Completion to Strong Confluence

Salvador Lucas[1,2] and Julia Pagan[2]

[1]DSIC & [2]VRAIN, Universitat Politècnica de València (UPV)

15th International Workshop on Confluence

IWC 2026

Lisboa, Portugal

Completion à la Knuth-Bendix: given an equational specification E

obtain a confluent and terminating TRS $\mathcal{R}$ whose equational theory $E_{\mathcal{R}}$ coincides with E. Then, compute with $\mathcal{R}$

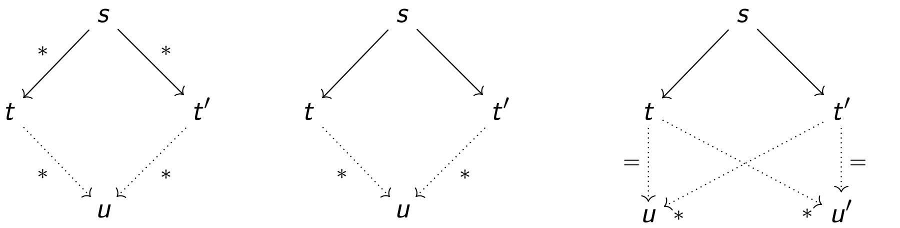
This may fail to work. The following equational specification

$$\begin{array}{rcl} \mathsf{h}(\mathsf{b},x) & = & \mathsf{c}(\mathsf{h}(\mathsf{b},\mathsf{b})) \quad (1) \\ \mathsf{f}(\mathsf{f}(x)) & = & \mathsf{b} \quad (2) \end{array}$$

yields (replace '=' by '->') a non-terminating TRS $\mathcal{R}$ with a non-joinable critical pair $\pi_{(2),1,(2)}$

$$\langle \mathsf{f}(\mathsf{b}),\mathsf{b}\rangle \quad (3)$$

Hence, $\mathcal{R}$ is not confluent



Confluence (CR) Local confluence (WCR) Strong confluence (SCR)

Knuth and Bendix completion [KB70] uses a *reduction ordering* $\succ$

(0) Orient $s = t$ in E as $s \to t$ if $s \succ t$ or $t \to s$ if $t \succ s$ (it is R_0); $i := 0$.
(1) If CPs of R_i are all joinable (WCR holds), then return R_i; else,
(2) Non-joinable CPs $\langle s, t\rangle$ become rules $\widehat{s} \to \widehat{t}$ if $\widehat{s} \succ \widehat{t}$ or $\widehat{t} \to \widehat{s}$ if $\widehat{t} \succ \widehat{s}$
(3) Add the obtained rule (if any!) to R_i to obtain R_{i+1} and go to (1).

If it suceeds, $\mathcal{R}_E = R_i$ is WCR and SN; by Newman's Lemma, CR holds

Completion to Strong Confluence: main idea

Replace WCR by SCR; $\succ$ is *not* necessary, as SCR $\Rightarrow$ CR holds

1. Strong confluence of linear Term Rewriting Systems
2. Completion to strong confluence of linear TRSs
3. Strong $\nrightarrow\!\!\!\!\!\!\!\mid\mid$-confluence of left-linear Term Rewriting Systems
4. Completion to strong $\nrightarrow\!\!\!\!\!\!\!\mid\mid$-confluence of left-linear TRSs
5. Implementation
6. Conclusions and future work

Strong confluence of linear TRSs

A Term Rewriting System (TRS) $\mathcal{R}$ is a set of rewrite rules $\ell \to r$

$\mathcal{R}$ is linear if both ℓ and r are always linear terms (no variable occurs twice)

Theorem ([Hue80, Lemma 3.2])

A linear TRS $\mathcal{R}$ is strongly confluent if for all CPs $\langle s, t \rangle$, there are terms u and u' such that the following holds (where $s \to^{=} t$ if $s \to t$ or $s = t$)

$$
\begin{tikzcd}
s \arrow[d, "="'] \arrow[dr, "*" near end] & t \arrow[d, "="] \arrow[dl, "*"' near end] \\
u & u'
\end{tikzcd}
$$

