Transformations of Conditional Rewrite Systems Revisited

Karl Gmeiner and Bernhard Gramlich

TU Wien, Austria
{gmeiner,gramlich}@logic.at

Abstract. We revisit known transformations of conditional rewrite systems to unconditional ones in a systematic way. We present a unified framework for describing, analyzing and classifying such transformations, discuss the major problems arising, and finally present a new transformation which has some advantages as compared to the approach of [6]. The key feature of our new approach (for left-linear confluent normal 1-CTRSs) is that it is backtracking-free due to an appropriate encoding of the conditions.

1 Background and Overview

Conditional term rewrite systems (CTRSs) and conditional equational specifications are very important in algebraic specification, prototyping, implementation and programming. They naturally occur in most practical applications. Yet, compared to unconditional term rewrite systems (TRSs), CTRSs are much more complicated, both in theory (especially concerning criteria and proof techniques for major properties of such systems like confluence and termination) and practice (implementing conditional rewriting in a clever way is far from being obvious, due to the inherent recursion when evaluating conditions). For these (theoretical and practical) reasons, transforming CTRSs into (unconditional) TRSs in an adequate way has been studied for a long time cf. e.g. [4, 9, 18, 12, 15, 5, 2, 6, 13, 17, 10]. In many other early papers (like [1, 8]) the issue of transforming conditional into unconditional TRSs is not studied in depth, but at least touched from a programming language point of view.

Roughly, all transformations work by translating the original syntax (signature and terms) into an extended or modified one using auxiliary function symbols, and by translating the rules in a corresponding way such that the evaluation of conditions and some control structure is (appropriately) encoded within the resulting unconditional TRS (in which in some cases reduction is additionally restricted, see below).

In the papers mentioned above certain of these issues have been investigated for particular (quite different) transformations and with different terminology. In order to better understand and relate the different approaches together with their results, we will propose a kind of unified terminology for such transformations and their properties.

In the second main part of the paper we will deal with the issue of backtracking and the question whether a transformed system is computationally adequate for simulating the original one. Here we will propose a new approach whose characteristic feature is “backtracking-freeness”. The underlying goal here is as follows: If, for some given conditional system, we start a simulation (a reduction in the transformed TRS) from an “initial” term and obtain a normal form in the transformed system, then the latter should correspond to a normal form of the initial term in the original CTRS (this property, together with a few other requirements, is called *computational equivalence* in [6]). Otherwise, some form of backtracking would be needed, because then we are stuck with a failed attempt of verifying conditions, and may need to try another conditional rule.

The rest of the paper is structured as follows. In Section 2 we introduce the necessary background about (conditional) term rewriting. Then, in Section 3 we present and discuss a unifying framework for describing transformations from CTRSs to TRSs. Furthermore known and new unsoundness phenomena are dealt with briefly. Then, in Section 4 we investigate how to design a transformation that avoids explicit backtracking during simulation in such a way that the transformed system still enjoys most desired preservation properties. We motivate the approach by a careful analysis, give a formal definition and present the main results. Finally, in Section 5 we report on some first experiments with our new transformation, briefly discuss related work, sketch possible optimizations, refinements and alternatives, and mention a few interesting perspectives. Due to lack of space, proofs are omitted in the paper.[1]

2 Preliminaries

We assume familiarity with the basic notations and terminology in rewriting, cf. e.g. [3]. We denote by $\mathcal{O}(t)$ the set of all subterm positions of a term t, that is partitioned into all variable positions $\mathcal{O}_{\mathcal{X}}(t) = \{p \in \mathcal{O}(t) \mid t|_p \text{ is a variable}\}$ and all non-variable positions $\overline{\mathcal{O}}(t) = \mathcal{O}(t) \setminus \mathcal{O}_{\mathcal{X}}(t)$. By $\mathcal{V}\text{ars}(t)$ we denote the set of all variables occurring in a term t. This notion is extended in the obvious way to rules and conditions. The set of normal forms of a rewrite system $\mathcal{R}$ is denoted by $\text{NF}(\mathcal{R})$. Left- and right-hand sides of rewrite rules are also abbreviated as lhs and rhs, respectively. Slightly abusing notation, we sometimes confuse a rewrite system $\mathcal{R} = (\mathcal{F}, R)$ and its set R of rewrite rules.

Definition 1 (Conditional rewrite system, conditional rewrite relation, depth of reductions). *A conditional term rewriting system (CTRS) $\mathcal{R}$ (over some signature $\mathcal{F}$) consists of rules $l \to r \Leftarrow c$ where c is a conjunction of equations $s_i = t_i$. Equality in the conditions may be interpreted (recursively) e.g. as $\leftrightarrow^*$ (*semi-equational *case), as $\downarrow$ (*join *case), or as $\to^*$ (*oriented *case). In the latter case, if all right-hand sides of conditions are ground terms that are irreducible*

[1] More theoretical results, complete proofs and more details about experiments, related work and possible optimizations and refinements can be found in the full version of this paper (forthcoming).

w.r.t. the unconditional version $\mathcal{R}_u = \{l \to r \mid l \to r \Leftarrow c \in \mathcal{R}\}$ *of* $\mathcal{R}$*, the system is said to be a* normal *one. Furthermore, according to the distribution of variables, a conditional rule* $l \to r \Leftarrow c$ *may satisfy* (1) $\mathcal{V}ars(r) \cup \mathcal{V}ars(c) \subseteq \mathcal{V}ars(l)$, (2) $\mathcal{V}ars(r) \subseteq \mathcal{V}ars(l)$, (3) $\mathcal{V}ars(r) \subseteq \mathcal{V}ars(l) \cup \mathcal{V}ars(c)$, *or* (4) *no variable constraints. If all rules of a CTRS* $\mathcal{R}$ *are of type (i),* $1 \leq i \leq 4$*, respectively, we say that* $\mathcal{R}$ *is an i-CTRS. The rewrite relation of an oriented CTRS* $\mathcal{R}$ *is recursively defined as follows:* $R_0 \stackrel{\mathrm{def}}{=} \emptyset$, $R_{n+1} \stackrel{\mathrm{def}}{=} \{l\sigma \to r\sigma \mid l \to r \Leftarrow s_1 \to^* t_1, \ldots, s_k \to^* t_k \in \mathcal{R}, s_i\sigma \to^*_{R_i} t_i\sigma$ *for all* $1 \leq i \leq k\}$, $\to_R \stackrel{\mathrm{def}}{=} \bigcup_{n \geq 0} R_n$.

In the rest of the paper we will mainly deal with **normal 1-CTRSs**.

3 A Unifying Approach to Transformations

3.1 Basic Transformation Approaches

Basically, two different lines of approaches can be distinguished, according to the way in which the conditions and the intermediate condition evaluation process are encoded. Consider a conditional rule of a given normal 1-CTRS and a term $s = s[l\sigma]$ to be reduced. Obviously, the actual reduction of $s = s[l\sigma]$ into $s' = s[r\sigma]$ has to be delayed until the conditions $s_i\sigma \to^* t_i\sigma = t_i$ have been verified. To this end, the condition evaluation needs to be initiated and performed, while keeping the relevant context, i.e., about the current rule, in order to be finally able to produce $r\sigma$. In one line of approaches (historically the earlier one), the latter information is encoded in an abstract way that hides any concrete structure of l, but keeps the variable bindings of the matching substitution σ. Using the latter, after successful verification of the conditions the corresponding instantiated right-hand side $r\sigma$ can be produced. This means, we need two rules, an *introduction* or *initialization* rule $\rho' : l \to U_\rho(s_1, \ldots, s_n, \mathcal{V}\mathrm{ars}(l))$ where $\mathcal{V}\mathrm{ars}(s)$ denotes the sequence of the set of all variables occurring in s (in an arbitrary, but fixed order) and the fresh function symbol U_ρ (of appropriate arity) stands for rule ρ, and an *elimination* (or *reducing*) rule $\rho'' : U_\rho(t_1, \ldots, t_n, \mathcal{V}\mathrm{ars}(l)) \to r$ that completes the successful rule application after the instantiated conditions $s_i\sigma \to^* t_i\sigma = t_i$ have been verified (by other rewrite steps in between). The most prominent representative of this type of approach are Marchiori's *unravelings* [12]. Early forerunners (with some restrictions/modifications or special cases) and successors of *unraveling* approaches in the literature are among others [4, 8, 15].

In the other main line of approaches, when trying to apply a conditional rule, the left-hand side is not completely abstracted away during verification of the conditions, but instead is kept in a modified form such that the conditions become additional arguments of some function symbol(s) in l, typically of the root function symbol. That means, the arity of this function symbol is increased appropriately by *conditional (argument) positions* which are used to represent the *conditional arguments*. Suppose $l = f(u_1, \ldots, u_k)$. Then f is modified into f' by increasing its arity to $k+n$. For example, for the rule $\rho' : f(x) \to x \Leftarrow x \to^* 0$ the *introduction* and *elimination* rules become $f'(x,\bot) \to f'(x,x)$ and $f'(x,0) \to x$, respectively. Here, the fresh constant $\bot$ stands for an uninitialized condition. In order to prevent trivial cases of non-preservation of termination[2] we will wrap conditional arguments in some fresh syntactic structure, e.g. as follows: $f'(x,\bot) \to f'(x,\langle x\rangle)$, $f'(x,\langle 0\rangle) \to x$.[3] Now, increasing the arity of some function symbols in general for storing conditional arguments there requires a more sophisticated construction of the transformed system, since for every occurrence of such function symbols in left- and right hand sides as well in the conditions one has to specify how these conditional arguments should be filled and dealt with during rewriting. And the basic transformation step has to be done for every conditional rule! The basic idea underlying this approach goes back at least till [1]. Yet, the work that inspired many later approaches in this direction is by Viry [18].[4] Other more recent transformation approaches along this line of reasoning include [2, 6, 16].

