Ground Confluence Proof with Pattern Complementation*

Takahito Aoto[1] and Yoshihito Toyama[2]

[1] Faculty of Engineering, Niigata University
aoto@ie.niigata-u.ac.jp
[2] RIEC, Tohoku University
toyama@nue.riec.tohoku.ac.jp

Abstract

In (Aoto&Toyama, FSCD 2016), we gave a procedure for proving ground confluence of many-sorted TRSs based on rewriting induction for proving bounded ground convertibility. The procedure needs to find a strongly quasi-reducible terminating set of rules from the given input TRS to make the rewriting induction work. It turns out, however, that such a subset of rewrite rules is often not present in the input TRS. In this note, we propose an improvement of the procedure; in the new procedure, firstly the lack of defining patterns is detected using pattern complementation procedure (Lazrek et al., I&C 1990), and then possible defining rules that can be appended to obtain a strongly quasi-reducible terminating TRS are searched. The new procedure is useful to prove ground confluence of some TRSs automatically which have been failed in the previous procedure.

1 Introduction

A term rewriting system (TRS for short) is ground confluent if all ground terms are confluent. Procedures for proving ground confluence have been studied in e.g. [2, 5, 3, 4]. In [1], a procedure for proving ground confluence of many-sorted TRSs based on rewriting induction, aiming for proving bounded ground convertibility. For making the rewriting induction work, the procedure needs to find a strongly quasi-reducible terminating set of rules from the given input TRS. It turns out, however, that such a subset of rewrite rules is often not present in the input TRS.

In this note, we propose an improvement of the procedure. In our new procedure, firstly the lack of defining patterns is detected using pattern complementation procedure [6]. Then the rewrite rules to define such pattern are searched by combining multiple rewrite steps of the input TRS. Then founded rewrite rules are added to the input TRS to form a strongly quasi-reducible terminating subset of rewrite rules so that the rewriting induction can work on it. We also report a result of preliminary experiment.

2 Preliminaries

We assume basic familiarity with (many-sorted) term rewriting (e.g. [7]).

The transitive reflexive (reflexive, symmetric, reflexive symmetric, equivalence) closure of a relation $\to$ is denoted by $\overset{*}{\to}$ (resp. $\overset{=}{\to}$, $\leftrightarrow$, $\overset{=}{\leftrightarrow}$, $\overset{*}{\leftrightarrow}$). For any quasi-order $\gtrsim$, we put $\succ\, = \,\gtrsim \setminus \lesssim$ and $\approx\, = \,\gtrsim \cap \lesssim$. A quasi-order $\gtrsim$ is *well-founded* if so is its strict part $\succ$.

Let $\mathcal{S}$ be a set of *sorts*. Each *many-sorted function* f is equipped with its *sort declaration* $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha_0$, where $\alpha_0, \ldots, \alpha_n \in \mathcal{S}$ $(n \geq 0)$. The set of *terms* over the set of many-sorted function symbols $\mathcal{F}$ and the set of variables $\mathcal{V}$ is denoted by $\mathrm{T}(\mathcal{F}, \mathcal{V})$. The set of function symbols (variables) contained in a term t is denoted by $\mathcal{F}(t)$ (resp. $\mathcal{V}(t)$). The set of *ground*

*This work is partially supported by JSPS KAKENHI (Nos. 15K00003, 25280025).

terms over $\mathcal{G} \subseteq \mathcal{F}$ is denoted by $\mathrm{T}(\mathcal{G})$. A *ground substitution* is a mapping from $\mathcal{V}$ to $\mathrm{T}(\mathcal{F})$. A *rewrite relation (quasi-order)* is a relation (resp. quasi-order) on terms *closed under contexts and substitutions*. A rewrite relation (quasi-order) is a *reduction* relation (resp. quasi-order) if it is well-founded. (Indirected) *equations* $l \doteq r$ and $r \doteq l$ are identified. A directed equation $l \to r$ is a *rewrite rule* if $l \notin \mathcal{V}$ and $\mathcal{V}(l) \supseteq \mathcal{V}(r)$ hold. A (many-sorted) *term rewriting system* (*TRS* for short) is a finite set of rewrite rules. The smallest rewrite relation containing $\mathcal{R}$ is denoted by $\to_\mathcal{R}$. The set of *critical pairs* of a TRS $\mathcal{R}$ is denoted by $\mathrm{CP}(\mathcal{R})$.

