Canonical Conditional Rewrite Systems*

Nachum Dershowitz[1], *Mitsuhiro Okada*[2], *and G. Sivakumar*[1]

[1]Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801, U.S.A.

[2]Department of Computer Science
Concordia University
Montreal, Quebec H3G 1M8, Canada

Abstract

Conditional equations have been studied for their use in the specification of abstract data types and as a computational paradigm that combines logic and function programming in a clean way. In this paper we examine different formulations of conditional equations as rewrite systems, compare their expressive power and give sufficient conditions for rewrite systems to have the "confluence" property. We then examine a restriction of these systems using a "decreasing" ordering. With this restriction, most of the basic notions (like rewriting and computing normal forms) are decidable, the "critical pair" lemma holds, and some formulations preserve canonicity.

1. Introduction

Conditional rewriting systems arise naturally in the algebraic specification of data types and have been studied largely from this perspective [Remy-82, Kaplan-84, Bergstra-Klop-82]. See also [Brand-Darringer-Joyner-78]. With differing restrictions on left-hand sides and conditions, useful results have been obtained about the confluence of such systems. More recently, conditional rewriting systems have been shown to provide a natural computational paradigm combining logic and functional programming [Dershowitz-Plaisted-85, Fribourg-85, Goguen-Meseguer-86]. A program is a set of conditional rules and a computation is the process of finding a substitution that makes two terms equal in the underlying equational theory.

---

* This research was supported in part by the National Science Foundation under Grant DCR 85-13417. The second author is also partly supported by the Grant of the Committee on Aid to Research Activity of Faculty of Engineering and Computer Science (Concordia University), Fonds pour la Formation de Chercheurs et l'Aide a la Recherche (Quebec) and the Natural Science and Engineering Research Council (Canada).

In this paper we study the theory of conditional rewrite systems. In Section 2, we present various formulations of conditional equations as rewrite systems and compare their expressive power. We also examine the differing restrictions under which we can prove confluence of the various systems and the equivalence of these formulations to the underlying equational theory if they are canonical. In Section 3, we identify a class of systems which are "decreasing". For these systems, the basic notions of rewriting are all decidable and the critical pair lemma holds as for unconditional systems. We show that decreasing systems extend the classes of "simplifying" and "reductive" systems that had been proposed earlier.

2. Conditional Equations

A *positive-conditional* equation is of the form

$$s_1{=}t_1 \wedge \cdots \wedge s_n{=}t_n\mathbf{:}\ s = t$$

where $n \geq 0$ and the $s_i = t_i$ are equations, containing universally quantified variables. The ":" may be thought of as implication with $s = t$ as the conclusion and $t_i = s_i$ as the premises. In this paper, we consider only *equational* consequences and proofs. Note that equational logic lacks a "law of excluded middle". The relation between equational proofs and first-order ones is studied in [Dershowitz-Plaisted-87].

We define the one-step replacement relation $\longleftrightarrow$ and its reflexive-transitive closure $\overset{*}{\longleftrightarrow}$ as follows: If $s_1{=}t_1 \wedge \cdots \wedge s_n{=}t_n\mathbf{:}\ s = t$ is a conditional equation, σ is a substitution, u is a term, π is a position in u and $t_i\sigma \overset{*}{\longleftrightarrow} s_i\sigma$ for $i = 1, \cdots, n$, then $u[s\sigma]_\pi \longleftrightarrow u[t\sigma]_\pi$ where $u[s\sigma]_\pi$ denotes the term u with $s\sigma$ as a subterm at position π and $u[t\sigma]_\pi$ is the term obtained by replacing $s\sigma$ by $t\sigma$. We write $E \vdash s = t$ if $s \overset{*}{\longleftrightarrow} t$ for a set E of positive-conditional equations.

For example, using equations

$$\begin{array}{rrcl} & 0 + y & = & y \\ x + y = z\mathbf{:} & s(x) + y & = & s(z) \end{array}$$

we have $s(0) + s(0) \longleftrightarrow s(s(0))$ since $0 + s(0) \longleftrightarrow s(0)$.

