Level-Confluence of Conditional Rewrite Systems with Extra Variables in Right-Hand Sides

Taro Suzuki Aart Middeldorp Tetsuo Ida

Institute of Information Sciences and Electronics
University of Tsukuba, Tsukuba 305, Japan

{taro,ami,ida}@score.is.tsukuba.ac.jp

ABSTRACT

Level-confluence is an important property of conditional term rewriting systems that allow extra variables in the rewrite rules because it guarantees the completeness of narrowing for such systems. In this paper we present a syntactic condition ensuring level-confluence for orthogonal, not necessarily terminating, conditional term rewriting systems that have extra variables in the right-hand sides of the rewrite rules. To this end we generalize the parallel moves lemma. Our result bears practical significance since the class of systems that fall within its scope can be viewed as a computational model for functional logic programming languages with local definitions, such as let-expressions and where-constructs.

1. Introduction

There is a growing interest in combining the functional and logic programming paradigms in a single language, see Hanus [12] for a recent overview of the field. The underlying computational mechanism of most of these integrated languages is (conditional) narrowing. Examples of such languages include BABEL [19] and K-LEAF [8]. In order to ensure the desirable completeness of narrowing strategies, restrictions have to be imposed on the programs, which for the purpose of this paper are viewed as conditional term rewriting systems, written in these languages. In this paper we are concerned with the level-confluence restriction, a key property (Giovannetti and Moiso [9], Middeldorp and Hamoen [18]) for ensuring the completeness of narrowing in the presence of so-called extra variables. Very few techniques are available for establishing level-confluence of conditional systems, this in contrast to the confluence property for which several sufficient criteria are known, e.g. [2, 3, 6, 7, 11, 17, 20, 21, 22]. We only know of an early paper by Bergstra and Klop. In [3] they show that orthogonal normal conditional systems are level-confluent. (Actually they show confluence—Giovannetti and Moiso [9] remark that the proof yields level-confluence.) Bergstra and Klop restrict the use of extra variables to the conditional part of the rewrite rules. Several authors remarked that it makes good sense to lift this restriction, since it enables a more natural and efficient way of writing programs in a functional logic language. For example, the Haskell program

```
divide 0     (y+1)          = (0, 0)
```

A preliminary version of this paper appeared in the Proceedings of the 6th International Conference on Rewriting Techniques and Applications, Kaiserslautern, Lecture Notes in Computer Science 914, pp. 179–193, 1995.

```
divide (x+1) (y+1) | x < y  = (0, x+1)
                   | x >= y = (q+1, r)
                     where (q, r) = divide (x - y) (y+1)
```

corresponds to the conditional term rewriting system

$$\begin{cases} div(0, S(x)) \to (0, 0) \\ div(S(x), S(y)) \to (0, S(x)) \Leftarrow x < y = true \\ div(S(x), S(y)) \to (S(q), r) \Leftarrow x \geqslant y = true, \\ \qquad\qquad\qquad\qquad\quad div(x - y, S(y)) = (q, r) \end{cases}$$

which has extra variables q and r in the right-hand side of the last rewrite rule.

The criterion—orthogonality together with normality—of Bergstra and Klop [3] is no longer sufficient when extra variables are permitted in right-hand sides. For instance, the orthogonal normal system

$$\begin{cases} a \to x \Leftarrow f(x) = true \\ f(b) \to true \\ f(c) \to true \end{cases}$$

from [14] is not confluent, let alone level-confluent, since the term a can be rewritten to the different normal forms b and c. In this paper we present a useful syntactic condition for level-confluence in the presence of extra variables in right-hand sides of rewrite rules.

The remainder of the paper is organized as follows. In the next section we recapitulate the basics of conditional term rewriting. In Section 3 we introduce and motivate our syntactic criterion. In Section 4 we prove that our criterion indeed implies level-confluence. In the next section we extend our result to the larger class of join conditional systems. In Section 6 we relate our result to the recent work of Avenhaus and Loría-Sáenz [2] and Hanus [13]. In the final section we discuss further extensions of our result.

2. Preliminaries

We assume the reader is familiar with term rewriting. (See [5] and [16] for extensive surveys.) In this preliminary section we recall only some less common definitions and introduce the basic facts concerning conditional term rewriting.

The set of function symbols $\mathcal{F}$ of a term rewriting system (TRS for short) $(\mathcal{F}, \mathcal{R})$ is partitioned into disjoint sets $\mathcal{F}_\mathcal{D}$ and $\mathcal{F}_\mathcal{C}$ as follows: a function symbol f belongs to $\mathcal{F}_\mathcal{D}$ if there is a rewrite rule $l \to r$ in $\mathcal{R}$ such that $l = f(t_1, \ldots, t_n)$ for some terms $t_1, \ldots, t_n$, otherwise $f \in \mathcal{F}_\mathcal{C}$. Function symbols in $\mathcal{F}_\mathcal{C}$ are called *constructors*, those in $\mathcal{F}_\mathcal{D}$ *defined* symbols. A term built from constructors and variables is called a *constructor* term. A left-linear TRS without critical pairs is called *orthogonal*.

The rules of a conditional TRS (CTRS for short) have the form $l \to r \Leftarrow c$. Here the conditional part c is a (possibly empty) sequence $s_1 = t_1, \ldots, s_n = t_n$ of equations. At present we only require that l is not a variable. A rewrite rule without conditions will be written as $l \to r$. Depending on the interpretation of the equality sign in the conditions of the rewrite rules, different rewrite relations can be associated with a given CTRS. In this paper we are

mainly concerned with what we will call *oriented* CTRSs. The rewrite relation $\to_{\mathcal{R}}$ associated with an oriented CTRS $\mathcal{R}$ is obtained by interpreting the equality signs in the conditional part of a rewrite rule as reachability ($\to^*$). Formally, $\to_{\mathcal{R}}$ is the smallest (w.r.t. inclusion) rewrite relation $\to$ with the property that $l\sigma \to r\sigma$ whenever there exist a rewrite rule $l \to r \Leftarrow c$ in $\mathcal{R}$ and a substitution σ such that $s\sigma \to^* t\sigma$ for every equation $s = t$ in c. The existence of $\to_{\mathcal{R}}$ is easily proved. For every oriented CTRS $\mathcal{R}$ we inductively define TRSs[1] $\mathcal{R}_n$ ($n \geqslant 0$) as follows:

$$\begin{aligned}\mathcal{R}_0 &= \varnothing,\\ \mathcal{R}_{n+1} &= \{\, l\sigma \to r\sigma \mid l \to r \Leftarrow c \in \mathcal{R} \text{ and } s\sigma \to^*_{\mathcal{R}_n} t\sigma \text{ for all } s = t \text{ in } c \,\}.\end{aligned}$$