Terms s and t are *joinable* w.r.t. the rewrite relation $\to_\mathcal{R}$ (denoted by $s \downarrow_\mathcal{R} t$) if $s \xrightarrow{*}_\mathcal{R} u$ and $t \xrightarrow{*}_\mathcal{R} u$ for some u. A TRS $\mathcal{R}$ is *(ground) confluent* if $s \downarrow_\mathcal{R} t$ holds for any (ground) terms s, t such that $u \xrightarrow{*}_\mathcal{R} s$ and $u \xrightarrow{*}_\mathcal{R} t$ for some (resp. ground) term u. Terms s and t are *ground convertible* if $s\sigma_g \overset{*}{\leftrightarrow}_\mathcal{R} t\sigma_g$ holds for any ground substitution σ_g. An equation $s \doteq t$ is an *inductive theorem* of a TRS $\mathcal{R}$, or *inductively valid* in $\mathcal{R}$, if s and t are ground convertible. We write $\mathcal{R} \models_{ind} E$ for a set E of equations (pairs, rewrite rules) if every equation $s \doteq t$ is an inductive theorem for any $s \doteq t$ ($\langle s, t\rangle$, $s \to t$) in E.

We consider a partition of function symbols into the set $\mathcal{D}$ of *defined symbols*, and the set $\mathcal{C}$ of *constructors* i.e. $\mathcal{F} = \mathcal{D} \uplus \mathcal{C}$. Terms in $\mathrm{T}(\mathcal{C}, V)$ are *constructor terms*. Then a mapping from $\mathcal{V}$ to $\mathrm{T}(\mathcal{C})$ is called a *ground constructor substitution*. The set of *ground basic terms* is defined by $\mathrm{T_B}(\mathcal{D},\mathcal{C}) = \{f(c_1,\ldots,c_n) \mid f \in \mathcal{D}, c_i \in \mathrm{T}(\mathcal{C})\}$. A TRS $\mathcal{R}$ is said to be *quasi-reducible* if no ground basic terms are normal. Clearly, if $\mathcal{R}$ is a quasi-reducible terminating TRS then for any ground term s there exists t such that $s \xrightarrow{*} t \in \mathrm{T}(\mathcal{C})$.

3 Ground Confluence Proof by Rewriting Induction

In [1], the authors give a system of rewriting induction for proving ground confluence of many-sorted term rewriting systems. The procedure is described as follows:

GCR Procedure 1

Input: TRS $\mathcal{R}$
Output: `YES` or `MAYBE`

1. Compute (possibly multiple) candidates for the partition $\mathcal{F} = \mathcal{D} \uplus \mathcal{C}$ of function symbols.
2. Compute (possibly multiple) candidates for strongly quasi-reducible $\mathcal{R}_0 \subseteq \mathcal{R}$.
3. Find a reduction quasi-order $\succsim$ such that $\mathcal{R}_0 \subseteq \succ$.
4. Run rewriting induction for proving bounded ground convertibility of $\mathrm{CP}(\mathcal{R}_0)$ with $\succsim$.
5. Run rewriting induction for proving $\mathcal{R}_0 \models_{ind} (\mathcal{R} \setminus \mathcal{R}_0)$.
6. Return `YES` if it succeeds in steps 4 and 5, otherwise `MAYBE`.

Note here that strong quasi-reducibility [1] and quasi-reducibility coincide when constructors are free, i.e. $\mathcal{D} = \{l(\epsilon) \mid l \to r \in \mathcal{R}\}$. To make the explanation simple, here after we only consider free constructors.

