerful*.

Other similar results can be obtained lifting the modularity results for simple termination (cf. [16, 25]), for instance:

Theorem 7.10 *Ultra simple termination is modular for CTRSs.*

Proof By Theorems 7.1, Lemma 5.4, and the modularity of simple termination for composable TRSs ([16]). □

Another important result that can be lifted from TRSs to CTRSs is that obtained by Middeldorp in [21]: he proved that if one of two terminating TRSs is both non-collapsing and non-duplicating, then their disjoint sum is terminating as well. Ohlebusch ([25]) extended this result to composable union of TRSs. When trying to lift this result to CTRSs, Middeldorp found it is no more true (even for ultra-terminating CTRSs). Being the situation in all similar to what mentioned above on the modularity of termination for non-duplicating CTRSs, the most natural expection was that imposing confluence on the other CTRS modularity is restored (as indeed conjectured in [22]). However, it was found in [25] that this conjecture is not true. All this can be seen as another confirmation that not only non-duplication for CTRSs as defined via the trivial unraveling u may not be the right extension (indeed, the mentioned counterexample involves an ultraduplicating CTRS), but once more of the fact that modularity for CTRSs is extremely complicated and deceptive, especially when trying to rely on previous results of TRSs.

Using unravelings, we obtain right away the following result even for composable CTRSs:

Theorem 7.11 *If one of two ultra-terminating CTRSs is both non-collapsing and non-ultraduplicating, then their composable union is ultra-terminating.*

Proof By Theorem 7.1, Lemmata 5.4, 5.1, and the aforementioned result of [25]. □

Finally, we tackle innermost termination: this property has been proved to be modular for disjoint CTRSs in [10]. No results are known for composable unions.

Theorem 7.12 *Ultra innermost termination is modular for composable CTRSs.*

Proof By Theorems 7.1, 4.5 and the modularity of innermost termination for composable TRSs ([7, 25]). □

Consistency Properties

So far, there were no results on the modularity of CON for CTRSs.

Theorem 7.13 *Ultra-*CON *is modular for CTRSs.*

Proof By Theorems 7.1, 7.3 and 4.3, since CON is modular for TRSs ([33]). □

7.2 Normal CTRSs

So far, there is *not a single result* specific for the modularity of normal CTRSs. This class has somehow been neglected in view of the fact, as said earlier, that every join CTRS can be simulated by a normal CTRS, and so it seemed that restricting to normal CTRSs was unuseful. However, we have seen that for normal CTRSs there is a fundamental difference to join CTRSs: unravelings manage to carry over *left-linearity*. In turn, this allows to state for the first time modularity results proper of normal CTRSs, showing that not only left-linearity *does* play a role for conditional rewriting, but also proving that the restriction to left-linearity is strict for normal CTRSs: normal CTRSs have been shown to be of the same logical strength than join CTRSs (cf. [5, 4]), but the modularity results here presented show that left-linear normal CTRSs have strictly less logical strength, thus providing a new intermediate degree in the hierarchy of CTRSs' expressive power.

Termination

No results are known for the modularity of decreasingness (again, but for the result of [24] showing the modularity of simplifyingness for composable CTRSs).

Theorem 7.14 *Decreasingness is modular for non-collapsing left-linear composable normal CTRSs.*

Proof By Lemma 6.6, Theorems 7.1, 6.12, and the modularity of termination for non-collapsing composable TRSs ([25]). □

Theorem 7.15 *Decreasingness is modular for non-overlapping left-linear composable normal CTRSs.*

Proof By Lemma 6.6, Theorems 7.1, 6.12, and the modularity of termination for non-overlapping composable TRSs ([2, 25]). □

Theorem 7.16 *Decreasingness is modular for non-ultraduplicating left-linear composable normal CTRSs.*

Proof By Lemma 6.6, Theorems 7.1, 6.12, and the modularity of termination for non-duplicating composable TRSs ([9, 25]). □

Theorem 7.17 *Ultra-$\mathcal{C}_\mathcal{E}$-termination is modular for left-linear normal CTRSs.*

Proof By Theorems 7.1, 6.12, and the modularity of $\mathcal{C}_\mathcal{E}$-termination for TRSs ([9, 28]). □

Analogously to what previously seen for ultra-$\mathcal{C}_\mathcal{E}$-termination of join CTRSs, it can be proved that *T is ultra-$\mathcal{C}_\mathcal{E}$-terminating (for left-linear normal CTRSs) iff $T \oplus \{or(X,Y) \to X, or(X,Y) \to Y\}$ is decreasing.*

Corollary 7.18 *Decreasingness is modular for ultra-$\mathcal{C}_\mathcal{E}$-terminating left-linear normal CTRSs.*

Proof By the above theorem, since ultra-$\mathcal{C}_\mathcal{E}$-termination implies decreasingness. □

Theorem 7.19 *Decreasingness is modular for left-linear* $\mathrm{CON}^{\to}$ *normal CTRSs.*

Proof By Theorems 6.12, 7.1, Lemma 6.6 and the modularity of termination for left-linear $\mathrm{CON}^{\to}$ TRSs ([19, 34]). □

Theorem 7.20 *If one of two decreasing left-linear normal CTRSs is both* $\mathrm{CON}^{\to}$ *and ultra-$\mathcal{C}_\mathcal{E}$-terminating, then their disjoint union is decreasing.*

Proof By Theorems 6.12, 7.1, Lemma 6.6 and the result proved in [17] for TRSs. □

Confluence

Confluence is modular for non-collapsing CTRSs ([22]), but no results were so far known for the modularity of confluence for non-collapsing CTRSs even in the constructor-sharing case.

Theorem 7.21 *Ultra-confluence is modular for left-linear non-collapsing composable normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.8, since confluence is modular for non-collapsing composable TRSs ([25]). □

Theorem 7.22 *Ultra-confluence is modular for left-linear constructor-sharing normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.8, since confluence is modular for left-linear constructor-sharing TRSs ([31]). □

Note that if confluence is proven to be modular for left-linear composable TRSs (a conjecture which is widely believed to hold), then we will automatically be able to say that ultra-confluence is modular for left-linear composable normal CTRSs.

