Disproving Confluence of Term Rewriting Systems by Interpretation and Ordering (extended abstract)

Takahito Aoto

RIEC, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, 980-8577, Japan
aoto@nue.riec.tohoku.ac.jp

Abstract

We present new criteria for ensuring non-joinability of terms based on interpretation and ordering, and report on an implementation of confluence disproving procedure based on some instances of the criteria. The experiment reveals that our methods can be applied to automatically disprove confluence of some term rewriting systems, on which state-of-the-art automated confluence provers fail.

1 Introduction

In contrast to many dedicated techniques that have been developed to prove confluence of term rewriting systems, not many techniques for disproving confluence are known. A typical approach to disprove confluence of (non-terminating) TRSs is first to construct some candidates of two terms that can be reduced from a common term, and then to show that these candidates are not joinable, i.e. they do not have a common reduct. In this scenario, as well as the selection of the candidates, proving non-joinability of terms is essential. So far, the only serious approach to prove the non-joinability of terms is to use approximation by tree automata [4, 7].

In this paper, we give new methods for proving that given two terms s, t are not joinable. The first method consists in giving an interpretation, e.g. a mapping from terms to natural numbers, that is preserved by the application of usable rules and such that the interpretation of s is different from that of t. The second method consists in giving an ordering $>$ such that $s > t$, and usable rules from s only increase or preserve w.r.t. $>$ and the usable rules from t only decrease or preserve w.r.t. $>$. These methods are implemented using polynomial interpretations and recursive path orderings—interpretations and orderings that are widely used in the literature for termination proving. The experiment reveals that our methods can be applied to automatically disprove confluence of some term rewriting systems, on which state-of-the-art automated confluence provers fail to disprove.

2 Preliminaries

We assume familiarity with standard notions and notations on term rewriting (see e.g. [3]). Below we explain some extra notations used in the paper. The *disjoint union* of two sets A and B is denoted by $A \uplus B$, and that of all A_i $(i \in I)$ by $\biguplus_{i \in I} A_i$. The set of terms over a set $\mathcal{F}$ of function symbols and the set $\mathcal{V}$ of variables is denoted by $\mathrm{T}(\mathcal{F}, \mathcal{V})$. The set of variables in a term t is denoted by $\mathcal{V}(t)$. We write $s \trianglelefteq t$ to denote that s is a subterm of t. We write $\mathrm{Unif}(s, t)$ to denote that the terms s and t are unifiable. A *rewrite rule* $l \to r$ is a pair of terms; we here drop the usual restriction that $l \notin \mathcal{V}$ and $\mathcal{V}(r) \subseteq \mathcal{V}(l)$. A *rewrite relation* is a

relation on terms that is closed under contexts and substitutions. A strict partial order (partial order, quasi-order) is a *rewrite strict partial order* (*rewrite partial order*, *rewrite quasi-order*, respectively) if it is a rewrite relation.

Given a term s, the sets of terms $\{t \in \mathrm{T}(\mathcal{F},\mathcal{V}) \mid s \xrightarrow{*} t\}$ and $\{t \in \mathrm{T}(\mathcal{F},\mathcal{V}) \mid t \xrightarrow{*} s\}$ are denoted by $[s](\xrightarrow{*})$ and $(\xrightarrow{*})[s]$, respectively. Terms s and t are said to be *joinable* if $[s](\xrightarrow{*}) \cap [t](\xrightarrow{*}) \neq \emptyset$, and *non-joinable* otherwise. We write $\mathrm{NJ}(s,t)$ to denote that the terms s and t are non-joinable. In order to disprove that a TRS $\mathcal{R}$ is confluent, we construct two terms s and t such that $(\xrightarrow{*})[s] \cap (\xrightarrow{*})[t] \neq \emptyset$ in some way, and then prove $\mathrm{NJ}(s,t)$. From here on, we concentrate on the problem of proving $\mathrm{NJ}(s,t)$, the non-joinability problem.

3 Proving Non-Joinability by Interpretation

In this section, we present several criteria to prove non-joinability of terms based on their interpretations in $\mathcal{F}$-algebras.

An *$\mathcal{F}$-algebra* $\mathcal{A} = \langle A, \langle f^{\mathcal{A}} \rangle_{f\in\mathcal{F}} \rangle$ is a pair of a set A and a tuple of functions $f^{\mathcal{A}} : A^n \to A$ for each n-ary function symbol $f \in \mathcal{F}$. The set A is called the *carrier set* of the $\mathcal{F}$-algebra $\mathcal{A}$ and is denoted by $|\mathcal{A}|$. A *valuation* on the $\mathcal{F}$-algebra $\mathcal{A}$ is a mapping $\mathcal{V} \to A$. Suppose an $\mathcal{F}$-algebra $\mathcal{A} = \langle A, (f^{\mathcal{A}})_{f\in\mathcal{F}} \rangle$ is fixed. Then the *interpretation* of a term under the valuation σ is denoted by $[\![t]\!]_\sigma$.

The notion of usable rules [2] is well-known in the literature for proving termination of TRSs. We introduce a notion of usable rules for non-joinability suitable for our setting. For this, the notion of TCAP [5] is used. For terms t, $\mathrm{TCAP}(t)$ is defined recursively like this: $\mathrm{TCAP}(x) = x'$, $\mathrm{TCAP}(f(t_1,\ldots,t_n)) = x'$ if $\mathrm{Unif}(f(u_1,\ldots,u_n), l)$ for some $l \to r \in \mathcal{R}$, and $\mathrm{TCAP}(f(t_1,\ldots,t_n)) = f(u_1,\ldots,u_n)$ otherwise, where $u_i = \mathrm{TCAP}(t_i)$ $(1 \le i \le \mathrm{arity}(f))$. Here, a new fresh variable is taken for x' every time it is used. Our notion of usable rules is obtained from the one for innermost termination [5] by replacing ICAP with TCAP.

Definition 1 (usable rules)**.** *The set of* usable rules *for non-joinability w.r.t. TRS $\mathcal{R}$ and a term s is the smallest set $\mathcal{U}_{\mathsf{nj}}(\mathcal{R},s) \subseteq \mathcal{R}$ satisfying two conditions: (i) for any $l \to r \in \mathcal{R}$ and non-variable subterm $f(u_1,\ldots,u_n) \trianglelefteq s$, if $\mathrm{Unif}(f(\mathrm{TCAP}(u_1),\ldots,\mathrm{TCAP}(u_n)), l)$ then $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s)$; (ii) if $l' \to r' \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s)$ and $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},r')$, then $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s)$.*

The following is a key lemma for proving our theorem given below.

Lemma 2. *Let $\mathcal{R}$ be a TRS, $l \to r \in \mathcal{R}$ and s,t terms. If $s \xrightarrow{*}_{\mathcal{R}} \circ \to_{\{l\to r\}} t$ then $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s)$.*

Theorem 3. *Let s,t be terms and $\mathcal{A} = \langle A, \langle f^{\mathcal{A}} \rangle_{f\in\mathcal{F}} \rangle$ an $\mathcal{F}$-algebra such that $A = \biguplus_{i\in I} A_i$. Suppose (i) for any valuation σ and $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R},t)$, if $[\![l]\!]_\sigma \in A_i$ then $[\![r]\!]_\sigma \in A_i$, (ii) for any $f \in \mathcal{F}$, $a \in A$ and $i,j \in I$, if $a \in A_i$ implies $f^{\mathcal{A}}(\ldots,a,\ldots) \in A_j$, then $f^{\mathcal{A}}(\ldots,b,\ldots) \in A_j$ for any $b \in A_i$ and (iii) $[\![s]\!]_\rho \in A_i$ and $[\![t]\!]_\rho \in A_j$ for some valuation ρ and $i \neq j$. Then $\mathrm{NJ}(s,t)$.*

The criterion of Theorem 3, in general, is not amenable for automation, and one has to use more concrete instances of the theorem such as given below.

Corollary 4. *Let $\mathcal{A}$ be an $\mathcal{F}$-algebra and s,t be terms. Suppose (i) $[\![l]\!]_\sigma = [\![r]\!]_\sigma$ for any valuation σ and $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R},s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R},t)$ and (ii) $[\![s]\!]_\rho \neq [\![t]\!]_\rho$ for some valuation ρ. Then $\mathrm{NJ}(s,t)$.*

Corollary 5. *Let s, t be terms and $\mathcal{A}$ an $\mathcal{F}$-algebra whose carrier set is a set of integers. Suppose there exists an integer $k \geq 2$ such that (i) for any valuation σ and $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$, $[\![l]\!]_\sigma \equiv [\![r]\!]_\sigma \pmod{k}$ and (ii) $[\![s]\!]_\rho \not\equiv [\![t]\!]_\rho \pmod{k}$ for some valuation ρ. Then* $\mathrm{NJ}(s, t)$.

In following examples, non-confluence is shown using these corollaries.

Example 6. *Let $\mathcal{R} = \{(1) : \mathsf{a} \to \mathsf{h}(\mathsf{c}), (2) : \mathsf{a} \to \mathsf{h}(\mathsf{f}(\mathsf{c})), (3) : \mathsf{h}(x) \to \mathsf{h}(\mathsf{h}(x)), (4) : \mathsf{f}(x) \to \mathsf{f}(\mathsf{g}(x))\}$. Let $s = \mathsf{h}(\mathsf{c})$ and $t = \mathsf{h}(\mathsf{f}(\mathsf{c}))$. As $\mathsf{a} \in (\xrightarrow{*})[s] \cap (\xrightarrow{*})[t]$, it suffices to show* $\mathrm{NJ}(s, t)$ *to disprove the confluence of $\mathcal{R}$. We have $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t) = \{(3), (4)\}$. Take an $\mathcal{F}$-algebra $\mathcal{A} = \langle \{0, 1\}, \langle f^{\mathcal{A}} \rangle_{f \in \mathcal{F}} \rangle$ as $\mathsf{a}^{\mathcal{A}} = \mathsf{c}^{\mathcal{A}} = 0$, $\mathsf{f}^{\mathcal{A}}(n) = 1 - n$, $\mathsf{h}^{\mathcal{A}}(n) = \mathsf{g}^{\mathcal{A}}(n) = n$. Then for any valuation σ, we have $[\![\mathsf{h}(x)]\!]_\sigma = \sigma(x) = [\![\mathsf{h}(\mathsf{h}(x))]\!]_\sigma$ and $[\![\mathsf{f}(x)]\!]_\sigma = 1 - \sigma(x) = [\![\mathsf{f}(\mathsf{g}(x))]\!]_\sigma$; thus, $[\![l]\!]_\sigma = [\![r]\!]_\sigma$ for each $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$. Take an arbitrary valuation ρ. Then $[\![s]\!]_\rho = [\![\mathsf{h}(\mathsf{c})]\!]_\rho = 0 \neq 1 = [\![t]\!]_\rho = [\![\mathsf{h}(\mathsf{f}(\mathsf{c}))]\!]_\rho$. Therefore,* $\mathrm{NJ}(s, t)$ *by Corollary 4.*

Example 7. *Let $\mathcal{R} = \{(1) : \mathsf{a} \to \mathsf{f}(\mathsf{c}), (2) : \mathsf{a} \to \mathsf{h}(\mathsf{c}), (3) : \mathsf{f}(x) \to \mathsf{h}(\mathsf{g}(x)), (4) : \mathsf{h}(x) \to \mathsf{f}(\mathsf{g}(x))\}$. Let $s = \mathsf{f}(\mathsf{c})$ and $t = \mathsf{h}(\mathsf{c})$. We have $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t) = \{(3), (4)\}$. Take an $\mathcal{F}$-algebra $\mathcal{A} = \langle \mathbb{N}, \langle f^{\mathcal{A}} \rangle_{f \in \mathcal{F}} \rangle$ as $\mathsf{a}^{\mathcal{A}} = \mathsf{c}^{\mathcal{A}} = 0$, $\mathsf{g}^{\mathcal{A}}(n) = n + 1$, $\mathsf{f}^{\mathcal{A}}(n) = n$, $\mathsf{h}^{\mathcal{A}}(n) = n + 1$. Then $[\![\mathsf{f}(x)]\!]_\sigma - [\![\mathsf{h}(\mathsf{g}(x))]\!]_\sigma = \sigma(x) - (\sigma(x) + 2) = -2$ and $[\![\mathsf{h}(x)]\!]_\sigma - [\![\mathsf{f}(\mathsf{g}(x))]\!]_\sigma = (\sigma(x) + 1) - (\sigma(x) + 1) = 0$. Take $k = 2$. Then $[\![\mathsf{f}(x)]\!]_\sigma \equiv [\![\mathsf{h}(\mathsf{g}(x))]\!]_\sigma \pmod{k}$ and $[\![\mathsf{h}(x)]\!]_\sigma \equiv [\![\mathsf{f}(\mathsf{g}(x))]\!]_\sigma \pmod{k}$ for any valuation σ. Furthermore, since we have $[\![s]\!]_\rho = [\![\mathsf{f}(\mathsf{c})]\!]_\rho = 0$ and $[\![t]\!]_\rho = [\![\mathsf{h}(\mathsf{c})]\!]_\rho = 1$, $[\![s]\!]_\rho \not\equiv [\![t]\!]_\rho \pmod{k}$. Hence,* $\mathrm{NJ}(s, t)$ *by Corollary 5.*

4 Proving Non-Joinability by Ordering

In Corollary 5, we considered the case that the carrier set is a set of integers. In such a case, another obvious choice to obtain a partition of the carrier set is to divide it as $A = \{n \in A \mid n < k\} \uplus \{n \in A \mid k \leq n\}$ for some k. We first formulate this idea in a more abstract setting, using the notion of ordered $\mathcal{F}$-algebra [10].

An *ordered $\mathcal{F}$-algebra* $\mathcal{A} = \langle A, \leq, \langle f^{\mathcal{A}} \rangle_{f \in \mathcal{F}} \rangle$ is a triple of a set A, a partial order $\leq$ on it and a tuple of functions $f^{\mathcal{A}} : A^n \to A$ for each n-ary function symbol $f \in \mathcal{F}$. We use $<$ to denote strict part of $\leq$, i.e. $< = \leq \setminus \geq$. An ordered $\mathcal{F}$-algebra $\mathcal{A} = \langle A, \leq, \langle f^{\mathcal{A}} \rangle_{f \in \mathcal{F}} \rangle$ is said to be *weakly monotone* if $a \leq b$ implies $f^{\mathcal{A}}(\ldots, a, \ldots) \leq f^{\mathcal{A}}(\ldots, b, \ldots)$ for any $a, b \in A$ and $f \in \mathcal{F}$.

Theorem 8. *Let $\mathcal{A}$ be a weakly monotone ordered $\mathcal{F}$-algebra and s, t be terms. Suppose (i) $[\![l]\!]_\sigma \leq [\![r]\!]_\sigma$ for any valuation σ and any $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s)$, (ii) $[\![l]\!]_\sigma \geq [\![r]\!]_\sigma$ for any valuation σ and any $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$ and (iii) $[\![s]\!]_\rho > [\![t]\!]_\rho$ for some valuation ρ. Then* $\mathrm{NJ}(s, t)$.

We next consider the case that term algebras are taken as $\mathcal{F}$-algebras, and formulate the theorem in a more general way using the notion of rewrite relation. For this, the following notion is useful.

Definition 9 (discrimination pair)**.** *A pair $\langle \gtrsim, \succ \rangle$ of two relations $\gtrsim$ and $\succ$ is said to be a* discrimination pair *if (i) $\gtrsim$ is a rewrite relation, (ii) $\succ$ is a strict partial order and (iii) $\gtrsim \circ \succ \subseteq \succ$ and $\succ \circ \gtrsim \subseteq \succ$.*

Clearly, for any rewrite quasi-order $\gtrsim$, the pair $\langle \gtrsim, \gtrsim \setminus \lesssim \rangle$ forms a discrimination pair.

Before presenting the next theorem, another notion from termination proving is required. An *argument filtering* [2] is a mapping such that $\pi(f) \in \{[i_1, \ldots, i_k] \mid i_1 < \cdots < i_k, 1 \leq i_1, \ldots, i_k \leq \mathrm{arity}(f)\} \cup \{i \mid 1 \leq i \leq \mathrm{arity}(f)\}$. Then the application t^π of the argument filtering π to terms t is given by $x^\pi = x$ for $x \in \mathcal{V}$, $f(t_1, \ldots, t_n)^\pi = f(t_{i_1}^\pi, \ldots, t_{i_k}^\pi)$ if $\pi(f) = [i_1, \ldots, i_k]$, $f(t_1, \ldots, t_n)^\pi = t_i^\pi$ if $\pi(f) = i$. For a TRS $\mathcal{R}$, we put $\mathcal{R}^\pi = \{l^\pi \to r^\pi \mid l \to r \in \mathcal{R}\}$.

Theorem 10. *Let $\mathcal{R}$ be a TRS and s, t terms. Suppose there exist a discrimination pair $\langle \gtrsim, \succ \rangle$ and an argument filtering π such that $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, s^\pi) \subseteq \lesssim$, $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, t^\pi) \subseteq \gtrsim$ and $s^\pi \succ t^\pi$. Then* $\mathrm{NJ}(s, t)$.

In terms of interpretations, Theorem 10 amounts to take term algebras as $\mathcal{F}$-algebras, while Theorem 8 allows to take any $\mathcal{F}$-algebra. On the other hand, in terms of discrimination pairs, Theorem 8 amounts to take a discrimination pair of the form $\langle \gtrsim, \gtrsim \setminus \lesssim \rangle$.

Example 11. *Let $\mathcal{R} = \{(1) : \mathsf{c} \to \mathsf{f}(\mathsf{c}, \mathsf{d}), (2) : \mathsf{c} \to \mathsf{h}(\mathsf{c}, \mathsf{d}), (3) : \mathsf{f}(x, y) \to \mathsf{h}(\mathsf{g}(y), x), (4) : \mathsf{h}(x, y) \to \mathsf{f}(\mathsf{g}(y), x)\}$. Let $s = \mathsf{h}(\mathsf{f}(\mathsf{c}, \mathsf{d}), \mathsf{d})$ and $t = \mathsf{f}(\mathsf{c}, \mathsf{d})$. Take an argument filtering π as $\pi(\mathsf{g}) = 1$, $\pi(\mathsf{f}) = [2]$ and $\pi(\mathsf{h}) = [1]$. Then we have $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, s^\pi) = \mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, t^\pi) = \{(3)^\pi, (4)^\pi\}$. The constraint $\{\mathsf{h}(\mathsf{f}(\mathsf{d})) \succ \mathsf{f}(\mathsf{d}), \mathsf{f}(y) \simeq \mathsf{h}(y), \mathsf{h}(x) \simeq \mathsf{f}(x)\}$ is satisfied by a discrimination pair $\langle \gtrsim_{rpo}, \gtrsim_{rpo} \setminus \lesssim_{rpo} \rangle$, where $\gtrsim_{rpo}$ is the recursive path order based on the precedence $\mathsf{f} \simeq \mathsf{h}$. Thus* $\mathrm{NJ}(s, t)$ *by Theorem 10.*

5 Implementations and Experiments

Implementations The following instances of presented criteria have been implemented. We assume below that we check non-joinability of ground terms s, t.

Cor. 5 ($k = 2, 3$) Corollary 5 applied for the polynomial interpretation with linear polynomials. In case $k = 2$, we check whether $[\![l]\!]_\sigma - [\![r]\!]_\sigma$ is even for all rewrite rules $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s) \cup \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$ and whether $[\![s]\!] - [\![t]\!]$ is odd. We encode these constraints in boolean formulas and check the constraints by an external SAT solver. We deal with integer variables of the range between 0 and 15. The case $k = 3$ is similar.

Th. 8 (poly) Theorem 8 applied for polynomial interpretation with linear polynomials. Similar to the case **Cor. 5 ($k = 2, 3$)**, we encode the constraints in boolean formulas and check the constraints by an external SAT solver. Our implementation tries two possible applications of the Theorem to show $\mathrm{NJ}(s, t)$, namely that (1) $[\![s]\!] > [\![t]\!]$, $[\![l]\!]_\sigma \geq [\![r]\!]_\sigma$ for $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$ and $[\![l]\!]_\sigma \leq [\![r]\!]_\sigma$ for $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s)$, and (2) $[\![t]\!] > [\![s]\!]$, $[\![l]\!]_\sigma \geq [\![r]\!]_\sigma$ for $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s)$ and $[\![l]\!]_\sigma \leq [\![r]\!]_\sigma$ for $l \to r \in \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, t)$.

Th. 10 (rpo) Theorem 10 applied for recursive path order with argument filtering. Similar to the cases **Cor. 5 ($k = 2, 3$)** and **Th. 8 (poly)**, we encode the constraints in boolean formulas and check the constraints by an external SAT solver. We approximate the set of usable rules $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, s^\pi)$ first by $\mathcal{S} = \mathcal{U}_{\mathsf{nj}}(\mathcal{R}, s)$ before encoding and then $\mathcal{U}(\mathcal{S}^\pi, s^\pi)$, the set of usable rules for dependency pairs [2], at the time of encoding (and similarly for $\mathcal{U}_{\mathsf{nj}}(\mathcal{R}^\pi, t^\pi)$).

Candidates for the non-joinability test are generated from the input TRS $\mathcal{R}$ like this: (1) first compute the one-step unfolding $\mathcal{R}'$ of $\mathcal{R}$ [8] and then (2) compute critical pairs of $\mathcal{R} \cup \mathcal{R}'$, and finally, (3) all critical pairs are sorted w.r.t. term size and at most 100 crucial pairs are considered for candidates for non-joinability test.

Experiments Experiments have been performed on our implementation and the state-of-the-art confluence provers ACP [1] (ver. 0.31), CSI [9] (ver. 0.2) and Saigawa [6] (ver. 1.4). Each test is performed on a PC with one 2.50GHz CPU and 4G memory; the timeout is set to 60 seconds. We have tested a collection of 23 new examples which includes Examples 6, 7, 11 and their

Table 1: Summary of experiments

	ACP	CSI	Saigawa	Cor. 5 ($k=2$)	Cor. 5 ($k=3$)	Th. 8 (poly)	Th. 10 (rpo)	all
Example 6	×	×	×	✓	✓	✓	✓	✓
Example 7	×	×	×	✓	✓	×	×	✓
Example 11	×	×	×	×	×	×	✓	✓
23 examples (success)	9	12	3	16	16	14	19	21
23 examples (time in sec.)	2	2107	228	25	293	206	26	84
35 examples (success)	18	21	17	17	16	17	17	16
35 examples (time in sec.)	71	485	482	318	562	446	106	761

variants, and a collection of 35 examples from the 1st Confluence Competition (CoCo 2012) that were not proved to be confluent by any of participating provers.

A summary of the experiments is shown in Table 1. The column below **all** denotes the result for the combination of the four instances. All provers ACP, CSI and Saigawa fail on Examples 6, 7 and 11. For the collection of 23 new examples, the following are observed: **Cor. 5** ($k = 2$) and **Cor. 5** ($k = 3$) succeed at the same examples. Examples handled by **Th. 8 (poly)** are also handled by **Th. 10 (rpo)** and also by **Cor. 5**. Examples handled by any of the provers ACP, CSI and Saigawa also are handled by **all**. For the collection of 35 examples from CoCo 2012, the following are observed: All instances succeed on the same examples, except for **Cor. 5** ($k = 3$), in which one timeouts. The numbers of examples on which ACP, CSI and Saigawa succeed but **all** fails are 4, 5, 3, respectively. Finally, the running time is observed like this: **Th. 10 (rpo)** $<$ **Cor. 5** ($k = 2$) $\ll$ **Th. 8 (poly)** $\ll$ **Cor. 5** ($k = 3$). All details of the experiments are available on the webpage: `http://www.nue.riec.tohoku.ac.jp/tools/acp/experiments/iwc13/all.html`.

References

[1] T. Aoto, Y. Yoshida, and Y. Toyama. Proving confluence of term rewriting systems automatically. In *Proc. of 20th RTA*, volume 5595 of *LNCS*, pages 93–102. Springer-Verlag, 2009.

[2] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1–2):133–178, 2000.

[3] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[4] T. Genet. Decidable approximations of sets of descendants and sets of normal forms. In *Proc. of 9th RTA*, volume 1379 of *LNCS*, pages 151–165. Springer-Verlag, 1998.

[5] J. Giesl, R. Thiemann, and P. Schneider-Kamp. Proving and disproving termination of higher-order functions. In *Proc. of 5th FroCoS*, volume 3717, pages 216–231. Springer-Verlag, 2005.

[6] N. Hirokawa and D. Klein. Saigawa: A confluence tool. In *Proc. of 1st IWC*, page 49, 2012.

[7] A. Middeldorp. Approximating dependency graphs using tree automata techniques. In *Proc. of the 1st IJCAR*, volume 2083 of *LNAI*, pages 593–610. Springer-Verlag, 2001.

[8] É. Payet. Loop detection in term rewriting using eliminating unfoldings. *Theoretical Computer Science*, 403:307–327, 2008.

[9] H. Zankl, B. Felgenhauer, and A. Middeldorp. CSI – A confluence tool. In *Proc. of 23rd CADE*, volume 6803 of *LNAI*, pages 499–505. Springer-Verlag, 2011.

[10] H. Zantema. Termination of term rewriting by semantic labelling. *Fundamenta Informaticae*, 24:89–105, 1995.