Intuitively, in both lines of approaches certain reductions in the transformed system during the evaluation of conditions do not correspond to what is done in the conditional system, e.g., reduction in the variable bindings $\mathcal{V}\text{ars}(l)$ of $U_\rho(s_1,\ldots,s_n,\mathcal{V}\text{ars}(l))$ for unravelings, reduction in the "original arguments" of $f'(x,\langle x\rangle)$, i.e., outside of $\langle x\rangle$, in the conditional argument approach, and reduction above "non-completed" conditional arguments (in both approaches). This phenomenon which may have (and indeed has) problematic consequences for transformations is well-known for a long time. For that reason several approaches in the literature impose some form of (context-sensitivity or strategy or order-sortedness) restrictions on rewriting in the transformed system, e.g. [10, 14, 16, 17, 18], which may lead to better results in theory and/or practice. We will keep this issue in mind, but not deepen it here due to lack of space.

What makes papers and results about transforming conditional systems sometimes hard to read and to compare, is the diversity of the terminology used to reason about their properties. In particular, *soundness* and *completeness* notions are usually defined in different ways. We will instead provide now a proposal for a unified description of such transformations including the relevant terminology.

3.2 A Unified Parameterized Description of Transformations

In view of the existing transformations and since one wants to simulate conditional rewriting in the original system by unconditional rewriting in the transformed one, extending the syntax appears to be unavoidable. So, generally instead of *original terms* from $\mathcal{T} \overset{\text{def}}{=} \mathcal{T}(\mathcal{F},\mathcal{V})$ the simulation will happen with

[2] Note that the introduction rule is obviously non-terminating, whereas the original conditional rule terminates (and is even decreasing cf. [7]).

[3] Strictly speaking, the symbols $\bot$ and $\langle\ldots\rangle$ here are variadic, since they have as many arguments as there are conditions in the respective rule. In a fixed-arity setting one would have to use k-adic symbols $\bot_k$ and $\langle\ldots\rangle_k$ instead, for appropriate arities k.

[4] Even though several main results (and proofs) in [18] are flawed, the ideas and the concrete approach developed there have been very influential.

terms from $\mathcal{T}' \overset{\text{def}}{=} \mathcal{T}(\mathcal{F}', \mathcal{V})$ over an extended or modified signature $\mathcal{F}'$. Moreover, it may be necessary to initially explicitly translate original terms into the new syntax and associate results obtained in the transformed system to original terms. For unravelings this would not be absolutely necessary, but still yields a generalized point of view that turns out to be beneficial for the analysis. For approaches with encoding conditional arguments at new conditional argument positions these mappings are essential, though.

Let us start with the general form of a transformation.[5]

Definition 2 (Transformations of CTRSs). *A* transformation *from a class of CTRSs into a class of TRSs is a total mapping T that associates to every conditional system $\mathcal{R} = (\mathcal{F}, R)$ from the class a triple $((\mathcal{F}', R', \to_{\mathcal{R}'}), \phi, \psi)$, where $(\mathcal{F}', R')$ is a TRS and $\to_{\mathcal{R}'}$ is a subset of the rewrite relation induced by $\mathcal{R}' = (\mathcal{F}', R')$.[6] We call $\phi\colon \mathcal{T} \to \mathcal{T}'$ the* initialization mapping *(or* encoding*) and $\psi\colon \mathcal{T}' \to \mathcal{T}$ the* backtranslation *(or* decoding*). Furthermore T has to satisfy the following requirements:*

(1) If $\mathcal{R} = (\mathcal{F}, R)$ is finite (i.e., both $\mathcal{F}$ and R are finite), then $\mathcal{R}' = (\mathcal{F}', R')$ is also finite.
(2) The restriction of T to finite systems (from the considered class of CTRSs) is effectively constructible.
(3) The initialization mapping $\phi\colon \mathcal{T} \to \mathcal{T}'$ is an injective total function.
(4) The backtranslation $\psi\colon \mathcal{T}' \to \mathcal{T}$ is a (partial) function that is defined at least on $\mathcal{T}'_r$, the set of all reachable terms[7] *which is given by $\mathcal{T}'_r \overset{\text{def}}{=} \{t' \in \mathcal{T}' \mid \phi(s) \to^*_{\mathcal{R}'} t' \text{ for some } s \in \mathcal{T}\}$.*
(5) The backtranslation $\psi\colon \mathcal{T}' \to \mathcal{T}$ satisfies $\psi(\phi(s)) = s$ for all $s \in \mathcal{T}$, i.e., on all initialized original terms it acts as inverse function w.r.t ϕ.

Discussion of requirements: Let us briefly discuss this abstract definition of transformation and in particular the requirements mentioned.

First, we parameterize transformations by the class of CTRSs that we want to transform, because this reflects the fact that for different types of CTRSs transformations are typically defined in different ways.

The transformation of $\mathcal{R}$ into $((\mathcal{F}', R', \to_{\mathcal{R}'}), \phi, \psi)$ allows to impose particular restrictions on $\to_{\mathcal{R}'}$ like innermost rewriting or context-sensitivity constraints. This ability is crucial in some existing transformation approaches. Intuitively, this happens there in order to simulate more accurately the evaluation of conditions in the transformed setting and to exclude computations that have no analogue in the original system. If such a restriction of ordinary reduction is involved in $\to_{\mathcal{R}'}$, we will mention this explicitly. Otherwise, again abusing notation, we will simply omit the third component in $(\mathcal{F}', R', \to_{\mathcal{R}'})$.

[5] Alternatively, instead of *transformation* also *encoding*, *embedding* or *simulation* are used in the literature.

[6] Actually, this is an abuse of notation. By writing $\to_{\mathcal{R}'}$ we simply want to cover the case, too, where the rewrite relation induced by $\mathcal{R}'$ is somehow restricted.

[7] The notion *reachable term* stems from [6]. In [2], *term of interest* was used instead.

Next, for a given CTRS of the respective class, the transformation does not only yield a transformed signature and rewrite system, but also an *initialization mapping* ϕ and a *backtranslation* ψ. For practical reasons, here the need for requirement (2) is obvious.

Requirement (1) actually is a precondition for (2): When starting with a finite system, we clearly don't want to end up with an infinite one after transformation which usually would lead to non-computability and undecidability problems. Without (1), the ideal and natural candidate for the transformed unconditional system would be $\mathcal{R}' = (\mathcal{F}, R')$ where $R' = \bigcup_{i \geq 0} R_i$ with R_i as in Definition 1 (over the same signature), and with ϕ and ψ the identity function. However, then in general $\mathcal{R}'$ would be infinite and its rewrite relation undecidable! Actually, from a logical point of view this choice of $\mathcal{R}'$ would be ideal, since virtually all typical problems with transformations (like unsoundness phenomena) would simply disappear, because the rewrite relations (of the original and the transformed system) are exactly the same in this case. Yet, for the mentioned reasons, this kind of transformation is practically useless.

Since in general the transformation may work on a modified or extended syntax, the original terms possibly need to be translated (via ϕ) before computation starts. Hence, ϕ should be total and also injective (3), since one clearly wants to be able to distinguish different original terms also after transformation. The other way round is more subtle. Namely, what should one be able to infer from derivations in the transformed system, in terms of the original one? First of all, the backtranslation ψ need not necessarily be total, because there may be terms in $\mathcal{T}'$ which do not correspond to intermediate results of computations in the transformed system. For such *garbage terms* nothing should be required. In particular, there need not exist corresponding original terms for them. However, for every reachable term in the transformed system we do require that there exists indeed some original term to which the former corresponds (4). In a sense, this condition is quite strong.[8] Yet, on an abstract level in general we do not know how the treatment of conditions (of $\mathcal{R}$) in $\mathcal{R}'$ actually works. The intuition behind (4) is that ψ should deliver those "original parts" of an intermediate result $t' \in \mathcal{T}'$ which can obviously be obtained by translating back (i.e., which directly correspond to original syntax). For the other "conditional parts" which correspond to started attempts of applying conditional rules, ψ should go back to the "beginning" of this attempt and recursively only translate back (conditional) rule applications that have been entirely completed. Since the former aspect requires solving reachability problems which are undecidable in general, due to (2) reasonable computable versions of ψ can only approximate "precise backtranslations". Injectivity of ψ would be too strong a requirement, because typically there exist many terms in $\mathcal{T}'$ corresponding in a natural way to a single original term. For initialized original terms $\phi(s), s \in \mathcal{T}$, it should be obvious and

[8] In this sense, (4) is perhaps the most debatable requirement in our abstract definition of transformation. Though this definition covers all the major approaches from the literature, it is conceivable that there exist other transformational approaches for which a step-by-step backtranslation need not make sense.

intuitive that via ψ we should get back the original term s (5), simply because the initialization ϕ is the standard or canonical way of translating original terms into the transformed setting.

Based on Definition 2 we will now define various important properties of transformations that are crucial not only in theory, but also in practical applications. Note that the resulting terminology differs from the existing literature.

