Saigawa: A Confluence Tool*

Nao Hirokawa
JAIST, Japan
hirokawa@jaist.ac.jp

Dominik Klein
JAIST, Japan
dominik.klein@jaist.ac.jp

Saigawa is a tool for automatically proving or disproving confluence of (ordinary) term rewrite systems (TRSs). The tool, written in OCaml, is freely available from

http://www.jaist.ac.jp/project/saigawa/

This system description is based on Saigawa version 1.4. The typical usage of the tool is: `saigawa <file>`. Here the input file is written in the standard WST format. The tool outputs `YES` if confluence of the input TRS is proved, `NO` if non-confluence is shown, and `MAYBE` if the tool does not reach any conclusion. The tool is based on the next four confluence criteria.

Theorem 1 ([1, Theorem 3]). *A left-linear TRS $\mathcal{R}$ is confluent if* $\mathsf{CP}(\mathcal{R}) \subseteq {\downarrow_{\mathcal{R}}}$ *and* $\mathsf{CPS}'(\mathcal{R})/\mathcal{R}$ *is terminating.*

Theorem 2 ([4, Theorem 2]). *Suppose $\mathcal{R}$ and $\mathcal{S}$ are strongly non-overlapping on each other, $\mathcal{S}$ is confluent, and $\mathcal{R}/\mathcal{S}$ is terminating. The TRS $\mathcal{R} \cup \mathcal{S}$ is confluent iff* $\mathsf{CP}_{\mathcal{S}}(\mathcal{R}) \subseteq {\downarrow_{\mathcal{R}\cup\mathcal{S}}}$.

Theorem 3 ([5]). *A TRS $\mathcal{R}$ is confluent if every critical peak is decreasing with respect to the rule labeling heuristic.*

Theorem 4 ([3]). *Suppose* $\mathcal{R}/\mathsf{AC}$ *is terminating. The TRS* $\mathcal{R} \cup \mathsf{AC}$ *is confluent if and only if* $\mathsf{CP}_{\mathsf{AC}}(\mathcal{R}) \subseteq \to^*_{\mathcal{R},\mathsf{AC}} \cdot \leftrightarrow^*_{\mathsf{AC}} \cdot {}_{\mathcal{R},\mathsf{AC}}^{*}\!\leftarrow$.

Our tool uses $\mathsf{T_TT_2}$ and MU-TERM to check (relative) termination.[1] When termination of $\mathcal{R}$ is proved, for every $(s,t) \in \mathsf{CP}(\mathcal{R})$ the joinability $s \downarrow_{\mathcal{R}} t$ is tested by comparing normal forms of s and t. In the other cases we test $s \to^m_{\mathcal{R}} \cdot {}^{n}_{\mathcal{R}}\!\leftarrow t$ for each $(s,t) \in \mathsf{CP}(\mathcal{R} \cup \mathcal{R}^{-1})$ and $1 \leqslant m,n \leqslant 5$. Unjoinability is detected by testing whether $\mathsf{TCAP}_{\mathcal{R}}(s)$ and $\mathsf{TCAP}_{\mathcal{R}}(t)$ do not unify [6]. In order to apply Theorem 2 we need to appropriately split a TRS into $\mathcal{R}$ and $\mathcal{S}$. In our tool this is done by simple enumeration, and confluence of $\mathcal{S}$ is checked in a recursive manner. A suitable rule labeling is searched by using the SMT solver MiniSmt,[2] see [1, Section 4] for details of automation. We are planning to support a commutation criterion [2] in the next version.

References

[1] N. Hirokawa and A. Middeldorp. Decreasing diagrams and relative termination. *Journal of Automated Reasoning*, 47:481–501, 2011.

[2] N. Hirokawa and A. Middeldorp. Commutation via relative termination. In *Proc. 2nd IWC*, pages 29–33, 2013.

[3] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal on Computing*, 15(4):1155–1194, 1986.

[4] D. Klein and N. Hirokawa. Confluence of non-left-linear TRSs via relative termination. In *Proc. 18th LPAR*, volume 7180 of *LNCS*, pages 258–273, 2012.

[5] V. van Oostrom. Confluence by decreasing diagrams — converted. In *Proc. 19th RTA*, volume 5117 of *LNCS*, pages 306–320, 2008.

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

*This work is supported by the JSPS KAKENHI Grant Number 25730004.

[1] http://colo6-c703.uibk.ac.at/ttt2/ and http://zenon.dsic.upv.es/muterm/

[2] http://cl-informatik.uibk.ac.at/software/minismt/