

IaCOP

Interface for the Administration of Cops*

Christian Nemeth
University of Innsbruck, Austria

Harald Zankl
University of Innsbruck, Austria

Nao Hirokawa
JAIST, Japan

Abstract

IaCOP maintains the problems considered for the competition of confluence tools. It allows the submission of new problems and supports automatic tagging of problems with certain properties. Problems can be filtered by boolean combinations of tags and the web interface comes with a mechanism to select problems for competitions.

1 Introduction

Beginning in the late 1980's the increasing power of computer hardware gave a tremendous boost to the area of automated deduction. Since then many tools have been developed to automatically reason about properties of programs. Nowadays many of these tools are successfully applied in software and hardware verification and are inevitable for the development of complex systems.

To spur the development of these tools and to give fair comparisons between them dedicated competitions take place regularly where tools compete against each other on a fixed (but previously typically unknown) selection of problems. Dating back to 1996 CASC¹ [4] has the longest tradition among such competitions. In CASC several provers try to prove or disprove validity of first-order logic formulas. Albeit undecidable in general, modern tools can establish a result in surprisingly many cases automatically. The first edition of the SAT competition² took place in 2002 where various SAT solvers investigate satisfiability of propositional logic formulas. Starting with a demo in 2003 the termination competition³ emerged where tools try to prove (refute) termination of programs automatically. While in the early years programs have been solely given as *term rewrite systems* nowadays termination of Prolog and Haskell programs as well as Java Bytecode is considered [2]. Satisfiability modulo Theories (SMT) extends propositional logic with various theories (arrays, bit-vectors, linear arithmetic, ...) and in 2005 the first SMT-COMP was conducted [1].

Obviously there are too many competitions to mention all of them but they have some properties in common. To make a competition thrilling a (large) database of interesting problems is essential and to run a competition within e.g. one day a suitable selection of problems must be performed. On the other hand tool authors want to test their tools on a subset of problems satisfying some given properties.

In 2012 the first edition of CoCo⁴ (*Confluence Competition*) will be run during the 1st International Workshop on Confluence, collocated with the 23rd International Conference on Rewriting Techniques and Applications (RTA 2012). The input problems considered for this competition—Cops,⁵ (*Confluence Problems*)—will be maintained by IaCOP, a web *Interface*

*Supported by the Austrian Science Fund (FWF) P22467 and the Grant-in-Aid for Young Scientists (B) 22700009 of the Japan Society for the Promotion of Science.

¹<http://www.cs.miami.edu/~tptp/CASC/>

²<http://www.satcompetition.org/>

³<http://termcomp.uibk.ac.at>

⁴<http://coco.nue.riec.tohoku.ac.jp/>

⁵<http://coco.nue.riec.tohoku.ac.jp/cops/>

```

46.trs
1 (VAR x)
2 (RULES
3   F(x,x) -> A
4   F(x,G(x)) -> B
5   C -> G(C)
6 )
7 (COMMENT from p.813 of \cite{Hue80})
8
tags: non confluent   non left linear   non linear   locally confluent
      non orthogonal  non terminating  non ground

```

Figure 1: A confluence problem displayed in IaCOP.

for the Administration of *Cops* [3]. This paper describes the main features of IaCOP, a test version of which is available from

<http://termcomp-devel.uibk.ac.at/csam1779/12iwc>

Cops is merely an archive of confluence problems (*.trs) in the old TRS syntax,⁶ and even lacks directory structures. However, IaCOP allows the submission of new problems and supports automatic *tagging* of problems with certain properties (see Figure 1). Problems can be filtered by boolean combinations of tags and the web interface comes with a mechanism to select problems for competitions. Hence IaCOP allows to maintain the database used for CoCo but is independent from the actual software running the competition. We anticipate that this design is more flexible and user friendly than combining both functionalities.

2 Features

IaCOP distinguishes between a plain *user mode* and an *administrator mode*. The functionality for users comprises the submission of new problems, searching of problems and downloading them. The administrator can (in addition) accept newly submitted problems, control the tag setting, and can initiate a selection of problems for competitions (depending on random seeds). Each item of the functionality is described in more detail below.

2.1 User Functionality

Submission: Users can upload single TRSs or zip archives in the old TRS syntax. The correct syntax is evaluated upon upload and only well-formed TRSs are uploaded. Useful auxiliary information (such as pointers to the literature or the origin of a problem) can be submitted in the `COMMENTS` field.

Search: IaCOP allows to search the content of problems as well as their tags by a full text search, i.e., words must be matched exactly. Search terms can be combined by boolean combinations, i.e., conjunctions, disjunctions, and complement. Hierarchic grouping of expressions is possible by braces.

