IaCOP

— Interface for the Administration of Cops —

http://coco.nue.riec.tohoku.ac.jp/

Christian Nemeth (University of Innsbruck)

Harald Zankl (University of Innsbruck)

Nao Hirokawa (JAIST)*

Background

- several confluence tools and certifier appeared

ACP **CeTA** **CSI** **Saigawa** **...**

Background

- several confluence tools and certifier appeared

ACP **CeTA** **CSI** **Saigawa** **...**

- 1st Confluence Competition (CoCo 2012) will run in next session

Background

- several confluence tools and certifier appeared

ACP **CeTA** **CSI** **Saigawa** **...**

- 1st Confluence Competition (CoCo 2012) will run in next session

- common problem set is necessary for fair evaluation

☞ **Con**fluence **P**roblem**s** database (**Cops**)

Conceptual Difference

competition	competition software	problem database
CASC	?	TPTP (.tptp)
termCOMP	TermExec	TPDB (.xml)
CoCo 2012	Zankl's tool	**Cops** (.trs)
CoCo 2013	StarExec	**Cops** (.trs)

Conceptual Difference

competition	**competition software**	**problem database**
CASC	?	TPTP (.tptp)
termCOMP	TermExec	TPDB (.xml)
CoCo 2012	Zankl's tool	**Cops** (.trs)
CoCo 2013	StarExec	**Cops** (.trs)

- often competition site provides interface for database
- **IaCOP** is interface for **Cops** but not competition tool

Cops

Cops

DESIGN POLICY

- **Cops** is designed for new entrants and working developers
- communication should be easy

Cops

DESIGN POLICY

- **Cops** is designed for new entrants and working developers
- communication should be easy

Cops adopts approach of TPDB (Termination Problem Database)

Cops

DESIGN POLICY

- **Cops** is designed for new entrants and working developers
- communication should be easy

Cops adopts approach of TPDB (Termination Problem Database)
but there are differences ☞ next slides

Problem Syntax

Cops use old WST input format (.trs) TPDB switched to XML...

Problem Syntax

Cops use old WST input format (.trs)   TPDB switched to XML...

```
(VAR x y)                                   immutable
(RULES                                      immutable
  F(x,x) -> A
  F(x,G(x)) -> B
  C -> G(C)
)
(COMMENT from p.813 of \cite{Hue80})   mutable
```

- one needs human readable and writable syntax for communication

File Names

- files in **Cops** are just numbered

  `1.trs`, `2.trs`, `...`, `125.trs`

File Names

- files in **Cops** are just numbered

  `1.trs`, `2.trs`, `...`, `125.trs`

- please describe explanation in `COMMENT` but not in file name

File Names

- files in **Cops** are just numbered

  `1.trs`, `2.trs`, `...`, `125.trs`

- please describe explanation in `COMMENT` but not in file name

- short names (e.g. `D33/33`, `z086` in TPDB) are good for communication

Directory Structures

- **Cops** has no directory structures ☞ scripting is very easy

$$\textbf{Cops} = \{\texttt{1.trs},\ \texttt{2.trs},\ \ldots,\ \texttt{125.trs}\}$$

Directory Structures

- **Cops** has no directory structures ☞ scripting is very easy

$$\textbf{Cops} = \{\texttt{1.trs},\ \texttt{2.trs},\ \texttt{...},\ \texttt{125.trs}\}$$

- easy to use problem set:

```
for i in *.trs; do acp $i > a/$i.out; done
```

Tags

tags indicate properties and some more information

```
(VAR x y) (RULES
  F(x,x) -> A
  F(x,G(x)) -> B
  C -> G(C)  )
(COMMENT from p.813 of \cite{Hue80})
(COMMENT %% Tags: non_confluent non_left_linear ..)
```

Tags

tags indicate properties and some more information

```
(VAR x y) (RULES
  F(x,x) -> A
  F(x,G(x)) -> B
  C -> G(C)  )
(COMMENT from p.813 of \cite{Hue80})
(COMMENT %% Tags: non_confluent non_left_linear ..)
```

- easy to compute subsets:

```
grep -l "%%.*\<non_left_linear\>" *.trs
```

Tags

tags indicate properties and some more information

```
(VAR x y) (RULES
  F(x,x) -> A
  F(x,G(x)) -> B
  C -> G(C)  )
(COMMENT from p.813 of \cite{Hue80})
(COMMENT %% Tags: non_confluent non_left_linear ..)
```