2.1. Conditional Rewrite Systems

Conditional rules are conditional equations with the equation in the conclusion oriented from left to right. A conditional rule is used to rewrite terms by replacing an instance of the left-hand side with the corresponding instance of the right-hand side (but not in the opposite direction) provided the conditions hold. A set of conditional rules is called a conditional rewrite system. Depending on what criterion is used to check conditions different rewrite relations are obtained for any given system R (see below). Once a criterion is chosen, we can define the one-step rewrite relation $\longrightarrow$ and its reflexive-transitive closure $\overset{*}{\longrightarrow}$ as follows:

$u[l\sigma]_\pi \rightarrow u[r\sigma]_\pi$ if $c: l \rightarrow r$ is a rule, σ is a substitution, u is a term, π is a position in u and $c\sigma$ satisfies the criterion.

A term t is *irreducible* (or in *normal* form) if there is no term s such that $t \rightarrow s$. We say that two terms s and t are *joinable*, denoted $s \downarrow t$, if $s \xrightarrow{*} v$ and $t \xrightarrow{*} v$ for some term v. A rewrite relation $\rightarrow$ is said to be *noetherian* if there is no infinite chain of terms $t_1, t_2, \cdots, t_k, \cdots$ such that $t_i \rightarrow t_{i+1}$ for all i. A rewrite relation $\rightarrow$ is said to be *confluent* if the terms u and v are joinable whenever $t \xrightarrow{*} u$ and $t \xrightarrow{*} v$. It is *locally confluent* if the terms u and v are joinable whenever $t \rightarrow u$ and $t \rightarrow v$ (in one step). A rewrite system R is *canonical* if its rewrite relation is both noetherian and confluent.

There are a fair number of different ways of formulating conditional equations as rewrite rules:

Semi–Equational systems

Here we formulate rules as $s_1 = t_1 \wedge \cdots \wedge s_n = t_n: l \rightarrow r$, where the conditions are still expressed as equations. To check if a condition holds we use the rules bidirectionally, as identities, and check if $s_i\sigma \longleftrightarrow^* t_i\sigma$.

Join systems

Here we express rules as $s_1 \downarrow t_1 \wedge \cdots \wedge s_n \downarrow t_n: l \rightarrow r$. The conditions are now checked in the rewrite system itself by checking if $s_i\sigma$ and $t_i\sigma$ are joinable. Note the circularity in the definition of $\rightarrow$. The base case, of course, is when unconditional rules are used or the conditions unify syntactically. This definition is the one most often used; see [Kaplan–84, Jouannaud–Waldmann–86, Dershowitz–Okada–Sivakumar–87].

Normal–Join systems

Here rules are written $s_1 \downarrow^! t_1 \wedge \cdots \wedge s_n \downarrow^! t_n: l \rightarrow r$. This is similar to join systems except that $s_i\sigma$ and $t_i\sigma$ are not only joinable, but also have a common reduct that is irreducible. (A sufficient condition for this is that the common reduct not contain any instance of a left–hand side.)

Normal systems

A special form of normal–join systems has all conditions of the form $s_i \xrightarrow{!} t_i$ (meaning that $s_i \xrightarrow{*} t_i$ and t_i is an irreducible ground term).

Inner–Join systems

Here rules are written $s_1 \downarrow^i t_1 \wedge \cdots \wedge s_n \downarrow^i t_n: l \rightarrow r$. We require that $s_i\sigma$ and $t_i\sigma$ are joinable by *innermost rewriting.* That is, in rewriting these terms, one applies a rule at some position only if all proper subterms are already in normal form.

Outer–Join systems

Here rules are written $s_1 \downarrow^o t_1 \wedge \cdots \wedge s_n \downarrow^o t_n$: $l \longrightarrow r$. We require that $s_i\sigma$ and $t_i\sigma$ are joinable by *outermost rewriting*. That is, in rewriting these terms, one applies a rule at some position only if no rule can be used above (at a superterm).