Completeness

Due to space limitations, we refer the interested reader to the full version of this paper for the whole listing of the modularity results for ultra-completeness obtained by using the corresponding results for TRSs, and by combining the results here obtained on the modularity of decreasingness with those on the modularity of confluence.

Consistency Properties

There are no results on the modularity of $CON^{\rightarrow}$ for CTRSs. Via unravelings, we can obtain the following important result:

Theorem 7.23 $CON^{\rightarrow}$ *is modular for left-linear normal CTRSs.*

Proof By Lemma 6.6, Corollary 7.2, and Theorems 7.3 and 6.10, since $CON^{\rightarrow}$ is modular for left-linear TRSs, as proved in [19] (see [34, 17] for alternative proofs). □

Normal Forms Properties

Theorem 7.24 *Ultra-*$UN^{\rightarrow}$ *is modular for left-linear SP normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.15, since $UN^{\rightarrow}$ is modular for left-linear TRSs ([19]). □

Theorem 7.25 *Ultra-*UN *is modular for left-linear SP normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.16, since UN is modular for TRSs ([20]). □

Theorem 7.26 *Ultra-*NF *is modular for left-linear SP normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.17, since NF is modular for left-linear TRSs ([22]). □

Note that in the above three results we have made implicitly use of the fact that SP is modular.

Normalization Properties

Normalization has been proved to be modular for disjoint CTRSs in [22]. No results are known for composable unions.

Theorem 7.27 *Ultra strict normalization is modular for left-linear composable normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.20, since normalization is modular for composable TRSs ([25]). □

Analogously, innermost normalization has been proved to be modular for disjoint CTRSs in [10], but no results on composable unions are known.

Theorem 7.28 *Ultra strict innermost normalization is modular for left-linear composable normal CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.21, since innermost normalization is modular for composable TRSs ([25]). □

Finally, semi-completeness: it is modular for constructor-sharing CTRSs ([26]), but no results are known for composable CTRSs.

Theorem 7.29 *Ultra strict semicompleteness is modular for composable CTRSs.*

Proof By Lemma 6.6 and Theorems 7.1, 7.3 and 6.21, since semicompleteness is modular for composable TRSs ([27, 25]). □

8 Hierarchical Combinations

In Section 7 we treated all the so far known composition operators, but for the most recent (and involved) case: hierarchical combinations. Hierarchical combinations are particularly interesting because they allow sharing of defined symbols in a much more flexible way than composable union, and are thus extremely more useful in practise. However, this flexibility also reflects the fact their analysis is extremely hard. It is not surprising, therefore, that so far there are no results on the modularity of hierarchical combinations (but for a slight extension stated in [30]). Even in this setting, however, unravelings allow to lift existing results from TRSs to CTRSs.

Formally, two TRSs R_1 and R_2 are said to form a hierarchical combination (cf. [25]) if R_1 does not have defined symbols of R_2, and no defined symbols of R_1 appears in the lhs's of the rules of R_2. R_1 is then said the *base* of the combination and R_2 the *extension*. Just like in the constructor-sharing and composable case, the word 'combination' is usually omitted when talking about modular properties.

Since no notion of 'hierarchical combination for CTRSs' has so far been defined, we define it by lifting the property from TRSs to CTRSs via unraveling.

For join CTRSs, the notion of hierarchical combination turns out to be just the one given by the trivial unraveling u (i.e. R_1 does not have defined symbols of R_2, and no defined symbols of R_1 appears in the lhs's of the rules of R_2).

For normal CTRSs, instead, usage of $\mathbb{U}_n$ leads to a slightly different definition:

Definition 8.1 Two normal CTRSs R_1 and R_2 are said to form a *hierarchical combination* if R_1 does not have defined symbols of R_2, and for every rule $l \to r \Leftarrow s_1 \to^*_! n_1, \ldots, s_k \to^*_! n_k$ of R_2 no defined symbol of R_1 appears in $l, n_1, \ldots, n_k$. □

The above definition is correct in the following sense, as it is easy to check:

Lemma 8.2 *Two normal CTRSs R_1 and R_2 form a hierarchical combination if and only if $\mathbb{U}_n(R_1)$ and $\mathbb{U}_n(R_2)$ form a hierarchical combination.*

As said above, Rao in [30] proved that simplifyingness is modular for hierarchical TRSs forming a 'proper extension'. From his result we obtain:

Theorem 8.3 *Ultra-simplifyingness is modular for hierarchical CTRSs forming a ultra proper extension.*

Using Lemma 5.6, we obtain that the above result is *strictly more powerful* than the result proved in [30], stating the modularity of simplifyingness for hierarchical CTRSs forming a proper extension.

We remark that, just like for non-ultraduplication (cf. Lemma 5.2) or for ultra-$\mathcal{C}_\mathcal{E}$-termination (cf. Subsection 7.1) one can express an ultraproperty in an equivalent way without mentioning the unraveling, the same occurs here: one can adapt the notions of the above papers (e.g. the concepts of proper extension, dependency relation etc. of Rao) extending them to the CTRSs case, thus obtaining a nice reformulation of the modularity result in term of the CTRS itself (cf. the full version of the paper).

As far as normal CTRSs are concerned, we can obtain some specific important results.

By Corollary 6.11, we can lift by unravelings also all the other results obtained for the completeness hierarchical combinations of TRSs: those of Dershowitz ([2]) and Rao ([29]) on completeness.

Another result is obtained by lifting the recent result of Verma ([37]) on the modularity of confluence: he treats a kind of combination, called *lr*-combination, that is even more general than those considered in the aforementioned papers. We can lift his result obtaining a sufficient criterion of termination for the *lr*-combination of left-linear normal CTRSs.

The last useful result copes with the general case of hierarchical combinations:

Theorem 8.4 *Ultra-confluence is modular for hierarchical left-linear normal CTRSs.*

Proof By Lemma 8.2 and Theorems 7.1, 7.3 and 6.8, since confluence is modular for hierarchical left-linear TRSs (cf. [31]). □

9 On the Power of Unraveling

We have seen that $\mathbb{U}$ and $\mathbb{U}_n$ are not complete for any of the major properties of CTRSs. The natural question is so: can we improve on the results that we have presented? Maybe this fact is just a weakness of the presented unravelings, and a more sophisticated unraveling, specifically tailored for a particular property (e.g. completeness, termination etc.) could improve on $\mathbb{U}$ and $\mathbb{U}_n$.

