COCO 2021

10th Confluence Competition

Aart Middeldorp Kiraku Shintani Naoki Nishida Johannes Waldmann

http://project-coco.uibk.ac.at/2021

Outline

1. **Acknowledgements**
2. **History**
3. **2021**
4. **Outlook**

Acknowledgements

- CoCo 2021 tool authors

Acknowledgements

- CoCo 2021 tool authors
- Fabian Mitterwallner

Acknowledgements

- CoCo 2021 tool authors
- Fabian Mitterwallner
- IWC 2021 chairs

Acknowledgements

- CoCo 2021 tool authors
- Fabian Mitterwallner
- IWC 2021 chairs
- CoCo 2021 panel
  - Dieter Hofbauer
  - Koji Nakazawa
  - Femke van Raamsdonk

Outline

1. Acknowledgements
2. **History**
3. 2021
4. Outlook

2012	TRS	CPF
ACP	✓	
CSI	✓	✓
Saigawa	✓	
CeTA		✓

2013	TRS	CPF
ACP	✓	
CSI	✓	✓
Saigawa	✓	
CeTA		✓

2014	TRS	CPF	CTRS
ACP	✓	✓	
CSI	✓	✓	
Saigawa	✓		
CeTA		✓	
CoLL	✓		
ConCon		✓	✓
CO3			✓

2015	TRS	CPF	CTRS	HRS	GCR	NRS
ACP	✓	✓				
CSI	✓	✓				
CeTA		✓				
CoLL-Saigawa	✓					
ConCon		✓	✓			
CO3			✓			
CoScart			✓			
ACPH				✓		
CSI^ho				✓		
AGCP					✓	
NoCo						✓

2016	TRS	CPF	CTRS	HRS	GCR	NRS	UN	CPF-CTRS
ACP	✓	✓						
CSI	✓	✓					✓	
CeTA		✓						✓
CoLL-Saigawa	✓							
ConCon			✓					✓
CO3			✓					
CoScart			✓					
ACPH				✓				
CSI^ho				✓				
AGCP					✓			
Nrbox						✓		
FORT					✓		✓	

2017	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC
ACP	✓	✓							
CSI	✓	✓				✓	✓		✓
CeTA		✓						✓	
CoLL-Saigawa	✓								
ConCon			✓					✓	
CO3			✓						
ACPH				✓					
CSI^ho				✓					
AGCP					✓				
FORT					✓	✓	✓		✓
SOL				✓					

2018	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC
ACP	✓	✓	✓						✓
CSI	✓	✓				✓	✓		✓
CeTA		✓						✓	
CoLL-Saigawa	✓								
ConCon			✓					✓	
CO3			✓						
CSI^ho				✓					
AGCP					✓				
FORT					✓	✓	✓		✓
SOL				✓					

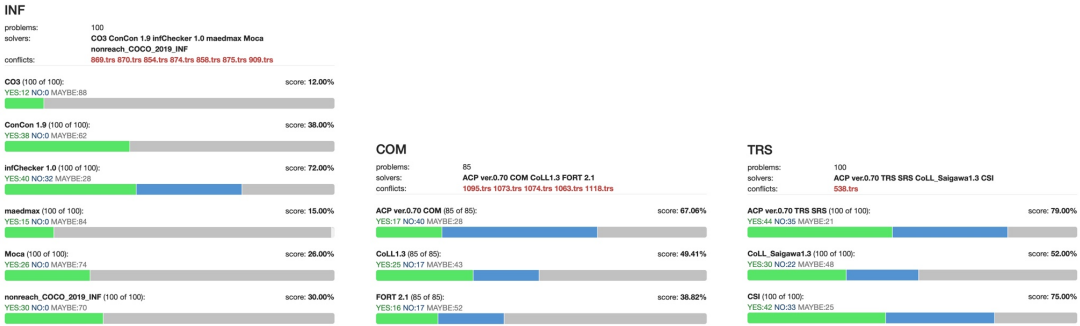
2019	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
ACP	✓	✓	✓						✓	✓		✓
CSI	✓	✓				✓	✓		✓			✓
CoLL										✓		
CeTA		✓						✓				
CoLL-Saigawa	✓											✓
ConCon			✓					✓			✓	
CO3			✓								✓	
infChecker											✓	
maedmax											✓	
CSI^ho				✓								
AGCP					✓							
Moca											✓	
FORT					✓	✓	✓		✓	✓		
noko-leipzig												✓
non-reach											✓	

2020	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
ACP	✓	✓	✓						✓	✓		✓
CSI	✓	✓				✓	✓		✓			✓
CoLL										✓		
CeTA		✓						✓				
CoLL-Saigawa	✓											✓
ConCon			✓					✓			✓	
CO3			✓								✓	
infChecker											✓	
CSI^ho				✓								
AGCP					✓							
Moca											✓	
FORT-h					✓	✓	✓		✓	✓		
SOL				✓								
non-reach											✓	

- CoCo is powered by StarExec

- CoCo is powered by StarExec
- exciting to watch

- CoCo is powered by StarExec
- exciting to watch, partly due to real-time yes / no conflicts



final slide CoCo 2019 presentation

- CoCo 2020 most likely in Paris, collocated with FSCD / IJCAR
- separate YES and NO results $\Longrightarrow$ more winners
- new T-shirt for participating tool authors

final slide CoCo 2019 presentation

- CoCo 2020 most likely in Paris, collocated with FSCD / IJCAR
- separate YES and NO results $\Longrightarrow$ more winners
- new T-shirt for participating tool authors