Meta–Conditional systems

Here we allow any (not necessarily recursively enumerable) predicate p in the conditions. For example, we may have conditions like $s \in S$ (for some term s and set S), $x \xrightarrow{!} x$ (x is already in normal form), or $l > r$ (for some ordering $>$). We write p: $l \longrightarrow r$.

Most of the formulations above have been considered by different authors with slight variations. For example, Bergstra and Klop in [Bergstra–Klop–86] restrict their attention to systems which are *left–linear* (no left–hand side has more than one occurrence of any variable) and non–overlapping (no left–hand side unifies with a renamed non–variable subterm of another left–hand side or with a renamed proper subterm of itself). With these restrictions on left–hand sides, they refer to semi–equational systems as of *Type I*, join systems as of *Type II* and normal systems as of *Type* III_n. They also prove that, with these restrictions on left–hand sides, *Type I* and *Type* III_n systems are confluent. Meta–conditional systems with *membership* conditions were proposed in [Toyama–87].

2.2. Sufficient Conditions for Confluence

An interesting question to address is what are the criteria under which each formulation is confluent. For syntactic criteria we have to consider overlaps between left–hand sides of rules called "critical pairs" defined below.

Let R be a rewrite system with rules c: $l \longrightarrow r$ and p: $g \longrightarrow d$ renamed to share no variables.

Definition 1. If g unifies with a non–variable subterm of l at position λ via a substitution σ, then the conditional equation $(c \wedge p)\sigma$: $r\sigma = l\sigma[d\sigma]_\lambda$, is a *critical pair* of the two rules.

A system is *non–overlapping* (or unambiguous) if it has no critical pairs. A critical pair c: $s = t$ is *feasible* if there is a substitution σ for which $c\sigma \xrightarrow{*} true$. A critical pair c: $s = t$ is *joinable* if for all feasible substitutions σ, there exists a term v such that $s\sigma \xrightarrow{*} v$ and $t\sigma \xrightarrow{*} v$. A critical pair is an *overlay* if the two left–hand sides unify at the root.

The *depth* of a rewrite is the depth of recursive evaluations of conditions needed to determine that the matching substitution is feasible. We define it formally for join systems.

Definition 2. The depth of an unconditional rewrite is 0; the depth of a rewrite using a conditional rule $u{\downarrow}v\colon l \to r$ and substitution σ is one more than the maximum of the depths of the two derivations used to show $u\sigma{\downarrow}v\sigma$. The depth of a n-step derivation $t \xrightarrow{*} s$ is the maximum of the depths of each of the n steps.

We write $t \underset{k}{\to} s$ if $t \to s$ and the depth of the rewrite step is no more than k. Similarly $t \overset{*}{\underset{k}{\to}} s$ will mean that the maximum depth in that derivation is at most k.

A critical pair between rules $c\colon l \to r$ and $p\colon g \to d$ is *shallow-joinable* if for each feasible substitution σ, $l\sigma \underset{m}{\to} r\sigma$, $l\sigma \underset{n}{\to} l\sigma[d\sigma]$ then there exists a term v, $r\sigma \underset{n}{\to} v$ and $l\sigma[d\sigma] \underset{n}{\to} v$. That is, the critical pair is joinable with the corresponding depths preserved. In particular, critical pairs between

unconditional rules should be joinable unconditionally.

For example, the rules $f(x) \to g_1(x)$ and $h(x){\downarrow}c\colon f(x) \to g_2(x)$ overlap to yield a critical pair $h(x){\downarrow}c\colon g_1(x) = g_2(x)$ (which is an also an overlay). If we also had a rule $h(0) \to c$, then we would have a feasible instance of this critical pair for the substitution $x \mapsto 0$. This instance would be shallow joinable, if for some term t, $g_1(0) \overset{*}{\underset{1}{\to}} t \overset{*}{\leftarrow} g_2(0)$ since $f(0) \to g_1(0)$ and $f(0) \underset{1}{\to} g_2(0)$.