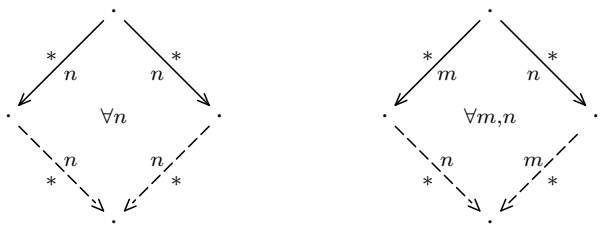
In the sequel we write $\mathcal{R}_n \vdash c\sigma$ instead of $s\sigma \to^*_{\mathcal{R}_n} t\sigma$ for every $s = t$ in c. Observe that $\mathcal{R}_n \subseteq \mathcal{R}_{n+1}$ for all $n \geqslant 0$. We have $s \to_{\mathcal{R}} t$ if and only if $s \to_{\mathcal{R}_n} t$ for some $n \geqslant 0$. The minimum such n is called the *depth* of $s \to t$. The depth of a reduction $s \to^*_{\mathcal{R}} t$ is the minimum n such that $s \to^*_{\mathcal{R}_n} t$. The depth of a 'valley' $s \downarrow_{\mathcal{R}} t$ is similarly defined. We abbreviate $\to_{\mathcal{R}_n}$ to $\to_n$. The same applies to the derived relations of $\to_{\mathcal{R}_n}$.

The TRS obtained from a CTRS $\mathcal{R}$ by dropping the conditions in rewrite rules is called the *underlying* TRS of $\mathcal{R}$ and denoted by $\mathcal{R}_u$. Concepts like left-linearity and constructor term are defined for CTRSs via the underlying TRS. Let $l_1 \to r_1 \Leftarrow c_1$ and $l_2 \to r_2 \Leftarrow c_2$ be renamed versions of rewrite rules of a CTRS $\mathcal{R}$ such that they have no variables in common. Suppose $(l_1)_{|p}$ is unifiable with l_2 for some non-variable position p in l_1, say with most general unifier σ. The conditional equation $l_1[r_2]_p\sigma = r_1\sigma \Leftarrow c_1\sigma, c_2\sigma$ is a conditional critical pair of $\mathcal{R}$, except in the case that the two rules differ a variable renaming and $p = \varepsilon$. A left-linear CTRS without conditional critical pairs is called orthogonal. A *normal* CTRS $\mathcal{R}$ is an oriented CTRS satisfying the additional restriction that every right-hand side of an equation in the conditions of the rewrite rules is a ground $\mathcal{R}_u$-normal form. Following [18], we classify rewrite rules $l \to r \Leftarrow c$ of CTRSs according to the distribution of variables among l, r, and c, as follows:

type	requirement
1	$\mathcal{V}ar(r) \cup \mathcal{V}ar(c) \subseteq \mathcal{V}ar(l)$
2	$\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$
3	$\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c)$
4	*no restrictions*

An n-CTRS contains only rules of type n. An *extra* variable x in a rewrite rule $l \to r \Leftarrow c$ satisfies $x \in (\mathcal{V}ar(r) \cup \mathcal{V}ar(c)) - \mathcal{V}ar(l)$. So a 1-CTRS contains no extra variables, a 2-CTRS may only contain extra variables in the conditions, and a 3-CTRS may also have extra variables in the right-hand sides provided these occur in the corresponding conditional part. Most of the literature on conditional term rewriting is concerned with 1 and 2-CTRSs. We are concerned with *level-confluence* of 3-CTRSs in this paper. An (oriented) CTRS $\mathcal{R}$ is called *level-confluent* if every TRS $\mathcal{R}_n$ ($n \geqslant 0$) is confluent. Level-confluence is illustrated in Figure 1. A related property is *shallow-confluence*. An (oriented) CTRS $\mathcal{R}$ is called *shallow-confluent* if, for all $m, n \geqslant 0$, $s \to^*_m t_1$ and $s \to^*_n t_2$ imply the existence of a term t_3 such that $t_1 \to^*_n t_3$ and $t_2 \to^*_m t_3$. A shallow-confluent CTRS is clearly level-confluent, but the reverse does not hold in general. Recently Andreas Werner [23] showed that basic conditional narrowing is complete for

[1] If $\mathcal{R}$ contains rewrite rules that have extra variables in their right-hand sides, the TRSs $\mathcal{R}_n$ may violate the usual restriction $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ imposed on (unconditional) rewrite rules. This doesn't cause us any concern.



level-confluence shallow-confluence

FIGURE 1.

terminating and shallow-confluent 3-CTRSs, but not for all terminating and level-confluent 3-CTRSs. We remark here that our syntactic condition for level-confluence actually gives shallow-confluence.

3. Syntactic Restrictions

In the introduction we saw that 3-CTRSs are not confluent in general, even if they are orthogonal and normal. In this section we present syntactic conditions that ensure (level-)confluence. The first consideration is that we have to severely restrict the many possible terms substituted for extra variables in right-hand sides of the rules.

DEFINITION 3.1. An oriented CTRS $\mathcal{R}$ is called *properly oriented* if every rewrite rule $l \to r \Leftarrow s_1 = t_1, \ldots, s_n = t_n$ with $\mathcal{V}ar(r) \not\subseteq \mathcal{V}ar(l)$ in $\mathcal{R}$ satisfies the following property:

$$\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(t_j)$$

for all $i \in \{1, \ldots, n\}$.

A properly oriented oriented CTRS is simply called a properly oriented CTRS. Clearly every 2-CTRS is properly oriented. For 3-CTRS proper orientedness guarantees that the value of extra variables in the right-hand side is determined by the values of the variables in the left-hand side. So extra variables in a properly oriented CTRS are not really 'extra'. The following example illustrates this point.

EXAMPLE 3.2. Consider the properly oriented 3-CTRS

$$\mathcal{R} = \begin{cases} f(x) & \to & g(x,y,z) & \Leftarrow & h(a,x) = i(y), \quad h(a,y) = i(z) \\ h(a,a) & \to & i(b) \\ h(a,b) & \to & i(c) \\ h(b,b) & \to & i(d) \end{cases}$$

Suppose we rewrite the term $f(a)$ by the first rewrite rule. In this rule y and z are extra variables. The value of y is determined by the condition $h(a,x) = i(y)$ since a is substituted for x and $h(a,a)$ reduces to $i(b)$. So y is bound to b. This determines the value of the extra variable z as $h(a,b)$ reduces to $i(c)$. Hence the term $f(a)$ rewrites only to $g(a,b,c)$.

Since we didn't impose any restrictions on the right-hand side of the conditions so-far, properly oriented orthogonal CTRSs are in general not normal. Bergstra and Klop [3] showed that orthogonal oriented 2-CTRSs are in general not confluent. Hence it is necessary to further restrict the class of properly oriented 3-CTRSs, before we can conclude level-confluence. In order to get a better understanding of such a restriction, we first present a number of counterexamples against the level-confluence of (properly) oriented orthogonal 2-CTRSs.

