

Transforming CRSs into TRSs

—*About Elimination of the Conditions*—

Mauricio Ayala Rincón
 Departamento de Matemática, Universidade de Brasília
 ayala@mat.unb.br

Abstract

We study cases in which Conditional Rewriting Systems (CRSs) can be transformed into equivalent Non-conditional Rewriting Systems (TRSs). In particular, we propose a syntactic transformation for CRSs with built-in predicates and present restrictions guaranteeing preservation of the rewriting properties (viz derivability, termination, confluence, conservativeness) for the transformation.

1 Introduction

In the last years interest in applications and theory of rewriting has been increased as can be inferred from the collection of open problems in [DJK95]. Rewriting systems are attractive because of their simple syntax and semantics, which facilitate a satisfactory mathematical analysis. Additionally, rewriting systems play a fundamental rôle in algebraic specification, computing in algebraic structures and theorem proving.

Term rewriting systems (TRSs) appear first in universal algebra as an operational approach to deduction in equational classes (varieties). Of growing importance in the field of rewriting are conditional rewriting systems (CRSs). Conditional equational logic originated in universal algebra, from the need to deal with conditional equations for algebraic structures (quasivarieties), as for instance a transitivity law: $\{x = y \ \& \ y = z\} \Rightarrow x = z$.

Equational classes have been proved sufficiently expressive to describe any computable algebra (with a decidable word problem). Therefore conditional equations do not present further decidable algebras. Consequently, all functions specified with convergent CRSs can be specified with TRSs (including possibly some hidden functions). Conditional equations were studied in the field of abstract data types, not only because they provide easier and more elegant specifications, but additionally because they have a greater expressive power (the class of quasivarieties is strictly greater than the class of varieties). However, since all relevant properties of rewriting have been studied in detail for TRSs, it is very important to determine when a CRS can be transformed into an equivalent TRS syntactically.

Giovannetti and Moiso proposed a conservative transformation from CRSs into TRSs [GM87] which, in this work, we extend for the class of CRSs with built in predicates as premises examined in [Aya94], whose algebraic properties were studied in [Aya95].

CRSs with built-in predicates are CRSs with conditional rules whose premises combine standard conditions and built-in predicates that evaluate the boolean value of terms by some mechanism independent of rewriting. We can dispose of a rewriting mechanism to solve this class of predicates but, if there is a better procedure, we need not use rewriting.

Related work: One way to obtain a transformation of a convergent CRSs into an equivalent TRS is transforming the CRS into a Turing machine, which can be coded as a single unconditional rewrite rule as proved by Dauchet [Dau92]. Of course, working with such Turing machine encodings is impractical. To our knowledge Bergstra and Klop were the first introducing systematically a syntactic translation from CRSs into TRSs [BK86]. They proposed a transformation, enlarging the original signature with hidden functions, preserving reduction for terms in the original signature but unfortunately, failing in the conservativeness; i.e., new equational theorems in the

original signature could come arise. The first ones to present a translation and restrictions guaranteeing conservativeness were Giovannetti and Moiso. For our understanding the fundamental idea of their translation is due to the detailed algebraic treatment for translating the “if-then-else” command introduced by Guessarian in [Gue87]. Sivakumar [Siv89] introduced a non-conservative transformation appropriated for handling non-decreasing critical pairs during the completion process, which was incorporated in the *Rewrite Rule Laboratory* [KZ89]. More recently, Hintermeier examined in [Hin95b] and [Hin95a] a transformation from convergent CRSs into equivalent convergent TRSs concluding that any computable function that can be realized by a ground-confluent strictly terminating CRS can be realized by a ground-confluent terminating TRS. Hintermeier’s transformation is based on order sorted conditional rewriting used to describe equationally the application of a conditional rule. To our knowledge finding a conservative transformation for CRSs with built-in predicates has not been done yet, however the combination of conditional rewriting and built-in decision algorithms for well-known theories such as integer arithmetic and boolean algebras is extremely relevant in the context of rewriting, see, for example, [Vor89], [DO90], [Aya93], [Bec94].

This paper is organized as follows: in the second section we give the denotational conventions, present various formulations of conditional equations as rewriting systems, consider conditional rewriting with built-in predicates and extend some classical results on conditional rewriting and equational systems. In the third section, we present briefly the transformations we know from CRSs into TRSs. Subsequently, we examine the transformation for CRSs with built-in predicates and determine conditions guaranteeing preservation of the rewriting properties. We present detailed proofs of the preservation properties which could be abbreviated for a camera ready version of the paper.

2 Background

We use notations that are consistent with the standard ones in the field of rewriting. For the basic concepts and results on rewriting, we recommend three surveys, viz [AM90],[DJ90] and [Klo92] and Avenhaus book [Ave95]. We assume some familiarity with the main notions in algebraic specification (see for example [EM85], [Wec92]).

2.1 Unconditional rewriting

We recall the basic concepts and notations on rewriting.

A **signature** (S, Σ) consists of a finite set S of name of domains, called **sorts**, and a finite family Σ of name of operators, equipped with an arity function on S .

Let (S, Σ) be a signature. An (S, Σ) -Algebra A consists of a family of sets $(A_s)_{s \in S}$ and a family of operators $(f^A)_{f \in \Sigma}$ such that if $f : s_1, \dots, s_n \rightarrow s$ then $f^A : A_{s_1} \times \dots \times A_{s_n} \rightarrow A_s$.

Example 2.1 Consider the signature (S_0, Σ_0) : $S_0 = \{nat\}$ and $\Sigma_0 = \{0 : \rightarrow nat, succ : nat \rightarrow nat\}$. The standard model of naturals $(\mathbb{N}, 0, succ)$ is an (S_0, Σ_0) -Algebra. \diamond

As usually, T_Σ denotes the algebra of well-formed terms of the signature (S, Σ) . X represents a family $(X_s)_{s \in S}$ of countably infinite sorted sets of variables. $T_\Sigma(X)$ is the algebra of the terms with variables.

The length of a term t , $\lambda(t)$, is defined by: $\lambda(x) = 1$, if x is a variable; $\lambda(f(t_1, \dots, t_n)) = 1 + \sum_{i=1}^n \lambda(t_i)$, if f is an n -ary function and t_1, \dots, t_n its arguments.

The set of variables occurring in a term t is denoted by $V(t)$. **Positions** of a term consist of sequences of natural numbers and they are compared by the usual lexicographical ordering. The set of all positions of a term t is denoted by $O(t)$. The subterm of t at position $\pi \in O(t)$ is denoted by $t|_\pi$. If $s = t|_\pi$ then s is called the **occurrence** of t at position π . The result of replacing in t the subterm at position π by s is denoted by $t[\pi \leftarrow s]$ or simply by $t[s]_\pi$. $t[s]_\pi$ is also used to remark that s is the subterm of t occurring at position π .

A **substitution** σ is a mapping from X to $T_\Sigma(X)$ such that its domain, $\{x \in X \mid x\sigma \neq x\}$, is finite. $\sigma|_Y$ denotes the restriction of the substitution σ to the domain $Y \subseteq X$. The homomorphic extension of a substitution σ to a mapping from $T_\Sigma(X)$ to $T_\Sigma(X)$ is also denoted by σ .

A **term rewriting system** (TRS) over signature $T_\Sigma(X)$ is a set of ordered pairs (l, r) of terms in $T_\Sigma(X)$ or (rewrite) rules, denoted by $l \rightarrow r$, with $l \notin X$ and $V(r) \subset V(l)$.