COCO
2020

Outline

1. Acknowledgements
2. History
3. **2021**
4. Outlook

CoCo 2021

COCO 2021

http://cocograph.uibk.ac.at/2021.html

2020	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
ACP	✓	✓	✓						✓	✓		✓
CSI	✓	✓				✓	✓		✓			✓
CoLL										✓		
CeTA		✓						✓				
CoLL-Saigawa	✓											✓
ConCon			✓					✓			✓	
CO3			✓								✓	
infChecker											✓	
CSI^ho				✓								
AGCP					✓							
Moca											✓	
FORT-h					✓	✓	✓		✓	✓		
SOL				✓								
non-reach											✓	

2021	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
ACP	✓	✓	✓						✓	✓		✓
CSI	✓	✓				✓	✓		✓			✓
CoLL										✓		
CeTA		✓						✓				
CoLL-Saigawa	✓											✓
CO3			✓								✓	
infChecker											✓	
FORTify	✓				✓		✓		✓	✓		
CONFident	✓		✓									✓
AGCP					✓							
NaTT											✓	
FORT-h					✓	✓	✓		✓	✓		

2021	TRS	CPF	CTRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
ACP	✓	✓	✓					✓	✓		✓
CSI	✓	✓			✓	✓		✓			✓
CoLL									✓		
CeTA		✓					✓				
CoLL-Saigawa	✓										✓
CO3			✓							✓	
infChecker										✓	
FORTify	✓			✓		✓		✓	✓		
CONFident	✓		✓								✓
AGCP				✓							
NaTT										✓	
FORT-h				✓	✓	✓		✓	✓		

2021	TRS	CPF	CTRS	GCR	NFP	UNR	UNC	COM	INF	SRS
ACP	✓	✓	✓				✓	✓		✓
CSI	✓	✓			✓	✓	✓			✓
CoLL								✓		
CeTA		✓								
CoLL-Saigawa	✓									✓
CO3			✓						✓	
infChecker									✓	
FORTify	✓			✓		✓	✓	✓		
CONFident	✓		✓							✓
AGCP				✓						
NaTT									✓	
FORT-h				✓	✓	✓	✓	✓		

Outline

1. Acknowledgements

2. History

3. **2021**

   Categories Rules Live View

4. Outlook

Categories

TRS	confluence of first-order term rewrite systems
CPF	certification for TRS
CTRS	confluence of first-order conditional term rewrite systems
HRS	confluence of higher-order rewrite systems
GCR	ground-confluence of many-sorted first-order rewrite systems
CPF-CTRS	certification for CTRS
NFP	normal form property of first-order rewrite systems
UNR	unique normal forms wrt reduction of first-order rewrite systems
UNC	unique normal forms wrt conversion of first-order rewrite systems
COM	commutation of first-order rewrite systems
INF	infeasibility
SRS	confluence of string rewrite systems

Outline

1. **Acknowledgements**
2. **History**
3. **2021**
   Categories Rules Live View
4. **Outlook**

Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES or NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2020 participate as demonstration tools in 2021

Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES or NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2020 participate as demonstration tools in 2021

- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
  - 4 secret problems submitted

Competition Rules

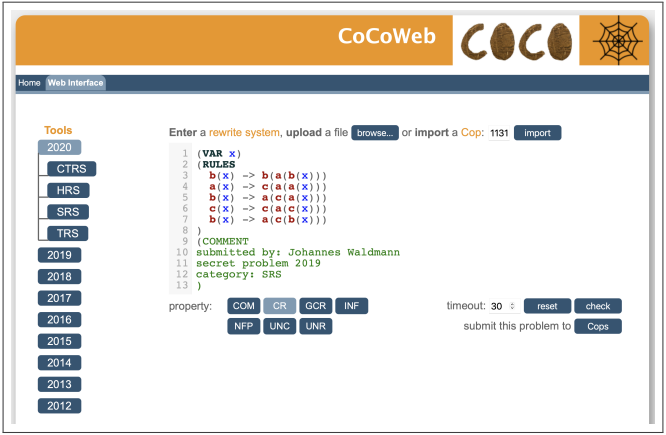
- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES or NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2020 participate as demonstration tools in 2021

- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
  - 4 secret problems submitted, including $\{\mathsf{f}(x) \to \mathsf{g}(\mathsf{b}),\ \mathsf{f}(x) \to \mathsf{g}(\mathsf{a}) \Leftarrow \mathsf{g}(x) \approx x\}$

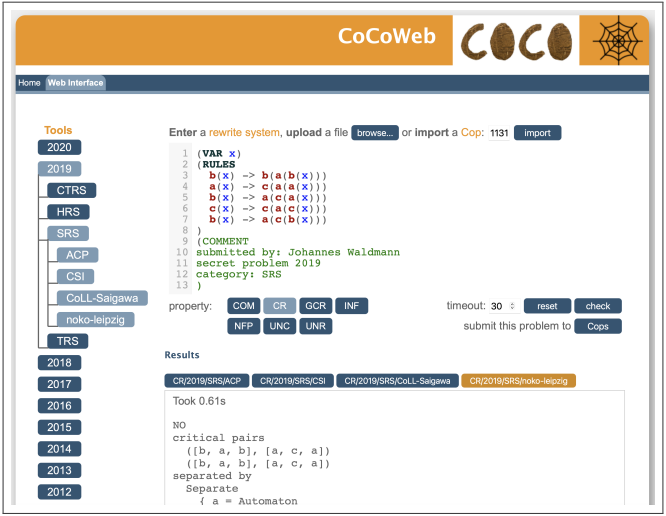
Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES or NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2020 participate as demonstration tools in 2021
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
  - 4 secret problems submitted, including $\{\mathsf{f}(x) \to \mathsf{g}(\mathsf{b}),\ \mathsf{f}(x) \to \mathsf{g}(\mathsf{a}) \Leftarrow \mathsf{g}(x) \approx x\}$
- Incorrect Results
  - tools with incorrect results (observed during live competition due to YES / NO conflict, or communicated afterwards by tool authors to SC) are excluded from results table