COUNTEREXAMPLE 3.3. Our first counterexample is a (properly oriented) orthogonal 1-CTRS:

$$\mathcal{R} = \left\{ \begin{array}{rcll} a & \to & b & \\ f(x) & \to & c & \Leftarrow \quad x = a \end{array} \right.$$

We can rewrite the term $f(a)$ both to $f(b)$ and c. These two reducts are not joinable since they are (different) normal forms. Observe that the right-hand side a of the condition $x = a$ is not a constructor term.

COUNTEREXAMPLE 3.4. Consider the (properly oriented) orthogonal 2-CTRS

$$\mathcal{R} = \left\{ \begin{array}{rcll} f(x,y) & \to & a & \Leftarrow \quad h(x,y) = h(z,z) \\ g(x) & \to & f(x,g(x)) & \\ c & \to & g(c) & \end{array} \right.$$

We have $c \to g(c) \to f(c,g(c)) \to f(g(c),g(c)) \to a$ and hence also $c \to g(c) \to^* g(a)$. It is not difficult to show that $g(a)$ does not rewrite to a. It follows that a and $g(a)$ do not have a common reduct. Hence $\mathcal{R}$ is not confluent. Note that the right-hand side $h(z,z)$ of the condition $h(x,y) = h(z,z)$ is not a linear term.

Counterexample 3.4 is a conditional variant of the famous example of Klop [15] showing that left-linearity is essential for confluence in the absence of critical pairs.

COUNTEREXAMPLE 3.5. Next consider the (properly oriented) orthogonal 2-CTRS

$$\mathcal{R} = \left\{ \begin{array}{rcll} a & \to & b & \\ f(x) & \to & c & \Leftarrow \quad a = y, \quad x = y \end{array} \right.$$

and the term $f(a)$ with the two $\mathcal{R}_1$-reducts $f(b)$ and c. These two terms have a common reduct c, but the step from $f(b)$ to c has depth 2. So $\mathcal{R}$ is not level-confluent. Observe how the two right-linear equations $a = y$ and $x = y$ in the conditional part of the second rewrite rule simulate the non-right-linear equation $h(a,x) = h(y,y)$.

COUNTEREXAMPLE 3.6. Finally, consider the (properly oriented) orthogonal 1-CTRS

$$\mathcal{R} = \left\{ \begin{array}{rcll} a & \to & b & \\ f(x) & \to & c & \Leftarrow \quad a = x \end{array} \right.$$

This CTRS is obtained from the CTRS $\mathcal{R}$ of Counterexample 3.3 by simply swapping both sides of the condition. The term $f(a)$ has the two $\mathcal{R}_1$-reducts $f(b)$ and c. These two terms are not joinable in $\mathcal{R}_1$. Note that the right-hand side of the condition has a variable in common with the left-hand side of the second rewrite rule.

Based on the above findings, we introduce the following restriction.

DEFINITION 3.7. A CTRS $\mathcal{R}$ is called *right-stable* if every rewrite rule $l \to r \Leftarrow s_1 = t_1, \ldots, s_n = t_n$ in $\mathcal{R}$ satisfies the following conditions:

$$(\mathcal{V}ar(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(s_j = t_j) \cup \mathcal{V}ar(s_i)) \cap \mathcal{V}ar(t_i) = \emptyset$$

and t_i is either a linear constructor term or a ground $\mathcal{R}_u$-normal form, for all $i \in \{1, \ldots, n\}$.

4. Level-Confluence

In this section we show that orthogonal properly oriented right-stable 3-CTRSs are level-confluent. It is not difficult to see that every normal 2-CTRS is right-stable. Hence our class of CTRSs properly extends the class of orthogonal normal 2-CTRSs (III_n systems in the terminology of [3]) of Bergstra and Klop. They showed that orthogonal normal 2-CTRSs satisfy the so-called parallel moves lemma. Hence these systems are confluent. Giovannetti and Moiso [9] observed that the confluence proof in [3] actually reveals level-confluence. Let us briefly recapitulate the result of Bergstra and Klop.

DEFINITION 4.1. Let $A\colon s \to_{[p,\, l \to r \Leftarrow c]} t$ be a rewrite step in a CTRS $\mathcal{R}$ and let $q \in \mathcal{P}os(s)$. The set $q\backslash A$ of *descendants* of q in t is defined as follows:

$$q\backslash A = \begin{cases} \{\, q \,\} & \text{if } q < p \text{ or } q \parallel p, \\ \{\, p{\cdot}p_3{\cdot}p_2 \mid r_{|p_3} = l_{|p_1} \,\} & \text{if } q = p{\cdot}p_1{\cdot}p_2 \text{ with } p_1 \in \mathcal{P}os_{\mathcal{V}}(l), \\ \varnothing & \text{otherwise.} \end{cases}$$

If $Q \subseteq \mathcal{P}os(s)$ then $Q\backslash A$ denotes the set $\bigcup_{q \in Q} q\backslash A$. The notion of descendant is extended to rewrite sequences in the obvious way.

DEFINITION 4.2. Let $\mathcal{R}$ be a CTRS. We write $s \Vdash_n t$ if t can be obtained from s by contracting a set of pairwise disjoint redexes in s by $\mathcal{R}_n$. We write $s \Vdash t$ if $s \Vdash_n t$ for some $n \geqslant 0$. The minimum such n is called the depth of $s \Vdash t$. The relation $\Vdash$ is called *parallel rewriting*.

The parallel moves lemma for orthogonal normal 2-CTRSs can now be stated as follows.

LEMMA 4.3. *Let $\mathcal{R}$ be an orthogonal normal 2-CTRS. If $t \Vdash_m t_1$ and $t \Vdash_n t_2$ then there exists a term t_3 such that $t_1 \Vdash_n t_3$ and $t_2 \Vdash_m t_3$. Moreover, the redexes contracted in $t_1 \Vdash_n t_3$ ($t_2 \Vdash_m t_3$) are the descendants in t_1 (t_2) of the redexes contracted in $t \Vdash_n t_2$ ($t \Vdash_m t_1$).* □

Unfortunately, the parallel moves lemma does not hold for our class of 3-CTRSs, as shown in the following example.

EXAMPLE 4.4. Consider the properly oriented right-stable 3-CTRS

$$\mathcal{R} = \begin{cases} f(x) \to y \Leftarrow x = y \\ a \to b \\ b \to c \end{cases}$$

and the term $f(a)$. We have $f(a) \to_1 a$ and $f(a) \to_2 c$. From a we can reach the normal form c, but not in a parallel step.

In the above example the depth of the non-parallel sequence from a to c is lower than the depth of the step from $f(a)$ to c. This is the key to level-confluence for orthogonal properly oriented right-stable 3-CTRSs. First we introduce a new relation on terms.