Proposition 1 ([1])**.** *If GCR Procedure 1 returns* `YES` *then* $\mathcal{R}$ *is ground confluent.*

As indicated above, for the procedure shown to work, it is required that there exists (strongly) quasi-reducible and terminating subset $\mathcal{R}_0 \subseteq \mathcal{R}$. Experiments in [1], however, reveal that there are cases that there does not exist such an $\mathcal{R}_0$.

Example 2 (Cops 128)**.** *Let* $\mathcal{F} = \{\mathsf{plus} : \mathsf{Nat} \times \mathsf{Nat} \to \mathsf{Nat}, \mathsf{s} : \mathsf{Nat} \to \mathsf{Nat}, \mathsf{0} : \mathsf{Nat}\}$ *and*

$$\mathcal{R} = \left\{ \begin{array}{lcll} \mathsf{plus}(\mathsf{0}, y) & \to & y & (a) \\ \mathsf{plus}(x, \mathsf{s}(y)) & \to & \mathsf{s}(\mathsf{plus}(x, y)) & (b) \\ \mathsf{plus}(x, y) & \to & \mathsf{plus}(y, x) & (c) \end{array} \right\}$$

Then there exists no quasi-reducible and terminating subsets of $\mathcal{R}$*.*

A natural candidate of quasi-reducible terminating $\mathcal{R}_0$ here would be

$$\mathcal{R}_0 = \left\{ \begin{array}{lcll} \mathsf{plus}(\mathsf{0}, y) & \to & y & (a) \\ \mathsf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathsf{plus}(x, y)) & (b') \end{array} \right\}$$

Indeed, the rewrite rule (b') is equationally valid as

$$\mathsf{plus}(\mathsf{s}(x), y) \to_{(c)} \mathsf{plus}(y, \mathsf{s}(x)) \to_{(b)} \mathsf{s}(\mathsf{plus}(y, x)) \to_{(c)} \mathsf{s}(\mathsf{plus}(x, y))$$

However, the rewrite rule (b') is not contained in $\mathcal{R}$ and thus the procedure given in [1] fails to prove ground confluence of this system.

4 Ground Confluence Proof with Pattern Complementation

A finite set P of basic terms is called a *pattern*. Intuitively, the set P can be regarded as expressing a set of ground terms given as $\mathrm{Inst}(P) = \{p\sigma_{gc} \mid p \in P, \sigma_{gc} : \mathcal{V} \to \mathrm{T}(\mathcal{C})\}$. A finite set Q of terms is said to be a *complement* (w.r.t. $\mathrm{T_B}(\mathcal{D}, \mathcal{C})$) of P if $\mathrm{Inst}(P) \uplus \mathrm{Inst}(Q) = \mathrm{T_B}(\mathcal{D}, \mathcal{C})$. We denote Q as $\mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus P$.

A pattern P is *linear* if so are all its elements. Theorem 1 of [6] gives an algorithm to compute Q from P (complementation algorithm) for any linear pattern P.

Example 3. *Let* $\mathcal{R}$ *be TRS in Example 6. Let* $P_0 = \{\mathsf{plus}(\mathsf{0}, y)\}$ *and* $P_1 = \{\mathsf{plus}(x, \mathsf{s}(y))\}$*. Then* $\mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus P_0 = \{\mathsf{plus}(\mathsf{s}(x), y)\}$ *and* $\mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus P_1 = \{\mathsf{plus}(x, \mathsf{0})\}$*. Furthermore, we have* $\mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus (P_0 \cup P_1) = \{\mathsf{plus}(\mathsf{s}(x), \mathsf{0})\}$.