- easy to compute subsets:

  ```
  grep -l "%%.*\<non_left_linear\>" *.trs
  ```

- TCT team had to write 150 lines of shell script for extracting

$$\{\mathcal{R} \in \textbf{TPDB} \mid \mathcal{R} \text{ is TRS and Oops answers } \texttt{NO}\}$$

IaCOP

IaCOP

IaCOP is interface for **Cops**

IaCOP

IaCOP is interface for **Cops**

FUNCTIONALITY

- display confluence problems

IaCOP

IaCOP is interface for **Cops**

FUNCTIONALITY

- display confluence problems
- tag search and download

IaCOP

IaCOP is interface for Cops

FUNCTIONALITY

- display confluence problems
- tag search and download
- problem submission

IaCOP

IaCOP is interface for **Cops**

FUNCTIONALITY

- display confluence problems
- tag search and download
- problem submission
- tag management

Tags

Cops has **tags** but no directory structures nor file names

Tags

Cops has **tags** but no directory structures nor file names

tags are used for

- properties of TRSs:

  `confluent`, `non_confluent`, `left_linear`, `non_left_linear`

Tags

Cops has **tags** but no directory structures nor file names

tags are used for

- properties of TRSs:

  `confluent`, `non_confluent`, `left_linear`, `non_left_linear`

- categories of problems:

  `literature`: examples from papers, articles, ...

  `crafted`: confluence problems that we manually made

  `generated`: by transformation methods, random generators, ...

  `debugging`: for debugging tools

Tags

Cops has **tags** but no directory structures nor file names

tags are used for

- properties of TRSs:

  `confluent`, `non_confluent`, `left_linear`, `non_left_linear`

- categories of problems:

`literature`:	examples from papers, articles, ...
`crafted`:	confluence problems that we manually made
`generated`:	by transformation methods, random generators, ...
`debugging`:	for debugging tools

- interesting subsets: `coco2012`, `certified`, ...

Current Tags

- properties (`non_..`):

  `confluent` `locally_confluent` `terminating`

  `left_linear` `linear` `ground`

  `orthogonal` `weakly_orthogonal`

Current Tags

- properties (non_..):

  confluent   locally_confluent   terminating

  left_linear   linear   ground

  orthogonal   weakly_orthogonal

- families:

  literature   crafted   generated   debugging

Current Tags

- properties (non_..):

  confluent locally_confluent terminating
  left_linear linear ground
  orthogonal weakly_orthogonal

- families:

  literature crafted generated debugging

- to be added (?):

  flat right_linear AC C
  coco2012 certified

Searching & Downloading

Definition

search expression:

$$E \ ::= \ \textit{tag} \ \mid \ E \ E \ \mid \ E \ \texttt{OR} \ E \ \mid \ \texttt{\{} \ E \ \texttt{\}} \ \mid \ \texttt{!} \ E$$

results can be downloaded

Searching & Downloading

Definition

search expression:

$$E \ ::= \ \mathit{tag} \ \mid \ E \ E \ \mid \ E \ \texttt{OR} \ E \ \mid \ \texttt{\{} \ E \ \texttt{\}} \ \mid \ \texttt{!} \ E$$

results can be downloaded

Example

search: $\boxed{\texttt{non\_left\_linear non\_terminating}}$

$$= \ \{\mathcal{R} \in \textbf{Cops} \mid \mathcal{R} \text{ is non-left-linear and (known) non-terminating}\}$$

$$= \ \{\texttt{1.trs}, \texttt{2.trs}, \texttt{3.trs}, \texttt{15.trs}, \ldots\}$$

Searching & Downloading

DEFINITION

search expression:

$$E \ ::= \ \mathit{tag} \mid E\ E \mid E\ \texttt{OR}\ E \mid \texttt{\{}\ E\ \texttt{\}} \mid \texttt{!}\ E$$

results can be downloaded

EXAMPLE

search: $\boxed{\texttt{non\_left\_linear non\_terminating}}$

$$= \{\mathcal{R} \in \textbf{Cops} \mid \mathcal{R} \text{ is non-left-linear and (known) non-terminating}\}$$
$$= \{\texttt{1.trs}, \texttt{2.trs}, \texttt{3.trs}, \texttt{15.trs}, \ldots\}$$

REMARK

`non_termianting` $\neq$ `! terminating` due to undecidability

How to Submit Problems?