DEFINITION 4.5. Let $\mathcal{R}$ be a CTRS. We write $s \Rightarrow\!\!\!\!\!\!\Rightarrow_n t$ if there exists a set $P \subseteq \mathcal{P}os(s)$ of pairwise disjoint positions such that for all $p \in P$ either
(1) $s_{|p}$ rewrites in a single *root* reduction step to $t_{|p}$ whose depth does not exceed n, or
(2) there exists a rewrite sequence from $s_{|p}$ to $t_{|p}$ whose depth is *less* than n.
Clearly $\Rightarrow\!\!\!\!\!\!\Rightarrow_0$ is the identity relation and $\Rightarrow\!\!\!\!\!\!\Rightarrow_1$ coincides with $\Rightarrow\!\!\!\!\!\!\Rightarrow_1$. We also have $\Rightarrow\!\!\!\!\!\!\Rightarrow_n \subseteq \to_n^* \subseteq \Rightarrow\!\!\!\!\!\!\Rightarrow_{n+1}$ for all $n \geqslant 0$. The infinite union $\Rightarrow\!\!\!\!\!\!\Rightarrow$ of the relations $\Rightarrow\!\!\!\!\!\!\Rightarrow_n$ $(n \geqslant 0)$ is called *extended parallel rewriting*. The depth of an extended parallel rewrite step $s \Rightarrow\!\!\!\!\!\!\Rightarrow t$ is the smallest n such that $s \Rightarrow\!\!\!\!\!\!\Rightarrow_n t$.

In the above example we have $a \Rightarrow\!\!\!\!\!\!\Rightarrow_2 c$. The following result states that in case of orthogonal properly oriented right-stable 3-CTRSs extended parallel rewriting satisfies the (important) first part of the parallel moves lemma.

THEOREM 4.6. *Let $\mathcal{R}$ be an orthogonal properly oriented right-stable 3-CTRS. If $t \Rightarrow\!\!\!\!\!\!\Rightarrow_m t_1$ and $t \Rightarrow\!\!\!\!\!\!\Rightarrow_n t_2$ then there exists a term t_3 such that $t_1 \Rightarrow\!\!\!\!\!\!\Rightarrow_n t_3$ and $t_2 \Rightarrow\!\!\!\!\!\!\Rightarrow_m t_3$.*

PROOF. We use induction on $m + n$. The case $m + n = 0$ is trivial. Suppose the result holds for all m and n such that $m + n < k$ for some $k \geqslant 1$ and consider the case $m + n = k$. Before we proceed, observe that the induction hypothesis implies the validity of the diagrams in Figure 2. If $m = 0$ then $t = t_1$ and we can take $t_3 = t_2$. The case $n = 0$ is just as simple.

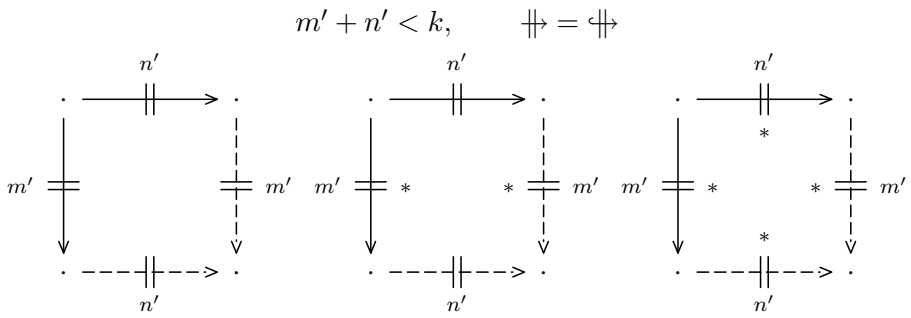


FIGURE 2.

So we may assume that $m, n > 0$. Let P (Q) be the set of positions justifying (according to Definition 4.5) $t \Rightarrow\!\!\!\!\!\!\Rightarrow_m t_1$ $(t \Rightarrow\!\!\!\!\!\!\Rightarrow_n t_2)$. Define a set U of pairwise disjoint positions as follows: $U = \{p \in P \mid \text{there is no } q \in Q \text{ with } q < p\} \cup \{q \in Q \mid \text{there is no } p \in P \text{ with } p < q\}$. Since U dominates all positions in $P \cup Q$, we have $t_1 = t[(t_1)_{|u}]_{u \in U}$ and $t_2 = t[(t_2)_{|u}]_{u \in U}$. Hence for every $u \in U$ we have extended parallel rewrite steps $t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_m (t_1)_{|u}$ and $t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_n (t_2)_{|u}$. If for every $u \in U$ we can show the existence of a term t^u such that $(t_1)_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_n t^u$ and $(t_2)_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_m t^u$, then we can take $t_3 = t[t^u]_{u \in U}$ since $t_1 \Rightarrow\!\!\!\!\!\!\Rightarrow_n t_3$ and $t_2 \Rightarrow\!\!\!\!\!\!\Rightarrow_m t_3$. Let $u \in U$. Without loss of generality we assume that $u \in P$. Let P_u (Q_u) be the set of positions in $t_{|u}$ contributing to $t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_m (t_1)_{|u}$ $(t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_n (t_2)_{|u})$. We have $P_u = \{\varepsilon\}$ and $Q_u = \{q \mid u{\cdot}q \in Q\}$ by assumption. Let m_u (n_u) be the depth of the extended parallel rewrite step $t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow (t_1)_{|u}$ $(t_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow (t_2)_{|u})$. We clearly have $m_u \leqslant m$ and $n_u \leqslant n$. If $m_u + n_u < k$ then we obtain a term t^u such that $(t_1)_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_{n_u} t^u$ and $(t_2)_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_{m_u} t^u$ from the induction hypothesis. Since $m_u \leqslant m$ and $n_u \leqslant n$ this implies the desired $(t_1)_{|u} \Rightarrow\!\!\!\!\!\!\Rightarrow_n t^u$

and $(t_2)_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_m t^u$. So we may assume that $m_u + n_u = k$. This implies that $m_u = m$ and $n_u = n$. By definition the extended parallel rewrite step $t_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_m (t_1)_{|u}$ is either a rewrite step of depth m at root position or a rewrite sequence $t_{|u} \to^* (t_1)_{|u}$ whose depth is $m-1$. In the latter case we have $t_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} (t_1)_{|u}$. The strengthened induction hypothesis (cf. Figure 2) yields a term t^u such that $(t_1)_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_n t^u$ and $(t_2)_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} t^u$. Clearly $(t_2)_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} t^u$ implies $(t_2)_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_m t^u$. So we may assume that $t_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_m (t_1)_{|u}$ is a rewrite step of depth m at root position. There exists a rewrite rule $l \to r \Leftarrow c$ and a substitution σ such that $t_{|u} = l\sigma$, $(t_1)_{|u} = r\sigma$, and $\mathcal{R}_{m-1} \vdash c\sigma$. Let c be the sequence of conditions $s_1 = t_1, \ldots, s_j = t_j$. For every $i \in \{0, \ldots, j\}$ let c_i be the subsequence of c consisting of the first i conditions. (So c_0 is the empty sequence.) We distinguish two cases.

