



## 9th Confluence Competition

Aart Middeldorp   Kiraku Shintani   Naoki Nishida   Johannes Waldmann

<http://project-coco.uibk.ac.at/2020>

# Outline

**1. Acknowledgements**

**2. History**

**3. 2020**

# Acknowledgements

- CoCo 2020 tool authors

# Acknowledgements

- CoCo 2020 tool authors
- Fabian Mitterwallner

# Acknowledgements

- CoCo 2020 tool authors
- Fabian Mitterwallner
- IWC 2020 chairs

# Acknowledgements

- CoCo 2020 tool authors
- Fabian Mitterwallner
- IWC 2020 chairs
- CoCo 2020 panel
  - Frédéric Blanqui
  - Cynthia Kop
  - Geoff Sutcliffe

# Outline

1. Acknowledgements










**2. History**

3. 2020

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

	TRS	CPF	CTRS	HRS	GCR	NRS
 ACP						
 CSI						
 CeTA						
 CoLL-Saigawa						
 ConCon						
 CO3						
 CoScart						
 ACPH						
 CSI <sup>ho</sup>						
 AGCP						
 NoCo						

2016

TRS CPF CTRS HRS GCR NRS UN CPF-CTRS

	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

	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

	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

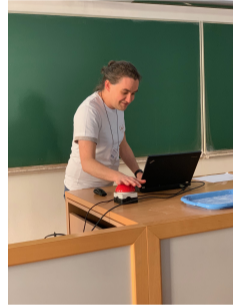
	TRS	CPF	CTRS	HRS	GCR	NFP	UNR	CPF-CTRS	UNC	COM	INF	SRS
 ACP	✓	✓	✓						✓	✓		✓
 CSI	✓	✓				✓	✓		✓			✓
 CoLL										✓		
 CeTA		✓						✓				
 CoLL-Saigawa	✓											✓
 ConCon			✓					✓			✓	
 CO3			✓								✓	
 infChecker											✓	
 maedmax											✓	
 CSI^ho				✓								
 AGCP					✓							
 Moca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


- CoCo is powered by StarExec



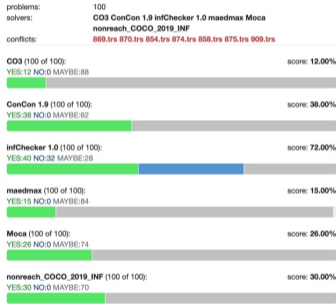


- CoCo is powered by StarExec
- **exciting** to watch

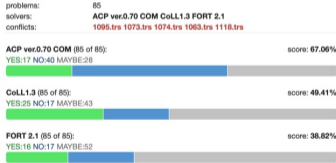


- CoCo is powered by StarExec 
- **exciting** to watch, partly due to real-time **yes/no conflicts**

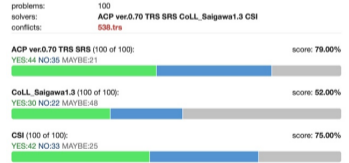
### INF



### COM



### TRS



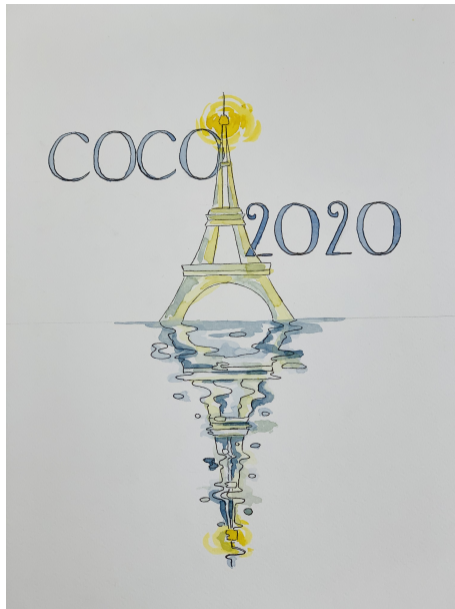
# final slide CoCo 2019 presentation

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





# Outline

## 1. Acknowledgements

## 2. History

## 3. 2020

Categories

Rules

Live View

Results
















# 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

















2019

TRS CPF CTRS HRS GCR NFP UNR CPF-CTRS UNC COM INF SRS

	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

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

# Outline

## 1. Acknowledgements

## 2. History

## 3. 2020

Categories

Rules

Live View

Results

# 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 2019 participate as demonstration tools in 2020

# 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 2019 participate as demonstration tools in 2020
- **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 2019 participate as demonstration tools in 2020
- **Secret Problems**
  - guaranteed to be selected
  - at most two problems per category per tool
  - 4 secret problems submitted, including  $a(b(a(x))) \rightarrow a(a(b(a(b(x))))))$