- input problem on Web interface with on-the-fly syntax checker, or

How to Submit Problems?

- input problem on Web interface with on-the-fly syntax checker, or
- upload `.trs` or `.zip` file

How to Submit Problems?

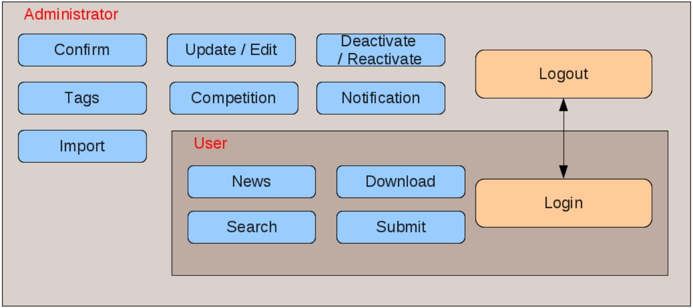
- input problem on Web interface with on-the-fly syntax checker, or
- upload `.trs` or `.zip` file
- problems are added after administrator confirms

How to Submit Problems?

- input problem on Web interface with on-the-fly syntax checker, or
- upload `.trs` or `.zip` file
- problems are added after administrator confirms
- property tags will be automatically added

Technical Design

Functionalities



Implementation

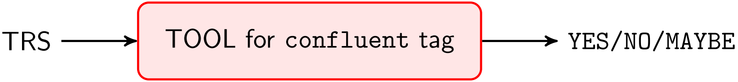
- server: PHP, grep (for searching), Flex and Bison (for syntax check)

Implementation

- server: PHP, grep (for searching), Flex and Bison (for syntax check)
- client: HTML, JavaScript (AJAX, jQuery) for syntax check

Implementation

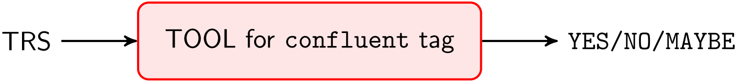
- server: PHP, grep (for searching), Flex and Bison (for syntax check)
- client: HTML, JavaScript (AJAX, jQuery) for syntax check
- each tag is maintained by tool:



TRS → TOOL for `confluent` tag → `YES/NO/MAYBE`

Implementation

- server: PHP, grep (for searching), Flex and Bison (for syntax check)
- client: HTML, JavaScript (AJAX, jQuery) for syntax check
- each tag is maintained by tool:



TRS → TOOL for `confluent` tag → `YES/NO/MAYBE`

$$\begin{cases} \texttt{confluent} & \text{if } \texttt{YES} \\ \texttt{non\_confluent} & \text{if } \texttt{NO} \\ \text{---} & \text{otherwise} \end{cases}$$

Conclusion and Further Work

- **Cops**: confluence problem database
- **laCOP**: interface for **Cops**, based on tags, online submission

Conclusion and Further Work

- **Cops**: confluence problem database
- **IaCOP**: interface for **Cops**, based on tags, online submission

extracting subsets is crucial for fostering new methods

Conclusion and Further Work

- **Cops**: confluence problem database
- **IaCOP**: interface for **Cops**, based on tags, online submission

extracting subsets is crucial for fostering new methods

FUTURE WORK

- import competition results

Conclusion and Further Work

- **Cops**: confluence problem database
- **IaCOP**: interface for **Cops**, based on tags, online submission

extracting subsets is crucial for fostering new methods

FUTURE WORK

- import competition results
- provide DOI or .bib

Conclusion and Further Work

- **Cops**: confluence problem database
- **IaCOP**: interface for **Cops**, based on tags, online submission

extracting subsets is crucial for fostering new methods

FUTURE WORK

- import competition results
- provide DOI or .bib
- new confluence problems are welcome!

Conclusion and Further Work

- **Cops**: confluence problem database
- **IaCOP**: interface for **Cops**, based on tags, online submission

extracting subsets is crucial for fostering new methods

FUTURE WORK

- import competition results
- provide DOI or .bib
- new confluence problems are welcome!
- call for administrators currently: Harald Zankl, and NH

starts soon

thank you for your attention