Confluence Competition 2015

Takahito Aoto[1], Nao Hirokawa[2], Julian Nagele[3], Naoki Nishida[4], and Harald Zankl[3]

[1] Tohoku University, Japan
[2] JAIST, Japan
[3] University of Innsbruck, Austria
[4] Nagoya University, Japan

Abstract. Confluence is one of the central properties of rewriting. Our competition aims to foster the development of techniques for proving/disproving confluence of various formalisms of rewriting automatically. We explain the background and setup of the 4th Confluence Competition.

1 Introduction

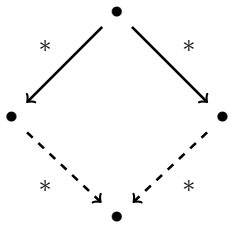

Fig. 1: Confluence

Confluence (Figure 1) provides a general notion of determinism and has been conceived as one of the central properties of rewriting [1]. Confluence has been investigated in many formalisms of rewriting such as first-order rewriting, lambda-calculi, higher-order rewriting, constrained rewriting, conditional rewriting, etc. More precisely, a rewrite system is a set of rewrite rules, and for each rewrite system $\mathcal{R}$, rewrite steps $s \to_{\mathcal{R}} t$ are associated. A rewrite system $\mathcal{R}$ is said to be confluent if for any $t_1 \,{}_{\mathcal{R}}\!\overset{*}{\leftarrow}\, t_0 \overset{*}{\to}_{\mathcal{R}} t_2$ there exists t_3 such that $t_1 \overset{*}{\to}_{\mathcal{R}} t_3 \,{}_{\mathcal{R}}\!\overset{*}{\leftarrow}\, t_2$, where $\overset{*}{\to}_{\mathcal{R}}$ is the reflexive transitive closure of $\to_{\mathcal{R}}$. The notions of rewrite rules, associated rewrite steps, and terms to be rewritten vary from one formalism to another. Confluence is also related to many important properties of rewriting such as the unique normal form property, ground confluence, etc.

The task of our competition is to foster the development of techniques for proving/disproving confluence automatically by setting up a dedicated and fair competition among confluence proving/disproving tools. The 4th Confluence Competition (CoCo 2015)[5] runs live during the *4th International Workshop on Confluence (IWC 2015)* collocated with the *25th International Conference on Automated Deduction (CADE-25)* in Berlin, Germany.

[5] `http://coco.nue.riec.tohoku.ac.jp`

$$\text{TRS:}\quad \left\{ \begin{array}{lll} +(0,y) & \to y, & \mathsf{sum}(\mathsf{nil}) \to 0 \\ +(\mathsf{s}(x),y) & \to \mathsf{s}(+(x,y)), & \mathsf{sum}(\mathsf{cons}(x,ys)) \to +(x,\mathsf{sum}(ys)) \end{array} \right\}$$

$$\text{CTRS:}\quad \left\{ \begin{array}{ll} +(0,y) & \to y \\ +(\mathsf{s}(x),y) & \to \mathsf{s}(+(x,y)) \\ \mathsf{fib}(0) & \to \mathsf{pair}(\mathsf{s}(0),0) \\ \mathsf{fib}(\mathsf{s}(x)) & \to \mathsf{pair}(w,y) \Leftarrow \mathsf{fib}(x) = \mathsf{pair}(y,z),\ +(y,z) = w \end{array} \right\}$$

$$\text{HRS:}\quad \left\{ \begin{array}{ll} \mathsf{map}\ (\lambda n.\ f\ n)\ \mathsf{nil} & \to \mathsf{nil} \\ \mathsf{map}\ (\lambda n.\ f\ n)\ (\mathsf{cons}\ x\ xs) & \to \mathsf{cons}\ (f\ x)\ (\mathsf{map}\ (\lambda n.\ f\ n)\ xs) \end{array} \right\}$$

Fig. 2: Three different formalisms of rewrite systems in CoCo 2015.

2 Categories

Since different formalisms capture different confluence problems and techniques for confluence proving, the competition is separated into several categories. Categories are divided into *competition* categories and *demonstration* categories.

Demonstration categories are a novelty of CoCo 2015. These categories are one-time events for demonstrating new attempts and/or merits of particular tools. Demonstration categories can be requested until 2 months prior to the competition.

In contrast to demonstration categories, competition categories are not only run in a single competition but also in future editions of the confluence competition. Competition categories can be requested until 6 months prior to the competition, in order to allow organizers to make a decision on the framework and semantics of the rewriting formalism and the input format of the problems. The following 4 competition categories are run in CoCo 2015. See Figure 2 for examples of the different rewriting formalisms.

TRS category This is a category for first-order term rewrite systems. The framework of first-order term rewrite systems is most fundamental in the theory of term rewriting (e.g. [1]).

CTRS category This is a category for *conditional* term rewriting. Conditional term rewriting allows to deal with conditions whose evaluation is defined recursively using the rewrite relation. Incorporation of conditional expressions is fundamental from the point of views of universal algebra (quasi-variety) and of functional programming. Depending on the interpretation of the conditions, 3 condition types are considered—namely, semi-equational, join and oriented types. We refer to the textbook [3] for details.