⁶<http://www.lri.fr/~marche/tpdb/format.html>

Example 1. The search string “`terminating locally_confluent`” yields all problems with tags `terminating` and `locally_confluent`. To search problems that are neither left-linear nor (known to be) terminating the search string “`!left_linear !terminating`” is used. This can also be expressed by “`{!left_linear OR terminating}`” where braces are used to group expressions.

Since some tags are undecidable (e.g. `terminating`, `confluent`), the lack of such a tag may indicate that the problem does not have the property or that it is unknown if the problem has this property. Hence problems are also tagged with the inverse (e.g. `non_terminating`, `non_confluent`) if they are known not to satisfy the property. Consequently the search string “`!confluent !non_confluent`” yields all problems whose confluence status is unknown.

Download: The result set of any search can be downloaded either including the tag information in the problem or without it. The former allows to process problems also outside of IaCOP and the latter option is useful to provide test sets for competitions, where tools should not make use of additional tag information.

2.2 Administrator Functionality

Tagging: To assist the administrator, IaCOP can employ external tools to compute the tags of problems automatically. Those tools are started upon upload of a problem. Tags can also be maintained manually by the administrator. Tags can be renamed and new tags can be added to problems at any time.

Problem Management: Once new problems are submitted, the administrator gets notified by email and can then decide to accept these problems into Cops. Afterwards they get attached a unique id and become visible to standard users. While (for backwards compatibility) the actual content of a problem cannot be changed—once it is accepted—this restriction does not apply to tags. However, tags used for competitions (see below) cannot be changed.

Problem Selection: To select a subset of the Cops for competitions the administrator can add a new manual tag in IaCOP. After providing a lower and an upper bound on the number of selected problems and a list of random seeds (which influence the selection) IaCOP tags problems for a competition. For fairness of the whole process the random seeds should be contributed by the participants of the competition. For transparency the seeds and the result of the selection are made public immediately and tags used for competitions cannot be changed once attached to a problem.

3 Implementation

To couple the problems as loosely as possible to IaCOP no database is employed. Problem files are stored in the file system of the server running IaCOP as plain text files. Tag information is stored in the comment field of TRS problems. This allows the extraction of problems with all existing tag information from IaCOP, maybe add new tags outside of IaCOP and import the updated problems again.

Since problems are stored in plain text the search functionality is implemented by the command line tool `grep`.

4 Related Work

In this section we report on how other communities handle issues concerning the problem database used for competitions. Our insight varies tremendously depending on the specific competition and is reflected in the discussion below. Hence we only report on TermComp and SMT-COMP for which we have submitted problems in recent years.

TermComp: Problems are submitted via Email and the steering committee decides which problems are added to the database. We are not aware that any submission has ever been rejected. However, submitted problems are not immediately available to the outside world. Every now and then the current problems can be downloaded from the competition website.

SMT-COMP: Problems are submitted via Email and easy problems (which every tool could solve) are ignored. Again, problems are not immediately available to others. However, typically the benchmarks can be downloaded before the competition takes place.

For both competitions we are not aware of sophisticated search functionalities which allow the user to extract a set of problems with specific properties.

Finally we mention the ongoing efforts of Stump, Sutcliffe, and Tinelli to build StarExec,⁷ a uniform platform for running and managing different competitions. The first edition of CoCo will not be run on StarExec but we hope to use this great service already for its next edition.

5 Conclusion

This note presented IaCOP, an Interface for the Administration of Cops. IaCOP makes the submission of new problems attractive to researchers via an online form and supports the administrator in maintaining the database by automatic support for determining the values of tags. Tags (as well as problem content) can be searched by a full text search, which allows to filter problems with certain properties. Finally, IaCOP can select problems for a competition based on a list of random seeds as input.

Concerning future work, one extension could be to run all current confluence provers on newly submitted problems and show the result immediately to the one uploading these problems.

Acknowledgments We are grateful to Aart Middeldorp and Aaron Stump for their valuable suggestions. We thank the anonymous reviewers for their detailed and helpful comments.

References

- [1] C. Barrett, M. Deters, L. de Moura, A. Oliveras, and A. Stump. 6 years of SMT-COMP. *Journal of Automated Reasoning*. doi: 10.1007/s10817-012-9246-5. Online First, to appear.
- [2] J. W. (ed.). Report on the termination competition 2008. In *Proc. 10th International Workshop on Termination*, pages 100–111, 2009.
- [3] C. Nemeth. Interface for the administration of confluence problems (IaCOP). Bachelor thesis, University of Innsbruck, 2012.
- [4] G. Sutcliffe. The CADE-23 automated theorem proving system competition – CASC-23. *AI Communications*, 25(1):49–63, 2012.

⁷<http://www.starexec.org/>