Given a TRS R , the rewrite relation \rightarrow_R for terms $s, t \in T_\Sigma(X)$ is defined as follows: $s \rightarrow_R t$ if there exists a rule $l \rightarrow r$ in R , a substitution σ and a position $\pi \in O(s)$, such that $s|_\pi = l\sigma$ and $t = s[r\sigma]_\pi$. For brevity we write \rightarrow when R is clear from the context. The symmetric and transitive-reflexive closures of \rightarrow are denoted by \leftrightarrow and \rightarrow^* , respectively. Analogously, \leftrightarrow^* denotes the symmetric reflexive transitive closure of \rightarrow .

Two terms s, t are **joinable** in R , denoted by $s \downarrow t$, if there exists a term u with $s \rightarrow^* u \leftarrow^* t$, where \leftarrow denotes the inverse of \rightarrow and \leftarrow^* its transitive reflexive closure. A term s is **irreducible** or is a **normal form** if there is no term t with $s \rightarrow t$.

A TRS R is **terminating** if \rightarrow is noetherian, i.e. if there is no infinite reduction sequence $s_1 \rightarrow s_2 \rightarrow \dots$. A TRS R is **confluent** if $(\leftarrow^* \circ \rightarrow^*) \subseteq (\rightarrow^* \circ \leftarrow^*)$, where \circ denotes the relation composition. R is said to be **locally confluent** if $(\leftarrow \circ \rightarrow) \subseteq (\rightarrow^* \circ \leftarrow^*)$. A confluent and terminating TRS is called *convergent* or complete.

If $l_1 \rightarrow r_1, l_2 \rightarrow r_2$ are rules of a TRS R , $\pi \in O(l_1)$ and $l_1|_\pi$ and l_2 are unifiable with most general unifier σ , then the ordered pair of terms $((l_1[r_2]_\pi)\sigma, r_1\sigma)$ is said to be a **critical pair** of R (obtained by overlapping $l_2 \rightarrow r_2$ with $l_1 \rightarrow r_1$ at position π).

It is well-known that for TRSs local confluence is equivalent to joinability of all critical pairs. This result is known as the critical pair lemma, which, originally, was proved by Knuth-Bendix using termination hypothesis [KB70]. Subsequently, Huet obtained the final version of the lemma without termination hypothesis [Hue80].

A TRS R is said to be **non-overlapping** if there are no critical pairs between rules of R . A term is **linear** if every variable occurs at most once. A TRS is **left-linear** if every left hand side of its rules is linear. A left-linear and non-overlapping TRS, R , is said to be **orthogonal**. It is well-known that orthogonal TRSs are confluent (without being necessarily terminating).

2.2 Conditional rewriting with built-in predicates

A system E of conditional equations or equational Horn clauses over a signature (S, Σ) , is a finite set of formulas of the form $s_1 = t_1 \wedge \dots \wedge s_n = t_n \Rightarrow l = r$, where s_i, t_i for $i = 1, \dots, n$ and l, r are terms of $T_\Sigma(X)$ belonging pairwise to the same sorts. (E, Σ, S) (or simply E) is called an S -sorted specification.

A model of E is an (S, Σ) -Algebra A such that, for any instantiation $\sigma : X \rightarrow A$, if $\forall i \in [1..n]$ $(s_i\sigma)^A = (t_i\sigma)^A$ then $(l\sigma)^A = (r\sigma)^A$.

An equation $s = t$ is said to be a logical consequence of E if $s = t$ is valid in all models of E , i.e., if for any model A of E and any instantiation $\sigma : X \rightarrow A$, $(s\sigma)^A = (t\sigma)^A$.

The aim with conditional rewrite systems is to characterize the set of all equations that are logical consequences of a given system of equational Horn clauses. Kaplan [Kap84] has introduced the standard way to interpret the '=' in the conditions of equational Horn clauses in order to give an operational semantics to conditional rewrite rules. This corresponds to the intuition about conditional rewriting and allows for recursive evaluation of the premises.

We define inductively the one-step replacement relation \leftrightarrow and its reflexive-transitive closure \leftrightarrow^* as follows: if $s_1 = t_1 \wedge \dots \wedge s_n = t_n : l = r$ is a conditional equation in R , σ is a substitution, u is a term, $\pi \in O(u)$ and $s_i\sigma \leftrightarrow^* t_i\sigma$ for $1 \leq i \leq n$, then $u[l\sigma]_\pi \leftrightarrow u[r\sigma]_\pi$. If $s \leftrightarrow^* t$ we write $R \vdash s = t$, or just $s = t$.

A **standard CRS** (for brevity CRS) over signature (S, Σ) is a (finite) set of rules of the form $s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$, meaning that an instance $l\sigma$ of l rewrites to $r\sigma$ if each $s_i\sigma$ can be reduced to the same term as the corresponding $t_i\sigma$.

For a standard rewrite system R^{std} , let R^{eqn} denote the underlying equational system. A result in [DO88] establishes the equivalence between the joinability property in confluent standard CRSs and equality in equational systems as follow:

For any confluent standard CRS R^{std} , $R^{std} \vdash p \downarrow q$ iff $R^{eqn} \vdash p = q$.

From classical rewriting theory, we know that for confluent CRSs any equational proof $p \leftrightarrow^* q$ can be replaced by a **rewrite or normal proof**: $p \downarrow q$. Thus for any confluent system and normal form t , $p \leftrightarrow^* t$ implies $p \rightarrow^* t$. So rewriting with (terminating) confluent systems can be used to find normal forms.

A standard system is called **decreasing** if there exists a well-founded extension \succ of the rewrite relation \rightarrow which satisfies the following additional properties:

- \succ contains the proper subterm relation \triangleright (i. e. if s is a proper subterm of t then $t \succ s$).
- for each rule $s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$, $l\sigma \succ s_i\sigma, t_i\sigma$, for all substitutions σ and indices $1 \leq i \leq n$.

In general, decreasing systems do not admit extra-variables [Siv89].

A **conditional equational system with built-in predicates** is a set of clauses of the form: $P_1(\vec{u}_1) \& \dots \& P_m(\vec{u}_m) \wedge s_1 = t_1 \wedge \dots \wedge s_n = t_n : l = r$, where $P_1 \dots P_m$ are built-in predicates, $\vec{u}_1 \dots \vec{u}_m$ are terms or tuples of terms that do not contain built-in predicates and ‘&’ is a (new) symbol for the logical conjunction of built-in predicates. All equational conditions, $s_i = t_i, 1 \leq i \leq n$ are called **standard conditions**. We assume that a built-in predicate evaluates the boolean value of a term by some mechanism that is independent of rewriting. We define (standard) **CRSs with built-in predicates** as sets of conditional rules of the form $P_1(\vec{u}_1) \& \dots \& P_m(\vec{u}_m) \wedge s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$, where the applicability of the rules is decided by joinability of the standard conditions and validity of the built-in premises.

Different logical “and” (‘&’ and ‘ \wedge ’) are needed, because we suppose that the built-in mechanism can resolve conjunctions of built-in predicates so that they can be seen as single ones.

When explicit mention of the premises of a rule is unnecessary, we write $c : l \rightarrow r$.

In order to give a formal treatment some definitions and restrictions are needed.

In general, for two deduction systems R_0 and R_1 over signatures (S_0, Σ_0) and (S_1, Σ_1) , respectively, where $S_0 \subseteq S_1$ and $\Sigma_0 \subseteq \Sigma_1$, R_1 is called a **conservative extension** of R_0 , if for every formula ϕ which is expressed in Σ_0 , $R_1 \vdash \phi$ iff $R_0 \vdash \phi$. For equational systems this means that R_1 doesn't add new equations to R_0 .

Let R_0 be a conditional equational system over signature (S_0, Σ_0) and let R_1 be a conditional equational system with built-in predicates over signature (S_1, Σ_1) such that $R_1 \supseteq R_0$ and all built-in predicate symbols in $\Sigma_1 \setminus \Sigma_0$ have domain sorts of S_0 .