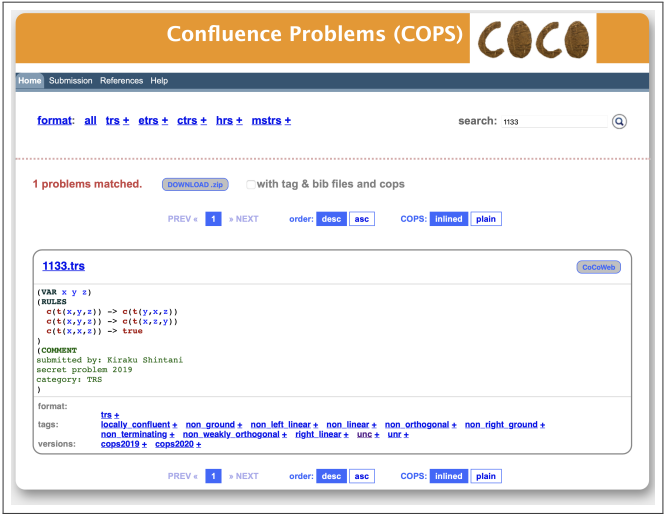
- (corrected) tools are available from CoCoWeb for testing



- (corrected) tools are available from CoCoWeb for testing



- (corrected) tools are available from CoCoWeb for testing
- confluence problems database (COPS) consists of 1155 problems



CoCo 2021 Live View

http://cocograph.uibk.ac.at/2021.html

2020 results

TRS category

- CSI reclaims top spot

2020 results

TRS category

- CSI reclaims top spot
- new participants in 2021: CONFident and (FORT-h+)FORTify

ACP (Automated Confluence Prover)

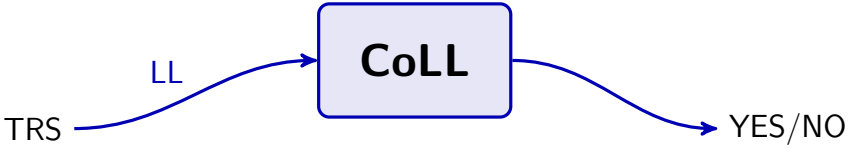
Takahito Aoto

- Entered to TRS/CPF-TRS/UNC/CTRS/COM/SRS
- Written in Standard ML of New Jersey (SML/NJ)
- Version: 0.10 (2009) $\cdots$ 0.70 (2019)
- Implementing multiple direct methods and divide-and-conquer methods
- For UNC, we use a decision procedure for shallow TRSs, based on:

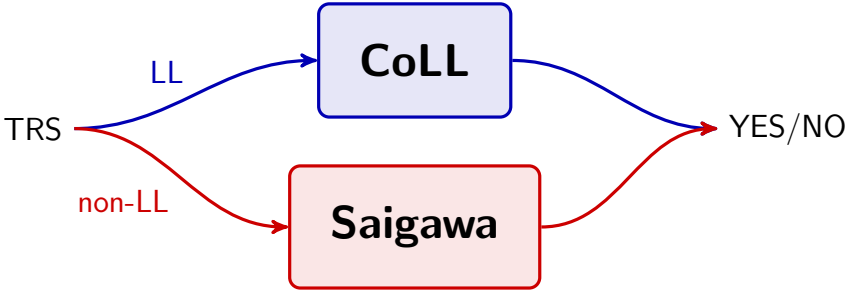
  A Fast Decision Procedure for UNC of Shallow TRS, M. Yamaguchi and T. Aoto, FSCD 2020

We couldn't make any efforts on ACP this year.

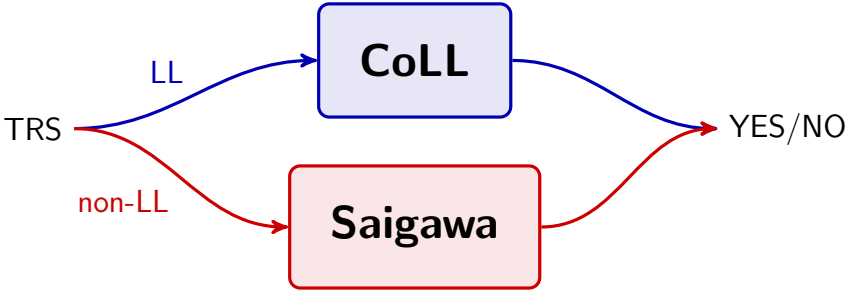
CoLL-Saigawa v1.6 (Shintani & Hirokawa, JAIST)



CoLL-Saigawa v1.6 (Shintani & Hirokawa, JAIST)

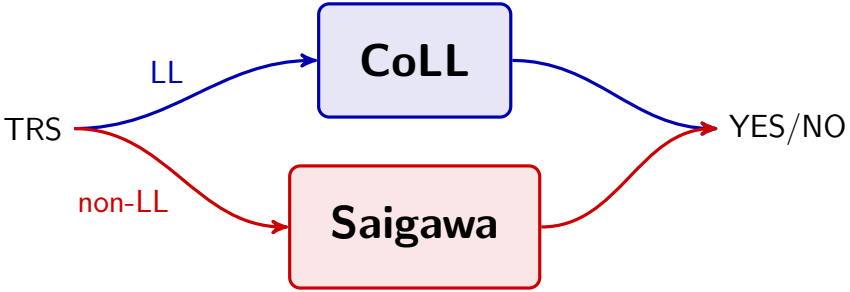


CoLL-Saigawa v1.6 (Shintani & Hirokawa, JAIST)



- CoLL-Saigawa uses MiniSmt, NaTT, and Z3

CoLL-Saigawa v1.6 (Shintani & Hirokawa, JAIST)



- CoLL-Saigawa uses MiniSmt, NaTT, and Z3
- no major changes from last year

Relatively New Features

Relatively New Features

- upside-parallel/outside closedness

Relatively New Features

- upside-parallel/outside closedness

Theorem (Oyamaguchi and Ohta 1997)

left-linear TRS is confluent if

- $\overset{p}{\leftarrow} \rtimes \overset{\epsilon}{\rightarrow} \subseteq \overset{\triangle p}{\leftarrow\!\!\!