Definition 3 (Properties of transformations). *Let T be a transformation from a class of CTRSs into the class of TRSs according to Definition 2 and let $\mathcal{R} = (\mathcal{F}, R)$ range over the former class of CTRSs. We define (two versions of) soundness and completeness properties relating (properties $\mathcal{P}$ of) the original and the transformed system (the rectangular brackets indicate the second versions; $\mathcal{P}$ may be e.g. confluence or termination):*

(a) *T is said to be* sound (for reduction) *(or* simulation sound*)* [w.r.t. reachable terms] *if for every $\mathcal{R} = (\mathcal{F}, R)$ with $T(\mathcal{R}) = ((\mathcal{F}', R', \rightarrow_{\mathcal{R}'}), \phi, \psi)$ we have: $\forall s, t \in \mathcal{T}\colon \phi(s) \rightarrow^*_{\mathcal{R}'} \phi(t) \Longrightarrow s \rightarrow^*_{\mathcal{R}} t$ [$\forall s', t' \in \mathcal{T}'_r\colon s' \rightarrow^*_{\mathcal{R}'} t' \Longrightarrow \psi(s') \rightarrow^*_{\mathcal{R}} \psi(t')$)].*

(b) *T is said to be* complete (for reduction) *(or* simulation complete*)* [w.r.t. reachable terms] *if for every $\mathcal{R} = (\mathcal{F}, R)$ with $T(\mathcal{R}) = ((\mathcal{F}', R', \rightarrow_{\mathcal{R}'}), \phi, \psi)$ we have: $\forall s, t \in \mathcal{T}\colon s \rightarrow^*_{\mathcal{R}} t \Longrightarrow \phi(s) \rightarrow^*_{\mathcal{R}'} \phi(t)$ [$\forall s' \in \mathcal{T}'_r, t \in \mathcal{T}\colon \psi(s') \rightarrow^*_{\mathcal{R}} t \Longrightarrow \exists t' \in \mathcal{T}'_r\colon s' \rightarrow^*_{\mathcal{R}'} t', \psi(t') = t$].*

(c) *T is said to be* sound for convertibility [w.r.t. reachable terms] *if for every $\mathcal{R} = (\mathcal{F}, R)$ with $T(\mathcal{R}) = ((\mathcal{F}', R', \rightarrow_{\mathcal{R}'}), \phi, \psi)$ we have: $\forall s, t \in \mathcal{T}\colon \phi(s) \leftrightarrow^*_{\mathcal{R}'} \phi(t) \Longrightarrow s \leftrightarrow^*_{\mathcal{R}} t$ [$\forall s', t' \in \mathcal{T}'_r\colon s' \leftrightarrow^*_{\mathcal{R}'} t' \Longrightarrow \psi(s') \leftrightarrow^*_{\mathcal{R}} \psi(t')$].*

(d) *T is said to be* sound for preserving normal forms [w.r.t. reachable terms] *if for every $\mathcal{R} = (\mathcal{F}, R)$ with $T(\mathcal{R}) = ((\mathcal{F}', R', \rightarrow_{\mathcal{R}'}), \phi, \psi)$ we have: $\forall s, t \in \mathcal{T}\colon \phi(s) \rightarrow^*_{\mathcal{R}'} \phi(t) \in NF(\mathcal{R}') \Longrightarrow t \in NF(\mathcal{R})$ [$\forall s' \in \mathcal{T}'_r\colon s' \rightarrow^*_{\mathcal{R}'} t' \in NF(\mathcal{R}') \Longrightarrow \psi(t') \in NF(\mathcal{R})$[9]].*

(e) *T is said to be* sound for $\mathcal{P}$ [w.r.t. reachable terms] *if $\mathcal{P}(\mathcal{R}')$ implies $\mathcal{P}(\mathcal{R})$ [if $\mathcal{P}(\mathcal{R}')$ on reachable terms implies $\mathcal{P}(\mathcal{R})$].*

(f) *T is said to be* complete for $\mathcal{P}$ [w.r.t. reachable terms] *if $\mathcal{P}(\mathcal{R})$ implies $\mathcal{P}(\mathcal{R}')$ [on reachable terms].*

The above preservation properties of T are "localized" for particular $\mathcal{R}$, $\mathcal{R}'$ and T in the obvious way.[10]

Discussion of terminology: Let us briefly discuss some aspects of these definitions. First, in general the two variants of "T sound / complete for $\mathcal{P}$" and "T sound / complete for $\mathcal{P}$ w.r.t. reachable terms" are not equivalent. We will see an example below. One main goal for the design of any transformation should be *soundness for reduction.* Technically, the stronger *soundness for reduction w.r.t. reachable terms* may be preferable due to proof-technical reasons. Concerning

[9] This is equivalent to: $\forall s \in \mathcal{T}\, \forall t' \in \mathcal{T}'\colon \phi(s) \rightarrow^*_{\mathcal{R}'} t' \in \mathrm{NF}(\mathcal{R}') \Longrightarrow \psi(t') \in \mathrm{NF}(\mathcal{R})$.

[10] Observe that, regarding termination, in practice one is typically interested in slightly different preservation properties, namely of the shape "$\mathcal{R}$ operationally terminating (on s) if (only if, iff) $\mathcal{R}'$ terminating (on $\phi(s)$)".

the computation of normal forms, *soundness for preserving normal forms* is in general strictly weaker than *soundness for preserving normal forms w.r.t. reachable terms*, because it is unrealistic to expect that the (or a) normal form of $\phi(s)$ (for $s \in \mathcal{T}$) has the shape $\phi(t)$ (for some $t \in \mathcal{T}$). Hence, in practice we need here the latter notion. It is a specialized version of *soundness for reduction*, but strengthened by the requirement that the property of being a normal form is preserved backward.

Now, which properties of a transformation T would one expect to have for a computationally feasible simulation of a given system $\mathcal{R}$. Assuming that $\mathcal{R}$ is confluent (a) and operationally terminating (b) (or, equivalently, decreasing, if we assume only one condition per rule), computing the final unique result $t \in \mathrm{NF}(\mathcal{R})$ for some initial $s \in \mathcal{T}$ could be done via $\mathcal{R}'$ as follows: Initialize s into $\phi(s)$, normalize $\phi(s)$ in $\mathcal{R}'$ yielding some t', and finally translate back into $\psi(t')$. For this to work properly, one needs (c) completeness for termination w.r.t. reachable terms, (d) soundness for preserving normal forms w.r.t. reachable terms, and (e) soundness for reduction w.r.t. reachable terms. Then we get $\phi(s) \to^*_{\mathcal{R}'} u' \in \mathrm{NF}(\mathcal{R}')$ for some u' by (b) and (c). Then, by (d) and (e) we obtain $s \to^*_{\mathcal{R}} \psi(u') \in \mathrm{NF}(\mathcal{R})$. This together with $s \to^*_{\mathcal{R}} t \in \mathrm{NF}(\mathcal{R})$ and (a) implies $\psi(u') = t$, i.e., the desired final result.

In fact, it is not difficult to view most transformations from the literature, in particular [12], [2] and [6], as instances of Definition 2 above (with corresponding ϕ and ψ), together with the appropriately adapted terminology according to Definition 3. Due to lack of space, we refrain from describing this in detail. Let us instead briefly discuss a few aspects of unsoundness.

3.3 On-Unsoundness Phenomena

All transformation approaches that do not strongly restrict rewriting in the transformed system (like [10, 14, 16, 17]) such that the simulation of conditional rule application corresponds very closely to what is done in the original CTRS, exhibit in general unsoundness phenomena. More precisely, soundness for reduction (simulation soundness) is violated in general. This has been shown for unravelings by Marchiori in the technical report version [11] of [12] via a tricky counterexample. We present here a slightly simplified variant of this counterexample.[11]

Example 1 (unsoundness (for reduction) in general). The normal 1-CTRS $\mathcal{R}$ consisting of the rules $a \to c$, $a \to d$, $b \to c$, $b \to d$, $c \to e$, $c \to k$, $d \to k$, $h(x,x) \to g(x,x,f(d))$, $g(d,x,x) \to A$, $f(x) \to x \Leftarrow x \to^* e$ is unraveled according to [12] into $\mathcal{R}'$ which is $\mathcal{R}$ with the conditional rule replaced by $f(x) \to U(x,x)$ and $U(e,x) \to x$. In $\mathcal{R}'$ we have the following derivation between original terms: $h(f(a),f(b)) \to^*_{\mathcal{R}'} h(U(c,d),U(c,d)) \to g(U(c,d),U(c,d),f(d)) \to^*_{\mathcal{R}'} g(d,U(k,k),U(k,k)) \to_{\mathcal{R}'} A$. In the original system, however, $h(f(a),f(b)) \to^*_{\mathcal{R}} A$ does not hold as is easily seen!

[11] Compared to our version, [11, Example 4.3] has two more rules and two more constants.

Note that Example 1 also constitutes a counterexample to soundness (in general) for most other transformation approaches including [2, 6]. We will not go into details about sufficient conditions that nevertheless ensure soundness for particular transformations (some of them are well-known to exist, cf. e.g. [11], [15], [2], [6]). Instead, let us ask – concerning our proposed terminology – whether soundness w.r.t reachable terms is strictly stronger than ordinary soundness. In fact, the answer is Yes, again for most approaches!