GCR Procedure 2

Input: TRS $\mathcal{R}$
Output: `YES` or `MAYBE`

1. Compute (possibly multiple) candidates for the partition $\mathcal{F} = \mathcal{D} \uplus \mathcal{C}$ of function symbols.
2. Find left-linear $\mathcal{R}_0 \subseteq \mathcal{R}$ and a reduction quasi-order $\succsim$ such that $\mathcal{R}_0 \subseteq \succ$.
3. Compute a complement $P = \mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus lhs(\mathcal{R}_0)$, where $lhs(\mathcal{R}_0) = \{l \mid l \to r \in \mathcal{R}_0\}$. For each $p \in P$ find p' such that $p \xrightarrow{*}_{\mathcal{R}} p'$ and $p \succ p'$. Let $\mathcal{R}_1 = \mathcal{R}_0 \cup \{p \to p' \mid p \in P\}$.
4. Run rewriting induction for proving bounded ground convertibility of $\mathrm{CP}(\mathcal{R}_1)$ with $\succsim$.
5. Run rewriting induction for proving $\mathcal{R}_1 \models_{ind} (\mathcal{R} \setminus \mathcal{R}_0)$.
6. Return `YES` if it succeeds in steps 4 and 5, otherwise `MAYBE`.

Table 1: Preliminary experiments

problem	added equation(s)	steps		
		#1	#2	#3
Cops 128	$+(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x)$	×	✓	✓
Cops 130	$\left\{\begin{array}{c}\mathsf{and3}(\mathsf{F},\mathsf{T},\mathsf{T}) \to \mathsf{F} \\ \mathsf{and3}(\mathsf{F},\mathsf{F},\mathsf{T}) \to \mathsf{F} \\ \mathsf{and3}(\mathsf{T},\mathsf{F},\mathsf{T}) \to \mathsf{F}\end{array}\right\}$	×	×	✓
Cops 133	$+(\mathsf{0}, \mathsf{s}(x)) \to \mathsf{s}(x)$	×	✓	✓
Cops 137	$\mathsf{max}(\mathsf{0}, \mathsf{s}(y)) \to \mathsf{s}(y)$	×	✓	✓
Cops 140	$+(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x)$	×	✓	✓
Cops 146	$+(\mathsf{0}, \mathsf{s}(x)) \to \mathsf{s}(x)$	×	✓	✓
Cops 165	$\mathsf{max}(\mathsf{0}, \mathsf{s}(y)) \to \mathsf{s}(y)$	×	✓	✓
Cops 174	$+(\mathsf{0}, \mathsf{s}(x)) \to \mathsf{s}(x)$	×	✓	✓
Cops 180	$+(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x)$	×	✓	✓
Cops 186	$+(\mathsf{0}, \mathsf{s}(x)) \to \mathsf{s}(x)$	×	✓	✓
Cops 197	$\mathsf{or}(\mathsf{F}, \mathsf{T}) \to \mathsf{T}$	×	✓	✓
Cops 210	$+(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x)$	×	✓	✓
Cops 234	$\mathsf{eq}(\mathsf{0}, \mathsf{0}) \to \mathsf{true}$	✓	✓	✓
	total time (seconds)	32.694	32.620	33.052

Theorem 4. *If GCR Procedure 2 returns* `YES` *then $\mathcal{R}$ is ground confluent.*

Proof. Let $\mathcal{R}' = \mathcal{R} \cup (\mathcal{R}_1 \setminus \mathcal{R}_0)$. Then we have $\to_{\mathcal{R}} \subseteq \to_{\mathcal{R}'} \subseteq \xrightarrow{*}_{\mathcal{R}}$ and thus the ground confluence of $\mathcal{R}$ follows from that of $\mathcal{R}'$. □

Example 5. *Let $\mathcal{R}$ be a TRS given in Example 6. Suppose $\precsim$ be the lpo based on the precedence $\mathsf{plus} \succ \mathsf{s} \succ \mathsf{0}$. Then the GCR Procedure 2 puts $\mathcal{R}_0 = \{(a), (b)\}$ and one has $P = \mathrm{T_B}(\mathcal{D}, \mathcal{C}) \ominus lhs(\mathcal{R}_0) = \{\mathsf{plus}(\mathsf{s}(x), \mathsf{0})\}$. By $\mathsf{plus}(\mathsf{s}(x), \mathsf{0}) \to \mathsf{plus}(\mathsf{0}, \mathsf{s}(x)) \to \mathsf{s}(x)$, one gets $\mathcal{R}_1 = \mathcal{R}_0 \cup \{\mathsf{plus}(\mathsf{s}(x), \mathsf{0}) \to \mathsf{s}(x)\}$. Then $\mathrm{CP}(\mathcal{R}_1) = \emptyset$ and one successfully proves $\mathcal{R}_1 \models_{ind} \{(c)\}$ by rewriting induction. Thus $\mathcal{R}$ is proved to be ground confluent.*