However, the following results show that this is not the case: we cannot find an unraveling which is complete for completeness or any other major property. Note that these results, stated for unravelings, also hold for normal unravelings (cf. the full version of this paper).

Theorem 9.1 *There is no unraveling which is complete for completeness.*

So, $\mathbb{U}$ and $\mathbb{U}_n$ represent for completeness one of the two different sides of the coin: they satisfactorily represents one face of it (termination) because no unraveling can grasp both of them.

We have seen that $\mathbb{U}_n$ behaves in a sense as the best possible unraveling for termination: this intuitively follows by Theorem 6.12 and the observation that every more powerful unraveling would give via (the proof of) Theorem 6.13 a more powerful criterion for effective termination than decreasingness.

This observation is also formally justified in view of the following result:

Theorem 9.2 *There is no unraveling which is complete for termination.*

As far as all the other properties are concerned, we have the same impossibility results:

Theorem 9.3 *There is no unraveling which is complete for confluence,* $\mathrm{UN}^{\rightarrow}$*,* UN*,* NF*, normalization, innermost termination, innermost normalization and semicompleteness.*

Note that, even if we have gathered the above impossibility results in a unique theorem for space reasons, each of the above properties requires a different separate proof.

Thus, the above analysis not only shows that in a sense the presented results are the best possible, but exactly clarifies what are the power and the intrinsic limit of the unraveling approach.

References

[1] J.A. Bergstra and J.W. Klop. Conditional rewrite rules: Confluence and termination. *JCSS*, 32(3):323-362, 1986.

[2] N. Dershowitz. Hierarchical termination. *Proc. 4th CTRS*, LNCS 968, 1995.

[3] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6, pages 243–320. Elsevier – MIT Press, 1990.

[4] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *TCS*, 75:111-138, 1990.

[5] N. Dershowitz, M. Okada, and G. Sivakumar. Canonical conditional rewrite systems. In *Proceedings of the 9th Conference on Automated Deduction*, volume 310 of *LNCS*, pages 538–549. Springer–Verlag, 1988.

[6] E. Giovannetti and C. Moiso. Notes on the elimination of conditions. *Proc. 1st CTRS*, LNCS 308, 1988.

[7] B. Gramlich. Relating innermost, weak, uniform and modular termination of term rewriting systems. *Proc. LPAR*, LNAI 624, pages 285–296. Springer-Verlag, 1992.

[8] B. Gramlich. Sufficient conditions for modular termination of conditional term rewriting systems. In *Proc. 3rd CTRS*, LNCS 656, pages 128–142. Springer-Verlag, 1993.

[9] B. Gramlich. Generalized sufficient conditions for modular termination of rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 5:131–158, 1994.

[10] B. Gramlich. On modularity of termination and confluence properties of conditional rewrite systems. *Proc. 4th Int. Conf. on Algebraic and Logic Programming*, vol. 850 of *LNCS*, pages 186–203. Springer-Verlag, 1994.

[11] C. Hintermeier. How to transform canonical decreasing CTRSs into equivalent canonical TRSs. *Proc. 4th CTRS*, LNCS 968, pages 186–205. Springer–Verlag, 1995.

[12] J.-P. Jouannaud and B. Waldmann. Reductive conditional term rewrite systems. In *3rd IFIP Working Conference on Formal Description of Programming Concepts*, pages 223–244, Ebberup, Denmark, 1986.

[13] S. Kaplan. Conditional rewrite rules. *Theoretical Computer Science*, 33(2):175–193, 1984.

[14] S. Kaplan. Simplifying conditional term rewriting systems. *J. of Symbolic Computation*, 4(3):295–334, 1987.

[15] J.W. Klop. Term rewriting systems. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 1–116. Clarendon Press, Oxford, 1992.

[16] M. Kurihara and A. Ohuchi. Modularity of simple termination of term rewriting systems. *Journal of IPS Japan*, 31(5):633–642, 1990.

[17] M. Marchiori. Bubbles in modularity. Technical Report 5, Dept. of Pure and Applied Mathematics, University of Padova, 1995. Submitted to *TCS*.

[18] M. Marchiori. Modularity of completeness revisited. In J. Hsiang, editor, *Proceedings of the Sixth International Conference on Rewriting Techniques and Applications*, volume 914 of *LNCS*, pages 2–10. Springer–Verlag, 1995.

[19] M. Marchiori. On the modularity of normal forms in rewriting. *J. of Symbolic Computation*, 1996. To appear.

[20] A. Middeldorp. Modular aspects of properties of term rewriting systems related to normal forms. *Proc. 3rd RTA*, volume 355 of *LNCS*, pages 263–277. Springer–Verlag, 1989.

[21] A. Middeldorp. A sufficient condition for the termination of the direct sum of term rewriting systems. In *Proceedings of the Fourth IEEE Symposium on Logic in Computer Science*, pages 396–401, 1989.

[22] A. Middeldorp. Modular properties of conditional term rewrite systems. *Information and Computation*, 104(1):110–158, 1993.

[23] A. Middeldorp and Y. Toyama. Completeness of combinations of constructor systems. *Journal of Symbolic Computation*, 15(3):331–348, 1993.

[24] E. Ohlebusch. Combinations of simplifying conditional term rewriting systems. *Proc. 3rd CTRS*, volume 656 of *LNCS*, pages 113–127. Springer–Verlag, 1993.

[25] E. Ohlebusch. *Modular Properties of Composable Term Rewriting Systems*. PhD thesis, Universität Bielefeld, 33501 Bielefeld, FRG, May 1994. Appeared as Technical Report 94-01.

[26] E. Ohlebusch. Modular properties of constructor-sharing conditional term rewriting systems. In *Proceedings 4th International Workshop on Conditional and Typed Rewriting Systems*, LNCS. Springer–Verlag, 1994.

[27] E. Ohlebusch. On the modularity of confluence of constructor-sharing term rewriting systems. In *Proc. 19th Colloquium on Trees in Algebra and Programming*, volume 787 of *LNCS*, pages 261–275. Springer–Verlag, 1994.