(1) Suppose $Q_u = \{\varepsilon\}$. By definition the extended parallel rewrite step $t_{|u} \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}_n (t_2)_{|u}$ is either a rewrite step of depth n at root position or a rewrite sequence $t_{|u} \to^* (t_2)_{|u}$ whose depth is $n-1$. The latter case follows from the strengthened induction hypothesis, exactly as above. So we may assume that $t_{|u}$ rewrites in a single root $\mathcal{R}_n$-step to $(t_2)_{|u}$. By orthogonality the employed rewrite rule of $\mathcal{R}$ must be $l \to r \Leftarrow c$. Let τ be a substitution such that $t_{|u} = l\tau$, $(t_2)_{|u} = r\tau$, and $\mathcal{R}_{n-1} \vdash c\tau$. If $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ then $r\sigma = r\tau$ and we simply define $t^u = r\sigma$. If the right-hand side r of the rewrite rule contains extra variables, we reason as follows. By induction on $i \in \{0, \ldots, j\}$ we show the existence of a substitution ρ_i such that

   (a) $\rho_i = \sigma = \tau \ [\mathcal{V}ar(l)]$,
   (b) $\mathcal{D}(\rho_i) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$,
   (c) $x\sigma \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{n-1} x\rho_i$ and $x\tau \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} x\rho_i$ for every variable $x \in \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$.

   The substitution $\rho_0 = \sigma{\restriction_{\mathcal{V}ar(l)}}$ clearly satisfies the requirements for $i = 0$. Suppose $i > 0$ and consider the i-th condition $s_i = t_i$. Because $\mathcal{R}$ is properly oriented we have $\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1})$. Using part (c) of the induction hypothesis, we easily obtain $s_i\sigma \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{n-1} s_i\rho_{i-1}$ and $s_i\tau \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} s_i\rho_{i-1}$. Two applications of the strengthened induction hypothesis (cf. Figure 2) yields a term s' such that $t_i\sigma \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{n-1} s'$ and $t_i\tau \mathrel{\Vdash\!\!\!\!\!\!\rightarrow}^*_{m-1} s'$, see Figure 3. From the right-stability of $\mathcal{R}$ we learn t_i is either a ground $\mathcal{R}_u$-normal form or a linear

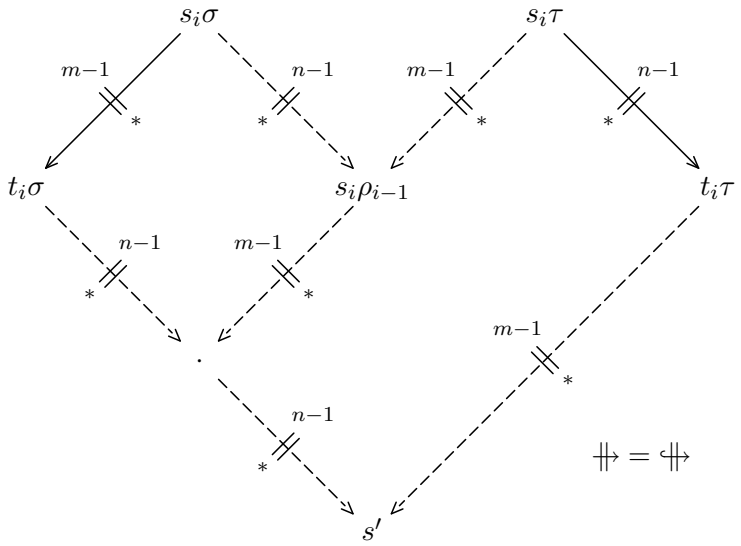


FIGURE 3.

constructor term. In the former case we have $t_i\sigma = s' = t_i\tau$ and hence the substitution

$\rho_i = \rho_{i-1}$ satisfies the three requirements. (Note that $\mathcal{V}ar(c_i) = \mathcal{V}ar(c_{i-1})$ in this case.) In the latter case there must be a substitution ρ such that $s' = t_i\rho$ with $\mathcal{D}(\rho) \subseteq \mathcal{V}ar(t_i)$. Right-stability yields $\mathcal{V}ar(t_i) \cap (\mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1}))) = \varnothing$. As a consequence $\rho_i = \rho_{i-1} \cup \rho$ is a well-defined substitution. It is easy to see that ρ_i satisfies the requirements (a)–(c). This concludes the induction step. Now consider the substitution ρ_j. Since $\mathcal{R}$ is a 3-CTRS we have $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_j)$. Hence, using property (c), we obtain $(t_1)_{|u} = r\sigma \Downarrow^*_{n-1} r\rho_j$ and $(t_2)_{|u} = r\tau \Downarrow^*_{m-1} r\rho_j$. Since $\Downarrow^*_{n-1} \subseteq \Downarrow_n$ and $\Downarrow^*_{m-1} \subseteq \Downarrow_m$, we can define $t^u = r\rho_j$.

(2) The second case is $Q_u \neq \{\varepsilon\}$. Using the orthogonality of $\mathcal{R}$ we obtain a substitution τ such that $(t_2)_{|u} = l\tau$, $\mathcal{D}(\tau) \subseteq \mathcal{V}ar(l)$, and $x\sigma \Downarrow_n x\tau$ for every $x \in \mathcal{V}ar(l)$. In this case there is no need to distinguish $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ from $\mathcal{V}ar(r) \not\subseteq \mathcal{V}ar(l)$. By induction on $i \in \{0, \ldots, j\}$ we show the existence of a substitution ρ_i such that

(a) $\rho_i = \tau\ [\mathcal{V}ar(l)]$,
(b) $\mathcal{D}(\rho_i) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$,
(c) $\mathcal{R}_{m-1} \vdash c_i\rho_i$, and
(d) $x\sigma \Downarrow_n x\rho_i$ for every variable $x \in \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$.

If $i = 0$ then $\rho_0 = \tau{\restriction_{\mathcal{V}ar(l)}}$ satisfies the requirements. Suppose $i > 0$ and consider the i-th condition $s_i = t_i$. From $\mathcal{R}_{m-1} \vdash c\sigma$ we infer that $s_i\sigma \to^*_{m-1} t_i\sigma$. Hence also $s_i\sigma \Downarrow^*_{m-1} t_i\sigma$. Let $V = \mathcal{V}ar(s_i) - (\mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1}))$ and define the substitution ρ as the (disjoint) union of ρ_{i-1} and $\sigma{\restriction_V}$. Using part (d) of the induction hypothesis we learn that $s_i\sigma \Downarrow_n s_i\rho$. The strengthened induction hypothesis (cf. Figure 2) yields a term t' such that $s_i\rho \Downarrow^*_{m-1} t'$ and $t_i\sigma \Downarrow_n t'$. From the right-stability of $\mathcal{R}$ we learn that t_i is either a ground $\mathcal{R}_u$-normal form or a linear constructor term. In the former case we have $t_i\sigma = t_i = t'$ and hence the substitution $\rho_i = \rho$ satisfies the four requirements. In the latter case there must be a substitution ρ' such that $t' = t_i\rho'$ with $\mathcal{D}(\rho') \subseteq \mathcal{V}ar(t_i)$. Right-stability yields $\mathcal{V}ar(t_i) \cap (\mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1}) \cup \mathcal{V}ar(s_i)) = \varnothing$. Hence $\rho_i = \rho \cup \rho'$ is a well-defined substitution. It is not difficult to check that ρ_i satisfies the requirements (a)–(d). For instance, we have $s_i\rho_i = s_i\rho$ and $t_i\rho_i = t_i\rho'$, hence $\mathcal{R}_{m-1} \vdash (s_i = t_i)\rho'$ follows from $s_i\rho \Downarrow^*_{m-1} t_i\rho'$. This concludes the induction step. Now consider the substitution ρ_j. We have $l\tau = l\rho_j$ by property (a) and $\mathcal{R}_{m-1} \vdash c\rho_j$ by property (c). Therefore $(t_2)_{|u} = l\tau \to_m r\rho_j$ and thus $(t_2)_{|u} \Downarrow_m r\rho_j$. Since $\mathcal{R}$ is a 3-CTRS we have $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_j)$. Hence, using property (d), we obtain $(t_1)_{|u} = r\sigma \Downarrow_n r\rho_j$. Thus $r\rho_j$ is the desired term t^u.