As for conditional equational systems, we define the one-step replacement relation \leftrightarrow and its reflexive-transitive closure \leftrightarrow^* as follows: If $P_1(\vec{u}_1) \& \dots \& P_m(\vec{u}_m) \wedge s_1 = t_1 \wedge \dots \wedge s_n = t_n : l = r$ is a conditional equation in R_1 , and σ is a substitution such that $t_i\sigma \leftrightarrow^* s_i\sigma$ for all $1 \leq i \leq n$ and for all $1 \leq j \leq m$ there exists \vec{u}'_j such that $R_0 \vdash \vec{u}_j\sigma = \vec{u}'_j$ and the built-in mechanism evaluates $P_j(\vec{u}'_j)$ as true then $u[l\sigma] \leftrightarrow u[r\sigma]$. If $s \leftrightarrow^* t$ we write $R_1 \vdash s = t$, or just $s = t$.

Let R_1 and R_0 be as before. R_1 is said to be **conservative** if for all terms s, t of sort in S_0 , $R_1 \vdash s = t$ iff $R_0 \vdash s = t$.

Example 2.2 Consider the conditional equational system $R_0 = \emptyset$ over the signature (S_0, Σ_0) of example 2.1. Let $S_1 = S_0 \cup \{\text{bool}, \text{list}\}$ and $\Sigma_1 = \Sigma_0 \cup \{\text{true}, \text{false} : \rightarrow \text{bool}, \text{nil} : \rightarrow \text{list}, \cdot : \text{nat} \times \text{list} \rightarrow \text{list}, \text{sort} : \text{list} \rightarrow \text{bool}, \text{less} : \text{nat} \times \text{nat}\}$. Let R_1 be the following set of built-in conditional equations over signature (S_1, Σ_1) :

$\text{sort}(\text{nil}) = \text{true}$	$\text{less}(x, y) : \text{sort}(x \cdot y \cdot L) = \text{sort}(y \cdot L)$
$\text{sort}(x \cdot \text{nil}) = \text{true}$	$\text{less}(y, x) : \text{sort}(x \cdot y \cdot L) = \text{false}$
	$x = y : \text{sort}(x \cdot y \cdot L) = \text{false}$

Obviously R_1 is conservative. Usually, by purely rewriting, we define a boolean predicate $<$ with the usual rules: $\text{succ}(x) < 0 \rightarrow \text{false}, \text{succ}(x) < \text{succ}(y) \rightarrow x < y, 0 < \text{succ}(x) \rightarrow \text{true}, 0 < 0 \rightarrow \text{false}$ intending to represent the natural numbers with zero, successor and predicate less. The applicability of the conditional rules can be decided more efficiently with a built-in mechanism for the predicate ‘<’ (for instance, *counting the number of s's*). In the example ‘<’ is replaced with *less* representing a built-in predicate (or mechanism), which evaluates truth values of instances of the premises more quickly than the original rewriting mechanism. \diamond

¹If $\vec{u}_j = (w_1, \dots, w_{k_j})$ and $\vec{u}'_j = (w'_1, \dots, w'_{k_j})$ then $R_0 \vdash \vec{u}_j\sigma = \vec{u}'_j$ is an abbreviation for $R_0 \vdash w_l\sigma = w'_l$ for all $1 \leq l \leq k_j$.

Let p be a new predicate in $\Sigma_1 \setminus \Sigma_0$. We suppose that the known built-in mechanism evaluates $p(t_1, \dots, t_n)$ correctly in the sense that if there is a tuple (u_1, \dots, u_n) of terms such that for all $i = 1, \dots, n$, $R_0 \vdash t_i = u_i$, then the evaluation of $p(t_1, \dots, t_n)$ and of $p(u_1, \dots, u_n)$ coincide.

Example 2.3 Consider the predicate *less* of example 2.2 and let t_1, t_2 be ground terms of sort *nat*. The rewriting system first executes *less*(t_1, t_2) to reach normal forms \bar{t}_1 and \bar{t}_2 , then the built-in mechanism evaluates *less*(\bar{t}_1, \bar{t}_2). Note that in some cases the built-in mechanism can evaluate a truth value for predicates with arguments which are not in ground form, for example the built-in mechanism can be strong enough to evaluate the conjunction *less*(t_1, t_2) & *less*(t_2, t_1) to false for all terms t_1, t_2 of sort *nat*. \diamond

In the rest of this work we consider only monadic built-in predicates. Generalizations of our definitions and results to n-ary predicates are trivial. Often we will omit arguments of the built-in predicates.

A CRS with built-in predicates is **decreasing** if there exists a well-founded extension \succ of the rewrite relation \rightarrow which satisfies:

- \succ contains the proper subterm relation \triangleright (i. e. if s is a proper subterm of t then $t \succ s$).
- for each rule $P_1(u_1) \& \dots \& P_m(u_m) \wedge s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$, $l\sigma \succ s_i\sigma, t_i\sigma, u_j\sigma$, for all substitutions σ and indices $1 \leq i \leq n$ and $1 \leq j \leq m$.

As consequence of the well-foundness of the ordering \succ , decreasing CRSs are terminating. In general (one sorted case), decreasingness restricts all variables in the condition of a rule to appear on the left hand side of the conclusion too. For simplicity, we write $l\sigma \succ P_j\sigma$ omitting the argument u_j .

Detailed proofs of the subsequent theorems appear in [Aya94].

The basic notions of rewriting for decreasing CRSs are decidable, as states the following theorem.

Theorem 2.1 *Let R be a decreasing standard CRS with built-in predicates. Then for any terms s, t , one-step reduction (“does $s \rightarrow t$ hold?”), finite reduction (“does $s \rightarrow^* t$ hold?”), joinability (“does $s \downarrow t$ hold?”) and normal form property (“is s irreducible?”) are all decidable.*

With the current assumption for built-in predicates we can easily extend some results of the classical theory that relate equational systems with standard CRSs.

Theorem 2.2 *For any confluent CRS R^{std} with built-in predicates:*

$$R^{std} \vdash p \downarrow q \text{ iff } R^{eqn} \vdash p = q.$$

The rewrite relation \rightarrow , the derivability relation \rightarrow^* , the joinability relation \downarrow and the normal form property are all decidable for decreasing CRSs as well as for decreasing CRSs with built-in predicates.

Decreasing CRSs with built-in predicates capture the finiteness of recursive evaluations of terms, in the following operational sense. For a given CRS R , let \rightsquigarrow be the relation defined by $t \rightsquigarrow t'$ if there is a rule in R , $P_1(u_1) \& \dots \& P_m(u_m) \wedge s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$ and a substitution σ such that $l\sigma = t|_\pi$ and t' is one of the $s_i\sigma, t_i\sigma$ for $1 \leq i \leq n$, or $u_j\sigma$ for $1 \leq j \leq m$. The relation $\rightarrow \cup \rightsquigarrow$ corresponds to one step of computation (we exclude the built-in evaluation from the computation) and its transitive closure $(\rightarrow \cup \rightsquigarrow)^+$ represents an arbitrary computation branch.

Theorem 2.3 *For any CRS R with or without built-in predicates, the relation $(\rightarrow \cup \rightsquigarrow)^+$ is well-founded iff R is decreasing.*

3 Transforming CRSs into TRSs

We consider syntactic transformations from CRSs into TRSs.

In general there are more than one equation in the condition of an equational Horn clause, but by using new non interpreted function symbols or *hidden functions* it is possible to represent every condition by only one equation. Let E be a set of equational Horn clauses over $T_\Sigma(X)$.