Relatively New Features

- upside-parallel/outside closedness

Theorem (Oyamaguchi and Ohta 2004)

left-linear TRS is confluent if

- $\xleftarrow{p} \rtimes \xrightarrow{\epsilon} \subseteq \xleftarrow{\not> p} \cup \xrightarrow{\epsilon}{}^{=}$ *for all* $p > \epsilon$ *and*
- $\xleftarrow{\epsilon} \rtimes \xrightarrow{\epsilon} \subseteq (\overset{>\epsilon}{\leftarrow\!\!\!\!\leftarrow} \cup \xrightarrow{\epsilon}) \cdot (\leftarrow\!\!\!\!\leftarrow \cup \xrightarrow{\epsilon})$

Relatively New Features

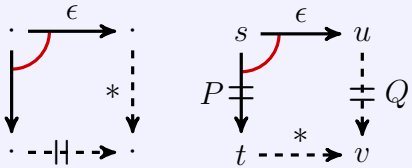
- upside-parallel/outside closedness
- criteria based on parallel/simultaneous critical pairs

Relatively New Features

- upside-parallel/outside closedness
- criteria based on parallel/simultaneous critical pairs

Theorem (Toyama 1981)

left-linear TRS is confluent if



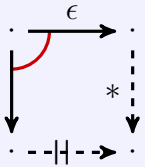
for some v *with* $\mathcal{V}\mathsf{ar}(v, Q) \subseteq \mathcal{V}\mathsf{ar}(s, P)$

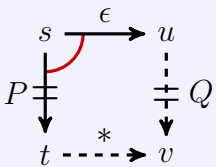
Relatively New Features

- upside-parallel/outside closedness
- criteria based on parallel/simultaneous critical pairs

Theorem (Toyama 1981)

left-linear TRS is confluent if

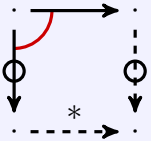




for some v *with* $\mathcal{V}\mathsf{ar}(v, Q) \subseteq \mathcal{V}\mathsf{ar}(s, P)$

Theorem (Okui 1998)

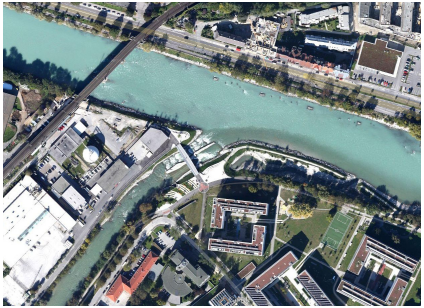
left-linear TRS is confluent if



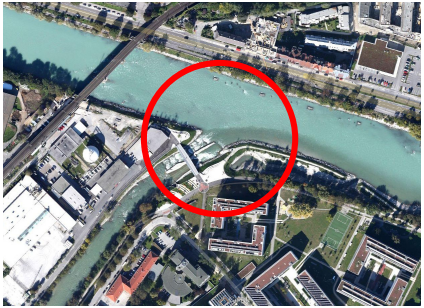
universität innsbruck

CoCo 2021 Participant: CSI 1.2.5

Fabian Mitterwallner Aart Middeldorp

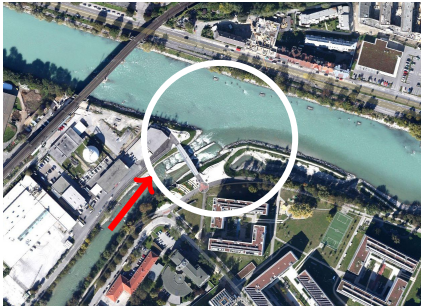


[Google Maps]



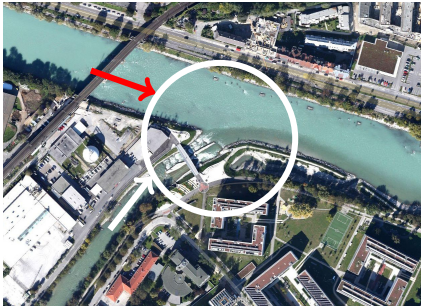
[Google Maps]

C



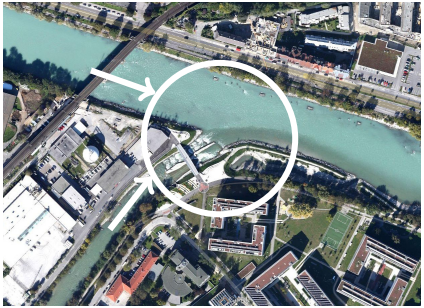
[Google Maps]

CS



[Google Maps]

CSI



[Google Maps]