5 Implementation and Experiment

A preliminary implementation has been done on AGCP so that when no strongly quasi-reducible subset is found it computes a complement of the defining patterns and adds defining rules. We used rewrite steps of length up to 7 to find p' such that $p \xrightarrow{*}_{\mathcal{R}} p'$ in the Step 3 of the GCR procedure 2. We tested our preliminary implementation on the collection of 121 ground confluence problems given in [1].

We found that 13 new examples can be handled using our preliminary implementation. The summary is presented in Table 1. Here, the column below "steps" shows results when length of rewrite steps to find p' are changed. Here, ✓ shows success and and × shows failure. All these examples are proved by ≤ 3 steps, one needs 3 steps only for Cops 130. Total time indicates the time required for running the prover on the collection of 121 ground confluence problems. Tests are performed on a PC with one 2.50GHz CPU and 4G memory. We impose 5 (resp. 1) seconds time limit rewriting induction proof (resp. computation of constructors). It turns out that changing the length from 1 step to 3 does not affect the total running time. However, with length 7, the total time exceeds 2 minutes and with length 8 we cannot complete the experiment within 10 minutes.

Example 6 (Cops 130)**.** *Let* $\mathcal{F} = \{\mathsf{and3} : \mathsf{Bool} \times \mathsf{Bool} \times \mathsf{Bool} \to \mathsf{Bool}, \mathsf{T} : \mathsf{Bool}, \mathsf{F} : \mathsf{Bool}\}$ *and*

$$\mathcal{R} = \left\{ \begin{array}{lcll} \mathsf{and3}(x, y, \mathsf{F}) & \to & \mathsf{F} & (a) \\ \mathsf{and3}(\mathsf{T}, \mathsf{T}, \mathsf{T}) & \to & \mathsf{T} & (b) \\ \mathsf{and3}(x, y, z) & \to & \mathsf{and3}(y, z, x) & (c) \end{array} \right\}$$

Let $\mathcal{D} = \{\mathsf{and3}\}$ *and* $\mathcal{C} = \{\mathsf{T}, \mathsf{F}\}$. *Take* $\mathcal{R}_0 = \{(a), (b)\}$. *Then one obtains* $\mathrm{T}_{\mathrm{B}}(\mathcal{D}, \mathcal{C}) \ominus lhs(\mathcal{R}_0) = \{\mathsf{and3}(\mathsf{F}, \mathsf{T}, \mathsf{T}), \mathsf{and3}(\mathsf{F}, \mathsf{F}, \mathsf{T}), \mathsf{and3}(\mathsf{T}, \mathsf{F}, \mathsf{T})\}$. *Then* $\mathsf{and3}(\mathsf{F}, \mathsf{T}, \mathsf{T}) \to_{\mathcal{R}} \mathsf{and3}(\mathsf{T}, \mathsf{T}, \mathsf{F}) \to_{\mathcal{R}} \mathsf{F}$ *and* $\mathsf{and3}(\mathsf{F}, \mathsf{F}, \mathsf{T}) \to_{\mathcal{R}} \mathsf{and3}(\mathsf{F}, \mathsf{T}, \mathsf{F}) \to_{\mathcal{R}} \mathsf{F}$. *But* $\mathsf{and3}(\mathsf{T}, \mathsf{F}, \mathsf{T}) \to_{\mathcal{R}} \mathsf{and3}(\mathsf{F}, \mathsf{T}, \mathsf{T}) \to_{\mathcal{R}} \mathsf{and3}(\mathsf{T}, \mathsf{T}, \mathsf{F}) \to_{\mathcal{R}} \mathsf{F}$. *Let us consider multiset path ordering with* $\mathsf{and3} \succ \mathsf{T} \succ \mathsf{F}$. *Then considering 2 steps at* $p \xrightarrow{*} p'$ *in the Step 3 of the procedure does not suffice as* $\mathsf{and3}(\mathsf{T}, \mathsf{F}, \mathsf{T}) \not\succ_{mpo} \mathsf{and3}(\mathsf{T}, \mathsf{T}, \mathsf{F})$. *By considering 3 steps at* $p \xrightarrow{*} p'$ *in the Step 3 of the procedure, we obtain a rewrite rule* $\mathsf{and3}(\mathsf{T}, \mathsf{F}, \mathsf{T}) \to \mathsf{F}$ *such that* $\mathsf{and3}(\mathsf{T}, \mathsf{F}, \mathsf{T}) \succ_{mpo} \mathsf{F}$.