For unconditional systems, the Critical Pair Lemma (see [Knuth-Bendix-70]) states that a system is locally confluent iff all its critical pairs are joinable. For conditional systems this is true only for some of the formulations above; in general, stronger restrictions are needed. By Newman's Lemma, a noetherian system is confluent iff it is locally confluent. We list below some sufficient conditions for the confluence of the various formulations.

Semi-Equational systems
Noetherian and critical pairs are joinable [proof straightforward].

Join systems
Decreasing (see next section for definition) and critical pairs are joinable [Section 3].

Join systems
Noetherian and all critical pairs are overlays and joinable [Dershowitz-Okada-Sivakumar-87].

Normal systems
Noetherian, left-linear and critical pairs are shallow-joinable [Dershowitz-Okada-Sivakumar-87].

Inner systems
Noetherian and critical pairs are joinable.

2.3. Strength of Rewrite systems

Let E be a set of conditional equations. By $E \vdash s = t$, we mean that $s \stackrel{*}{\longleftrightarrow} t$ is provable in E. Similarly, if R is a rewrite system (in any of the formulations), we use $R \vdash s{\downarrow}t$, to mean that s and t are joinable using the rules in R.

Definition 3. R and E have the same *logical strength* if $E \vdash s = t$ *iff* $R \vdash s{\downarrow}t$. Similarly, two rewrite systems R and R' have the same logical strength if $R \vdash s{\downarrow}t$ *iff* $R' \vdash s{\downarrow}t$. We say that R is *stronger* than R' if any two terms joinable using R' are joinable using R, but not the converse.

Figure 1 depicts the relative strength of the various formulations. In the figure, $A \longrightarrow B$, means that A is stronger than B in general. That is, if we take a system of type B and just change the connective in conditions to convert to a system of type A (for example, $s \downarrow^{i} t$ to $s{\downarrow}t$ to covert an inner–join to a join system), then we have that what is provable in B is also provable in A. In particular, if B is canonical then so is A. The converse is, of course, not true in general.

We now state and prove some of the equivalences and relationships between the various systems.

Proposition 1: If a join system is noetherian, then it is equivalent to the corresponding normal–join system (obtained by changing conditions of the form $s{\downarrow}t$ to $s{\downarrow}^{!}t$).

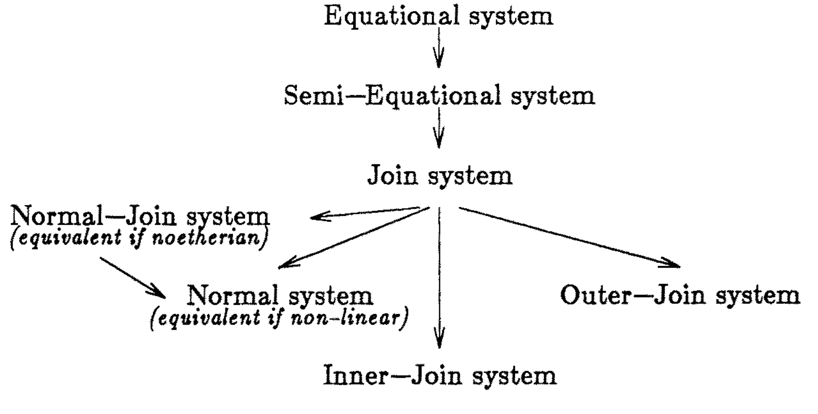

Figure 1

Proposition 2: We can convert a join system R to an equivalent normal system R' by a conservative extension (using new function symbols) provided that we allow the normal system to be non-left-linear (have repeated variables in left-hand sides).