So for every $u \in U$ we could define the desired term t^u. This concludes the proof. □

The main result of this paper is an immediate consequence of the above theorem.

COROLLARY 4.7. *Orthogonal properly oriented right-stable 3-CTRSs are level-confluent.* □

5. Join Systems

We extend the result of the previous section to join 3-CTRSs. The (only) difference between join and oriented CTRSs is that the equality signs in the conditions of the rewrite rules is interpreted differently: $\downarrow$ in the case of join CTRSs and $\to^*$ in the case of oriented CTRSs. In the following we make the explicit notational convention of writing $\mathcal{R}^j$ ($\mathcal{R}^o$) if the CTRS $\mathcal{R}$ is considered as a join (oriented) CTRS.

For an arbitrary CTRS $\mathcal{R}$, the rewrite relation associated with the oriented CTRS $\mathcal{R}^o$ is in general a proper subset of the one associated with the join variant $\mathcal{R}^j$. Consider for example the CTRS

$$\mathcal{R} = \left\{ \begin{array}{lcll} a & \to & x & \Leftarrow \quad c = x \\ b & \to & c & \end{array} \right.$$

We have $a \to_{\mathcal{R}^j} b$ because $c \downarrow_{\mathcal{R}^j} b$, but $a \to_{\mathcal{R}^o} b$ does not hold because c does not rewrite to b (in $\mathcal{R}^o$). Nevertheless $a \downarrow_{\mathcal{R}^o} b$ holds since $a \to_{\mathcal{R}^o} c$ and $b \to_{\mathcal{R}^o} c$. This relationship ($\to_{\mathcal{R}^j} \subseteq \downarrow_{\mathcal{R}^o}$) holds for all orthogonal properly oriented right-stable 3-CTRSs, as will be shown below. First we prove a special case.

LEMMA 5.1. *Let $\mathcal{R}$ be an orthogonal properly oriented right-stable 3-CTRS. If $s \to_{\mathcal{R}^j_n} t$ by application of a rewrite rule $l \to r \Leftarrow c$ with substitution σ such that $s'\sigma \downarrow_{\mathcal{R}^o_{n-1}} t'\sigma$ for every equation $s' = t'$ in c then $s \downarrow_{\mathcal{R}^o_n} t$.*

PROOF. Let c be the sequence of conditions $s_1 = t_1, \ldots, s_j = t_j$. For every $i \in \{1, \ldots, j\}$ let c_i be the subsequence of c consisting of the first i conditions. By induction on $i \in \{0, \ldots, j\}$ we show the existence of a substitution τ_i such that

(a) $\tau_i = \sigma\ [\mathcal{V}ar(l)]$,
(b) $\mathcal{D}(\tau_i) \subseteq \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$,
(c) $\mathcal{R}^o_{n-1} \vdash c_i\tau_i$, and
(d) $x\sigma \to^*_{\mathcal{R}^o_{n-1}} x\tau_i$ for every variable $x \in \mathcal{V}ar(l) \cup \mathcal{V}ar(c_i)$.

The substitution $\tau = \sigma|_{\mathcal{V}ar(l)}$ clearly satisfies the requirements for $i = 0$. Suppose $i > 0$ and consider the i-th condition $s_i = t_i$. By assumption $s_i\sigma$ and $t_i\sigma$ have a common $\mathcal{R}^o_{n-1}$-reduct, say t'. Let $V = \mathcal{V}ar(s_i) - (\mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1}))$ and define the substitution ρ as $\tau_{i-1} \cup \sigma|_V$. Using part (d) of the induction hypothesis we obtain an $\mathcal{R}^o_{n-1}$-rewrite sequence from $s_i\sigma$ to $s_i\rho$. According to Corollary 4.7 $\mathcal{R}^o$ is level-confluent. This implies that the $\mathcal{R}^o_{n-1}$-reducts t' and $s_i\rho$ of $s_i\sigma$ have a common $\mathcal{R}^o_{n-1}$-reduct, say t''. From the right-stability of $\mathcal{R}$ we learn that t_i is either a ground $\mathcal{R}_u$-normal form or a linear constructor term. In the former case we have $t_i\sigma = t_i = t' = t''$ and hence the substitution $\tau_i = \rho$ satisfies the four requirements. In the latter case there must be a substitution ρ' such that $t'' = t_i\rho'$ with $\mathcal{D}(\rho') \subseteq \mathcal{V}ar(t_i)$. Right-stability yields $\mathcal{V}ar(t_i) \cap (\mathcal{V}ar(l) \cup \mathcal{V}ar(c_{i-1})) = \varnothing$. Hence $\tau_i = \rho \cup \rho'$ is a well-defined substitution which is easily seen to be satisfying the requirements (a)–(d). This concludes the induction step. Consider the substitution τ_j. We have an $\mathcal{R}^o_{n-1}$-rewrite sequence from $r\sigma$ to $r\tau_j$ as a consequence of property (d). From properties (a) and (c) we learn that $l\sigma = l\tau_j \to_{\mathcal{R}^o_{n-1}} r\tau_j$. Therefore $s \downarrow_{\mathcal{R}^o_{n-1}} t$. □

LEMMA 5.2. *Let $\mathcal{R}$ be an orthogonal properly oriented right-stable 3-CTRS. If $s \to_{\mathcal{R}^j_n} t$ then $s \downarrow_{\mathcal{R}^o_n} t$.*

PROOF. We use induction on n. If $n = 0$ then we have nothing to prove, so suppose $n > 0$. Let $l \to r \Leftarrow c$ be the rewrite rule and σ the substitution used in $s \to_{\mathcal{R}^j_n} t$. We have $\mathcal{R}^j_{n-1} \vdash c\sigma$. Using Corollary 4.7, we obtain $\mathcal{R}^o_{n-1} \vdash c\sigma$ from the induction hypothesis by a routine induction argument. Lemma 5.1 yields the desired $s \downarrow_{\mathcal{R}^o_n} t$. □

The main result of this section is an easy consequence of Corollary 4.7 and Lemma 5.2.