Sometimes computation of $p \xrightarrow{*} p'$ diverges. The next example illustrates this.

Example 7. *Cops 62 contains the following rewrite rules:*

$$\begin{array}{lcllllll} \mathsf{mod}(0, y) & \to & 0 & (a) & \mathsf{mod}(x, 0) & \to & x & (c) \\ \mathsf{mod}(x, \mathsf{s}(y)) & \to & \mathsf{if}(<(x, \mathsf{s}(y)), x, \mathsf{mod}(-(x, \mathsf{s}(y)), \mathsf{s}(y))) & (b) & & & & \end{array}$$

Then $\mathcal{R}_0 = \{\ldots, (a), (c), \ldots\}$ *and* $(b) \notin \mathcal{R}_0$ *due to ordering restriction. Then* $\mathsf{mod}(\mathsf{s}(x), \mathsf{s}(y)) \in P$, *and the procedure searches some rewrite rule* $\mathsf{mod}(\mathsf{s}(x), \mathsf{s}(y)) \to r$. *However, the set* $\{r \mid \mathsf{mod}(\mathsf{s}(x), \mathsf{s}(y)) \xrightarrow{*}_{\mathcal{R}} r\}$ *is infinite and there is no* r *satisfying* $\mathsf{mod}(\mathsf{s}(x), \mathsf{s}(y)) \succ r$.

6 Conclusion

We have shown how the procedure for proving ground confluence of many-sorted TRSs in [1] is improved by constructing new rewrite rules necessary for making the rewriting induction work. We have presented our new procedure and shown its correctness. We have reported on our preliminary implementation and experiment. There are 13 new examples for which ground confluence can be proved from the collection of 121 examples, where the previous procedure can prove 86 problems.

References

[1] T. Aoto and Y. Toyama. Ground confluence prover based on rewriting induction. In *Proc. of 1st FSCD*, volume 52 of *LIPIcs*, pages 33:1–12, 2016.

[2] K. Becker. Proving ground confluence and inductive validity in constructor based equational specifications. In *Proc. of 4th TAPSOFT*, volume 668 of *LNCS*, pages 46–60. Springer-Verlag, 1993.

[3] A. Bouhoula. Simultaneous checking of completeness and ground conflunce for algebraic specifications. *ACM Transactions on Computational Logic*, 10(2):20:1–33, 2009.

[4] H. Ganzinger. Ground term confluence in parametric conditional equational specifications. In *Proc. of 4th STACS*, volume 247 of *LNCS*, pages 286–298, 1987.

[5] R. Göbel. Ground confluence. In *Proc. of 2nd RTA*, volume 256 of *LNCS*, pages 156–167, 1987.

[6] A. Lazrek, P. Lescanne, and J. J. Thiel. Tools for proving inductive equalities, relative completeness, and ω-completeness. *Information and Computation*, 84:47–70, 1990.

[7] Terese. *Term Rewriting Systems.* Cambridge University Press, 2003.