[28] E. Ohlebusch. On the modularity of termination of term rewriting systems. *TCS*, 136(2):333-360, 1994.

[29] K. Rao. Completeness of hierarchical combinations of term rewriting systems. In *Proc. STACS*, volume 761 of *LNCS*, pages 125–139. Springer–Verlag, 1993.

[30] K. Rao. Simple termination of hierarchical combinations of term rewriting systems. In M. Hagiya and J.C. Mitchell, editors, *Proc. TACS*, volume 789 of *LNCS*, pages 203–223. Springer–Verlag, 1994.

[31] K.-C. Raoult and J. Vuillemin. Operational and semantic equivalence between recursive programs. *Journal of the ACM*, 27(4):772–796, 1980.

[32] M. Rusinowitch. On termination of the direct sum of term rewriting systems. *IPL*, 26:65–70, 1987.

[33] M. Schmidt-Schauß. Unification in a combination of arbitrary disjoint equational theories. *Journal of Symbolic Computation*, 8(1,2):51–99, 1989.

[34] M. Schmidt-Schauß, M. Marchiori, and S.E. Panitz. Modular termination of r-consistent and left-linear term rewriting systems. *Theoretical Computer Science*, 149(2):361–374, 1995.

[35] Y. Toyama, J.W. Klop, and H.P. Barendregt. Termination for the direct sum of left-linear term rewriting systems. In N. Dershowitz, editor, *Proc. 3rd RTA*, volume 355 of *LNCS*, pages 477–491, Springer–Verlag, 1989.

[36] Y. Toyama, J.W. Klop, and H.P. Barendregt. Termination for direct sums of left-linear complete term rewriting systems. *Journal of the ACM*, 42(6):1275–1304, November 1995.

[37] R.M. Verma. Unique normal forms and confluence of rewrite systems: Persistence. In *Proc. 14th IJCAI*, volume 1, pages 362–368, 1995.

A Selected Proofs

Many results appear in the paper without proofs, due to space limitations. All the proofs, together with other new results, can be found in the full version of this paper, appeared as Technical Report n.8, Dept. of Pure and Applied Mathematics, University of Padova, December 1995. However, for the sake of completeness, we include here some selected proofs.

Note: for better readability, when unraveled CTRSs have only one $\mathcal{U}_\rho$ symbol we omit the subscript ρ, writing simply $\mathcal{U}$.

[Proof of Theorem 4.2]
We will prove the stronger implication $s \underset{R}{\rightarrow} t \Rightarrow s \underset{\mathbb{U}(R)}{\longrightarrow}^{+} t$.

So, suppose the rule used in $s \underset{R}{\rightarrow} t$ is $\rho: l \to r \Leftarrow s_1{\downarrow}t_1, \ldots, s_k{\downarrow}t_k$, and that $s = C[l\sigma]$, $t = C[r\sigma]$ for some substitution σ. The proof is by induction on the depth of the reduction $s \underset{R}{\rightarrow} t$:

- If the depth is 0, then the rule ρ is of the form $l \to r$, and so $\mathbb{U}(\rho) = l \to r \in \mathbb{U}(R)$, which gives right away $s \underset{\mathbb{U}(R)}{\longrightarrow}^{+} t$.
- If the depth is greater than 0, then by induction $s_1\sigma{\downarrow}_{\mathbb{U}(R)}t_1\sigma, \ldots, s_k\sigma{\downarrow}_{\mathbb{U}(R)}t_k\sigma$. So, since $\mathbb{U}(\rho)$ gives the two rules $\rho_1 : l \to \mathcal{U}_\rho(s_1, t_1, \ldots, s_k, t_k, \mathrm{VAR}(r))$ and $\rho_2 : \mathcal{U}_\rho(X_1, X_1, \ldots, X_k, X_k, \mathrm{VAR}(r)) \to r$, we have

$$\begin{array}{rcl} s = C[l\sigma] & \underset{\rho_1}{\rightarrow} & C[\mathcal{U}_\rho(s_1\sigma, t_1\sigma, \ldots, s_k\sigma, t_k\sigma, \mathrm{VAR}(r)\sigma)] \\ & \underset{\mathbb{U}(R)}{\longrightarrow}^{*} & C[\mathcal{U}_\rho(u_1, u_1, \ldots, u_k, u_k, \mathrm{VAR}(r)\sigma)] \quad \text{(for some } u_1, \ldots, u_k\text{)} \\ & \underset{\rho_2}{\rightarrow} & C[r\sigma] = t \end{array}$$

[Proof of $s \underset{R}{\rightarrow}^{*} t \not\Leftarrow s \underset{\mathbb{U}(R)}{\longrightarrow}^{*} t$]
Take the CTRS

$$\mathcal{R} = \left\{ \begin{array}{l} f(X) \to X \Leftarrow X{\downarrow}e \\ h(X,X) \to g(X,X,f(k)) \\ g(d,X,X) \to A \\ a \to c \\ b \to c \\ a \to d \\ b \to d \\ c \to e \\ c \to l \\ d \to m \\ k \to l \\ k \to m \end{array} \right.$$

Its unraveled CTRS is

$$\mathbb{U}(\mathcal{R}) = \left\{ \begin{array}{l} f(X) \to \mathcal{U}(X,e,X) \\ \mathcal{U}(X_1,X_1,X) \to X \\ h(X,X) \to g(X,X,f(k)) \\ g(d,X,X) \to A \\ a \to c \\ b \to c \\ a \to d \\ b \to d \\ c \to e \\ c \to l \\ d \to m \\ k \to l \\ k \to m \end{array} \right.$$

Now, in $\mathbb{U}(\mathcal{R})$ there is the following reduction:

$$h(f(a),f(b)) \to^* h(\mathcal{U}(c,e,d),f(b)) \to^* h(\mathcal{U}(c,e,d),\mathcal{U}(c,e,d)) \to$$
$$g(\mathcal{U}(c,e,d),\mathcal{U}(c,e,d),f(k)) \to g(\mathcal{U}(e,e,d),\mathcal{U}(c,e,d),f(k)) \to g(d,\mathcal{U}(c,e,d),f(k)) \to^*$$
$$g(d,\mathcal{U}(l,e,m),f(k)) \to^* g(d,\mathcal{U}(l,e,m),\mathcal{U}(l,e,m)) \to A$$

That is to say, $h(f(a),f(b)) \underset{\mathbb{U}(\mathcal{R})}{\longrightarrow}^* A$.