Proposition 3: Let R (with conditions of the form $s{\downarrow}t$) be a canonical (confluent and noetherian) join system R' the corresponding semi-equational systems (change conditions to $s = t$) and E the underlying equational system (change conditions to $s = t$ and $l \to r$ to $l = r$). The following are equivalent:

(1) $u = v$ is provable in E, that is, $E \vdash u = v$

(2) u and v have a common reduct in R,that is, $R' \vdash u{\downarrow}v$

(3) u and v have a common reduct in R', that is, $R \vdash u{\downarrow}v$

Proof. Proposition 1 is easy to see, for the noetherian property implies that if two terms are joinable, then they have a common reduct that is irreducible. The translation mechanism for Proposition 2 uses two new function symbols eq and $true$. We add a new rule $eq(x,x) \to true$ and change conditions to the form $eq(s,t){\downarrow}true$. With this translation, it is easy to prove that for any two terms s and t not having the new function symbols eq and $true$, we have $R \vdash s{\downarrow}t$ *iff* $R' \vdash s{\downarrow}t$. The argument for Proposition 3 is by induction on the depth of a proof. The interesting case is when $u = v$ is provable in E and we wish to show $u{\downarrow}v$ in R. By induction on the depth we first show that the subproofs in E can be replaced by rewrite proofs and then using the confluence of R we can show that $u{\downarrow}v$. □

Under the assumption of canonicity the various weaker formulations of join systems are also equivalent to the corresponding join system and, hence, to the underlying equational system. We prove this below for inner systems.

Theorem 1. *Let R be a canonical inner-join (or outer-join, normal-join) system and R' the corresponding join system obtained by replacing conditions of the form $s \downarrow^i t$ ($s \downarrow^o t$, $s \downarrow^! t$, respectively) by $s{\downarrow}t$. R is equivalent to R'. That is for any terms s and t, $R \vdash s{\downarrow}t$ iff $R' \vdash s{\downarrow}t$.*

It is easy to see that if s and t have a common reduct in an inner-join system, then they have a common reduct in the join system (the same proof holds). To show the other direction we prove an even stronger version.

Lemma 1. *For any terms s and t, if $s{\downarrow}t$ in the inner-join system R' then $s \downarrow^i t$ in R. That is, s and t are joinable by innermost rewriting in the inner join system.*

Proof. The proof of this lemma is carried out by induction on the depth of $s{\downarrow}t$ in R'. If s joins t at depth 0 in R', the same rewrite sequence is also valid in the inner-join system since we do not use any conditional rules. So, $s{\downarrow}t$ in R too. Since R is canonical, this also means that $s \downarrow^i t$ in R. Otherwise, if s and t are joinable at depth $n+1$, every condition, say $u \downarrow v$, used for rewriting is provable in depth at most n in R'. By the induction hypothesis, the corresponding conditions $(u \downarrow^i v)$ are all provable in R. Therefore, the rewrite steps also apply in R, hence $s \downarrow t$ in R. Again, since R is confluent and noetherian, $s \downarrow t$ in R implies $s \downarrow^i t$ in R. □

We can sum up by saying that if any formulation as a rewrite system is canonical, then it is equivalent to the corresponding equational system.

3. Decreasing Systems

By the *reduction ordering* $>_R$ of a rewrite system, we mean the irreflexive-transitive closure $\xrightarrow{+}$ of the reduction relation. That is, $t_1 >_R t_2$ if $t_1 \xrightarrow{+} t_2$. The reduction ordering is *monotonic*. That is, if $t >_R s$ then $u[t] >_R u[s]$ for any context $u[\cdot]$. By the *proper subterm ordering* $>_s$ we mean the well-founded ordering $u[t] >_s t$ for any term t and non-empty context $u[\cdot]$. In this section, we will use the join system formulation of conditional rules to illustrate definitions and results.

Definition 4. A conditional rewrite system is *decreasing* if there exists a well-founded extension $>$ of the proper subterm ordering such that $>$ contains $>_R$ and $l\sigma > s_1\sigma, \cdots, t_n\sigma$ for each rule $s_1 \downarrow t_1 \wedge \cdots \wedge s_n \downarrow t_n$: $l \rightarrow r$ $(n \geq 0)$ and substitution σ.

Note that the second condition restricts all variables in the condition to also appear on the left-hand side. In general, a decreasing ordering need not be monotonic.

Proposition 4: If a rewrite system is decreasing then it has the following properties:

(1) The system is terminating.