CSI 1.2.5

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

CoCo 2020 Categories

- CPF-TRS NFP TRS UNC UNR SRS

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

CoCo 2020 Categories

- CPF-TRS NFP TRS UNC UNR SRS 1st place

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

CoCo 2021 Categories

- CPF-TRS NFP TRS UNC UNR SRS

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

CoCo 2021 Categories

- CPF-TRS NFP TRS UNC UNR SRS

New Features

- extended upside-parallel-closed criterion with relaxed root overlap condition (Oyamaguchi and Ohta, RTA 1997)

- open source
- convenient web interface
- `http://cl-informatik.uibk.ac.at/software/csi/`

CoCo 2021 Categories

- CPF-TRS NFP TRS UNC UNR SRS

New Features

- extended upside-parallel-closed criterion with relaxed root overlap condition (Oyamaguchi and Ohta, RTA 1997)
- added criterion for confluence of outside-closed rewrite systems (Oyamaguchi and Ohta, IEICE TIS 2004)

2020 results

CPF-TRS category

universität
innsbruck

CoCo 2021 Participant: CeTA 2.40

René Thiemann

CeTA 2.40

- CeTA: certifier of various properties, verified in Isabelle/HOL
- confluence part of CeTA developed by
  - Bertram Felgenhauer
  - Julian Nagele
  - Christian Sternagel
  - Thomas Sternagel
  - René Thiemann
  - Sarah Winkler
  - Akihisa Yamada
  - Harald Zankl
- several confluence techniques supported, see a complete list at:

  `http://cl-informatik.uibk.ac.at/software/ceta/`
- no new confluence techniques in CeTA 2.40 in comparison to 2020
- improved support for confluence techniques based on (relative) termination
  - included multiset-comparisons in weighted path order

  $\longrightarrow$ new potential to optimise your certified confluence strategy

2020 results

CTRS category

- ConCon regains top spot

2020 results

CTRS category

- ConCon regains top spot
- new participant in 2021: CONFident

CO3 (Ver. 2.2)

a **CO**nverter for proving **CO**nfluence of **CO**nditional TRSs

Naoki Nishida
Nagoya University, Japan

Overview

CO3 proves confluence of 3-DCTRSs or infeasibility of conditions by using

- very simple termination/confluence criteria for TRSs,
- the improved sequential unraveling $\mathbb{U}_{conf}$ [Gmeiner et al, 13], and
- narrowing trees [Nishida & Maeda, 18]
- reduction of confluence of join or semi-equational CTRSs to that of oriented ones

Infeasibility Criterion for Condition c w.r.t. DCTRSs $\mathcal{R}$

$\mathbb{U}_{conf}(\mathcal{R})$ is right-linear and a narrowing tree for c defines $\emptyset$ [Maeda et al, 19]

Confluence Criteria for syntactically deterministic 3-CTRSs $\mathcal{R}$

- $\mathcal{R}$ is weakly-left-linear and $\mathbb{U}_{conf}(\mathcal{R})$ is confluent [Gmeiner et al, 13], or
- $\mathbb{U}_{conf}(\mathcal{R})$ is terminating and right-linear, and
  $\forall\langle s,t\rangle \Leftarrow c \in CP(\mathcal{R}),\ (c = \epsilon \wedge s = t) \vee$ "c is infeasible" [Maeda et al, 19]

CONFident **at CoCo 2021**

Miguel Vítores[1] Raúl Gutiérrez[2] Salvador Lucas[1]

BUENOS AIRES, JULY 23RD, 2021

[1] Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València
Spain
[2] Universidad Politécnica de Madrid
Spain

Description

- CONFident is a tool for checking **(non-)confluence** of systems based on rewriting by means of its logical representation.
- Specially designed to (dis)prove confluence of
  - Conditional Term Rewriting Systems (join, oriented, and, soon, semi-equational).
  - Context-Sensitive Term Rewriting Systems.
  - Conditional Context-Sensitive Rewriting Systems.
- Also able to (dis)prove confluence of
  - Term Rewriting Systems and
  - String Rewriting Systems

Implementation

- The tool is available here: `http://zenon.dsic.upv.es/confident/`.
- It is written in Haskell and it is based on a divide-and-conquer schema where different techniques are applied recursively using a pre-defined strategy.
- The logical representation of the systems is well-suited to check **(non-)joinability** of (conditional) critical pairs:
  - Proving and disproving **conditional joinability** by translation into combinations of (in)feasibility problems.
  - Feasibility problems solved using infChecker
  - Termination and operational termination of (C)TRSs is proved using MU-TERM.

Strategy and Results

- Our **proof strategy** is:
  1. we apply simplification and modular techniques;
  2. we analyze the system to extract good properties (external tools like MU-TERM can be used to check termination and operational termination);
  3. we compute the set of conditional critical pairs;
  4. we apply the semantic based techniques to prove or disprove the joinability of the conditional critial pairs.
- CoCo SC agreed to host a **CSR subcategory** in 2022.
- Bibliography:

GLV21 R. Gutiérrez, S. Lucas, and M. Vítores. Confluence of Conditional Rewriting in Logic Form. Submitted, 2021.

GL20 R. Gutiérrez and S. Lucas. Automatically Proving and Disproving Feasibility Conditions. In Proc. of IJCAR'2020, LNCS 12167:416–435. Springer, 2020.

Luc19 S. Lucas. Proving semantic properties as first-order satisfiability. Artificial Intelligence 277, paper 103174, 24 pages, 2019.

2020 results

- new participant in 2021: (FORT-h+)FORTify

AGCP (Automated Ground Confluence Prover)

Takahito Aoto

A ground confluence prover for many-sorted TRSs

- **An entrant of GCR category**
- **Written in Standard ML of New Jersey (SML/NJ)**
- **Methods:**
  - **rewriting induction for bounded convertibility**
  - **equivalent transformations, disproving methods**
- **Based on:**

  Improvements of the Rewriting Induction Approach for proving GCR, Aoto/Toyama/Kimura, FSCD 2017

We couldn't make any efforts on AGCP this year.