COROLLARY 5.3. *Orthogonal properly oriented right-stable join 3-CTRSs are level-confluent.* □

6. Related Work

Bertling and Ganzinger [4] defined the class of *quasi-reductive* and *deterministic* 3-CTRSs. Deterministic CTRSs are very similar to properly oriented CTRSs, the difference being that we don't impose the restrictions ($i \in \{1, \ldots, n\}$)

$$\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(s_j = t_j)$$

when the right-hand side r of the rewrite rule $l \to r \Leftarrow s_1 = t_1, \ldots, s_n = t_n$ doesn't contain extra variables. So deterministic CTRSs are a proper subclass of properly oriented CTRS. Quasi-reductivity is an (undecidable) criterion guaranteeing termination. Bertling and Ganzinger showed that every quasi-reductive deterministic 3-CTRS has a decidable rewrite relation.

In a recent paper Avenhaus and Loría-Sáenz [2] provide a simple but powerful decidable condition for quasi-reductivity. Moreover, they show that every *strongly* deterministic and quasi-reductive 3-CTRS with joinable critical pairs[2] is confluent. A strongly deterministic CTRS $\mathcal{R}$ is a deterministic one with the additional property that for every right-hand side t of the equations in the conditional parts of the rewrite rules and every normalized substitution σ, the term $t\sigma$ is a normal form. Observe that both constructor terms and ground $\mathcal{R}_u$-normal forms satisfy this property. Our result remains valid (with the same proof) if we relax the second condition in the definition of right-stability to the requirement that every right-hand side t of the equations in the conditional parts of the rewrite rules is a *linear* term with the property that $t\sigma$ is a normal form whenever σ is a normalized substitution. Note though that this is no longer a decidable criterion. Every right-stable deterministic CTRS is strongly deterministic but not vice-versa, e.g., the 3-CTRS of Counterexample 3.4 is strongly deterministic but not right-stable. So the class of strongly deterministic quasi-reductive 3-CTRSs is incomparable with our class of orthogonal properly oriented right-stable 3-CTRSs. The essential difference however is that we do not impose the termination restriction. From a programming point of view, the termination assumption is quite severe. Strongly deterministic quasi-reductive 3-CTRSs with joinable critical pairs are in general not level-confluent. For instance, the 3-CTRS

$$\mathcal{R} = \begin{cases} a \to b \\ a \to c \\ b \to c \Leftarrow d = e \\ d \to e \end{cases}$$

is clearly strongly deterministic. The reduction order $a \succ b \succ c \succ d \succ e$ can be used to show quasi-reductivity. We have $a \to_1 b$ and $a \to_1 c$, but the step from b to c has depth 2. Observe that $\mathcal{R}$ is properly oriented and right-stable. Hence we know that the system cannot be orthogonal, and indeed there is a critical pair between the first two rules. Finally, we would like to mention that the termination assumption makes the task of proving confluence easier.

Very recently (and independently) Hanus [13] presented a sufficient condition for the level-confluence of properly oriented orthogonal 3-CTRSs with *strict equality*. Strict equality means that a substitution σ satisfies a condition $s = t$ of a rewrite rule only if $s\sigma$ and $t\sigma$ reduce to the same ground constructor term. CTRSs with strict equality can be viewed as a special case

[2] Overlays obtained from a rewrite rule and (a renamed version of) itself don't have to be considered.

of normal CTRSs. The proof in [13] is however insufficient since it is based on the parallel moves lemma for orthogonal normal 2-CTRSs (Bergstra and Klop [3]), which, as we have seen in Section 4, is not valid for 3-CTRSs. Our proof method can be specialized to complete Hanus' proof. Actually, Hanus considers *almost orthogonal* ([1]) CTRSs. These are left-linear systems in which the non-overlapping restriction is relaxed by allowing trivial overlays. Although we only considered orthogonal CTRSs in this paper, our result immediately extends to almost orthogonal systems.

7. Concluding Remarks

In this final section we discuss some further extensions of our sufficient condition. First of all, Dershowitz *et al.* [7] and Gramlich [11], among others, distinguish feasible from infeasible critical pairs. A critical pair is *infeasible* if the instantiated combination of the two conditional parts of the rewrite rules that induce the critical pair is unsolvable. Infeasibility is undecidable in general, but see González–Moreno *et al.* [10] for a decidable sufficient condition. Infeasible critical pairs are harmless, so orthogonality can be strengthened by allowing infeasible critical pairs. This is important in practice since it permits systems like

$$\left\{\begin{array}{rcll} div(0,S(x)) & \to & (0,0) & \\ div(S(x),S(y)) & \to & (0,S(x)) & \Leftarrow \quad x < y = true \\ div(S(x),S(y)) & \to & (S(q),r) & \Leftarrow \quad x \geqslant y = true, \\ & & & \quad div(x-y,S(y)) = (q,r) \end{array}\right.$$

with disambiguating conditions. Because the condition $x < y = true,\ x \geqslant y = true$ has no solutions, the critical pair between the last two rules is infeasible. The proofs in this paper remain valid if we allow infeasible critical pairs. Secondly, satisfiability of the conditional part of a rewrite rule is independent of the order of its equations. Hence we can relax proper orientedness and right-stability by allowing any permutation of the equations in the conditions to satisfy the requirements in Definitions 3.1 and 3.7. Finally, and more in spirit with the results of this paper, we consider relaxing the proper orientedness restriction. Proper orientedness requires that extra variables in the left-hand side of some equation in the conditional part of some rewrite rule occur in the preceding equations. This requirement is not necessary, however, if the extra variable under consideration does not affect the values of extra variables that occur in the right-hand side of the rewrite rule. This leads us to the following definition.

Definition 7.1. In an *extended properly oriented* CTRS the conditional part c of every rewrite rule $l \to r \Leftarrow c$ with $\mathcal{V}ar(r) \not\subseteq \mathcal{V}ar(l)$ can be written as $s_1 = t_1, \ldots, s_m = t_m, s'_1 = t'_1, \ldots, s'_n = t'_n$ such that the following two conditions are satisfied:

$$\mathcal{V}ar(s_i) \subseteq \mathcal{V}ar(l) \cup \bigcup_{j=1}^{i-1} \mathcal{V}ar(t_j)$$

for all $i \in \{1, \ldots, m\}$, and

$$\mathcal{V}ar(r) \cap \mathcal{V}ar(s'_i = t'_i) \subseteq \mathcal{V}ar(l) \cup \bigcup_{j=1}^{m} \mathcal{V}ar(t_j)$$

for all $i \in \{1, \ldots, n\}$.