Example 2 (soundness is weaker than soundness w.r.t. reachable terms). Consider the normal 1-CTRS $\mathcal{R} = \{a \to b,\ a \to c,\ f(x) \to x \Leftarrow x \to^* c\}$ that, via [2], is transformed into $\mathcal{R}' = \{a \to b,\ a \to c,\ f'(x, \bot) \to f'(x, \langle x \rangle),\ f'(x, \langle c \rangle) \to x\}$. In $\mathcal{R}'$ we have $f'(a, \bot) \to_{\mathcal{R}'} f'(a, \langle a \rangle) \to^*_{\mathcal{R}'} f'(b, \langle c \rangle) \to b$. However, backtranslation of the last two reachable terms here yields $\psi(f'(b, \langle c \rangle)) = f(b)$ and $\psi(b) = b$. Yet, in $\mathcal{R}$, we clearly do not have $f(b) \to^*_{\mathcal{R}} b$. Hence the transformation is not sound w.r.t. reachable terms. Note, however, that we do not obtain ordinary unsoundness here, because $\phi(f(a)) = f'(a, \bot) \to^*_{\mathcal{R}'} \phi(b) = b$ implies indeed $f(a) \to^*_{\mathcal{R}} b$.

Furthermore let us just mention that unravelings are in general also unsound for convertibility (a counterexample is obtained by extending Example 1 above.

And finally, we observe that some transformations in the literature (e.g. [4] and the technical report version [11] of [12]) are inherently unsound, because they do not transmit all variable bindings in the encoding process (in the introduction rule). Counterexamples for these cases can easily be constructed by adding a non-left-linear rule.

4 The New Transformation

4.1 Motivation, Goal and Basic Idea

Our transformation will be based on the approach of [2]. The transformation in [2] is complete, sound w.r.t. reachable terms and sound for preserving normal forms w.r.t. reachable terms for "constructor-based" (normal 1-)CTRSs that are left-linear and confluent. However, for other CTRSs it may be unsound or incomplete for confluence.

Example 3 (Incompleteness for confluence cf. [2, Example 4]). Consider the confluent CTRS $\mathcal{R} = \{g(s(x)) \to g(x)\ ,\ \ f(g(x)) \to x \ \Leftarrow\ x \to^* 0\}$. The transformation of [2] returns the TRS $\mathcal{R}' = \{g(s(x)) \to g(x),\ f'(g(x), \bot) \to f'(g(x), \langle x \rangle),\ f'(g(x), \langle 0 \rangle) \to x\}$. In $\mathcal{R}'$ we have the derivation $f'(g(s(0), \bot) \xrightarrow{(1)}_{\mathcal{R}'} f'(g(s(0), \langle s(0) \rangle) \xrightarrow{(2)}_{\mathcal{R}'} f'(g(0), \langle s(0) \rangle)$. The latter term is irreducible and does not rewrite to 0 although $f(g(0)) \to_{\mathcal{R}} 0$. $f'(g(s(0)), \bot)$ corresponds to $f(g(s(0)))$ in $\mathcal{R}$ that is matched by the lhs of the conditional rule l with the matcher $\tau = \{x \mapsto s(0)\}$. After introducing the introduction step (1) the unconditional g-rule is applied to the g-subterm (2). $f'(g(0), \langle s(0) \rangle)$ corresponds to $f(g(0))$ in the original CTRS that is matched by l with the matcher $\sigma = \{x \mapsto 0\}$. Since there are no derivations $x\tau \rightarrow^*_{\mathcal{R}} x\sigma$ ($x \in \mathcal{V}\mathrm{ars}(l)$) the conditional argument is "outdated" and "inconsistent" with the original argument.

Such inconsistencies may block further derivations as in Example 3 or lead to unsound derivations as in [2, Example 6]. We will refer to derivations that reproduce a term that is matched by the lhs of a conditional rule as being *pattern preserving.* Our goal is to provide a transformation that conservatively extends the transformation of [2] and does not require explicit propagation of reset information such that for confluent normal 1-CTRSs no explicit backtracking is needed. This means that in particular "critical" pattern preserving derivations are dealt with correctly.

In the derivation of Example 3, the unconditional rule $g(s(x)) \rightarrow g(x)$, that is applied in (2) and leads to the "outdated" conditional argument, should eliminate (and re-initialize) the conditional argument that has been introduced in (1). Yet the conditional argument is "out of reach" for the unconditional rule. By encoding the conditional argument in the g-subterm of the conditional rule, however, we can eliminate the conditional argument in the unconditional g-rule. This way $\mathcal{R}$ is transformed into $\mathcal{R}' = \{g'(s(x),z) \rightarrow g'(x,\bot),\ f(g'(x,\bot)) \rightarrow f(g'(x,\langle x\rangle)),\ f(g'(x,\langle 0\rangle)) \rightarrow x\}$. Now the conditional argument can be reintroduced: $f(g'(s(0),\bot)) \rightarrow f(g'(s(0),\langle s(0)\rangle)) \rightarrow f(g'(0,\bot)) \rightarrow^* 0$.

Following this example our strategy is to encode the conditions in all subterms of the lhs of a conditional rule that otherwise would give rise to "inconsistencies" of conditional and original arguments. To avoid confusion we will refer to subterms that directly (i.e., as additional argument of the root function symbol) contain a conditional argument as subterms encoding a conditional argument.

In certain conditional rules several subterms subterm of the lhs may lead to pattern preserving derivations. Then we have to encode the conditions multiple times:

Example 4 (Multiple conditional arguments). Consider the CTRS

$$\mathcal{R} = \left\{ \begin{array}{cc} g(s(x)) \rightarrow g(x) & h(s(x)) \rightarrow h(x) \\ f(g(x),h(y)) \rightarrow i(x,y) & \Leftarrow\ x \rightarrow^* 0, y \rightarrow^* 0 \end{array} \right\}$$

Both subterms $g(x)$ and $h(y)$ of the lhs of the conditional rule lead to a critical pattern preserving derivation. Therefore we need to add a conditional argument to the g-subterm and the h-subterm.

Whenever a conditional argument was eliminated and reinitialized, hence has become $\bot$, we have to reintroduce both conditional arguments via an introduction step. Therefore we need one introduction rule for each conditional argument. Only if both conditional arguments "satisfy" the conditions we may reproduce the corresponding rhs. Hence the transformed TRS $\mathcal{R}'$ here should be

$$\mathcal{R}' = \left\{ \begin{array}{r} g'(s(x),z) \rightarrow g'(x,\bot) \qquad h'(s(x),z) \rightarrow h'(x,\bot) \\ f(g'(x,\bot),h'(y,z_2)) \rightarrow f(g'(x,\langle x,y\rangle),h'(y,\langle x,y\rangle)) \\ f(g'(x,z_1),h'(y,\bot)) \rightarrow f(g'(x,\langle x,y\rangle),h'(y,\langle x,y\rangle)) \\ f(g'(x,\langle 0,0\rangle,z_2),h'(y,z_1,\langle 0,0\rangle)) \rightarrow i(x,y) \qquad\qquad\qquad \end{array} \right\}$$

Consider a conditional rule $\rho : l \to r \Leftarrow s_1 \to^* t_1, \ldots, s_n \to^* t_n$ of a CTRS $\mathcal{R}$. Subterms where possibly "critical" pattern preserving derivations start are always (instances of) non-variable subterms $l|_p$ ($p \in \overline{\mathcal{O}}(l)$) of l, because inconsistencies of conditional arguments and original arguments in variables can be resolved by rewrite steps in the variable bindings (for confluent CTRSs). However, in order to detect all possibilities of overlaps (also after rewriting in 'variable parts'), we must linearize l into l^{lin}. Then we can identify possible lhs's that overlap into l^{lin} by systematic unification of all non-variable subterms of l with lhs's of $\mathcal{R}$ and encode the conditions in such subterms of l.

It is not necessary to encode conditions in all subterms of l^{lin} that are unifiable with some lhs. If a subterm does not contain any variables that occur in the conditions, rewrite steps in this subterm do not influence the satisfiability of the conditions and therefore it is not necessary to introduce the conditions here. Yet, we must consider the case that (only) after some rewrite steps in such subterms a rule may become applicable that did not overlap into l^{lin} initially.

Example 5 (Iterative abstraction of overlapping subterms). Consider the CTRS

$$\mathcal{R} = \left\{ \begin{array}{cc} a \to b & g(s(x), k(b)) \to g(x, h(a)) \\ h(b) \to k(b) & f(g(x, h(a))) \to x \Leftarrow x \to^* 0 \end{array} \right\}$$

The only subterm of the linearized lhs of the conditional rule l^{lin} into which an lhs of some rule overlaps is the constant a which does not contain any variable of the condition. Since there are no other overlaps, we would just encode the conditions in the root symbol of the conditional rule:

$$\mathcal{R}' = \left\{ \begin{array}{c} a \to b \qquad g(s(x), k(b)) \to g(x, h(a)) \qquad h(b) \to k(b) \\ f'(g(x, h(a)), \bot) \to f'(g(x, h(a)), \langle x \rangle) \qquad f'(g(x, h(a)), \langle s(0) \rangle) \to x \end{array} \right\}$$

In $\mathcal{R}'$ we have the unsound derivation

$$\begin{aligned} f'(g(s(0), h(a)), \bot) &\to f'(g(s(0), h(a)), \langle s(0) \rangle) \to f'(g(s(0), h(b)), \langle s(0) \rangle) \\ &\to f'(g(s(0), k(b)), \langle s(0) \rangle) \to f'(g(0, h(a)), \langle s(0) \rangle) \to 0\,. \end{aligned}$$