(2) The basic notions are decidable. That is, for any terms s, t
i) one-step reduction ("does $s \rightarrow t$?")
ii) finite reduction ("does $s \xrightarrow{*} t$?")
iii) joinability ("does $s \downarrow t$?")
iv) normal form or reducibility ("is s irreducible?")
are all decidable.

Proof. That the system is terminating is obvious from the well-foundedness of $>$. The decidability of basic notions is proved by transfinite induction on $>$, as follows. We first consider the following property: "Given a term t we can find the set of normal forms of t." If t has no instance of a left-hand side of any rule as a subterm then t is irreducible and it is its only normal form. Otherwise, let $t = u[l\sigma]$ for some rule $u_1 \downarrow v_1 \wedge \cdots \wedge u_n \downarrow v_n$: $l \rightarrow r$ By our two conditions on decreasingness we have that $t = u[l\sigma] > l\sigma$ and $l\sigma > u_i\sigma$, $v_i\sigma$. By induction, since $t > u_i$, v_i, we can compute the set of normal forms of u_i, v_i for each i and check if the rule applies. If it does then $t \rightarrow u[r\sigma]$. Similarly (using each matching rule) we can compute all the terms, say $s_1, \ldots, s_n$, that t rewrites to in one-step. By induction hypothesis one can enumerate the normal forms for each s_i. Then the union of these is the set of normal forms for t.

Other basic properties can be shown decidable likewise. □

The following are some sufficient conditions for decreasingness:

Simplifying systems [Kaplan-84, Kaplan-87]
A conditional rewrite system R is *simplifying* if there exists a simplification ordering $>$ (in the sense of [Dershowitz-82]) such that $l\sigma > r\sigma, s_1\sigma \cdots, t_n\sigma$, for each rule $s_1 \downarrow t_1 \wedge \cdots \wedge s_n \downarrow t_n$: $l \rightarrow r$ $(n \geq 0)$.

Reductive systems [Jouannaud-Waldmann-86]
A conditional rewrite system is *reductive* if there is a well-founded monotonic ordering $>$ such that $>$ contains the reduction ordering $>_R$ and $l\sigma > s_1\sigma, \cdots, t_n\sigma$ for each rule $s_1 \downarrow t_1 \wedge \cdots \wedge s_n \downarrow t_n$: $l \rightarrow r$ $(n \geq 0)$.

Both simplifying systems and reductive systems are special cases of decreasing ones. To see this for simplifying systems, note that simplification orderings contain the subterm ordering, by definition. For reductive systems, note that no monotonic well-founded ordering can have $s > t$ for a proper subterm s of t. So we can extend the monotonic ordering with the subterm property and get a well-founded ordering as in [Jouannaud-Waldmann-86]. The following is an example of a system that is decreasing but neither simplifying nor general reductive:

$$
\begin{array}{rcrcl}
 & & b & \rightarrow & c \\
 & & f(b) & \rightarrow & f(a) \\
b \downarrow c & : & a & \rightarrow & c
\end{array}
$$

This is not reductive because there is no monotonic extension of the reduction ordering (which has $f(b) >_R f(a)$) that can have $a > b$.

Decreasing systems also satisfy the critical pair lemma.

Theorem 2. *For any decreasing system, if every critical pair is joinable, then the system is confluent, hence canonical.*

Proof. The proof is implicit in [Jouannaud-Waldmann-86] where they impose stronger conditions for their definition of a reductive system (as explained earlier) but use essentially the same conditions that we have for decreasingness in their proof. □

Were we to omit "the subterm property" in our definition of a decreasing ordering, then the critical pair lemma no longer holds. We illustrate this with the counter-example shown in Table 1. All critical pairs are joinable, yet the term $h(f(a))$ has two normal forms d and $k(f(b))$. Although we have $h(f(a)) \rightarrow h(f(b))$ and $h(g(b)) \rightarrow k(g(b)) \rightarrow d$, to determine if $h(f(b)) \rightarrow h(g(b))$ using the last rule we have to check the condition $h(f(b)) \downarrow d$

		c	$\rightarrow$	$k(f(a))$
		c	$\rightarrow$	d
		a	$\rightarrow$	b
		$h(x)$	$\rightarrow$	$k(x)$
		$h(f(a))$	$\rightarrow$	c
		$k(g(b))$	$\rightarrow$	d
$h(f(x)) \downarrow d$	:	$f(x)$	$\rightarrow$	$g(x)$