But, this reduction cannot occur in $\mathcal{R}$, as we will now prove.

If $h(f(a),f(b))$ rewrites to A, it must use the rule $h(X,X) \to g(X,X,f(k))$. So, $f(a)$ and $f(b)$ must be first reduced to a common term: we have the possibilities $f(c),f(d),f(l),f(m),c,d,e,l,m$. But then to arrive to A we must also employ the rule $g(d,X,X) \to A$, which means that this common term must be rewritten to d.

For the terms $f(c),f(d),f(l)$ and $f(m)$, this can happen only by applying the rule $f(X) \to X \Leftarrow X{\downarrow}e$: so, the choice narrows to the single term $f(c)$. But then $f(c)$ reduces to c, which cannot be rewritten to d.

On the other hand, when considering the terms c,d,e,l,m the choice narrows to d. So, we have a reduction of the form $h(f(a),f(b)) \to^* h(d,d) \to g(d,d,f(k)) \to \cdots$. But to apply the rule $g(d,X,X) \to A$ we need to join d and $f(k)$, which is impossible since d rewrites only to m, and $f(k)$ rewrites only to $f(m)$.

Hence, the statement $\forall s,t \in$ TERMS. $s \underset{R}{\to}^* t \Leftrightarrow s \underset{\mathbb{U}(R)}{\longrightarrow}^* t$ does *not* hold.

Since A is a normal form in $\mathcal{R}$, the counterexample also falsifies the statement $\forall s,t \in$ TERMS. $s{\downarrow}_R t \Leftrightarrow s{\downarrow}_{\mathbb{U}(R)} t$.

....

[Proof of Theorem 6.7]

First, we introduce some notation that will be needed in the sequel.

Definition A.1 The $\mathcal{U}$*-rank* of a term is inductively defined as follows:

$\mathcal{U}\text{-}rank(a) = 0$	if a is either a constant or a variable
$\mathcal{U}\text{-}rank(f(t_1,\ldots,t_k)) = 1 + \max_{i\in[1,k]}\{\mathcal{U}\text{-}rank(t_i)\}$	if f is an $\mathcal{U}$ symbol
$\mathcal{U}\text{-}rank(f(t_1,\ldots,t_k)) = \max_{i\in[1,k]}\{\mathcal{U}\text{-}rank(t_i)\}$	if f is not an $\mathcal{U}$ symbol

Analogously, the $\mathcal{U}$*-depth* of a reduction $s_1 \to \ldots \to s_n$ is $\max_{j\in[1,n]}\{\mathcal{U}\text{-}rank(s_j)\}$. □

Following a Gentzen-like style, call '$\mathcal{U}$-introduction' the first rule produced by $\mathbb{U}_n$ and '$\mathcal{U}$-elimination' the second (they vaguely correspond to the rules in natural deduction that introduce and eliminate constructs). Analogously, we talk about elimination and introduction when an $\mathcal{U}$ symbol is introduced via an $\mathcal{U}$-introduction rule, and when an $\mathcal{U}$ symbol is eliminated via an $\mathcal{U}$-elimination rule.

We call $\mathcal{U}$-term a term having as top symbol an $\mathcal{U}$ symbol. An $\mathcal{U}$-term is said *resolvable* (w.r.t. a CTRS R) if there is a reduction in $\mathbb{U}_n(R)$ that eliminates the descendant of its top $\mathcal{U}$ symbol (equivalently, there is a reduction in $\mathbb{U}_n(R)$ such that the last step is an $\mathcal{U}$-elimination that gets rid of the top $\mathcal{U}$ symbol). The CTRS R will me mostly considered understood.

Given an $\mathcal{U}$-term $\mathcal{U}_\rho(u_1,\ldots,u_k,v_1,\ldots,v_m)$ (with $\rho : l \to r \Leftarrow s_1 \to_!^* n_1,\ldots,s_k \to_!^* n_k$), we call *test terms* the first k terms $u_1,\ldots,u_k$, and *non-test terms* the other $(v_1,\ldots,v_m)$.

It is trivial to see that an $\mathcal{U}$-term is resolvable iff there is a reduction that eliminates the descendant of its top $\mathcal{U}$ symbol *and* it doesn't apply any rule to the non-test terms. So, when we talk about *resolving* an $\mathcal{U}$-term we will mean applying any such reduction.

We now define the useful notion of balanced reduction in the unraveled CTRS, that in some sense is the direct equivalent of a reduction in a CTRS.

Definition A.2 Given a CTRS R, a reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t$ is *balanced* if:

1. When an introduction is performed on a subterm, there can be other $\mathcal{U}$-terms only above it.
2. No rewrite step is applied to non-test subterms.
3. Every descendant of an $\mathcal{U}$-term is eliminated.

□

Balanced reductions have a nice compositional behaviour:

Lemma A.3 *If $s \in$ TERMS, and two reductions $s \to^* s'$ and $s' \to^* t$ are balanced, then their composed reduction $s \to^* s' \to^* t$ is balanced.*

Proof Routine. □

The following result shows that balanced reduction are in direct correspondence with reductions in the original CTRS:

Lemma A.4 *Suppose R is a left-linear normal CTRS. If $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ is a balanced reduction, and $s \in$ TERMS, then $s \xrightarrow[R]{}{}^* t$.*

Proof The proof is by double induction: primary induction on the $\mathcal{U}$-*depth* of the reduction, and secondary induction on the length of the reduction.

The base case ($s \xrightarrow[\mathbb{U}_n(R)]{}{}^* s$) via the empty reduction) is trivial.

So, consider a balanced reduction $s \xrightarrow[\mathbb{U}_n(R)]{} s' \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$.