Although (the lhs of) the g-rule does not overlap into l^{lin}, it is applicable after some rewrite steps. Therefore, we abstract all non-variable subterms of l^{lin}, that are unifiable with some lhs of $\mathcal{R}$, into fresh variables iteratively and try to unify the non-variable subterms of the resulting terms with lhs's of $\mathcal{R}$. In the example, the a-rule overlaps into $l_0 = f(g(x, h(a)))$ so that we abstract it into $l_1 = f(g(x, h(y)))$. Because of the overlap with the h-rule this term then is abstracted into $l_2 = f(g(x, z))$. Now the g-rule overlaps into the g-subterm of l_2 that contains the variable x that also occurs in the conditions. We therefore encode a conditional argument in the g-subterm instead of f and obtain the transformed TRS

$$\mathcal{R}' = \left\{ \begin{array}{c} a \to b \qquad g'(s(x), k(b), z) \to g'(x, h(a), \bot) \qquad h(b) \to k(b) \\ f(g'(x, h(a), \bot)) \to f(g'(x, h(a), \langle x \rangle)) \qquad f(g'(x, h(a), \langle 0 \rangle)) \to x \end{array} \right\}$$

In CTRSs that give rise to multiple conditional arguments, it may be possible to "recombine" them in an inconsistent way, if we do not iterate the sketched construction:

Example 6 (Recombination of conditional arguments). Consider the CTRS

$$\mathcal{R} = \left\{ \begin{array}{ccc} i(0, s(0)) \to 0 & i(s(0), 0) \to 0 & f(s(x), y) \to s(x) \\ f(x, s(y)) \to s(y) & g(s(x)) \to g(x) & h(s(x)) \to h(x) \\ \multicolumn{3}{c}{f(s(g(x)), s(h(y))) \to i(x, y) \Leftarrow i(x, y) \to^* 0} \end{array} \right\}$$

The g-rule and the h-rule overlap into the lhs of the conditional rule, therefore we would encode the condition in both subterms:

$$\mathcal{R}' = \left\{ \begin{array}{lll} i(0, s(0)) \to 0 & i(s(0), 0) \to 0 & f(s(x), y) \to s(x) \\ f(x, s(y)) \to s(y) & g'(s(x), z) \to g'(x, \bot) & h'(s(x), z) \to h'(x, \bot) \\ \multicolumn{3}{l}{\quad f(s(g'(x, \bot)), s(h'(y, z_2))) \to f(s(g'(x, \langle i(x, y)\rangle)), s(h'(y, \langle i(x, y)\rangle)))} \\ \multicolumn{3}{l}{\quad f(s(g'(x, z_1)), s(h'(y, \bot))) \to f(s(g'(x, \langle i(x, y)\rangle)), s(h'(y, \langle i(x, y)\rangle)))} \\ \multicolumn{3}{l}{f(s(g'(x, \langle 0\rangle)), s(h'(y, \langle 0\rangle))) \to i(x, y)} \end{array} \right\}$$

In $\mathcal{R}'$, we now have the following derivation:

$$\begin{array}{l} f(f(s(g'(0, \bot)), s(h'(s(0), \bot))), f(s(g'(s(0), \bot)), s(h'(0, \bot)))) \\ \quad \to^* f(f(s(g'(0, t_1)), s(h'(s(0), t_1))), f(s(g'(s(0), t_2)), s(h'(0, t_2)))) \\ \quad \to^* f(s(g'(0, t_1)), s(h'(0, t_2))) \to^* f(s(g'(0, \langle 0\rangle)), s(h'(0, \langle 0\rangle))) \to i(0, 0) \end{array}$$

with $t_1 = \langle i(0, s(0))\rangle$ and $t_2 = \langle i(s(0), 0)\rangle$, whereas in $\mathcal{R}$ we have

$$f(f(s(g(0)), s(h(s(0)))), f(s(g(s(0))), s(h(0)))) \not\to^*_{\mathcal{R}} i(0, 0)\,.$$

In order to avoid that "fragments" of introduction steps can be inconsistently rearranged, we will iteratively abstract (in parallel) all non-variable subterms of the lhs of the conditional rule, that (after transformation) contain conditional arguments, into new variables. This way the lhs of the conditional rule $f(s(g(x), h(y)))$ is abstracted into $f(s(z_1), s(z_2))$. But then we have an overlap with the unconditional f-rules at root position. Hence, we will also encode the conditional argument at root position. Thus the above problem disappears.

4.2 Definition of the Transformation

In our transformation we iteratively abstract "overlapping" subterms of lhs's of conditional rules into new variables and append conditional arguments to such subterms provided they contain variables that also occur in the conditions. Additionally we have to take into account that also rules into which the lhs of a conditional rule overlaps may lead to inconsistencies. Before defining our transformation we define some mappings to increase (decrease) the arity of function symbols:

Definition 4 (Initialization mapping and backtranslation)

$$\phi_f^{\perp}(t) = \begin{cases} f'(\phi_f^{\perp}(t_1), \dots, \phi_f^{\perp}(t_n), \perp) & \text{if } t = f(t_1, \dots, t_n) \\ g(\phi_f^{\perp}(t_1), \dots, \phi_f^{\perp}(t_m)) & \text{if } t = g(t_1, \dots, t_m),\ g \neq f \\ t & \text{if } t \text{ is a variable} \end{cases}$$

$$\psi_f(t') = \begin{cases} f(\psi_f(t_1), \dots, \psi_f(t_n)) & \text{if } t' = f'(t_1, \dots, t_n, u_1) \\ g(\psi_f(t_1), \dots, \psi_f(t_m)) & \text{if } t' = g(t_1, \dots, t_m),\ g \neq f' \\ t' & \text{if } t' \text{ is a variable} \end{cases}$$

$$\phi_f^{\mathcal{X}}(t) = \begin{cases} f'(\phi_f^{\mathcal{X}_1}(t_1), \dots, \phi_f^{\mathcal{X}_n}(t_n), z) & \text{if } t = f(t_1, \dots, t_n) \\ g(\phi_f^{\mathcal{X}_1}(t_1), \dots, \phi_f^{\mathcal{X}_m}(t_m)) & \text{if } t = g(t_1, \dots, t_m),\ g \neq f \\ t & \text{if } t \text{ is a variable} \end{cases}$$

where $\mathcal{X}$ is an infinite set of new variables, $z \in \mathcal{X}$ and $\mathcal{X}_i$ $(i \geq 1)$ are infinite disjoint subsets of $\mathcal{X}$ such that $z \notin \mathcal{X}_i$. We abbreviate multiple applications of these mappings: $\phi^{\perp}_{f_1,\dots,f_n}(t) = \phi^{\perp}_{f_2,\dots,f_n}(\phi^{\perp}_{f_1}(t))$, $\psi_{f_1,\dots,f_n}(t) = \psi_{f_2,\dots,f_n}(\psi_{f_1}(t))$ and $\phi^{\mathcal{X}}_{f_1,\dots,f_n}(t) = \phi^{\mathcal{X} \setminus \mathcal{V}ars(\phi^{\mathcal{X}}_{f_1}(t))}_{f_2,\dots,f_n}(\phi^{\mathcal{X}}_{f_1}(t))$.

By abuse of notation we assume in the following that, if $\phi^{\mathcal{X}}$ is used multiple times in a term, then always only mutually distinct variables are inserted.

Definition 5 (Definition of the transformation T). *Let $\mathcal{R}$ be a normal 1-CTRS so that the rules are arranged in some arbitrary but fixed total order $<$. Let $\rho : l_\rho \to r_\rho \Leftarrow s_{\rho,1} \to^* t_{\rho,1}, \dots, s_{\rho,n_\rho} \to^* t_{\rho,n_\rho}$ be a conditional rule of $\mathcal{R}$. Let l_i and P_i be the following*

$$\begin{aligned} l_0 &= l_\rho^{lin} & l_{i+1} &= l_i[z_1]_{q_1} \dots [z_m]_{q_m} \\ P_0 &= \emptyset & P_{i+1} &= P_i \cup \{q \in \overline{Q} \mid \mathcal{V}ars(l_\rho|_q) \cap \mathcal{V}ars(s_{\rho,1}, \dots, s_{\rho,n_\rho}) \neq \emptyset\} \end{aligned}$$

where $Q = \{q \in \overline{\mathcal{O}}(l_i) \mid l_i\sigma = l_i\sigma[l_{\rho'}\sigma]_q, \rho' \in \mathcal{R}, \rho' \neq \rho \ \vee \ q \neq \epsilon\}$ are all positions of l_i that are unifiable with some lhs $l_{\rho'}$ (except $\rho' = \rho$ at root position), $\{q_1, \dots, q_m\} = \overline{Q} = \{q \in Q \mid \nexists q' \in Q : q < q'\}$ are the innermost positions of Q and $z_1, \dots, z_m$ are fresh new variables.

Let $\overline{l_\rho} = l_j$ be the first l_j such that $l_j = l_{j+1}$. Then the set of conditional positions P_ρ is

$$P_\rho = \begin{cases} P_j \cup \{\epsilon\} & \text{if } \exists \rho', l_{\rho'}\sigma = l_{\rho'}\sigma[\overline{l_\rho}\sigma]_q \text{ with } q \in \overline{\mathcal{O}}(l_{\rho'}) \text{ and } \rho' \neq \rho \vee q \neq \epsilon, \\ & \text{or } P_j = \emptyset \text{ and } \rho \text{ is a conditional rule} \\ P_j & \text{otherwise} \end{cases}$$

Let $\{p_{\rho,1}, \dots, p_{\rho,k_\rho}\} = P_\rho$ and $f_{\rho,j} = \mathrm{root}(l|_{p_{\rho,j}})$. Then the position of the conditional argument encoded in $l_\rho|_{p_{\rho,j}}$ is

$$i_{\rho,j} = \mathrm{arity}(f_{\rho,j}) + 1 + |\{\langle \rho', j' \rangle \mid \langle \rho', j' \rangle <_{lex} \langle \rho, j \rangle, f_{\rho',j'} = f_{\rho,j}\}|$$

Let $\mathcal{R} = \{\rho_1, \ldots, \rho_m\}$. The initialization mapping ϕ is $\phi^{\perp}_{f_{\rho_1,1},\ldots,f_{\rho_m,k_{\rho_m}}}$, $\phi^{\mathcal{X}}$ is $\phi^{\mathcal{X}}_{f_{\rho_1,1},\ldots,f_{\rho_m,k_{\rho_m}}}$ and the backtranslation ψ is $\psi_{f_{\rho_1,1},\ldots,f_{\rho_m,k_{\rho_m}}}$.