Table 1

which leads to a cycle. Note that if we converted this to a semi–equational system we would have that $h(f(b)) \overset{*}{\longleftrightarrow} d$ and the last rule can be applied and the system is confluent.

This example satisfies the conditions for decreasing systems except the subterm property. The reduction ordering of the above rewrite system is embeddable into the well–founded ordering $<_\infty$ (which, however, does not have the subterm property) of Takeuti's system $O(2,1)$ of ordinal diagrams (one of the two major systems of proof theoretic ordinals). See [Okada–Takeuti–87] for definitions. Also, $<_\infty$ satisfies the additional condition for decreasingness (each term in the condition– d and $h(f(x))$– is smaller than the left–hand side—$f(x)$—of that rule). So it is clear that well–foundedness alone is not sufficient. For more details see [Okada–87, Dershowitz–Okada–88].

We saw in the previous section that, while the confluence of a join system implies the confluence of the corresponding semi–equational system (without any other restriction), the converse is not true in general. We now show that if we restrict our attention to decreasing systems the converse does hold. That is, under the assumption of decreasingness the two formulations, semi–equational systems and join systems make no difference with respect to confluence.

Theorem 3. *If a decreasing semi–equational system (conditions of the form $s = t$) is confluent, then the corresponding join system (with conditions changed to $s \downarrow t$) is also confluent.*

It is convenient to introduce the following notations. By a *direct proof* of $s = t$, we mean a rewrite proof of the form $s \downarrow t$. That is, s and t are joinable. By a *completely direct* proof of $s = t$, we mean a direct proof of $s = t$ (i.e, of $s \downarrow t$) in which every subproof of the conditions (during application of conditional rules) is also direct. For instance, if a substitution instance of the form $s_1\sigma = t_1\sigma \wedge \cdots \wedge s_n\sigma = t_n\sigma$: $l\sigma \rightarrow r\sigma$ of a conditional rule is used in the proof with subproofs of $s_i\sigma = t_i\sigma$, then each of these subproofs is also direct. If for a given proof $\mathcal{P}$ of $s = t$ there is a completely direct proof $\mathcal{P}'$ (of $s \downarrow t$), then we say that the proof $\mathcal{P}$ is completely normalizable.

Lemma 2 (Complete Normalizability Lemma). *For any confluent and decreasing semi-equational system, every proof is completely normalizable.*

Proof. This is proved using transfinite induction on the well-founded decreasing ordering. It is easily seen that every proof in a decreasing system can be made direct if the system is confluent. By using the properties of decreasingness, we can show that every top-level subproof is smaller in the decreasing ordering and, hence, can be made completely direct by the induction hypothesis. Then this lemma follows. □

Theorem 3 is a direct consequence of the Complete Normalizability Lemma. Thus, if a decreasing system is confluent in the semi-equational formulation, then it is confluent as a join system.

4. Conclusion

We have studied different formulations of conditional rewrite rules and their expressive power and identified a class of decreasing systems for which most of the interesting notions of rewriting are decidable and which satisfy the critical pair lemma. Decreasing systems have weaker restrictions than the simplification systems or reductive systems studied previously. We have shown that straightforward attempts at weakening these restriction further (by dropping the subterm property) do not work. For this class of systems we have also shown that two of the formulations as rewrite systems are equivalent with respect to confluence.

REFERENCES

[Bergstra-Klop-82] Bergstra, J. A., and Klop, J. W. "Conditional rewrite rules: Confluence and termination", Report IW 198/82 MEI, Mathematische Centrum, Amsterdam, 1982.