Let "true" be a new constant symbol; let "eq" and "&" be new binary function symbols. Let $\Sigma' = \Sigma \cup \{true, \&\} \cup \{eq_s \mid s \in S\}$. E can be transformed into a set E' of equational Horn clauses over $T_{\Sigma'}(X)$, where the conditions are conformed by only one equation of the form $p = true$ as follows: each clause $u_1 = v_1 \wedge \dots \wedge u_n = v_n \Rightarrow s = t$ is replaced with $(eq_{s_1}(u_1, v_1) \& \dots \& eq_{s_n}(u_n, v_n)) = true \Rightarrow s = t$, where $s_i \in S$ is the sort of terms $u_i, v_i, i = 1, \dots, n$. Moreover, E' contains new nonconditional equations:

$$eq_s(x, x) = true, \quad \forall s \in S$$

$$x \& true = x$$

Of course, the new function symbol "&" is associative and commutative.

It is easy to see that E' is a conservative extension of E .

The words "equivalent" and "correct" are often used making reference to conservative extensions.

In the sequel we consider only equational Horn clauses (with built-in predicates), which contain a sole equational condition.

It is possible to simulate implications using pure equational clauses by means of new function symbols " $if_s, s \in S$ " including the following axioms:

$$if_s(true, x) = x, \quad \forall s \in S$$

By simplicity, only new symbols "eq" and "if" without sort subindices are used.

Consider the transformation of equational Horn clauses of the form $u = v \Rightarrow s = t$ into the clause $s = if(eq(u, v), t)$. This elementary transformation is incorrect as illustrates the following example.

Example 3.1 Let $E = \{a = b \Rightarrow c = d, a = b \Rightarrow e = d\}$. Consider its corresponding transformation, E' :

$eq(x, x) = true$	$c = if(eq(a, b), d)$
$if(true, x) = x$	$e = if(eq(a, b), d)$

Note that $E' \vdash c = e$ but not $E \vdash c = e$. In fact, $c = if(eq(a, b), d) = e$ is a proof in E' but there is no proof of $c = e$ in E . ◇

The suggested transformation can be modified to obtain a conservative set of equational clauses for any set of equational Horn clauses. It can be achieved replacing every conditional clause of the form $u = v \Rightarrow s = t$ with the nonconditional clause $if(eq(u, v), s) = if(eq(u, v), t)$.

Example 3.2 (Continuing previous example) The above transformation gives the following set of clauses for the example, E'' :

$eq(x, x) = true$	$if(eq(a, b), c) = if(eq(a, b), d)$
$if(true, x) = x$	$if(eq(a, b), e) = if(eq(a, b), d)$

Note that $E'' \vdash if(eq(a, b), c) = if(eq(a, b), e)$ holds but $E'' \vdash c = e$ does not. ◇

The previous transformation generates a conservative extension always. Unfortunately, in general this transformation is not appropriate from the viewpoint of the generation of convergent rewriting systems. However, this transformation results useful when some conditional rules cannot be manipulated by completion techniques [Siv89], as is made frequently by the well-known *Rewrite Rule Laboratory (RRL)* system of Zhang and Kapur [KZ89].

Without hidden functions it is impossible to generate a finite conservative extension for any set of equational Horn clauses. Moreover, restricting the premises to be decidable, this is impossible too (this was contrarily conjectured by Classen in [Cla88]). The following counterexample, originally presented in [Aya93], illustrates this fact.

Example 3.3 Consider the following set of equational Horn clauses

$$E = \{f(g(x)) = c, f(h(g(x))) = c, f(h(x)) = f(x) \Rightarrow f(h(h(x))) = f(h(x))\}$$

The conditional clause has a decidable premise; in fact, observe that $f(h(y)) = f(y)$ if and only if $y \in \{h^n(g(x)) \mid n \geq 0\}$.

An infinite equational system, E' , equivalent to E consists of the following set of axioms:

$$\{f(h^n(g(x))) = c \mid n \geq 0\},$$

where h^n abbreviates n -composition of h .

The original signature can be extended with a copy of the natural numbers obtaining a finite presentation of the theory, but we search for an equivalent equational system without hidden functions.

To show that there does not exist a finite equational system equivalent to E , assume contrarily the existence of one, say $E'' = \{s_1 = r_1, \dots, s_n = r_n\}$, where, $s_i, r_i, i = 1, \dots, n$ are terms in the original signature.

Let $f(h^k(g(x)))$ be a term longer than each s_i and r_i ; i.e., $\lambda(f(h^k(g(x)))) > \max(\{\lambda(s_i) \mid 1 \leq i \leq n\} \cup \{\lambda(r_i) \mid 1 \leq i \leq n\})$. By assumption, $E'' \vdash f(h^k(g(x))) = c$. But $f(h^k(g(x)))$ cannot be simplified applying equations of E'' , because with equations conformed with terms shorter than $f(h^k(g(x)))$, only three cases of simplification are possible: either subterms of the form $f(h^j(x))$ or $h^j(x)$ or $h^j(g(x))$ for $1 \leq j < k$ of $f(h^k(g(x)))$ can be simplified. These three cases of simplification are impossible since the equivalence classes of $f(h^j(c))$, $h^j(x)$ and $h^j(g(x))$ wrt E are singletons. \diamond

Now we consider the transformation for CRSs suggested by Bergstra and Klop [BK86].

Every rule $s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$ in R is replaced with two rules of the form:

$$l \rightarrow \delta(s_1, t_1, \dots, s_n, t_n, r), \quad \delta(x_1, x_1, \dots, x_n, x_n, x) \rightarrow x$$

In this way a new unconditional rewriting system R' is obtained. Obviously, $s \rightarrow_{R'}^* t$ implies $s \rightarrow_R^* t$. Unfortunately this transformation is not conservative as can be seen in the following example.

Example 3.4 Consider the following CRS:

$$R = \{h(y) \downarrow a : f(x, y) \rightarrow g(x), \quad h(d) \rightarrow h(c)\}$$

The transformation gives the following unconditional rewriting system:

$$R' = \{f(x, y) \rightarrow \delta(h(y), a, g(x)), \quad \delta(y, y, x) \rightarrow x, \quad h(d) \rightarrow h(c)\}$$

Note that $f(x, c) \leftrightarrow_{R'}^* f(x, d)$ but not $f(x, c) \leftrightarrow_R^* f(x, d)$. In fact, $f(x, c) \rightarrow_{R'} \delta(h(c), a, g(x))$ and $\delta(h(c), a, g(x)) \xrightarrow{R'} \delta(h(d), a, g(x)) \xrightarrow{R'} f(x, d)$. \diamond

The above suggested transformation does not preserve left-linearity of the original CRS. Consequently, the transformation of a confluent orthogonal CRSs could give rise to a non confluent TRS as the following example illustrates.

Example 3.5 Consider the following orthogonal CRS:

$$R = \{g(x) \downarrow r(x) : f(x) \rightarrow a, r(x) \rightarrow g(x), \quad b \downarrow a : g(x) \rightarrow b\}$$

By simple inspection R can be proved convergent. The transformation gives the following TRS:

$$R' = \{f(x) \rightarrow \delta_f(g(x), r(x), a), \delta_f(x, x, y) \rightarrow y, r(x) \rightarrow g(x), g(x) \rightarrow \delta_g(b, a, b), \delta_g(x, x, y) \rightarrow y\}$$

R' is not convergent. In fact, $f(x) \rightarrow_{R'} \delta_f(g(x), r(x), a) \rightarrow_{R'} \delta_f(g(x), g(x), a) \rightarrow_{R'} a$ and $f(x) \rightarrow_{R'} \delta_f(g(x), r(x), a) \rightarrow_{R'} \delta_f(\delta_g(b, a, b), r(x), a) \rightarrow_{R'} \delta_f(\delta_g(b, a, b), g(x), a)$. \diamond

A simple restriction proposed in [BK86] in order to preserve left-linearity of the original CRS is to work with **normal CRSs**, which are CRSs consisting of rules of the form $s_1 \downarrow t_1 \wedge \dots \wedge s_n \downarrow t_n : l \rightarrow r$, for all whose conditions $s_i \downarrow t_i$, t_i is a ground normal form. A *normal rule*, as the preceding one, is replaced with the following two rules:

$$l \rightarrow \delta(s_1, \dots, s_n, r), \delta(t_1, \dots, t_n, x) \rightarrow x$$

In this way, from a normal CRS, R , it is obtained a TRS R' , which preserves the reduction but which is not necessarily a conservative extension of R (example 3.4 applies). Moreover, since orthogonal TRSs and normal orthogonal CRSs are confluent and the translation R' preserves orthogonality of the original normal CRS R , R' is also confluent. Note that orthogonal CRSs are not necessarily confluent, as illustrates the following example.