A rule $\rho \in \mathcal{R}$ is transformed into the rules

$$\rho'_{\rho,j} : \phi^{\mathcal{X}}(l_\rho)[\perp]_{p_{\rho,j}.i_{\rho,j}} \to \phi^{\mathcal{X}}(l_\rho)[\langle \phi^{\perp}(s_{\rho,1}), \ldots, \phi^{\perp}(s_{\rho,n_\rho})\rangle]_{p_{\rho,1}.i_{\rho,1},\ldots,p_{\rho,k_\rho}.i_{\rho,k_\rho}}$$
$$\rho'_{\rho,k_\rho+1} : \phi^{\mathcal{X}}(l_\rho)[\langle \phi^{\mathcal{X}}(t_{\rho,1}), \ldots, \phi^{\mathcal{X}}(t_{\rho,n_\rho})\rangle]_{p_{\rho,1}.i_{\rho,1},\ldots,p_{\rho,k_\rho}.i_{\rho,k_\rho}} \to \phi^{\perp}(r_\rho)$$

$\rho'_{\rho,1}, \ldots, \rho'_{\rho,k_\rho}$ are the introduction rules *and $\rho'_{k_\rho+1}$ is the* elimination rule *of ρ. The transformed TRS $T(\mathcal{R})$ then is $(\mathcal{R}', \phi, \psi)$ where $\mathcal{R}' = \{\rho'_{\rho_0,1}, \ldots, \rho'_{\rho_m,k_{\rho_m}+1}\}$.*

In constructor-based normal 1-CTRSs, P_ρ is $\{\epsilon\}$ for all conditional rules ρ so that in this case our transformation coincides with the transformation of [2], except for the additional wrapping $\langle \ldots \rangle$ of the conditional arguments. In unconditional rules ρ, $P_\rho = \emptyset$.

The following example of [6] can be interpreted as a "self-sorting" list structure:

Example 7 (Sorting CTRS of [6]). Consider the CTRS

$$\mathcal{R} = \left\{ \begin{array}{c} 0 \le y \to tt \qquad s(x) \le 0 \to ff \qquad s(x) \le s(y) \to x \le y \\ f(x, f(y, ys)) \to f(y, f(x, ys)) \ \Leftarrow\ x \le y \to^* ff \end{array} \right\}$$

The (linear) lhs of the conditional rule $l_0 = f(x, f(y, ys))$. The conditional rule overlaps into itself at position $q = 2$ such that $P_1 = \{2\}$ and $l_1 = f(x, z)$. Now only the conditional rule itself overlaps into l_1 at root position, therefore $\bar{l} = l_1 = f(x, z)$. Since $\bar{l}$ overlaps into the lhs of the conditional rule at some non-root position $P_\rho = \{2\} \cup \{\epsilon\}$. For both positions the root symbol is f and the arity of f is increased by 2. The transformed TRS then is

$$\mathcal{R}' = \left\{ \begin{array}{c} 0 \le y \to tt \qquad s(x) \le 0 \to ff \qquad s(x) \le s(y) \to x \le y \\ f(x, f(y, ys, z_1, z_2), z_3, \perp) \to f(x, f(y, ys, \langle x \le y\rangle, z_2), z_3, \langle x \le y\rangle) \\ f(x, f(y, ys, \perp, z_2), z_3, z_4) \to f(x, f(y, ys, \langle x \le y\rangle, z_2), z_3, \langle x \le y\rangle) \\ f(x, f(y, ys, \langle ff\rangle, z_2), z_3, \langle ff\rangle) \to f(y, f(x, ys, \perp, \perp), \perp, \perp) \end{array} \right\}$$

4.3 Properties of the Transformation

In the following we assume that $\mathcal{R}$ always denotes a normal 1-CTRS, $\mathcal{R}'$ its transformed TRS using our transformation T and ρ a conditional rule with lhs l that leads to k conditional positions $p_1.i_1, \ldots, p_k.i_k$.

The following result contains a selection of syntactical preservation properties of our transformation. Properties (2) - (5) are not satisfied by the transformation of [6].

Lemma 1 (Syntactic properties)

(1) The transformation is sound and complete for being left-linear.
(2) The transformation is sound and complete for being non-collapsing.

(3) If $\mathcal{R}$ is non-overlapping, then $\mathcal{R}'$ is weakly non-overlapping.
(4) The transformation is sound and complete for being an overlay system.
(5) If $\mathcal{R}$ is orthogonal, $\mathcal{R}'$ is weakly orthogonal.

In order to show that our transformation has nice preservation properties for certain CTRSs, we have to guarantee that inconsistencies of conditional arguments and original arguments do not occur or are not "critical".

In a derivation D starting from some initialized term every conditional argument originates from an introduction step. A conditional argument originating from a certain introduction step may be viewed as being inconsistent if the redex of the introduction step is modified at some original argument that is not inside the matcher of the introduction step (rewrite steps in the matcher can be reconstructed in the conditional arguments, at least directly after their introduction). The redex of the "modifying" rewrite step overlaps with the redex of the introduction step so that, according to our transformation, there is (at least) one conditional argument inside the matcher of the "modifying" redex (unless the rewrite step is not potentially dangerous, i.e., it does not modify any variable that occurs in the conditions or it is an introduction step). We will only consider those conditional arguments as being inconsistent, that are inside such a redex, and refer to the overlapping rewrite step as the rewrite step in which the conditional arguments *become* inconsistent.

Definition 6 (Inconsistent conditional arguments). *Let D be a derivation $\phi(s) \to^* u_0 \to_{q_0,\rho'_0} u_1 \to_{q_1,\rho'_1} \cdots$ $(s \in \mathcal{T})$ in $\mathcal{R}'$. Let ρ'_0 be an introduction rule of ρ and $u_n|_{q.i_j}$ be a conditional argument that is a descendant of $u_1|_{q_0.p_j.i_j}$. The conditional argument in $u_n|_{q.i_j}$ is* inconsistent w.r.t. *D, if there is an elimination step $u_m = u_m[l'_m\sigma]_{q_m} \to_{q_m,\rho'_m} u_m[r'_m\sigma]_{q_m} = u_{m+1}$ in D such that $u_m|_{q'_m.p_j.i_j}$ is a descendant of $u_1|_{q_0.p_j.i_j}$ and an ancestor of $u_n|_{q.i_j}$, and the elimination step "overlaps with" the descendant $u_m|_{q'_m}$ of $u_0|_{q_0}$ above p_j, i.e., $q_m < q'_m.p_j$ and there is no $q' \in \mathcal{O}_{\mathcal{X}}(l'_m)$ such that $q_m.q' \leq q'_m$. A conditional argument that is not inconsistent (w.r.t. D) is* consistent (w.r.t. *D*).

Lemma 2 (Iterative abstraction, inconsistent conditional arguments) *Let $D\colon \phi(s) \to^*_{\mathcal{R}'} s' \to_{\mathcal{R}'} t'$ $(s \in \mathcal{T})$ be a derivation in $\mathcal{R}'$ such that* $\mathrm{root}(s'|_q) = \mathrm{root}(\phi(l|_{p_j}))$ *and $s'|_{q.i_j} \neq \bot$ for some $j \in \{1, \ldots, k\}$. If the conditional argument $s'|_{q.i_j}$ becomes inconsistent w.r.t. D in the last rewrite step $s' \to t'$ of D, then there is a $q' \in \mathcal{O}(s')$ such that $q'.p_j = q$ and in the iteration of Definition 5 for ρ we obtain some l_i that is unifiable with $\psi(s'|_{q'})$, and there is some conditional position $p.i \in \{p_1.i_1, \ldots, p_k.i_k\}$ of ρ such that $p < p_j$.*

If inconsistent conditional arguments "block" or "are used in" elimination steps, we may obtain incompleteness for confluence or unsoundness. We will refer to those CTRSs where this cannot happen as *consistently transformable* ones.