If the rule applied in $s \xrightarrow[\mathbb{U}_n(R)]{} s'$ is not an introduction, then the same rule is present in R (since $\mathbb{U}_n(T) = T$ for every TRS T), and so $s \xrightarrow[R]{} s'$. The reduction $s' \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ has same $\mathcal{U}$-*depth* than the original reduction, and minor length. So, since $s' \in$ TERMS, by induction we have $s' \xrightarrow[R]{}{}^* t$ and consequently $s \xrightarrow[R]{}{}^* t$.

The other case is when the rule applied in $s \xrightarrow[\mathbb{U}_n(R)]{} s'$ is an introduction. Suppose the introduction rule is $l \to \mathcal{U}_\rho(s_1, \ldots, s_k, \mathrm{VAR}(r))$ (obtained by the rule $\rho : l \to r \Leftarrow s_1 \to_!^* n_1, \ldots, s_k \to_!^* n_k$ of R), and it is applied with matching substitution σ. Since the reduction is balanced, every descendant of $\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(r)\sigma)$ is eliminated (Point 3 of Def. A.2). Hence, extract from a given descendant in $s' \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ the subreductions $s_1\sigma \xrightarrow[\mathbb{U}_n(R)]{}{}^* n_1, \ldots, s_k\sigma \xrightarrow[\mathbb{U}_n(R)]{}{}^* n_k$. Now, $s_1\sigma, \ldots, s_k\sigma \in$ TERMS (since the reduction is balanced), and all the above subreductions have $\mathcal{U}$-*depth* strictly less than the original reduction $s \xrightarrow[\mathbb{U}_n(R)t]{}{}^*$: hence, by induction hypothesis $s_1\sigma \xrightarrow[R]{}{}^* n_1, \ldots, s_k\sigma \xrightarrow[R]{}{}^* n_k$. Also, by Point 2, when a descendant is eliminated the resulting subterm is just $\mathrm{VAR}(r)\sigma$.

Consider the smallest reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* s''$ of $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ having no descendants of the $\mathcal{U}$-term $\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(r)\sigma)$ in s''. By Point 1, all the reductions steps that are not applied to the test terms of the descendants of $\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(r)\sigma)$ cannot introduce other $\mathcal{U}$-symbols, and therefore they are unconditional rules (belonging to R).

So, in the corresponding reduction of R we skip every rule applied to (the descendants of) $\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(r)\sigma)$, until in the original reduction an elimination rule was applied. In this case, we apply the conditional rule ρ (it can be applied since $s_1\sigma \xrightarrow[R]{}{}^* n_1, \ldots, s_k\sigma \xrightarrow[R]{}{}^* n_k$), obtaining just $r\sigma$, as in the original reduction. Thus, the reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* s''$ can be mimicked in R obtaining $s \xrightarrow[R]{}{}^* s''$.

Now consider the remaining part of the original reduction, $s'' \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$: it obviously has not greater $\mathcal{U}$-*depth* than the original reduction, and minor length. So, since $s'' \in$ TERMS, by induction we have $s'' \xrightarrow[R]{}{}^* t$ and consequently $s \xrightarrow[R]{}{}^* t$. □

Next, we prove an extremely powerful lemma, which will be needed in the sequel to obtain important results on $\mathbb{U}_n$:

Lemma A.5 *For every left-linear normal CTRS R, there are two operators $\blacktriangle_R$ and $\blacktriangledown_R$ from* TERMS$^+$ *to* TERMS *that are the identity on* TERMS *and such that*

$$\forall s \in \text{TERMS}, t \in \text{TERMS}^+.\ s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t \Rightarrow s \xrightarrow[R]{}{}^* \blacktriangledown_R(t)$$
$$\forall s \in \text{TERMS}^+, t \in \text{TERMS}.\ s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t \Rightarrow \blacktriangle_R(s) \xrightarrow[R]{}{}^* t$$

Proof We will first prove the existence of $\blacktriangledown_R$.

So, suppose to have a reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ with $s \in$ TERMS, $t \in$ TERMS$^+$.

It is not restrictive to assume that in t there are no resolvable $\mathcal{U}$-(sub)terms.

Indeed, if there is an $\mathcal{U}$-(sub)term in t which is resolvable, resolve it. This way, we obtain a new reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t \xrightarrow[\mathbb{U}_n(R)]{}{}^* t'$ Now, the number of $\mathcal{U}$-terms in t' that are resolvable is the same as in t minus one (readily, no other $\mathcal{U}$-terms are introduced, and we have resolved one). Hence, repeating this process on the reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t'$ leads, ultimately, to a reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* s'$ such that s' has no resolvable $\mathcal{U}$-terms.

We will show that from every reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t$ (where t has no resolvable $\mathcal{U}$-terms), we can obtain a balanced reduction $s \xrightarrow[\mathbb{U}_n(R)]{}{}^* t'$, where $t = t'$ if $t \in$ TERMS.

The proof is by induction on the $\mathcal{U}$-depth of the derivation.

The base case ($\mathcal{U}$-depth=0) is trivial (since $\mathbb{U}_n(T) = T$ for every TRS T by definition).

So it remains the induction step.

Consider the first $\mathcal{U}$ symbol introduced in the reduction. There are two possible cases:

Case 1: All its descendants are eliminated.

Suppose that the rule introducing the $\mathcal{U}$ symbol is $\rho : l \to r \Leftarrow s_1 \to_!^* n_1, \ldots, s_k \to_!^* n_k$, with matching substitution σ. Since at least one descendant of $\mathcal{U}_\rho$ is eliminated, for that descendants there are subreductions $s_1\sigma \to^* n_1, \ldots, s_k\sigma \to^* n_k$. Since by assumption $s_1\sigma, \ldots, s_k\sigma \in$ TERMS, by depth induction there are corresponding balanced reductions $s_1\sigma \underset{R}{\to}^* n'_1 = n_1, \ldots, s_k\sigma \underset{R}{\to}^* n'_k = n_k$.

So, we apply these reductions soon afterwards the $\mathcal{U}_\rho$-introduction, obtaining the subterm $\mathcal{U}_\rho(n_1), \ldots, n_k, \mathrm{VAR}(l)\sigma)$. Then, we apply the corresponding elimination rule, obtaining $r\sigma$.