2020 results

NFP category

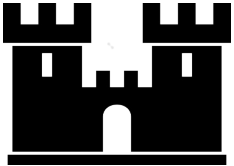
universität innsbruck

supported by FWF project P30301

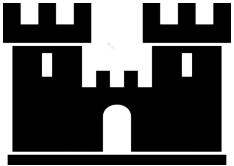
CoCo 2021 Participant: FORT-h 1.1

Fabian Mitterwallner Jamie Hochrainer Aart Middeldorp

FORT-h



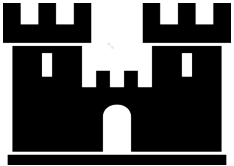
FORT-h

property →  ← TRS

↓

yes | no | ?

FORT-h



linear variable-separated TRS

property

$$\forall s\, \exists t\, (s \to^* t \wedge \neg \exists u\, (t \to u) \implies \exists v\, (s \not\to v \vee v \xrightarrow{\epsilon} t))$$

yes | no | ?

property is arbitrary formula in first-order theory of rewriting

CoCo 2020 Categories (FORT-h)

GCR NFP UNC UNR COM

CoCo 2020 Categories (FORT-h)

GCR NFP UNC UNR COM

most YES results

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

Differences FORT and FORT-h (2020)

- modified decision procedure
- supports linear variable-separated TRSs
- more expressive theory ($\rightarrow_{>\epsilon}$)
- goal: certified results

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

Differences 2020 and 2021

- certified results!
- FORTify can certify proofs

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

Differences 2020 and 2021

- certified results!
- FORTify can certify proofs
- optimized signature extension results (IWC 2021)

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

Differences 2020 and 2021

- certified results!
- FORTify can certify proofs
- optimized signature extension results (IWC 2021)
- faster/smaller automata constructions via eager epsilon eliminations

CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

Differences 2020 and 2021

- certified results!
- FORTify can certify proofs
- optimized signature extension results (IWC 2021)
- faster/smaller automata constructions via eager epsilon eliminations

`https://fortissimo.uibk.ac.at/fort(ify)/`

2020 results

- new participant in 2021: (FORT-h+)FORTify

universität innsbruck

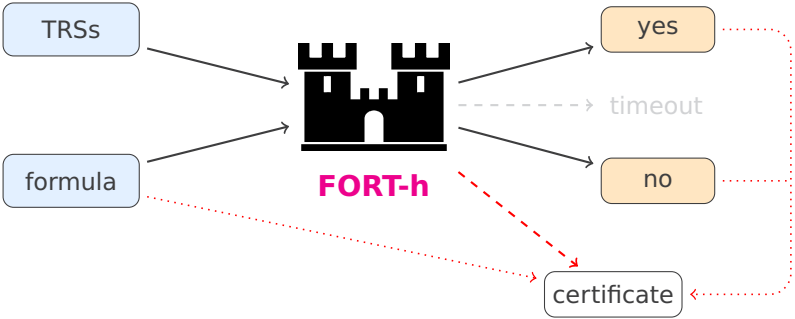
supported by FWF project P30301

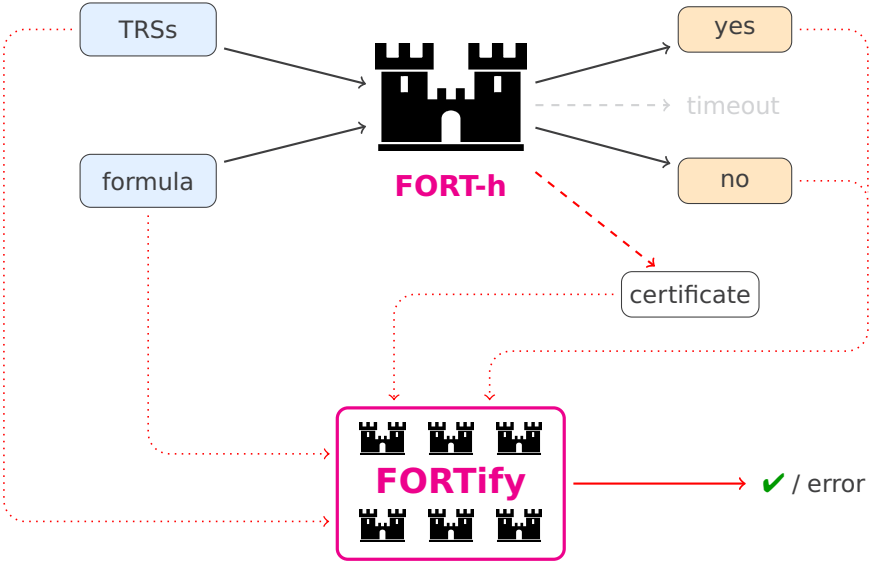
FORTify

Alexander Lochmann Fabian Mitterwallner

Aart Middeldorp

University of Innsbruck


TRSs
formula
FORT-h
yes
timeout
no
certificate


TRSs
formula
FORT-h
yes
timeout
no
certificate
FORTify
✔ / error

CoCo 2021 Categories (FORT-h + FORTify)

COM TRS GCR UNC UNR

CoCo 2021 Categories (FORT-h + FORTify)

COM TRS GCR UNC UNR

FORTify

code generated via Isabelle/HOLs code generation facility consisting of

- data type declarations for formulas and certificates

CoCo 2021 Categories (FORT-h + FORTify)

COM TRS GCR UNC UNR

FORTify

code generated via Isabelle/HOLs code generation facility consisting of

- data type declarations for formulas and certificates
- function that checks the correctness of certificates