Definition 7 (Consistently transformable). *$\mathcal{R}$ is* consistently transformable, *if for every derivation $D\colon \phi(s) \to^*_{\mathcal{R}'} s'$ $(s \in \mathcal{T})$ such that $\psi(s'|_q) = l\sigma$, either $s'|_{q.p_j.i_j} = \bot$ for some j or $s'|_{q.p_j.i_j}$ is consistent w.r.t. D for all $j \in \{1, \ldots, k\}$.*

Unfortunately, not all CTRSs are consistently transformable:

Example 8 (Inconsistent conditional arguments in collapsing systems). Consider the CTRS

$$\mathcal{R} = \left\{ \begin{array}{ll} i(a,a) \to a & g(f(x,b)) \to x \\ f(g(x),y) \to h(x) & \Leftarrow\ i(x,y) \to^* a \end{array} \right\}$$

We obtain $P_2 = \{\epsilon, 1\}$ and hence the transformed TRS is

$$\mathcal{R}' = \left\{ \begin{array}{l} i(a,a) \to a \qquad g'(f'(x,b,z_1),z_2) \to x \\ f'(g'(x,\bot),y,z_2) \to f'(g'(x,\langle i(x,y)\rangle),y,\langle i(x,y)\rangle) \\ f'(g'(x,z_1),y,\bot) \to f'(g'(x,\langle i(x,y)\rangle),y,\langle i(x,y)\rangle) \\ f'(g'(x,\langle a\rangle),y,\langle a\rangle) \to h(x) \end{array} \right\}$$

$\mathcal{R}$ has two infeasible conditional critical pairs and is therefore confluent (it is also decreasing). Yet, in $\mathcal{R}'$ we obtain D: $f'(g'(f'(g'(a,\bot),b,\bot),\bot),a,\bot) \to^* f'(g'(f'(g'(a,t_1),b,t_1),t_2),a,t_2) \to f'(g'(a,t_1),t_2)$ where $t_1 = \langle i(a,b)\rangle$ and $t_2 = \langle i(f'(g'(a,\bot),b,\bot),a)\rangle$. The inner conditional argument is inconsistent w.r.t. D while the outer conditional argument is consistent. Usually, we would expect that the inner conditional argument is set to $\bot$, but since the g-rule is collapsing, no conditional argument is set to $\bot$ in its rhs. Hence, the inconsistent (w.r.t. D) conditional argument blocks further reductions.

Lemma 3 (Inconsistent conditional arguments and collapsing rules) *Let D: $\phi(s) \to^*_{\mathcal{R}'} s'$ be a derivation in $\mathcal{R}'$ such that $\psi(s'|_q) = l\sigma$ and $s'|_{q.p_j.i_j} \neq \bot$ for all $j \in \{1,\ldots,k\}$. If some conditional argument $s'|_{q.p_j.i_j}$ is inconsistent w.r.t. D, then $\mathcal{R}$ is collapsing.*

Theorem 1 (Sufficient syntactic conditions for consistent transformability (a)). *$\mathcal{R}$ is consistently transformable, if $\mathcal{R}$ is non-collapsing or all $\rho \in \mathcal{R}$ only lead to pairwise parallel subterms encoding conditional arguments.*

Theorem 2 (Sufficient syntactic conditions for consistent transformability (b)). *$\mathcal{R}$ is consistently transformable, if it is non-collapsing, a constructor system, a system where all left-hand sides of conditional rules are constructor terms, or a left-linear overlay system.*

Theorem 3 (Soundness w.r.t. reachable terms). *If $\mathcal{R}$ is consistently transformable and confluent, then for all reachable $s', t' \in \mathcal{T}'_r$ $s' \to^*_{\mathcal{R}'} t' \Rightarrow \psi(s') \to^*_{\mathcal{R}} \psi(t')$.*

Theorem 4 (Completeness w.r.t. reachable terms). *If $\mathcal{R}$ is consistently transformable, left-linear and confluent, then for all $s' \in \mathcal{T}'_r, t \in \mathcal{T}$ such that $\psi(s') \to_{\mathcal{R}} t$ there is a $t' \in \mathcal{T}'_r$ with $s' \to^+_{\mathcal{R}'} t'$ and $\psi(t') = t$.*

Theorem 5 (Soundness for preserving normal forms w.r.t. reachable terms). *If $\mathcal{R}$ is consistently transformable, left-linear and confluent, then $\mathcal{R}'$ is sound for preserving normal forms w.r.t. reachable terms.*

Theorem 6 (Preservation of termination). *If $\mathcal{R}$ is consistently transformable, decreasing and confluent, then $\mathcal{R}'$ is terminating on reachable terms.*

Observe that in Theorem 4 and, consequently, also in Theorem 5 we have left-linearity as additional assumption! The reason is that non-left-linear rules may lead to an incomplete (w.r.t. reachable terms) behaviour of our transformation.

Example 9 (Non-left-linearity may lead to incompleteness w.r.t. reachable terms) Consider the CTRS $\mathcal{R} = \{g(0) \to 0,\ f(x,x) \to a,\ f(g(x),y) \to a \ \Leftarrow\ x \to^* 0, y \to^* 0\}$ that is transformed into $\mathcal{R}'$ consisting of

$$g'(0,z) \to 0 \quad f'(x,x,z) \to a \qquad f'(g'(x,\bot),y,z_2) \to f'(g'(x,\langle x,y\rangle),y,\langle x,y\rangle)$$
$$f'(g'(x,z_1),y,\bot) \to f'(g'(x,\langle x,y\rangle),y,\langle x,y\rangle) \qquad f'(g'(x,\langle 0,0\rangle),y,\langle 0,0\rangle) \to a$$

For $s = f(g(a),g(a))$ we have in $\mathcal{R}$ just one normalizing step $f(g(a),g(a)) \to_{\mathcal{R}} a$. In $\mathcal{R}'$ we get the corresponding reduction $\phi(s) = f'(g'(a,\bot),g'(a,\bot),\bot) \to_{\mathcal{R}'} a$, but also $\phi(s) \to_{\mathcal{R}'} f'(g'(a,\langle a,g'(a,\bot)\rangle),g'(a,\bot),\langle a,g'(a,\bot)\rangle)$ where the latter term does not rewrite to a and is even irreducible w.r.t. $\mathcal{R}'$. Hence, in this example the transformation is neither complete w.r.t. reachable terms nor sound for preserving normal forms w.r.t. reachable terms.

Our transformation satisfies many properties only for consistently transformable CTRSs. Although it is undecidable, whether a CTRS is consistently transformable, we can show for large sub-classes of CTRSs that they are consistently transformable, cf. e.g. Theorem 2. According to Theorem 1 every system, that is not consistently transformable, must in particular be collapsing. For such CTRSs we will now show how to handle them via an additional preprocessing transformation that makes all rules non-collapsing, but retains a one-to-one-correspondence between rewrite steps.

4.4 Transformation for Non-consistently Transformable CTRSs

For the fully general case, consider a collapsing rule $l \to x \ \Leftarrow\ c$ of a CTRS $\mathcal{R}$. An intuitive method to transform $\mathcal{R}$ into a non-collapsing CTRS is to wrap x into a new symbol C: $l \to C(x) \ \Leftarrow\ c$. In order to retain a one-to-one correspondence of rewrite steps, it is necessary to wrap all non-variable terms in C consistently.

Definition 8 (Transformation into non-collapsing CTRSs T_C) *Let $\mathcal{R}$ be some CTRS with signature $\mathcal{F}$, $C \notin \mathcal{F}$ be some new unary function symbol, ρ be a rule $l \to r \ \Leftarrow\ s_1 \to^* t_1, \ldots, s_n \to^* t_n \in \mathcal{R}$ and let ϕ'_ρ be the auxiliary mapping*

$$\phi'_\rho(t) = \begin{cases} C(f(\phi'_\rho(t_1),\ldots,\phi'_\rho(t_n))) & \text{if } t = f(t_1,\ldots,t_n) \\ C(r) & \text{if } t = r \text{ and } r \text{ is a variable} \\ x & \text{if } t = x \text{ and } x \neq r \end{cases}$$

Then the transformed rule ρ_C of ρ is

$$\phi'_\rho(l) \to \phi'_\rho(r) \ \Leftarrow\ \phi'_\rho(s_1) \to^* \phi'_\rho(t_1), \ldots, \phi'_\rho(s_n) \to^* \phi'_\rho(t_n)$$

The transformed CTRS $\mathcal{R}_C$ then is $\mathcal{R}_C = \{\rho_C \mid \rho \in \mathcal{R}\}$ with signature $\mathcal{F} \cup \{C\}$ and the initialization mapping ϕ_C and backtranslation ψ_C are

$$\phi_C(t) = \begin{cases} C(f(\phi_C(t_1), \dots, \phi_C(t_n))) & \text{if } t = f(t_1, \dots, t_n) \\ C(t) & \text{if } t \text{ is a variable} \end{cases}$$

$$\psi_C(t) = \begin{cases} f(\psi_C(t_1), \dots, \psi_C(t_n)) & \text{if } t = f(t_1, \dots, t_n) \\ \psi_C(t') & \text{if } t = C(t') \\ t & \text{if } t \text{ is a variable} \end{cases}$$

Using this transformation collapsing rules are replaced by non-collapsing rules into which all other rules overlap. For reachable terms of T_C (these are all $\phi_C(s)$), these overlaps are joinable within one step (from both sides).

Definition 9 (Combined transformation $T \circ T_C$). *Let $\mathcal{R}$ be a CTRS such that $T_C(\mathcal{R}) = (\mathcal{R}_C, \phi_C, \psi_C)$ and $T(\mathcal{R}_C) = (\mathcal{R}'_C, \phi', \psi')$. Then the combined transformation $T \circ T_C$ is $(\mathcal{R}'_C, \phi' \circ \phi_C, \psi_C \circ \psi')$.*

Using this combined transformation, it is easily verified that the "blocking problem" in Example 8 disappears.[12]

Theorem 7 (Properties of the combined transformation). *Let $\mathcal{R}$ be a normal 1-CTRS such that $T \circ T_C(\mathcal{R}) = (\mathcal{R}'_C, \phi, \psi)$.*

(1) If $\mathcal{R}$ is confluent, then $\mathcal{R}'_C$ is sound w.r.t. reachable terms.
(2) If $\mathcal{R}$ is left-linear and confluent, then $\mathcal{R}'_C$ is complete w.r.t. reachable terms.
(3) If $\mathcal{R}$ is left-linear and confluent, then $\mathcal{R}'_C$ is sound for preserving normal forms w.r.t. reachable terms.
(4) If $\mathcal{R}$ is decreasing and confluent, then $\mathcal{R}'_C$ is terminating on reachable terms.