Example 3.6 Consider the following orthogonal standard CRS:

$$R = \{x \downarrow g(x) : g(x) \rightarrow a, b \rightarrow g(b)\}$$

Note that $g(g(b))$ rewrites to a and $g(a)$, since $g(b) \downarrow g(g(b))$ and $b \downarrow g(b)$. \diamond

4 Transforming CRSs with built-in predicates into TRSs

The transformation of Giovannetti and Moiso [GM87] for CRSs and its correctness restrictions are extended for CRSs with built-in predicates. Without loss of the generality, conditional rules with only one standard condition and a built-in premise will be considered.

Definition 4.1 Let R be a CRS with built-in predicates. R is called **transformable** if for all rules $P \wedge u \downarrow v : l \rightarrow r$ in R , $V(P), V(u) \subseteq V(l)$, $V(l) \cap V(v) = \emptyset$ and $V(r) \subseteq V(l) \cup V(v)$.

Subsequently, we describe the translation of a CRS with built-in predicates R into a TRS R_c .

The precise algebraic treatment for translating the “if-then-else” into equational axioms presented by Guessarian [Gue87] and its extension to the “case-of” motivate our transformation². A CRS with built-in predicates, R , is canonically partitioned into a family of sets $R = R_1 \cup \dots \cup R_q \cup U_R$, where U_R is the set of unconditional rules in R and each R_j , for $j = 1, \dots, q$ consists of a maximal set of rules with the same left hand side in the conclusion and in the standard condition, i. e. a set of rules of the form:

$$\{P_1 \wedge u_j \downarrow v_1 : l_j \rightarrow r_1, P_2 \wedge u_j \downarrow v_2 : l_j \rightarrow r_2, \dots, P_{|R_j|} \wedge u_j \downarrow v_{|R_j|} : l_j \rightarrow r_{|R_j|}\}.$$

If a rule has no built-in predicate, then this is considered as the empty predicate, which is valid. A new boolean sort, *bool* and boolean constants *true* and *false* are included. In order to separate the built-in logic from the rest of the specification a new operator D is used. For a built-in predicate, P , $D[P]$ is *true* or *false* depending on the logical value of P . Deciding a predicate will be considered as a reduction step in the new system R_c . For every set R_j of the partition it is defined a new operator symbol *if_j*. R_c , consists of the union of sets of unconditional rules $U_1 \cup \dots \cup U_q \cup U_R$, where U_j , for $j = 1, \dots, q$, is obtained from R_j as follows:

$$\boxed{\begin{array}{l} P_1 \wedge u_j \downarrow v_1 : l_j \rightarrow r_1, \\ P_2 \wedge u_j \downarrow v_2 : l_j \rightarrow r_2, \\ \vdots \\ P_{|R_j|} \wedge u_j \downarrow v_{|R_j|} : l_j \rightarrow r_{|R_j|} \end{array}} \rightsquigarrow \boxed{\begin{array}{l} l_j \rightarrow \text{if}_j(x_1, \dots, x_{n_j}, u_j, D[P_1], D[P_2], \dots, D[P_{|R_j|}]) \\ \text{if}_j(x_1, \dots, x_{n_j}, v_1, \text{true}, y_2, \dots, y_{|R_j|}) \rightarrow r_1, \\ \text{if}_j(x_1, \dots, x_{n_j}, v_2, y_1, \text{true}, \dots, y_{|R_j|}) \rightarrow r_2, \\ \vdots \\ \text{if}_j(x_1, \dots, x_{n_j}, v_{|R_j|}, y_1, y_2, \dots, \text{true}) \rightarrow r_{|R_j|} \end{array}}$$

²The definition $f(\vec{x}) \equiv \text{if } p(\vec{x}) \text{ then } r(\vec{x}) \text{ else } s(\vec{x})$, where \vec{x} is a list of variables, can be replaced with the following set of unconditional rules: $\{f(\vec{x}) = f'(\vec{x}, p(\vec{x})), f'(\vec{x}, \text{true}) = r(\vec{x}), f'(\vec{x}, \text{false}) = s(\vec{x})\}$. Analogously, a “case-of” instruction of the form $l = \text{case } u \text{ of } v_1 : r_1; \dots; v_n : r_n$ can be replaced with the following set of conditional rules: $\{u \downarrow v_1 : l \rightarrow r_1, \dots, u \downarrow v_n : l \rightarrow r_n\}$. Subsequently, the last set is replaced with the following set of unconditional rules: $\{l \rightarrow \text{if}(V(l), u), \text{if}(V(l), v_1) \rightarrow r_1, \dots, \text{if}(V(l), v_n) \rightarrow r_n\}$.

where $V(l_j) = \{x_1, \dots, x_{n_j}\}$ and $y_1, y_2, \dots, y_{|R_j|}$ are boolean variables.

The signature of R_c should be restricted admitting only terms with root symbol if_j with final sequences of boolean arguments of the form $D[P_1]\sigma, D[P_2]\sigma, \dots, D[P_{|R_j|}]\sigma$ and their derivations. In this way we avoid trivial divergence; namely, $r_1 \xrightarrow{R_c} if_j(x_1, \dots, x_{n_j}, true, true, \dots, true) \xrightarrow{R_c} r_2$, where R_j corresponds to a set of rules without standard condition.

To guarantee preservation of the rewriting properties, we give additional restrictions.

Definition 4.2 A transformable CRS with built-in predicates, R , is called **quasi-normal** if and only if for every conditional rule $P \wedge u \downarrow v : l \rightarrow r$ and every rule (conditional or nonconditional) $c : l' \rightarrow r'$ in R , v is linear and if d is a nonvariable subterm of v then d and l' do not unify.

Lemma 4.1 Let R be a quasi-normal CRS with built-in predicates. Then for every rule $P \wedge u \downarrow v : l \rightarrow r$ in R and every substitution σ , if $u\sigma \downarrow v\sigma$ then there exists a substitution $\delta : V(l) \cup V(r) \rightarrow T_\Sigma(X)$, such that $\sigma|_{V(l)} = \delta|_{V(l)}$ and $u\delta \rightarrow^* v\delta$.

Proof: Suppose that $u\sigma \rightarrow^* z \leftarrow^* v\sigma$. By the condition over variables for transformable standard CRSs, $V(u) \cap V(v) = \emptyset$. By quasi-normality of R , the derivation $v\sigma \rightarrow^* z$ does not change the structure of v , i.e. $z = v\sigma'$ for some substitution $\sigma' : V(v) \rightarrow T_\Sigma(X)$. Finally, the substitution $\delta = \sigma|_{V(l)} \cup \sigma'$ satisfies $u\delta = u\sigma \rightarrow^* z = v\delta$. \square

Observe that decreasing quasi-normal CRSs cover the class of normal CRSs.