CoCo 2021 Categories (FORT-h + FORTify)

COM TRS GCR UNC UNR

FORTify

code generated via Isabelle/HOLs code generation facility consisting of

- data type declarations for formulas and certificates
- function that checks the correctness of certificates

details in "Certifying Proofs in the First-Order Theory" (TACAS 2021)

CoCo 2021 Categories (FORT-h + FORTify)

COM TRS GCR UNC UNR

FORTify

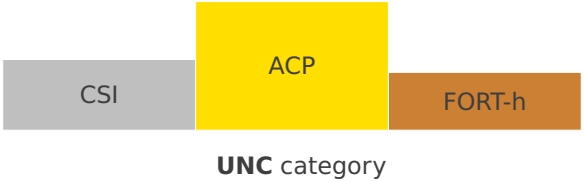
code generated via Isabelle/HOLs code generation facility consisting of

- data type declarations for formulas and certificates
- function that checks the correctness of certificates

details in "Certifying Proofs in the First-Order Theory" (TACAS 2021)

`https://fortissimo.uibk.ac.at/fort(ify)/`

2020 results



- new participant in 2021: (FORT-h+)FORTify

2020 results

COM category

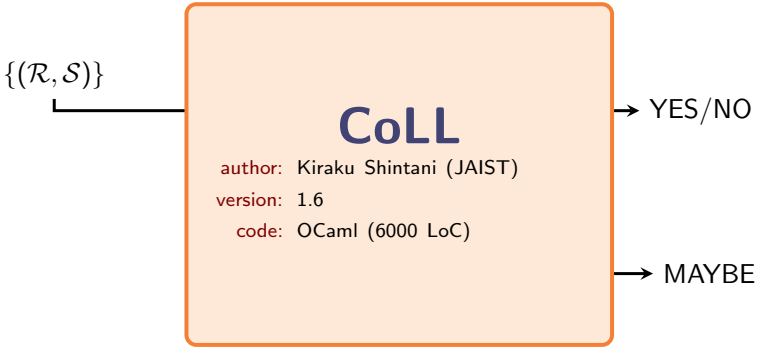
- first win for ACP

2020 results

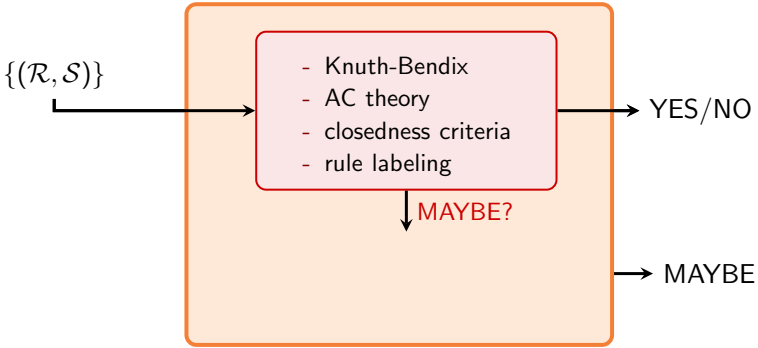
COM category

- first win for ACP
- new participant in 2021: (FORT-h+)FORTify

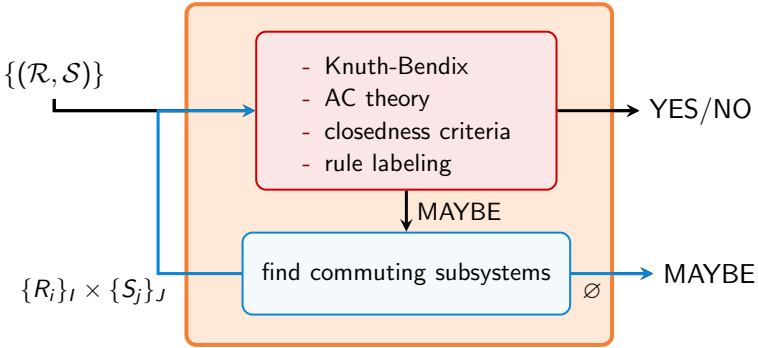
CoLL: Commutation Tool for Left-Linear TRSs



CoLL: Commutation Tool for Left-Linear TRSs

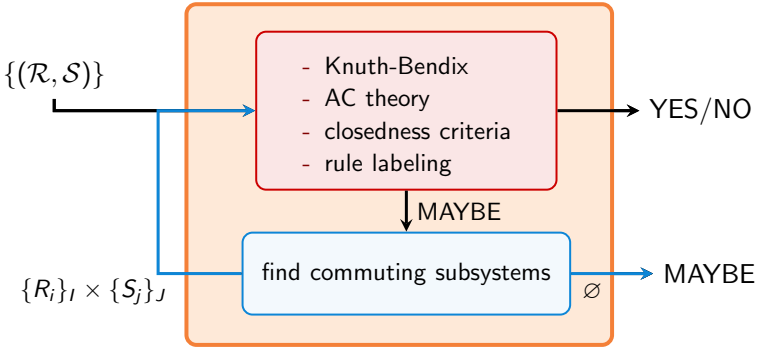


CoLL: Commutation Tool for Left-Linear TRSs



- based on Hindley's commutation lemma:
  $\bigcup_i \mathcal{R}_i$ and $\bigcup_j \mathcal{S}_j$ commute if $\mathcal{R}_i$ and $\mathcal{S}_j$ commute for all i, j

CoLL: Commutation Tool for Left-Linear TRSs



- based on Hindley's commutation lemma:
  $\bigcup_i \mathcal{R}_i$ and $\bigcup_j \mathcal{S}_j$ commute if $\mathcal{R}_i$ and $\mathcal{S}_j$ commute for all i, j
- left-linearity is often essential for commutation

