Hakusan 0.11: A Confluence Tool

Fuyuki Kawano, Nao Hirokawa, and Kiraku Shintani

JAIST, Japan
{f-kawano,hirokawa}@jaist.ac.jp, s.kiraku@gmail.com

Hakusan (https://www.jaist.ac.jp/project/saigawa/) is a confluence tool for left-linear term rewrite systems (TRSs). It analyzes confluence by successive application of *rule removal* criteria [6, 4] based on rule labeling [5, 8] and critical pair systems [3]. Confluence proofs of Hakusan are now verifiable by CeTA [7], see [2].

With a small example we illustrate confluence analysis in our tool. Let $\mathcal{R}$ be a TRS and $\mathcal{S} \subseteq \mathcal{R}$ a subsystem of $\mathcal{R}$. We write $\mathsf{PCPS}(\mathcal{R},\mathcal{S})$ for the *parallel critical pair system* given by $\{s \to t, s \to u \mid t \;{}_{\mathcal{R}}{\leftarrow\!\!\!\!\!\parallel}\; s \xrightarrow{\epsilon}_{\mathcal{R}} u$ is a parallel critical peak but not $t \leftrightarrow^*_{\mathcal{S}} u\}$. Moreover, we write $\mathcal{R}\!\restriction_{\mathcal{S}}$ for the TRS $\{\ell \to r \in \mathcal{R} \mid \mathcal{F}\mathsf{un}(\ell) \subseteq \mathcal{F}\mathsf{un}(\mathcal{S})\}$.

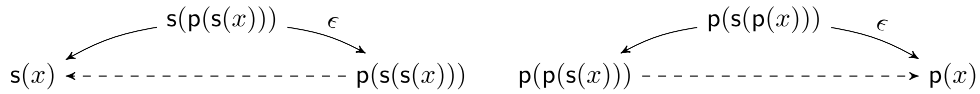
Theorem 1 ([4])**.** *Let $\mathcal{R}$ be a left-linear TRS and $\mathcal{S} \subseteq \mathcal{R}$. If every parallel critical pair of $\mathcal{R}$ is joinable, $\mathsf{PCPS}(\mathcal{R},\mathcal{S})/\mathcal{R}$ is terminating, and $\mathcal{R}\!\restriction_{\mathcal{S}} \subseteq \to^*_{\mathcal{S}}$ then $\mathcal{R}$ and $\mathcal{S}$ are equi-confluent.*

Example 1. *Consider the left-linear TRS $\mathcal{R}$:*

$$1\colon\ \mathsf{s}(\mathsf{p}(x)) \to \mathsf{p}(\mathsf{s}(x)) \qquad 2\colon\ \mathsf{p}(\mathsf{s}(x)) \to x \qquad 3\colon\ \infty \to \mathsf{s}(\infty)$$

We show the confluence of $\mathcal{R}$ by using the rule removal criteria based on parallel critical pair system and rule labeling.

(i) The TRS $\mathcal{R}$ admits two parallel critical peaks and the corresponding critical pairs join:



Let $\mathcal{S} = \{3\}$. The parallel critical pair system $\mathsf{PCPS}(\mathcal{R},\mathcal{S})$ consists of the four rules:

$$\begin{array}{ll} \mathsf{s}(\mathsf{p}(\mathsf{s}(x))) \to \mathsf{s}(x) & \mathsf{p}(\mathsf{s}(\mathsf{p}(x))) \to \mathsf{p}(\mathsf{p}(\mathsf{s}(x))) \\ \mathsf{s}(\mathsf{p}(\mathsf{s}(x))) \to \mathsf{p}(\mathsf{s}(\mathsf{s}(x))) & \mathsf{p}(\mathsf{s}(\mathsf{p}(x))) \to \mathsf{p}(x) \end{array}$$

By taking the linear polynomial interpretation $\mathcal{A}$ on $\mathbb{N}$ with

$$\mathsf{s}_{\mathcal{A}}(n) = 2n \qquad \mathsf{p}_{\mathcal{A}}(n) = n + 1 \qquad \infty_{\mathcal{A}} = 0$$

the inclusions $\mathsf{PCPS}(\mathcal{R},\mathcal{S}) \subseteq >_{\mathcal{A}}$ and $\mathcal{R} \subseteq \geqslant_{\mathcal{A}}$ hold. Thus, $\mathsf{PCPS}(\mathcal{R},\mathcal{S})/\mathcal{R}$ is terminating. As $\mathcal{F}\mathsf{un}(\mathcal{S}) = \{\mathsf{s}, \infty\}$ implies $\mathcal{R}\!\restriction_{\mathcal{S}} = \{3\} = \mathcal{S}$, we obtain $\mathcal{R}\!\restriction_{\mathcal{S}} \subseteq \to^*_{\mathcal{S}}$. Therefore, by Theorem 1 the TRSs $\mathcal{R}$ and $\mathcal{S}$ are equi-confluent.

(ii) As $\mathcal{S}$ admits no parallel critical peaks, the rule removal criterion based on rule labeling [4] proves the equi-confluence of $\mathcal{S}$ and $\varnothing$.

(iii) The empty TRS $\varnothing$ is trivially confluent.

Hence the original TRS $\mathcal{R}$ is confluent.

As a final remark, our tool employs the SMT solver Z3 [1] for automating the compositional confluence criteria and the reduction method.

References

[1] L. de Moura and N. Bjørner. Z3: An efficient SMT solver. In *Proceedings of 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *LNCS*, pages 337–340, 2008. The website of Z3 is: https://github.com/Z3Prover/z3. doi:10.1007/978-3-540-78800-3_24.

[2] N. Hirokawa, D. Kim, K. Shintani, and R. Thiemann. Certification of confluence- and commutation-proofs via parallel critical pairs. In *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 147–161, 2024. doi:10.1145/3636501.3636949.

[3] N. Hirokawa and A. Middeldorp. Decreasing diagrams and relative termination. *Journal of Automated Reasoning*, 47:481–501, 2011. doi:10.1007/s10817-011-9238-x.

[4] N. Hirokawa and K. Shintani. Rule removal for confluence. In *Proceedings of 13th International Workshop on Confluence*, 2024. In this proceedings.

[5] V. van Oostrom. Confluence by decreasing diagrams, converted. In *Proceedings of 19th International Conference on Rewriting Techniques and Applications*, volume 5117 of *LNCS*, pages 306–320, 2008. doi:10.1007/978-3-540-70590-1_21.

[6] K. Shintani and N. Hirokawa. Compositional confluence criteria. *Logical Methods in Computer Science*, 20:6:1–6:28, 2024. doi:10.46298/lmcs-20(1:6)2024.

[7] R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics*, volume 5674 of *LNCS*, pages 452–468, 2009.

[8] H. Zankl, B. Felgenhauer, and A. Middeldorp. Labelings for decreasing diagrams. *Journal of Automated Reasoning*, 54(2):101–133, 2015. doi:10.1007/s10817-014-9316-y.