HRS category Many expressive formal systems such as systems of predicate logics, λ-calculi, process calculi, etc. need variable binding. Higher-order rewriting is a framework that extends first-order term rewriting by a binding mechanism. Various formalisms of higher-order rewriting have been proposed in the literature. This category deals with one of the most classical frameworks of higher-order rewriting, namely *higher-order rewriting systems* [2].

	number of tools	number of tool authors	categories
CoCo 2012	4	8	TRS/CPF
CoCo 2013	4	10	TRS/CPF
CoCo 2014	7	15	TRS/CTRS/CPF

Fig. 3: Statistics in the previous competitions.

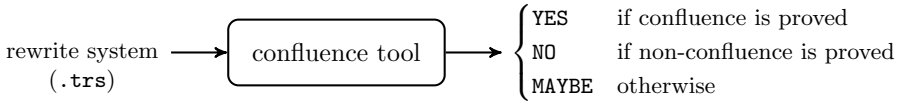

Fig. 4: Input and output scheme of a confluence tool.

CPF category This category is for the certification of confluence proofs based on interactive theorem provers. Here confluence tools must produce machine-checkable proofs which are checked by trustable certifiers in a second step.

In Figure 3, we list statistics and categories of the previous competitions. The HRS category has been incorporated for the first time in CoCo 2015.

3 Problems and Evaluation Process

We maintain a database of *confluence problems* (Cops), dealing with the three rewriting formalisms reflected in the competition categories. The community can submit problems prior to the competition. For the competition, only problems from the literature are considered, where this family collects examples from the literature (articles, papers, technical notes, and so on) dealing with confluence, avoiding test examples generated automatically or tend to have a similar structure. The actual problem sets for the competition are selected randomly from these problems considering the time balance. For the demonstration categories, the participants are requested to prepare the problems for the competition.

Figure 4 shows the input and output scheme required for tool participants. In the competition a set of confluence problems is submitted to each participating tool. Each tool is supposed to answer whether the given rewrite system is confluent or not. Tools must be able to run on the designated execution platform and read problems as input. The output of the tools must contain an answer in the first line followed by some proof argument understandable for human experts. Valid answers are `YES` (the input is confluent) and `NO` (the input is not confluent). Any other answer (such as `MAYBE`) is interpreted as the tool could not determine the status of the input. The timeout for each problem is set to 60 seconds for all categories. Every problem in the CTRS category is classified by the pair of a condition type (oriented, join, semi-equational), and a type of CTRS (type 1, 2, 3, or 4). A tool for the CTRS category should output `UNSUPPORTED` in

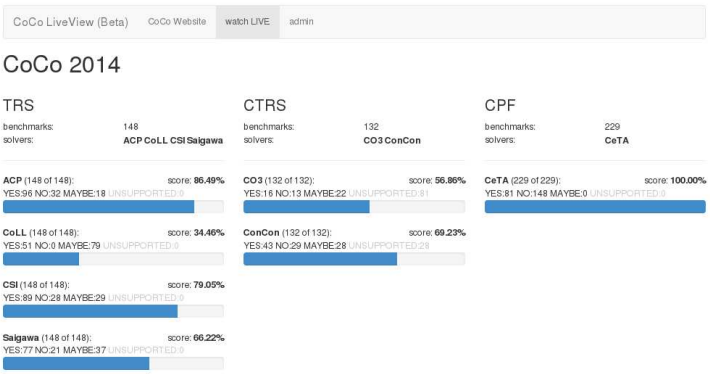

Fig. 5: LiveView of CoCo 2014.

place of `YES`/`NO` for input CTRSs in classes that the tool does not support. The unsupported classes of each tool must be declared at the time of registration. For the CPF category, each participant should output certifiable proofs as well as the result of certification: `YES` if confluence proof is certified and `NO` if non-confluence proof is certified.

The score is computed in percent of solved vs. supported problems (i.e. number of `YES`/`NO` answers vs. total number of `UNSUPPORTED` answers). In case of a draw there might be more winners. The tool with the maximal score wins. An answer is plausible if it was not falsified (automatically or manually). A tool with at least one non-plausible answer cannot be a winner.

4 Competition Platform and LiveView

The competition runs on a dedicated high-end cross-community competition platform *StarExec* [4]. The progress of the live competition is shared with the audience visually through the *LiveView* tool which interacts with StarExec. A screenshot of the LiveView is shown in Figure 5.

References

1. Baader, F., Nipkow, T.: Term Rewriting and All That. Cambridge University Press (1998)
2. Mayr, R., Nipkow, T.: Higher-order rewrite systems and their confluence. Theoretical Computer Science 192(1), 3–29 (1998)
3. Ohlebusch, E.: Advanced Topics in Term Rewriting Systems. Springer (2002)
4. Stump, A., Sutcliffe, G., Tinelli, C.: Starexec: a cross-community infrastructure for logic solving. In: Proc. 7th IJCAR. LNAI, vol. 8562, pp. 367–373 (2014)