What's new?

supported several closedness criteria

1. commutation
   - almost parallel closedness (Toyama 1988)
2. self commutation (confluence)
   - extension of parallel closedness (Toyama 1981)
   - simultaneous closedness (Okui 1998)
   - upside parallel closedness (Ohta and Oyamaguchi 1997)
   - outside closedness (Ohta and Oyamaguchi 2003)

What's new?

supported several closedness criteria

1. commutation
   - extension of parallel closedness (Toyama 1981)

2. self commutation (confluence)
   - ~~extension of parallel closedness (Toyama 1981)~~
   - simultaneous closedness (Okui 1998)
   - upside parallel closedness (Ohta and Oyamaguchi 1997)
   - outside closedness (Ohta and Oyamaguchi 2003)

What's new?

supported several closedness criteria

1. commutation
   - extension of parallel closedness (Toyama 1981)
   - simultaneous closedness (Okui 1998)
2. self commutation (confluence)
   - ~~extension of parallel closedness (Toyama 1981)~~
   - ~~simultaneous closedness (Okui 1998)~~
   - upside parallel closedness (Ohta and Oyamaguchi 1997)
   - outside closedness (Ohta and Oyamaguchi 2003)

2020 results

INF category

- most contested category with 6 participating tools

2020 results

INF category

- most contested category with 6 participating tools
- new participant in 2021: NaTT

infChecker **at CoCo 2021**

Raúl Gutiérrez[1] Salvador Lucas[2] Miguel Vítores[2]

BUENOS AIRES, JULY 23RD, 2021

[1] Universidad Politécnica de Madrid
Spain
[2] Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València
Spain

Description

- infChecker is a tool for checking **(in)feasibility of goals** $\mathcal{G} = \{F_i\}_{i=1}^{m}$, where $F_i = (s_{ij} \bowtie_{ij} t_{ij})_{i=1}^{n_i}$.
- $\bowtie_{ij}$ represents **predicates** on terms defined by provability of goals $s \bowtie_{ij} t$ with respect to a *first-order theories* $\mathrm{Th}_{\bowtie_{ij}}$.
- $\bowtie_{ij}$ can be one of the following predicates:
  - One (CS-)rewriting step (`->`, `\->`).
  - Zero or more (CS-)rewriting steps (`->*`, `\->*`).
  - One or more (CS-)rewriting steps (`->+`, `\->+`).
  - Subterm (`|>=`) and strict subterm (`|>`).
  - (CS-)Joinability (`->*<-`, `\->*<-/`).
  - One (CS-)convertibility step (`<->`, `<-/\->`).
  - Zero or more (CS-)convertibility steps (`<->*`, `<-/\->*`).

Implementation

- The tool is available here: `http://zenon.dsic.upv.es/infChecker/`.
- It is written in Haskell and provides a first implementation of the **Feasibility Framework**, where four **processors** have been implemented:
  - P^{Sat} integrates a satisfiability approach to **prove infeasibility using model generators** as AGES and Mace4 to find a proof.
  - P^{UR} **simplifies** problems by removing non-usable rules.
  - P^{Prov} integrates a logic-based approach to program analysis to **prove feasibility by theorem proving**. In infChecker, we use the theorem prover Prover9.
  - P^{NC} adapt the processor that **narrow conditions** in the 2D DP framework for proving operational termination of CTRs to be used with feasibility sequences.

Strategy and Results

- Our **proof strategy** is:
  1. we apply P^{UR} whenever it is sound and complete;
  2. we try to prove feasibility using P^{Prov};
  3. if P^{Prov} fails, we apply P^{Sat};
  4. if P^{Sat} fails, we apply P^{NC};
  5. if P^{NC} succeeds and modifies the feasibility sequence, we repeat the strategy, otherwise we return `MAYBE`.
- Bibliography:

GL20 R. Gutiérez and S. Lucas. Automatically Proving and Disproving Feasibility Conditions. In Proc. of IJCAR'2020, LNCS 12167:416–435. Springer, 2020.

Luc19 S. Lucas. Proving semantic properties as first-order satisfiability. Artificial Intelligence 277, paper 103174, 24 pages, 2019.

LG18 S. Lucas and R. Gutiérrez. Use of Logical Models for Proving Infeasibility in Term Rewriting. Information Processing Letters, 136:90-95, 2018.

NaTT in CoCo

Akihisa Yamada (AIST, Tokyo Waterfront)

- a **termination prover** for **plain** term rewriting
- participating **INF** with an easy checker [Sternagel & Yamada '19]
  - deigned for approximating dependency graphs & usable rules
  - weak but fast
  - ignores conditions
- Changes for participating CoCo: **Parsing** COPS format for INF
  - employed a generic rule-based parser
  - eventually, the termCOMP format can be directly read

https://www.trs.cm.is.nagoya-u.ac.jp/NaTT/

2020 results

- new participants in 2021: CONFident and (FORT-h+)FORTify

and the 2021 winners are . . .

http://cocograph.uibk.ac.at/2021.html

Outline

1. Acknowledgements
2. History
3. 2021
4. **Outlook**

Outlook

- CoCo 2022 during FLoC 2022

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)
- new category: confluence of context-sensitive rewriting

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)
- new category: confluence of context-sensitive rewriting
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)
- new category: confluence of context-sensitive rewriting
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)
- new category: confluence of context-sensitive rewriting
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS

Outlook

- CoCo 2022 during FLoC 2022 (olympic games?)
- new category: confluence of context-sensitive rewriting
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS