ACPH: System Description for CoCo 2017

Kouta Onozawa[1], Kentaro Kikuchi[1], Takahito Aoto[2], and Yoshihito Toyama[1]

[1] RIEC, Tohoku University
{ `onozawa`, `kentaro`, `toyama` }`@nue.riec.tohoku.ac.jp`
[2] Faculty of Engineering, Niigata University
`aoto@ie.niigata-u.ac.jp`

Higher-order rewriting systems (HRSs) is a formalism of rewriting with variable binding and higher-order functions [2]. Higher-order rewriting deals with simply-typed lambda-terms with constants, which are identified modulo $\beta\eta$-equality. HRSs are a set of rewrite rules whose left-hand sides are restricted to patterns.

ACPH (Automated Confluence Prover for HRSs) is a tool for proving confluence of HRSs. If the tool succeeds to prove that an input HRS is confluent, it outputs YES. If the tool succeeds to prove that an input HRS is not confluent, it outputs NO. If the tool can not determine whether an input HRS is confluent or not, it outputs MAYBE. The tool uses following criteria for proving confluence and non-confluence of HRSs [1].

- If a HRS $\mathcal{R}$ is weakly orthogonal (left-linear and all critical pairs are trivial), then $\mathcal{R}$ is confluent.

- If a HRS $\mathcal{R}$ is terminating, then all critical pairs are joinable iff $\mathcal{R}$ is confluent.

The algorithms used in the program are based on those described in [1, 2]. For proving termination of HRSs, a higher-order termination tool WANDA[3] is used. ACPH program is written in Standard ML of New Jersey, and ACPH is provided as a heap image that can be loaded into SML/NJ runtime systems. It can be used from the command line by typing the following command:

```
$ sml @SMLload=acph.x86-linux <filename>
```

No new (non-)confluence criterion has been incorporated from the one submitted for CoCo 2016.

References

[1] Tobias Nipkow, Functional unification of higher-order patterns, *Proceedings of eighth annual IEEE symposium on logic in computer science*, pp.64-74, 1993.

[2] Richard Mayr, Tobias Nipkow, Higher-order rewrite systems and their confluence, *Theoretical computer science 192*, pp. 3-29, 1998.

[3] WANDA: A Higher-Order Termination Tool, http://wandahot.sourceforge.net/index.html