Definition 4.3 Let R be a CRS with decidable built-in predicates as conditions. R is said to be **safely transformable** if and only if it is decreasing, confluent, quasi-normal and **conditionally superposition free** which means:

- If $P \wedge u_j \downarrow v : l_j \rightarrow r$ and $P' \wedge u_j \downarrow v' : l_j \rightarrow r'$ are different rules in R_j (for some $j \in \{1, \dots, q\}$) and σ is a unifier of v and v' then $D[P\sigma] \wedge D[P'\sigma] = \text{false}$.
- If $P \wedge u_j \downarrow v : l_j \rightarrow r$ and $P' \wedge u_i \downarrow v' : l_i \rightarrow r'$ are rules in R_j and R_i , respectively, with $i \neq j$, then l_j and l_i do not overlap.
- Conditional and nonconditional rules of R do not overlap.

The transformation R_c of a safely transformable CRSs with built-in predicates, R , preserves reduction, convergence and joinability, as states the following theorem.

Theorem 4.1 (Conservation for CRSs with built-in predicates) Let R be a safely transformable CRS with built-in predicates. Then its transformation R_c (as was above described) is convergent and satisfies the following preservation properties:

- R_c preserves reduction, i.e. for all pair of terms s, t in the signature of R , if $t \xrightarrow{*}_R s$ then $t \xrightarrow{*}_{R_c} s$.
- Let s, t be terms in the signature of R , then $t \downarrow_R s$ if and only if $t \downarrow_{R_c} s$.

Proof: Firstly, we prove that R_c preserves reduction. Let s, t be terms in the original signature such that $t \rightarrow_R s$; it will be proved by transfinite induction that $t \xrightarrow{*}_{R_c} s$. Suppose that for every term t' smaller than t , if $t' \rightarrow_R^* s'$ then $t' \xrightarrow{*}_{R_c} s'$. If $t \rightarrow s$ by applying a rule at position π then $t|_\pi \rightarrow s|_\pi$ and by induction hypothesis $t \rightarrow_{R_c} s$, since $t \succ t|_\pi$. Suppose $t \rightarrow_R s$ by applying a rule at top position. We should consider two cases as follows:

- On one side, if $t \rightarrow_R s$ by applying a rule in U_R , then $t \rightarrow_{R_c} s$, because $U_R \subseteq R_c$.
- On the other side, suppose that $t \rightarrow_R s$ by applying a conditional rule in R_j , $P_i \wedge u_j \downarrow v_i : l_j \rightarrow r_i$, with substitution σ . Then $t \equiv l_j\sigma$, $P_i\sigma$ holds or more exactly, $P_i\sigma \rightarrow_R^* P'_i$ and $D[P'_i] = \text{true}$, $u_j\sigma \downarrow v_i\sigma$ and $r_i\sigma \equiv s$. R_c includes the subset of rules U_j corresponding to the set R_j , which includes the following two rules:

$$l_j \rightarrow if_j(\vec{x}_j, u_j, D[P_1], \dots, D[P_i], \dots, D[P_{|R_j|}]) \text{ and } if_j(\vec{x}_j, v_i, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$$

where \bar{x}_j abbreviates x_1, \dots, x_{n_j} . By lemma 4.1 there exists a substitution δ with domain $V(v_i) \cup V(l_j)$ such that $u_j \delta \rightarrow_R^* v_i \delta$. By induction hypothesis $u_j \delta \rightarrow_{R_c}^* v_i \delta$ because $t \succ l_j \delta \succ u_j \delta$ and $P_i \sigma \equiv P_i \delta \rightarrow_{R_c}^* P_i' \delta$ because $t \succ P_i \sigma \succ P_i' \delta$. Then we can build the following R_c -derivation: $t \equiv l_j \delta \rightarrow_{R_c} i f_j(\bar{x}_j \delta, u_j \delta, D[P_1 \delta], \dots, D[P_i \delta], \dots, D[P_{|R_j|} \delta]) \rightarrow_{R_c}^* i f_j(\bar{x}_j \delta, v_i \delta, D[P_1 \delta], \dots, true, \dots, D[P_{|R_j|} \delta]) \rightarrow_{R_c} r_i \delta \equiv s$.

We conclude that for all terms s, t in the original signature, $t \rightarrow_R^* s$ implies $t \rightarrow_{R_c}^* s$.

Secondly, we prove the convergence of R_c . We prove termination and subsequently confluence.

To prove termination, we extend the decreasingness ordering \succ to the new signature. For all terms, t , and built-in predicates, P , if $t \succ P$ then $t \succ D[P]$ and $D[P] \succ P \succ true, false$. The multiset ordering³ \succ of the extension is used to compare terms in the signature of R_c . Two terms in the new signature, say s and t , are compared according to their associated multisets, $MS(s) \succ MS(t)$, defined as follows. For all terms, t , in the initial signature, it is associated the multiset $MS(t) = \{\{t, t\}\}$; for terms of the form $i f_j(t_1, \dots, t_{n_j}, s, w_1, \dots, w_{|R_j|})$, the multiset $\{\{l_j[x_1 \leftarrow t_1, \dots, x_{n_j} \leftarrow t_{n_j}]\} \cup MS(s) \cup MS(w_1) \cup \dots \cup MS(w_{|R_j|})\}$; for terms of the form $D[P]$, it is associated the multiset $\{\{D[P]\}\}$; for $true$ and $false$, $\{\{true\}\}$ and $\{\{false\}\}$ respectively; for terms of the form $t[\dots i f_j(t_1, \dots, t_{n_j}, s, w_1, \dots, w_{|R_j|}) \dots]$, the multiset $\{\{t[\dots l_j[x_1 \leftarrow t_1 \dots x_{n_j} \leftarrow t_{n_j}] \dots], MS(i f_j(t_1, \dots, t_{n_j}, s, w_1, \dots, w_{|R_j|}))\}$. Since \succ is a well-founded extension of \rightarrow_R , \succ is a well-founded extension of \rightarrow_{R_c} , which, in addition, is compatible with the original ordering and satisfies the subterm property in the signature of R_c . Moreover, for all substitutions, σ , the rules of R_c are well-oriented wrt \succ . In fact, for all rules $l \rightarrow r \in U_R$ and substitutions σ , $l \sigma \succ r \sigma$ implies $\{\{l \sigma, l \sigma\}\} \succ \{\{r \sigma, r \sigma\}\}$; For all rules of the form $l_j \rightarrow i f_j(\bar{x}_j, u_j, D[P_1], \dots, D[P_i], \dots, D[P_{|R_j|}])$ and substitutions σ , $\{\{l_j \sigma, l_j \sigma\}\} \succ \{\{l_j \sigma, u_j \sigma, D[P_1 \sigma], \dots, D[P_{|R_j|} \sigma]\}\}$ (because $l_j \sigma \succ u_j \sigma, P_1 \sigma, \dots, P_{|R_j|} \sigma$); for all rules of the form $i f_j(\bar{x}_j, v_i, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$ and substitutions σ , $\{\{l_j \sigma, u_i \sigma, y_1 \sigma, \dots, true, \dots, y_{|R_j|} \sigma\}\} \succ \{\{r_i \sigma, r_i \sigma\}\}$ (because $l_j \sigma \succ r_i \sigma$). Consequently, R_c is decreasing and hence terminating.

To prove that R_c is confluent, it is enough to show joinability of its critical pairs, since R_c has been proved to be terminating and, consequently, the critical pair lemma implies its confluence. Since R is conditionally superposition free and quasi-normal, there are no possible critical pairs between rules of the following forms:

- $l \rightarrow r$ and $l_j \rightarrow i f_j(\bar{x}_j, u_j, D[P_1], \dots, D[P_{|R_j|}])$ (because there are no overlaps between rules in U_R and R_j);
- $l_j \rightarrow i f_j(\bar{x}_j, u_j, D[P_1], \dots, D[P_{|R_j|}])$ and $l_i \rightarrow i f_i(\bar{x}_i, u_i, D[P_1], \dots, D[P_{|R_i|}])$ (because there are no overlaps between rules in R_j and R_i);
- $i f_j(\bar{x}_j, v_i, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$ and rules in $U_R \cup \bigcup_{j=1}^q R_j$, (because there are no overlaps between rules in R and right-hand sides of the standard conditions of rules in R);
- $i f_j(\bar{x}_j, v_i, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$ and $i f_k(\bar{x}_k, v_k, y_1, \dots, true, \dots, y_{|R_k|}) \rightarrow r_k$, $j \neq k$.