[Bergstra-Klop-86] Bergstra, J. A., and Klop, J. W. "Conditional rewrite rules: confluency and termination", *JCSS* **32** pp. 323–362 , 1986.

[Brand-Darringer-Joyner-78] Brand, D., Darringer, J. A., Joyner, W. J. "Completeness of conditional reductions". Report RC 7404, IBM Thomas J. Watson Research Center, December 1978.

[Dershowitz-82] Dershowitz, N. "Orderings for term-rewriting systems". *Theoretical Computer Science* **17**(3), pp. 279–301 (March 1982).

[Dershowitz-Okada-88] Dershowitz, N., and Okada, M. "Proof-theoretic techniques for term rewriting theory". *Proceedings of the third annual Symposium on Logic in Computer Science,* Edinburgh, July 1988 (to appear).

[Dershowitz-Okada-Sivakumar-87] Dershowitz, N., Okada, M., and Sivakumar, G. "Confluence of Conditional Rewrite Systems". *First International Workshop on Conditional Rewriting Systems*, Orsay, France (July 1987) (to appear in Lecture Notes in Computer Science, Springer, Berlin).

[Dershowitz-Plaisted-85] Dershowitz, N., and Plaisted, D. A. "Logic programming *cum* applicative programming". *Proceedings of the 1985 Symposium on Logic Programming*, Boston, MA, pp. 54–66 (July 1985).

[Dershowitz–Plaisted–87] Dershowitz, N., and Plaisted, D. A. "Equational programming". In: *Machine Intelligence 11*, J. E. Hayes, D. Michie and J. Richards, eds., 1987 (to appear).

[Fribourg–85] Fribourg, L. "SLOG: A logic programming language interpreter based on clausal superposition and rewriting". *Proceedings of the 1985 Symposium on Logic Programming*, Boston, MA (July 1985), pp. 172–184.

[Goguen–Meseguer–86] Goguen, J. A., and Meseguer, J. "EQLOG: Equality, types and generic modules for logic programming". In *Logic Programming: Functions, relations and equations* (D. DeGroot and G. Lindstrom, eds.), Prentice–Hall, Englewood Cliffs, NJ, pp. 295–363, 1986.

[Jouannaud–Waldmann–86] Jouannaud, J. P., and Waldmann, B. "Reductive Conditional term rewriting systems". *Proceedings of the Third IFIP Working Conference on Formal Description of Programming Concepts*, Ebberup, Denmark.

[Kaplan–84] Kaplan, S. "Fair conditional term rewriting systems: Unification, termination and confluency", Laboratoire de Recherche en Informatique, Université de Paris–Sud, Orsay, France, November 1984.

[Kaplan–87] Kaplan, S. "Simplifying conditional term rewriting systems: Unification, termination and confluence", *Journal of Symbolic Computation* (to appear).

[Knuth–Bendix–70] Knuth, D. E., and Bendix, P. B. "Simple word problems in universal algebras". In: *Computational Problems in Abstract Algebra*, J. Leech, ed. Pergamon Press, Oxford, U. K., 1970, pp. 263–297.

[Okada–87] Okada, M. "A logical analysis for theory of conditional rewriting". *First International Workshop on Conditional Rewriting Systems*, Orsay, France (July 1987) (to appear in Lecture Notes in Computer Science, Springer, Berlin).

[Okada–Takeuti–87] Okada, M. and Takeuti, G. "On the theory of quasi–ordinal diagrams", in *Logic and Combinatorics*, ed. S.Simpson, Contemporary Mathematics 68 (1987), American Mathematical Society.

[Remy–82] Rémy J.–L., "Etude des systèmes de réécriture conditionnels et applications aux types abstraits algébriques" Thèse, Institut National Polytechnique de Lorraine, July 1982.

[Toyama–87] Toyama, Y. "Term rewriting systems with membership conditions". *First International Workshop on Conditional Rewriting Systems*, Orsay, France (July 1987) (to appear in Lecture Notes in Computer Science, Springer, Berlin).