5 Experiments, Related Work and Discussion

In our experiments we compared our transformation (T) with other transformations, especially the one of [6] (T_{SR}). The "sorting list" of Example 7 is representative for most of our results. Sorting the descending list $f(s^{n-1}(0), f(s^{n-2}(0), \cdots, f(0, nil) \cdots))$ needs the following number of rewrite steps to obtain the sorted, irreducible list:

[12] As remarked by one of the referees, our construction of introducing separating C-layers in terms combined with the previous transformation appears to have some similarity with another complex transformation in [5, 3.4] (based on a different approach), an early forerunner of [2] and [6]. However, in [5, 3.4] these layers are used to propagate reset information to outer positions similar to [6].

		no sharing				maximal sharing			
		innermost		outermost		innermost		outermost	
	n	left	right	left	right	left	right	left	right
$T_{SR}(\mathcal{R})$	34	15893	15893	39269	125509	15893	15893	33285	125509
$T(\mathcal{R})$		27863	27863	34967	46903	14773	14773	9349	19043
$T_{SR}(\mathcal{R})$	55	62863	62863	166319	820763	62863	62863	140084	820763
$T(\mathcal{R})$		115335	115335	144538	196955	59895	59895	35144	76537

When using outermost rewriting, T_{SR} requires more rewrite steps, because it resets conditional arguments "too often": In $T(\mathcal{R})$ every condition is evaluated and if it is not satisfied, the conditional arguments is "cached" in terms like $f(0, f(s(0), f(s(s(0)), \ldots), \langle tt\rangle, \langle tt\rangle), \bot, \langle tt\rangle)$. In $T_{SR}(\mathcal{R})$ this term corresponds to $\{f(0, f(s(0), f(s(s(0)), \ldots), \langle tt\rangle), \langle tt\rangle)\}$. If we exchange two elements at inner positions, the reset-operator is propagated to outer positions in $T_{SR}(\mathcal{R})$, so that all conditional arguments are reset and must be reintroduced and reevaluated: $f(0, f(s(0), \{\ldots\}, \langle tt\rangle), \langle tt\rangle) \rightarrow^* \{f(0, f(s(0), f(s(s(0)), \ldots), \bot), \bot)\}$.

Our transformation is rather complex, because we have to iteratively check terms for unifiability in order to determine the subterms in which we have to encode conditional arguments. We can approximate these subterms via defined symbols: For $\rho : l \rightarrow r \Leftarrow s_1 \rightarrow^* t_1, \ldots, s_n \rightarrow^* t_n$ just take $P_\rho = \{p \in \overline{\mathcal{O}}(l_\rho) \mid root(l_\rho|_p) \in \mathcal{D}, \mathcal{V}\text{ars}(l_\rho|_p) \cap \mathcal{V}\text{ars}(s_{\rho,1}, \ldots, s_{\rho,n_\rho}) \neq \emptyset\}$ for encoding. This approximation clearly yields an "approximation from above", cf. Definition 5.

In order to transform *deterministic CTRSs* (DCTRSs) we can use a strategy that is easily adaptable to other transformations like e.g. [6] or [15]. A deterministic rule is a rule $l \rightarrow r \Leftarrow s_1 \rightarrow^* t_1, \ldots, s_n \rightarrow^* t_n$ that may contain extra variables, yet all extra variables "depend" directly or indirectly on variables in l: $\mathcal{V}\text{ars}(s_i) \subseteq \mathcal{V}\text{ars}(l, t_1, \ldots, t_{i-1})$ and $\mathcal{V}\text{ars}(r) \subseteq \mathcal{V}\text{ars}(l, t_1, \ldots, t_n)$. Let w.l.o.g. the conditions $s_1 \rightarrow^* t_1, \ldots, s_m \rightarrow^* t_m$ be those satisfying $\mathcal{V}\text{ars}(s_1, \ldots, s_m) \subseteq \mathcal{V}\text{ars}(l)$. By transforming the rule ρ: $l \rightarrow r \Leftarrow s_1 \rightarrow^* t_1, \ldots, s_m \rightarrow^* t_m$ we obtain introduction rules $\rho'_1, \ldots, \rho'_k$ and an elimination rule ρ'_{k+1}: $l' \rightarrow r'$ without extra variables. By adding the remaining conditions to the elimination rule we obtain the deterministic conditional rule $l' \rightarrow r' \Leftarrow \phi(s_{m+1}) \rightarrow^* \phi^{\mathcal{X}}(t_{m+1}), \ldots, \phi(s_n) \rightarrow^* \phi^{\mathcal{X}}(t_n)$ with strictly less conditions than the original rule. By repeatedly applying the above strategy, we finally obtain an unconditional TRS. In [13] a similar iterative approach for unravelings is presented.

Regarding future work, we intend to investigate various further aspects of our transformation, especially whether soundness for left-linear (consistently transformable) normal 1-CTRSs holds, and whether we can somehow get rid of the left-linearity requirement in Theorems 4, 5 and 7(2)-(3). Moreover we want to explore the optimizations and refinements sketched above. We also plan to extend our preliminary practical evaluations and comparison with related approaches. Another perspective is to analyze possible applications of our approach like conditional narrowing or inversion of rewrite systems ([14]).

6 Conclusion

We have presented a general framework for analyzing transformations of CTRSs into TRSs as well as a new approach whose characteristic feature is backtracking-freeness. It works for left-linear confluent normal 1-CTRSs and extends the approach of [2] to non-constructor systems. Compared to [6] (which also works for confluent, but not necessarily left-linear systems) with a "reset"-operator and an explicit (rule-based) propagation of "reset"-information, our approach directly incorporates necessary reset information in the transformation.

Acknowledgements. We are grateful to the anonymous referees for their detailed feedback, hints and criticisms. In particular, one of them exhibited examples (similar to Ex. 8 and 9) that triggered a partial revision of our original analysis in Section 4.

References

[1] Aida, H., Goguen, J., Meseguer, J.: Compiling concurrent rewriting onto the rewrite rule machine. In: Okada, M., Kaplan, S. (eds.) CTRS 1990. LNCS, vol. 516, pp. 320–332. Springer, Heidelberg (1991)

[2] Antoy, S., Brassel, B., Hanus, M.: Conditional narrowing without conditions. In: Proc. PPDP (2003), August 27-29, pp. 20–31. ACM Press, New York (2003)

[3] Baader, F., Nipkow, T.: Term rewriting and All That. Cambridge University Press, Cambridge (1998)

[4] Bergstra, J., Klop, J.: Conditional rewrite rules: Confluence and termination. Journal of Computer and System Sciences 32(3), 323–362 (1986)

[5] Braßel, B.: Bedingte Narrowing-Verfahren mit verzögerter Auswertung. Master's thesis, RWTH Aachen (1999)

[6] Şerbănuţă, T.-F., Roşu, G.: Computationally equivalent elimination of conditions. In: Pfenning, F. (ed.) RTA 2006. LNCS, vol. 4098, pp. 19–34. Springer, Heidelberg (2006)

[7] Dershowitz, N., Okada, M., Sivakumar, G.: Confluence of conditional rewrite systems. In: Kaplan, S., Jouannaud, J.-P. (eds.) CTRS 1987. LNCS, vol. 308, pp. 31–44. Springer, Heidelberg (1988)

[8] Dershowitz, N., Plaisted, D.: Equational programming. In: Hayes, J.E., Michie, D., Richards, J. (eds.) Machine Intelligence 11: The logic and acquisition of knowledge ch. 2, pp. 21–56 (1988)

[9] Giovanetti, E., Moiso, C.: Notes on the elimination of conditions. In: Kaplan, S., Jouannaud, J.-P. (eds.) CTRS 1987. LNCS, vol. 308, pp. 91–97. Springer, Heidelberg (1988)

[10] Lucas, S., Meseguer, J., Marché, C., Urbain, X.: Proving operational termination of membership equational programs. Higher-Order and Symbolic Computation 21(10), 59–88 (2008)

[11] Marchiori, M.: Unravelings and ultra-properties. Technical Report 8, University of Padova, Italy (1995)

[12] Marchiori, M.: Unravelings and ultra-properties. In: Hanus, M., Rodríguez-Artalejo, M. (eds.) ALP 1996. LNCS, vol. 1139, pp. 107–121. Springer, Heidelberg (1996)

[13] Nishida, N., Mizutani, T., Sakai, M.: Transformation for refining unraveled conditional term rewriting systems. ENTCS, vol. 174(10), pp. 75–95 (2007)

[14] Nishida, N., Sakai, M., Sakabe, T.: Partial inversion of constructor term rewriting systems. In: Giesl, J. (ed.) RTA 2005. LNCS, vol. 3467, pp. 264–278. Springer, Heidelberg (2005)

[15] Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer, Heidelberg (2002)

[16] Rosu, G.: From conditional to unconditional rewriting. In: Fiadeiro, J.L., Mosses, P.D., Orejas, F. (eds.) WADT 2004. LNCS, vol. 3423, pp. 218–233. Springer, Heidelberg (2005)

[17] Schernhammer, F., Gramlich, B.: On proving and characterizing operational termination of deterministic conditional rewrite systems. In: Hofbauer, D., Serebrenik, A. (eds.) Proc. WST (2007), pp. 82–85 (2007)

[18] Viry, P.: Elimination of conditions. J. Symb. Comput. 28(3), 381–401 (1999)