The proofs in this paper can be adapted to deal with extended properly oriented CTRSs. Hence all orthogonal extended properly oriented right-stable 3-CTRSs are level-confluent. An interesting application is the class of 3-CTRSs with strict equality and so-called *local definitions*. With respect to Definition 7.1 this means that t'_i denotes the constant *true* or *false* and s'_i the strict equation $s'_{1i} \equiv s'_{2i}$, for $i \in \{1, \ldots, n\}$. Here $\equiv$ is a function that tests whether its two arguments denote the same ground constructor term. The local definitions $s_1 = t_1, \ldots, s_m = t_m$ support (potentially) infinite data structures. An interesting example is the following Haskell like program:

```
filter n r []     = (-1, [])
filter n r (x:xs) = (x, xs),    if divide x n == (q, r)
                  = (y, x:ys),  otherwise
                    where (y, ys) = filter n r xs
```

The idea is that evaluating `filter n r l` filters out a natural number x such that x *mod* `n` $=$ `r` from the list of natural numbers `l`. This program can handle (potentially) infinite lists. The function `filter` corresponds to the following orthogonal (in the sense of having only infeasible critical pairs) 3-CTRS with strict equality and local definitions:

$$\left\{\begin{array}{lcll} \mathit{filter}(n, r, \mathit{nil}) & \to & (-1, \mathit{nil}) & \\ \mathit{filter}(n, r, x : xs) & \to & (x, xs) & \Leftarrow \quad \mathit{div}(x, n) \equiv (q, r) = \mathit{true} \\ \mathit{filter}(n, r, x : xs) & \to & (y, x : ys) & \Leftarrow \quad \mathit{filter}(n, r, xs) = (y, ys), \\ & & & \qquad\quad \mathit{div}(x, n) \equiv (q, r) = \mathit{false} \end{array}\right.$$

This system is extended properly oriented and right-stable but not properly oriented due to the presence of the extra variable q in the last rewrite rule.

Acknowledgements. We thank Michael Hanus for scrutinizing a previous version of the paper. Vincent van Oostrom simplified most of the counterexamples in the (LNCS version of the) paper.

References

1. S. Antoy and A. Middeldorp, *A Sequential Reduction Strategy*, Proceedings of the 4th International Conference on Algebraic and Logic Programming, Madrid, Lecture Notes in Computer Science **850**, pp. 168–185, 1994.

2. J. Avenhaus and C. Loría-Sáenz, *On Conditional Rewrite Systems with Extra Variables and Deterministic Logic Programs*, Proceedings of the 5th International Conference on Logic Programming and Automated Reasoning, Kiev, Lecture Notes in Artificial Intelligence **822**, pp 215–229, 1994.

3. J.A. Bergstra and J.W. Klop, *Conditional Rewrite Rules: Confluence and Termination*, Journal of Computer and System Sciences **32**(3), pp. 323–362, 1986.

4. H. Bertling and H. Ganzinger, *Completion-Time Optimization of Rewrite-Time Goal Solving*, Proceedings of the 3rd International Conference on Rewriting Techniques and Applications, Chapel Hill, Lecture Notes in Computer Science **355**, pp. 45–58, 1989.

5. N. Dershowitz and J.-P. Jouannaud, *Rewrite Systems*, in: Handbook of Theoretical Computer Science, Vol. B ed. J. van Leeuwen), North-Holland, pp. 243–320, 1990.

6. N. Dershowitz and M. Okada, *A Rationale for Conditional Equational Programming*, Theoretical Computer Science **75**, pp. 111–138, 1990.

7. N. Dershowitz, M. Okada, and G. Sivakumar, *Confluence of Conditional Rewrite Systems*, Proceedings of the 1st International Workshop on Conditional Term Rewriting Systems, Orsay, Lecture Notes in Computer Science **308**, pp. 31–44, 1987.

8. E. Giovannetti, G. Levi, C. Moiso, and C. Palamidessi, *Kernel-LEAF: A Logic plus Functional Language*, Journal of Computer and System Sciences **42**(2), pp. 139–185, 1991.

9. E. Giovannetti and C. Moiso, *A Completeness Result for E-Unification Algorithms based on Conditional Narrowing*, Proc. Workshop on Foundations of Logic and Functional Programming, Trento, Lecture Notes in Computer Science **306**, pp. 157–167, 1986.

10. Juan Carlos González-Moreno, M.T. Hortalá-González, M. Rodríguez-Artalejo, *Denotational versus Declarative Semantics for Functional Programming*, Proceedings of the 5th Workshop on Computer Science Logic, Berne, Lecture Notes in Computer Science **626**, pp. 134–148, 1992.

11. B. Gramlich, *On Termination and Confluence of Conditional Rewrite Systems*, Proceedings of the 4th International Workshop on Conditional Term Rewriting Systems, Jerusalem, Lecture Notes in Computer Science, 1995. To appear.

12. M. Hanus, *The Integration of Functions into Logic Programming: From Theory to Practice*, Journal of Logic Programming **19 & 20**, pp. 583–628, 1994.

13. M. Hanus, *On Extra Variables in (Equational) Logic Programming*, report MPI-I-94-246, Max-Planck-Institut für Informatik, 1994. To appear in the Proceedings of the 12th International Conference on Logic Programming, Tokyo, MIT Press, 1995.

14. T. Ida and S. Okui, *Outside-In Conditional Narrowing*, IEICE Transactions on Information and Systems, **E77-D**(6), pp. 631–641, 1994.

15. J.W. Klop, *Combinatory Reduction Systems*, Ph.D. thesis, Mathematical Centre Tracts **127**, Centre for Mathematics and Computer Science, Amsterdam, 1980.

16. J.W. Klop, *Term Rewriting Systems*, in: Handbook of Logic in Computer Science, Vol. II (eds. S. Abramsky, D. Gabbay and T. Maibaum), Oxford University Press, pp. 1–116, 1992.

17. A. Middeldorp, *Completeness of Combinations of Conditional Constructor Systems*, Journal of Symbolic Computation **17**, pp. 3–21, 1994.

18. A. Middeldorp and E. Hamoen, *Completeness Results for Basic Narrowing*, Applicable Algebra in Engineering, Communication and Computing **5**, pp. 213–253, 1994.

19. J.J. Moreno-Navarro and M. Rodrìguez-Artalejo, *Logic Programming with Functions and Predicates: The Language BABEL*, Journal of Logic Programming **12**, pp. 191–223, 1992.

20. E. Ohlebusch, *Modular Properties of Constructor-Sharing Conditional Term Rewriting Systems*, Proceedings of the 4th International Workshop on Conditional Term Rewriting Systems, Jerusalem, Lecture Notes in Computer Science, 1995. To appear.

21. P. Padawitz, *Generic Induction Proofs*, Proceedings of Third International Workshop on Conditional Term Rewriting Systems, Pont–à-Mousson, Lecture Notes in Computer Science **656**, pp. 175–197, 1993.

22. Y. Toyama and M. Oyamaguchi, *Church-Rosser Property and Unique Normal Form Property of Non-Duplicating Term Rewriting Systems*, Proceedings of the 4th International Workshop on Conditional Term Rewriting Systems, Jerusalem, Lecture Notes in Computer Science, 1995. To appear.

23. A. Werner, personal communication, January 1995.