# 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 2019 participate as demonstration tools in 2020
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
  - 4 secret problems submitted, including  $a(b(a(x))) \rightarrow a(a(b(a(b(x))))))$
- **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

The screenshot shows the CoCoWeb web interface. At the top, there is a navigation bar with the text "CoCoWeb" and a logo consisting of the letters "COCO" in a stylized font next to a spiderweb icon. Below the navigation bar, there is a header "Home Web Interface".

The main content area is titled "Tools" and features a vertical sidebar on the left with buttons for the years 2019, 2018, 2017, 2016, 2015, 2014, 2013, and 2012. Under the year 2019, there are buttons for "CTRS", "HRS", "SRS", and "TRS".

The central part of the interface is a text editor for a rewrite system. It contains the following code:

```
1 (VAR x)
2 (RULES
3  a(b(x)) -> b(c(x))
4  c(b(x)) -> b(c(x))
5  c(b(x)) -> c(c(x))
6  b(b(x)) -> a(c(x))
7  a(b(x)) -> a(b(x))
8  c(c(x)) -> c(b(x))
9  a(c(x)) -> c(a(x))
10 )
11 (COMMENT
12 submitted by: Johannes Waldmann
13 )
14 )
```

Below the code editor, there is a "property:" label followed by buttons for "COM", "CR", "GCR", "INF", "NFP", "UNC", and "UNR". To the right of these buttons, there is a "timeout:" field set to "30", and buttons for "reset", "check", and "submit this problem to Cops".



- (corrected) tools are available from **CoCoWeb** for testing

**Tools**

- 2019
  - CTRS
  - HRS
  - SRS
  - ACP
  - CSI
  - CoLL-Saigawa
  - noko-leipzig
- 2018
- 2017
- 2016
- 2015
- 2014
- 2013
- 2012

Enter a **rewrite system**, upload a file  or import a Cop:

```

1 (VAR x)
2 (RULES
3  a(b(x)) -> b(c(x))
4  c(b(x)) -> b(c(x))
5  c(b(x)) -> c(c(x))
6  b(b(x)) -> a(c(x))
7  a(b(x)) -> a(b(x))
8  c(c(x)) -> c(b(x))
9  a(c(x)) -> c(a(x))
10 )
11 (COMMENT
12  submitted by: Johannes Waldmann
13 )
14 )

```

property:

timeout:     
submit this problem to

**Results**

```

Took 0.02s
NO
not joinable
CP
{ lhs = [b, c, b]
, rhs = [a, a, c]
..* = f_{a, b} ~ f_{b, a}

```

- (corrected) tools are available from CoCoWeb for testing
- confluence problems database (**COPS**) consists of 1150 problems

Confluence Problems (Cops)

[Home](#) [Submission](#) [References](#) [Help](#)

format: [all](#) [trs](#) [etrs](#) [ctrs](#) [hrs](#) [mstrs](#) [+](#) search:

---

**1 problems matched.** [DOWNLOAD .zip](#) with tag & bib files and cops

PREV = [1](#) = NEXT    order: [desc](#) [asc](#)    COPS: [inlined](#) [plain](#)

[1133.trs](#)
[CoCoWeb](#)

```

(VAR x y z)
(RULES
  c(t(x,y,z)) -> c(t(y,x,z))
  c(t(x,y,z)) -> c(t(x,z,y))
  c(t(x,x,z)) -> true
)
(COMMENT
submitted by: Kiraku Shintani
secret problem 2019
category: TRS
)
format: trs +
tags: locally\_confluent non\_ground non\_left\_linear non\_linear non\_orthogonal non\_right\_ground non\_terminating non\_weakly\_orthogonal right\_linear unc unr +
versions: cops2019 +

```

PREV = [1](#) = NEXT    order: [desc](#) [asc](#)    COPS: [inlined](#) [plain](#)