Additionally, by restriction on the signature of R_c there are no possible critical pairs between rules of the form $i f_j(\bar{x}_j, v_j, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$ and $i f_j(\bar{x}_j, v_k, y_1, \dots, true, \dots, y_{|R_j|}) \rightarrow r_i$ with $j \neq k$. Consequently, only critical pairs from nonconditional rules are possible. Suppose $\langle s, t \rangle$ is a critical pair from overlapping rules in U_R . By confluence hypothesis, $s \rightarrow_R^* w \xrightarrow{R_c}^* t$, for some w in the signature of R . By preservation of the reduction $s \rightarrow_{R_c}^* w \xrightarrow{R_c}^* t$.

Finally, we prove that for terms in the original signature $t \downarrow_R s$ if and only if $t \downarrow_{R_c} s$.

$t \downarrow_R s$ implies $t \downarrow_{R_c} s$ because of preservation of the reduction. Conversely, $t \downarrow_{R_c} s$ implies $t \downarrow_R s$ is not trivial because if $t \rightarrow_{R_c}^* u \xrightarrow{R_c}^* s$ then u could be a term of the extended signature.

³Multisets orderings were first proposed in [DM79] and are extensively applied to prove termination of rewriting systems. Intuitively, a multiset M over a set S is an unordered collection of elements of S , with possibly repetitions. More formally, a multiset M is a mapping from S to \mathbb{N} associating with each element in S the number of its occurrences in M . We enclose multisets in double parenthesis ($\{\{\}\}$), to distinguish from sets. We will use the well-known fact that multiset orderings created from well-founded orderings are well-founded too.

We can suppose that u is an R_c -normal form, because R_c is convergent. For example, u could contain R_c -irreducible subterms of the form $if_j(\bar{x}_j\sigma, u_j\sigma, false, \dots, false)$ which have been derived from $l_j\sigma$ for some substitution σ , for which none of the σ -instances of the premises $P_1, \dots, P_{|R_j|}$ hold (or more exactly, the σ -instance of the corresponding premise does not hold). We define a function n_R , that is used to convert R_c -irreducible terms into R -irreducible terms. We denote the R - and R_c -normal forms by $NF_R(t)$ and $NF_{R_c}(t)$, respectively. The function n_R decodes normal forms of R_c into the corresponding normal forms of R ; i.e., n_R transforms subterms of the form $if_j(\bar{x}_j\sigma, u_j\sigma, false, \dots, false)$ into $l_j\sigma$. If u is a term in the original signature then $NF_R(u) = n_R(NF_{R_c}(u))$. Suppose that the terms s and t in the original signature converge to the R_c -normal form u ; i.e. $s \rightarrow_{R_c}^* u \xleftarrow{*}_{R_c} t$. Then $s \rightarrow_{R_c}^* n_R(u) \xleftarrow{*}_{R_c} t$. Consequently, $t \rightarrow_R^* n_R(u) \xleftarrow{*}_R s$ \square

Remark: Note that the convergence property of R_c and the second property of the theorem imply that R_c is a conservative extension of R . \bullet

Obviously confluence and decreasingness of R and decidability of the built-in predicates are necessary conditions to guarantee the conservation theorem. The other conditions of safely transformable CRS are also necessary as it is illustrated in the following examples.

Example 4.1 Consider the CRS $R = \{c \downarrow 0 : f(x, a) \rightarrow 0, d \downarrow 1 : f(b, x) \rightarrow 1\}$.

Observe that R is confluent. However, note that R is not conditionally superposition free, because $f(x, a)$ and $f(b, x)$ overlap. The transformation gives the following divergent TRS:

$$R_c = \{f(x, a) \rightarrow if_1(x, c), if_1(x, 0) \rightarrow 0, f(b, x) \rightarrow if_2(x, d), if_2(x, 1) \rightarrow 1\}$$

In fact, $f(b, a)$ has two normal forms; namely, $if_1(b, c) \xleftarrow{*}_{R_c} f(b, a) \xrightarrow{*}_{R_c} if_2(a, d)$.

Analogously, consider the CRS with built-in predicates in the theory of integers with the usual predicate \leq , $R = \{x \leq 0 : f(x, a) \rightarrow 0, 1 \leq x : f(b, x) \rightarrow 1\}$.

As before, R is not safely transformable. The transformation gives the following TRS:

$$R_c = \{f(x, a) \rightarrow if_1(x, D[x \leq 0]), if_1(x, true) \rightarrow 0, f(b, x) \rightarrow if_2(x, D[1 \leq x]), if_2(x, true) \rightarrow 1\}$$

$f(a, b)$ diverges. In fact, $if_1(b, D[b \leq 0]) \xleftarrow{*}_{R_c} f(b, a) \xrightarrow{*}_{R_c} if_2(a, D[1 \leq a])$. \diamond

The following example combines also built-in predicates and standard conditions.

Example 4.2 Consider the CRS with built-in predicates in the theory of integers with the usual predicate \geq , $R = \{d \downarrow c \wedge n \geq 0 : f(n) \rightarrow 0, d \downarrow c \wedge 0 \geq n : f(n) \rightarrow 1\}$.

Obviously R is confluent. However, R does not satisfy the first condition of conditionally superposition free, because for every substitution σ such that $n\sigma = 0$, $D[n\sigma \geq 0] = D[0 \geq n\sigma] = true$. The transformation gives the following system:

$$R_c = \{f(n) \rightarrow if_1(n, d, D[n \geq 0], D[0 \geq n]), if_1(n, c, true, y) \rightarrow 0, if_1(n, c, y, true) \rightarrow 1\}$$

The divergence arises trivially from the admissible term $if_1(n\sigma, c, D[n\sigma \geq 0], D[0 \geq n\sigma])$, where $n\sigma = 0$. In effect, $if_1(0, c, D[0 \geq 0], D[0 \geq 0]) \xrightarrow{*}_{R_c} if_1(0, c, true, true)$ and the last term rewrites to 0 and 1. \diamond

The next example illustrates the necessity of quasi-normality in order to obtain a conservative extension.

Example 4.3 Consider the non quasi-normal CRS $R = \{i(x) \rightarrow x, g(x) \downarrow i(y) : f(x) \rightarrow y\}$.

The transformation gives the following system:

$$R_c = \{i(x) \rightarrow x, f(x) \rightarrow if_1(x, g(x)), if_1(x, i(y)) \rightarrow y\}$$

R_c is not conservative; namely, $f(x) \rightarrow_R^* g(x)$ holds but $f(x) \not\xrightarrow{*}_{R_c} g(x)$. \diamond

Remark: The strong restriction for terms in the extended signature with root symbol if_j to be of the form $if_j(\vec{x}, x, D[P_1\sigma], \dots, D[P_{|R_j|}\sigma])$ and their derivations with respect to the basic theory is not syntactically natural. Evidently, every R_c -derivation from a term in the original signature gives rise to a term, which satisfies this operational restriction; but there exist terms syntactically correct in the extended signature that can give rise to divergence. In effect, consider a term of the form $if_j(\vec{x}_j\sigma, v, D[P_1\sigma], \dots, D[P_1\sigma], \dots, D[P_k\sigma], \dots, D[P_{|R_j|}\sigma])$ and suppose, contrarily to the first restriction of conditionally superposition free, that $v \equiv v_k\sigma \equiv v_l\sigma$ and $D[P_l\sigma] \wedge D[P_k\sigma] = true$ (this situation does not contradict the convergence hypothesis of R because $u_j\sigma \not\rightarrow_R^* v^4$). Divergence trivially arises as follows:

$$r_k\sigma \stackrel{*}{\leftarrow}_{R_c} if_j(\vec{x}_j\sigma, v, D[P_1\sigma], \dots, D[P_1\sigma], \dots, D[P_k\sigma], \dots, D[P_{|R_j|}\sigma]) \rightarrow_{R_c}^* r_l\sigma.$$

This restriction is operationally coherent, but it should arise naturally (from syntactic constraints). An alternative restriction in this sense that does not alter substantially the above proof (however, that does not release the operational restrictions over the signature of R_c), becomes by changing the first condition in the definition of conditionally superposition free as follows:

- Let $P \wedge u_j \downarrow v : l_j \rightarrow r$ and $P' \wedge u_j \downarrow v' : l_j \rightarrow r'$ be different rules in R_j (for some $j \in \{1, \dots, q\}$) and let σ be a unificator of v and v' such that $D[P\sigma] \wedge D[P'\sigma] = true$ then $r\sigma \downarrow r'\sigma$.

5 Conclusion

We have presented a syntactic transformation from CRSs into TRSs and proved in detail preservation of rewriting properties (derivation, termination, confluence) for the class of safely transformable (decreasing, confluent, quasi-normal and conditionally superposition free) CRSs with built-in predicates. Restricting the extended signature of the transformation we can assure the conservativeness of the transformation too.

Since specifying computable algebras using conditional equations with built-in predicates is easier and more elegant than with purely nonconditional equations, we consider our transformation of practical applicability for the reuse of software designed for TRSs.

References

- [AM90] J. Avenhaus and K. Madlener. Term Rewriting and Equational Reasoning. In R. B. Banerji, editor, *Formal Techniques in Artificial Intelligence*, chapter 1, pages 1–43. Elsevier Science Publishers B. V. (North-Holland), 1990.
- [Ave95] J. Avenhaus. *Reduktionssysteme — Rechnen und Schliessen in gleichungsdefinierten Strukturen*. Springer-Lehrbuch. Springer, 1995. In German.
- [Aya93] M. Ayala. *Expressiveness of Conditional Equational Systems with Built-in Predicates*. PhD thesis, Universität Kaiserslautern, Kaiserslautern (Germany), December 1993.
- [Aya94] M. Ayala. Confluence of Conditional Rewriting Systems with Built-in Predicates and Standard Premises as Conditions. In *Anais XXI Semindrio Integrado de Software e Hardware, Cazambú, Brazil*, August 1994.
- [Aya95] M. Ayala. A Deductive Calculus for Conditional Equational Systems with Built-in Predicates as Premises — Extended Abstract —. In *XV International Conference of the Chilean Computer Science Society, Arica, Chile*, November 1995.
- [Bec94] K. Becker. *Rewrite Operationalization of Clausal Specifications with Predefined Structures*. PhD thesis, Universität Kaiserslautern, Kaiserslautern (Germany), April 1994.
- [BK86] J. A. Bergstra and J. W. Klop. Conditional Rewrite Rules: Confluence and Termination. *Journal of Computer and Systems Sciences*, 32, 1986.

⁴If $u_j\sigma \rightarrow_R^* v \equiv v_k\sigma \equiv v_l\sigma$ and $D[P_l\sigma] \wedge D[P_k\sigma] = true$ then both rules apply and then $r_k\sigma \leftarrow l_j\sigma \rightarrow_R r_l\sigma$. Convergence hypothesis guarantees that $r_k\sigma \downarrow r_l\sigma$.

- [Cla88] I. Classen. Algebraische Grundlagen der Termersetzung mit bedingten Gleichungen. Technical Report 88-04, TU Berlin, 1988. In German.
- [Dau92] M. Dauchet. Simulation of Turing Machines by a Regular Rewrite Rule. *Theoretical Computer Science*, 103:409–420, 1992.
- [DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume 2, chapter 6, pages 244–320. Elsevier Science Publishers B. V. (North-Holland), 1990.
- [DJK95] N. Dershowitz, J.-P. Jouannaud, and J. W. Klop. Problems in Rewriting III. In J. Hsiang, editor, *Proc. Sixth International conference on Rewriting Techniques and Applications, Kaiserslautern, Germany*, volume 914 of LNCS. Springer, April 1995.
- [DM79] N. Dershowitz and Z. Manna. Proving termination with multiset orderings. *Comm. of the ACM*, 22(8):465–476, 1979.
- [DO88] N. Dershowitz and M. Okada. Conditional Equational Programming and the Theory of Conditional term Rewriting. In ICOT (Institute for New Generation Computer Technology), editor, *Proc. of the International Conference on Fifth Generation Computer Systems*, volume 2, pages 337–346. Springer, 1988.
- [DO90] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.
- [EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1*. EATCS Monographs on Theoretical Computer Science. Springer, 1985.
- [GM87] E. Giovannetti and C. Moiso. Notes on the elimination of conditions. In S. Kaplan and J-P Jouannaud, editors, *Proceedings of the First International Workshop on Conditional Term Rewriting Systems, Orsay, France*, volume 308 of LNCS. Springer, 1987.
- [Gue87] I. Guessarian. Equational axiomatization of if-then-else. In S. Kaplan and J-P Jouannaud, editors, *Proceedings of the First International Workshop on Conditional Term Rewriting Systems, Orsay, France*, volume 308 of LNCS. Springer, 1987.
- [Hin95a] C. Hintermeier. *Déduction avec Sortes Ordonnées et Égalités*. PhD thesis, Université Henri Poincaré — Nancy I, Centre de Recherche en Informatique de Nancy, INRIA—Lorraine, October 1995. In English.
- [Hin95b] C. Hintermeier. How to Transform Canonical Conditional TRS's into Canonical TRS's. In N. Dershowitz and N. Lindenstrauss, editors, *Proc. fourth Int. Workshop on Conditional and Typed Rewriting Systems CTRS-94, Jerusalem, Israel, July, 1994*, volume 968 of LNCS. Springer, 1995.
- [Hue80] G. Huet. Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems. *Journal of the Association for Computing Machinery*, 27(4):797–821, October 1980.
- [Kap84] S. Kaplan. Simplifying conditional term rewriting systems: Unification, termination and confluence. Laboratoire de Recherche en Informatique, Université de Paris-Sud, Orsay, France, Rapport de Recherche N. 316, November 1984.
- [KB70] D. E. Knuth and P. B. Bendix. *Computational Problems in Abstract Algebra*, chapter Simple Words Problems in Universal Algebras, pages 263–297. J. Leech, ed. Pergamon Press, Oxford, U. K., 1970.
- [Klo92] J. W. Klop. Term Rewriting Systems. In S. Abramski, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 1–116. Oxford Science Publications, 1992.
- [KZ89] D. Kapur and H. Zhang. An overview of Rewrite Rule Laboratory (RRL). In N. Dershowitz, editor, *Proc. Third Int. Conf. on Rewriting techniques and Applications, Chapel-Hill, NC*, volume 355 of LNCS. Springer, April 1989.
- [Siv89] G. Sivakumar. *Proofs and Computations in Conditional Equational Theories*. PhD thesis, University of Illinois at Urbana-Champaign, 1989.
- [Vor89] S. G. Vorobyov. Conditional rewrite rule systems with Built-in Arithmetic and Induction. In N. Dershowitz, editor, *Proc. Third Int. Conf. on Rewriting techniques and Applications, Chapel-Hill, NC*, volume 355 of LNCS, pages 492–512. Springer, April 1989.
- [Wec92] W. Wechler. *Universal Algebra for Computer Scientists*. EATCS Monographs on Theoretical Computer Science. Springer, 1992.