This way, we have anticipated the elimination of $\mathcal{U}_\rho$, performing the reduction in advance. The neat effect is that in place of $\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(l)\sigma)$ we have $r\sigma$.

So, every rewrite step that in the original reduction acted on (a descendant of) the $\mathcal{U}$ term has to be replaced with a corresponding mimicked rewrite step on $r\sigma$.

Suppose that $|\mathrm{VAR}(r)| = |X_1, \ldots, X_m| = m$.

Consider an operator π_ρ defined this way:

$$\pi_\rho(\mathcal{U}_\rho(u_1, \ldots, u_k, , v_1, \ldots, v_m)) = r\{X_1/v_1, \ldots, X_m/v_m\} \tag{1}$$

During this mimicking, we will always have that in the original reduction a descendant of $\mathcal{U}_\rho$ is t, then in the mimicked reduction the corresponding subterm is $\pi_\rho(t)$.

For the base case (i.e. when no reduction steps have been applied to a descendant of $\mathcal{U}_\rho$), the equation (1) is true, since we have just $\pi_\rho(\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(l)\sigma)) = r\sigma$.

If in the original reduction a rewrite step was applied to (a descendant of) one of the test terms $s_i\sigma, t_i\sigma$, we simply ignore it (the intuition is that we have performed in advance the reduction on these terms). This maintains (1) true, since π_ρ does not depend on the test terms of $\mathcal{U}_\rho$.

On the other hand, if a rewrite step occurred in the i-th non-test term (i.e. in the $k+i$-th term of $\mathcal{U}_\rho$): if the original subterm was $\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_1, \ldots, v_m)$ and was rewritten into $\mathcal{U}_\rho(u_1, \ldots, u_k, , v_1, \ldots, v'_i, \ldots, v_m)$ we act with the rewrite step

$$r\{Y_1/v_1, \ldots, Y/v_i, \ldots, Y_m/v_m\} \to r\{Y_1/v_1, \ldots, Y/v'_i, \ldots, Y_m/v_m\}$$

that readily maintains (1) true.

This way we have obtained a reduction composed by a balanced reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}^* s_1$ (from the beginning till the elimination of the $\mathcal{U}$ symbol) and a subsequent reduction $s_1 \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t_1$. The number of introductions performed in the latter is the number of introductions performed in the original reduction is at most $s \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t$ minus one, since readily our modifications do not introduce new introductions. So, since $s_1 \in$ TERMS, by induction we can transform the reduction $s_1 \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t$ into a balanced reduction $s_1 \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t'$. Therefore, applying Lemma A.3, we obtain that the composed reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}^* s_1 \underset{\mathbb{U}_n(R)}{\longrightarrow}^* t'$ is balanced.

Case 2: There are no descendants that are eliminated.

This time, we do not perform the $\mathcal{U}$ introduction.

Again, we build up a mimicking reduction for each descendant until it disappears (via an erasing rule). So, every rewrite step that in the original reduction acted on (a descendant of) the $\mathcal{U}$ term has to be replaced with a corresponding mimicked rewrite step on $r\sigma$.

Suppose that $|\mathrm{VAR}(r)| = |X_1, \ldots, X_m| = m$.

Consider an operator π'_ρ defined this way:

$$\pi'_\rho(\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_m)) = l\{X_1/v_1, \ldots, X_m/v_m\} \tag{2}$$

During this mimicking, we will always have that in the original reduction a descendant of $\mathcal{U}_\rho$ is t, then in the mimicked reduction the corresponding subterm is $\pi'_\rho(t)$.

For the base case (i.e. when no reduction steps have been applied to a descendant of $\mathcal{U}_\rho$), the equation (2) is true, since we have just $\pi'_\rho(\mathcal{U}_\rho(s_1\sigma, \ldots, s_k\sigma, \mathrm{VAR}(r)\sigma)) = l$.

Suppose the descendant is of the form $\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_m)$.

If a reduction step is applied to the 'test terms' $u_1, \ldots, u_k$ of the descendant, we just ignore it. Readily, (2) still holds since it does not depend on the values of these terms.

On the other hand, if a reduction step is applied to v_i, passing from $\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_i, \ldots, v_m)$ to $\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_i', \ldots, v_m)$, then we perform the same reduction step on each corresponding v_i in $l\{X_1/v_1, \ldots, X_i/v_i, \ldots, X_m/v_m\}$, that is to say

$$l\{X_1/v_1, \ldots, X_i/v_i, \ldots, X_m/v_m\} \rightarrow^* l\{X_1/v_1, \ldots, X_i/v_i', \ldots, X_m/v_m\}$$

But $l\{X_1/v_1, \ldots, X_i/v_i', \ldots, X_m/v_m\} = \pi_\rho'(\mathcal{U}_\rho(u_1, \ldots, u_k, v_1, \ldots, v_i', \ldots, v_m))$, and so (2) is satisfied.

This way we have obtained a reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t_1$: the number of introductions performed in this reduction is at most $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t$ minus one, since readily our modifications do not introduce new introductions. So, by induction we can transform the reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t_1$ into a balanced reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t'$.

We have seen that from $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t$ (where t has no resolvable $\mathcal{U}$-terms), we can obtain a balanced reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t'$. What is the relationship between t and t'? We have modified the original reduction iteratively, via the modifications discussed in Case 1 and 2. When we transform a reduction using the modification of Case 1, before invoking the induction step we have left the last term of the reduction unmodified, as it is trivial to see. On the other hand, when we transform a reduction using the modification of Case 1, before invoking the induction step we have substituted in the original last term of the reduction every descendant d of the selected $\mathcal{U}_\rho$-term with $\pi_\rho'(d)$.

So, the final effect is that we have replaced every $\mathcal{U}_\rho$-(sub)term in d with the corresponding subterm $\pi_\rho'(d)$.

Hence, if we have an arbitrary reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t$, with $s \in$ Terms and $t \in$ Terms$^+$, we can obtain a balanced reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* t'$ where t' is obtained from t this way:

1. First, we resolve every resolvable $\mathcal{U}$-term in t.
2. Next, we replace every $\mathcal{U}_\rho$-term d with $\pi_\rho'(d)$

Note that this mapping is independent on the particular reduction, depending only on the CTRS R: so, we define $\blacktriangledown_R$ to be just this mapping. Note that readily $\blacktriangledown_R$ is the identity on Terms.