This condition can be checked as the *feasibility* of the sequence

$$s^{\downarrow} \to^{=} x, s^{\downarrow} \to^{*} x', t^{\downarrow} \to^{*} x, t^{\downarrow} \to^{=} x'$$

That is, the existence of a substitution satisfying the rewriting conditions above (where $s^{\downarrow}$ is s with variables x replaced by fresh constants c_x)

Completion to Strong Confluence of linear TRSs

Algorithm 1: Completion to Linear, Strongly $\rightarrow$-Confluent TRSs

Initialization:

$i := 0$; $oldCPs_0 := \{\}$, $R_0 := \{\ell \to r \mid (\ell \approx r) \in E \cup E^{-1} \wedge \ell \notin \mathcal{X} \wedge \mathcal{V}ar(r) \subseteq \mathcal{V}ar(\ell)\}$

repeat

$R_{i+1} := R_i$; $oldCPs_{i+1} := oldCPs_i$;

for all $\langle s, t\rangle \in (CP(R_i) - oldCPs_i)$ **do**

if *s is not linear* **or** *t is not linear* **then**

Terminate with output `Fail`

end

if $SJoin(\langle s, t\rangle) =$`False` **or** $SJoin(\langle s, t\rangle) =$`Maybe` **then**

if $s \notin \mathcal{X} \wedge \mathcal{V}ar(t) \subseteq \mathcal{V}ar(s)$ **then**

$R_{i+1} := R_{i+1} \cup \{s \to t\}$;

else

if $t \notin \mathcal{X} \wedge \mathcal{V}ar(s) \subseteq \mathcal{V}ar(t)$ **then**

$R_{i+1} := R_{i+1} \cup \{t \to s\}$;

else

Terminate with output `Fail`

end

end

end

$oldCPs_{i+1} := oldCPs_{i+1} \cup \{\langle s, t\rangle, \langle t, s\rangle\}$

od

$i := i + 1$;

until $R_i = R_{i-1}$;

output R_i;

Knuth and Bendix completion does not work with the specification E

```
(VAR x)
(EQUATIONS
  h(b,x) = c(h(b,b))
  f(f(x)) = b
)
```

Now, an E-equivalent [Pag25] Strongly Confluent TRS $\mathcal{R}_E$ is obtained:

```
(VAR x)
(RULES
  h(b,x) -> c(h(b,b))
  f(f(x)) -> b
  f(b) -> b
)
```

Strong $\nrightarrow$-confluence of left-linear TRSs

Given a TRS $\mathcal{R}$, $\mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow}_{\mathcal{R}}$ is simultaneous rewriting on disjoint redex positions

A TRS $\mathcal{R}$ is left-linear if ℓ is linear for all rules $\ell \to r$

Theorem ([Toy88, cf. Corollary 3.2])

Given a left-linear TRS $\mathcal{R}$, $\mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow}_{\mathcal{R}}$ is strongly confluent (and hence $\mathcal{R}$ is confluent) if for all critical pairs $\langle s, t\rangle$ with critical position p,

- *if $p = \Lambda$, then s and t are strongly $\mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow}_{\mathcal{R}}$-joinable; and*
- *if $p \neq \Lambda$, then $s \mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow}_{\mathcal{R}} t$ (parallel-closedness).*

Again, these conditions can be checked as the *feasibility* of

$$s^{\downarrow} \mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow} u,\ s^{\downarrow} \to^{*} u',\ t^{\downarrow} \to^{*} u,\ t^{\downarrow} \mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow} u' \quad (4)$$

$$s^{\downarrow} \mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow} t^{\downarrow} \quad (5)$$

Algorithm 2

The definition of the corresponding completion to strong $\mathrel{+\!\!\!\shortparallel\!\!\!\rightarrow}$-confluence algorithm is similar to Algorithm 1

Consider the specification E (cf. [HM11, Example 8], see COPS/113.trs)

$$
\begin{array}{rcl}
\mathsf{f}(\mathsf{a},\mathsf{a}) & = & \mathsf{c} \\
\mathsf{f}(\mathsf{b},x) & = & \mathsf{f}(x,x) \\
\mathsf{f}(x,\mathsf{b}) & = & \mathsf{f}(x,x) \\
\mathsf{a} & = & \mathsf{b}
\end{array}
$$

Algorithm 1 does not work, but Algorithm 2 obtains:

```
(VAR x)
(RULES
  f(a,a) -> c
  f(b,x) -> f(x,x)
  f(x,b) -> f(x,x)
  a -> b
  f(a,b) -> c
  f(b,a) -> c
  f(b,b) -> c
)
```

Implementation

Our algorithms, implemented on top of TRS.Tool [Arn13], are here:

`http://zenon.dsic.upv.es/trstool/CompSCR.aspx`

We use infChecker [GL25] to test feasibility conditions

Benchmarks: COPS search sequences:

Alg. 1: `linear non_terminating non_confluent`

Alg. 2: `left_linear non_linear non_terminating non_weakly_orthogonal`

Table: Completion to Strong Confluence: Experiments

		Added Rules												
Alg.	**Success**	**0**	**1**	**2**	**3**	**4**	**6**	**8**	**9**	**11**	**12**	**No Result**	**Fail**	**Total**
1	39 (33%)	0	19	8	5	3	2	1	0	1	0	79	0	118
2	13 (27%)	1	3	2	2	1	2	0	1	0	1	31	2	46

Conclusions and future work

We have introduced new procedures for completion to different kind of *strong confluence* properties

The experimental evaluation of the approach is encouraging

As far as we know, the idea of *completion to strong confluence* and its implementation in COMPSCR are novel contributions of this paper.

Future work

1. Improve the implementation
2. Use parallel critical pairs [Gra96] in completion.

Thanks!

Completion to Strong Confluence

Salvador Lucas[1,2] and Julia Pagan[2]

[1]DSIC & [2]VRAIN, Universitat Politècnica de València (UPV)

15^{th} International Workshop on Confluence

IWC 2026

Lisboa, Portugal

Salvador Arnal.
Term Rewriting Systems .Net Framework (master thesis).
Master's thesis, Departamento de Sistemas Informáticos y Computación. Universitat Politècnica de València, Spain, July 2013.

Raúl Gutiérrez and Salvador Lucas.
Proving and disproving feasibility with infChecker.
In Raúl Gutérrez and Naoki Nishida, editors, *14th International Workshop on Confluence, IWC 2025*, pages 38–44, 2025.

Bernhard Gramlich.
Confluence without Termination via Parallel Critical Pairs.
In Hélène Kirchner, editor, *Trees in Algebra and Programming - CAAP'96, 21st International Colloquium, Proceedings*, volume 1059 of *Lecture Notes in Computer Science*, pages 211–225. Springer, 1996.
doi:10.1007/3-540-61064-2_39.

Nao Hirokawa and Aart Middeldorp.
Decreasing diagrams and relative termination.

J. Autom. Reason., 47(4):481–501, 2011.
doi:10.1007/S10817-011-9238-X.

Gérard P. Huet.
Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems.
J. ACM, 27(4):797–821, 1980.
doi:10.1145/322217.322230.

Donald E. Knuth and Peter E. Bendix.
Simple Word Problems in Universal Algebra.
In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

Julia Pagan.
Completion of Term Rewriting Systems to Ensure Strong Confluence (master thesis).
Master's thesis, Departamento de Sistemas Informáticos y Computación. Universitat Politècnica de València, Spain, September 2025.

Yoshihito Toyama.
Commutativity of Term Rewriting Systems.
In Kazuhiro Fuchi and Laurent Kott, editors, *Programming of Future Generation Computer, II. Proceedings of the Second Franco-Japanese Symposium on Programming of Future Generation Computers, Cannes, France, 9-11, November, 1987*, pages 393–407.
North-Holland, 1988.