Now, since we have the balanced reduction $s \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* \blacktriangledown_R(t)$, by Lemma A.4 we obtain that $s \underset{R}{\rightarrow}{}^* \blacktriangledown_R(t)$.

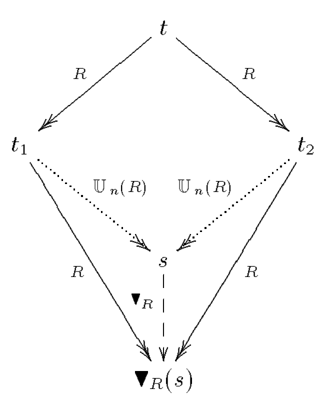
The proof of the existence of $\blacktriangle_R$ is completely analogous. The difference is that now, dually, we have to get rid of the $\mathcal{U}$-terms in the first term of the reduction. So, if such a term $\mathcal{U}_\rho(\cdots)$ is resolvable, we replace it with its resolvent (i.e. with $\pi_\rho(\mathcal{U}_\rho(\cdots))$, and if it is not, we replace it by $\pi_\rho'(\mathcal{U}_\rho(\cdots))$. ☐

Now that we have developed this machinery, we can easily prove **Theorem 6.7**: The "$\Rightarrow$" direction is given by (the proof of) Theorem 6.4. The "$\Leftarrow$" direction follows from Lemma A.5, since $\blacktriangledown_R$ (or, equivalently, $\blacktriangle_R$) are the identity on Terms.

. . . .

[Proof of Theorem 6.8]

Let R be a left-linear normal CTRS and $\mathbb{U}_n(R)$ be confluent. Suppose a term $t \in$ Terms rewrites in R to t_1 and t_2. By (the proof of) Theorem 6.2, the same reductions hold in $\mathbb{U}_n(R)$ too. So, by confluence of $\mathbb{U}_n(R)$ we have $t_1 \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* s$, $t_2 \underset{\mathbb{U}_n(R)}{\longrightarrow}{}^* s$ (with $s \in$ Terms$^+$). Then, applying Lemma A.5 we obtain $t_1 \underset{R}{\rightarrow}{}^* \blacktriangledown_R(s)$, $t_2 \underset{R}{\rightarrow}{}^* \blacktriangledown_R(s)$, that is to say $t_1 \downarrow_R t_2$. The situation is illustrated by the following diagram:



. . . .

[Proof of Theorem 6.10]

The soundness follows by Theorem 6.3. So, we have to prove preservation.

Let R be a left-linear normal CTRS, and suppose $\mathbb{U}_n(R)$ is not CON$^{\rightarrow}$. This means there is a term $t \in$ TERMS^{+} such that $t \xrightarrow[\mathbb{U}_n(R)]{}^* X$ and $t \xrightarrow[\mathbb{U}_n(R)]{}^* Y$. So, by Lemma A.5 we have that $\blacktriangle_R(t) \xrightarrow[R]{}^* X$ and $\blacktriangle_R(t) \xrightarrow[R]{}^* Y$, that is to say R is not CON$^{\rightarrow}$.

. . . .

[Proof of Theorem 9.2]

Ab absurdo, suppose there is an unraveling $\mathbf{U}$ which is complete for termination.

Take the rule $\rho : a \to b \Leftarrow a{\downarrow}c$.

Since in $a \to c \cup \rho$ we have $a{\downarrow}b$, then in $\mathbf{U}(a \to c \cup \rho) = a \to c \cup \mathbf{U}(\rho)$ we have $a{\downarrow}_{a\to c\cup\mathbf{U}(\rho)}b$ as well. So, we have two reductions $a \xrightarrow[a\to c\cup\mathbf{U}(\rho)]{}^* n$ and $b \xrightarrow[a\to c\cup\mathbf{U}(\rho)]{}^* n$.

Suppose $b \xrightarrow[a\to c\cup\mathbf{U}(\rho)]{}^* n$ uses the rule $a \to c$. This means $b \xrightarrow[\mathbf{U}(\rho)]{}^* C[a]$ (for some context $C[\,]$). Hence, $a \to b \cup \rho$ is terminating $\Rightarrow$ $a \to b \cup \mathbf{U}(\rho)$ is terminating, an absurdum since in $a \to b \cup \mathbf{U}(\rho)$ there is the derivation $b \to^* C[a] \to C[b]$.

So, $a \xrightarrow[a\to c\cup\mathbf{U}(\rho)]{}^* n$ and $b \xrightarrow[\mathbf{U}(\rho)]{}^* n$.

Suppose that $a = n$: $b \xrightarrow[\mathbf{U}(\rho)]{}^* d$.

Since $a \to b \cup \mathbf{U}(\rho)$ is terminating, this is a contradiction since in this TRS we have $a \to b \to^* a$.

So, $a \neq n$, and thus $a \xrightarrow[a\to c\cup\mathbf{U}(\rho)]{}^+ n$.

Being $a \to c \cup \mathbf{U}(\rho)$ terminating, in this TRS no reduction starting from a can be of the form $a \to^+ C[a]$. Thus, two possibilities remain:

1. $a \to c \xrightarrow[\mathbf{U}(\rho)]{}^* n$
2. $a \xrightarrow[\mathbf{U}(\rho)]{}^+ n$

Case 1: Then $c{\downarrow}_{\mathbf{U}(\rho)}b$. So, $f(X,X) \to f(c,b) \cup \rho$ is terminating implies that $f(X,X) \to f(c,b) \cup \mathbf{U}(\rho)$ is terminating as well, whilst in this TRS there is the reduction $f(c,b) \to^* f(n,n) \to f(c,b)$, a contradiction.

Case 2: Then $a{\downarrow}_{\mathbf{U}(\rho)}b$. So, $f(X,X) \to f(a,b) \cup \rho$ is terminating implies that $f(X,X) \to f(a,b) \cup \mathbf{U}(\rho)$ is terminating as well, while in this TRS there is the reduction $f(a,b) \to^* f(n,n) \to f(a,b)$, a contradiction.