<http://cocograph.uibk.ac.at/2020.html>

# Outline

## 1. Acknowledgements

## 2. History

## 3. 2020

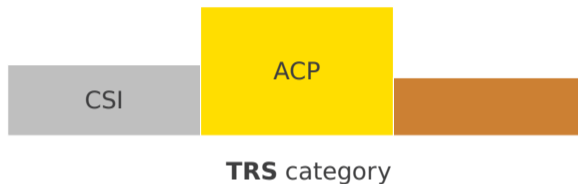
Categories

Rules

Live View

Results

# 2019 results



- ACP breaks 3-year hegemony of CSI
- incorrect answers by CoLL-Saigawa

# ACP (Automated Confluence Prover)

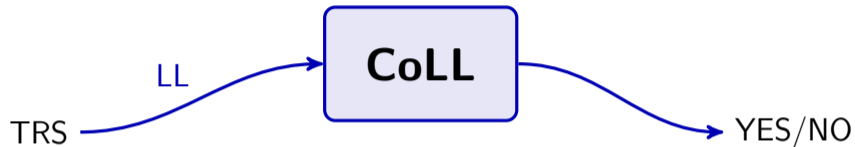
Takahito Aoto

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

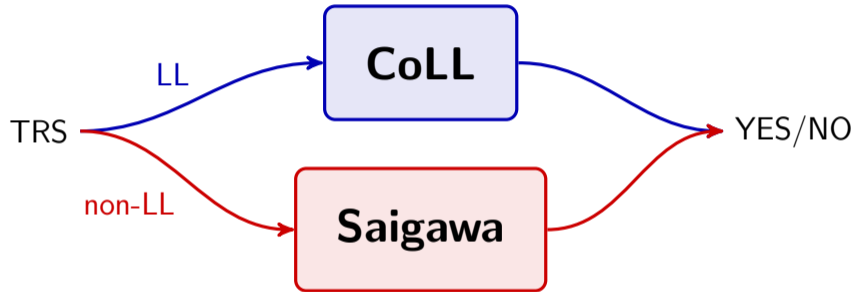
*A Fast Decision Procedure for UNC of Shallow TRS,  
M. Yamaguchi and T. Aoto, FSCD 2020*

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

# CoLL-Saigawa v1.5 (Shintani & Hirokawa, JAIST)

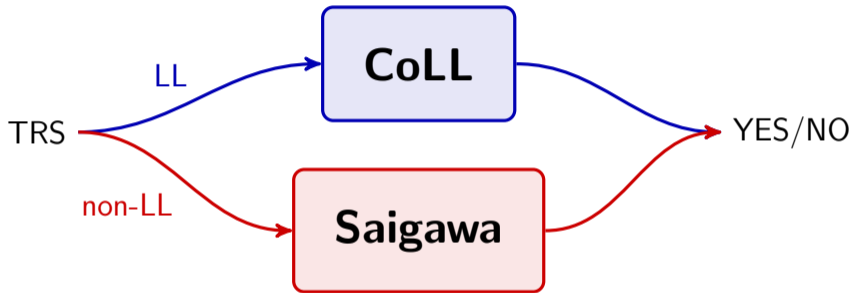


# CoLL-Saigawa v1.5 (Shintani & Hirokawa, JAIST)



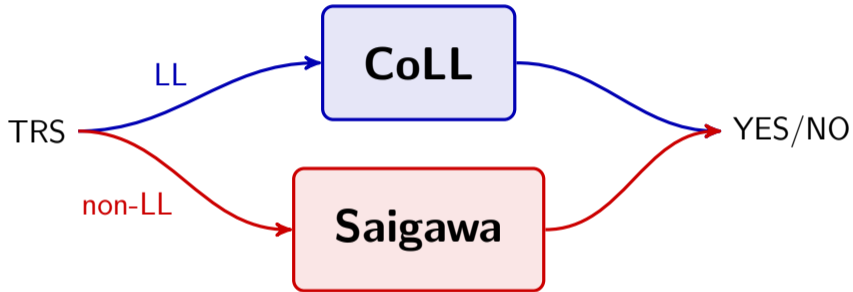


# CoLL-Saigawa v1.5 (Shintani & Hirokawa, JAIST)



- CoLL-Saigawa uses MiniSmt, NaTT, and Z3

# CoLL-Saigawa v1.5 (Shintani & Hirokawa, JAIST)



- CoLL-Saigawa uses MiniSmt, NaTT, and Z3
- it no longer uses  $T_1T_2$  (= CSI)

# New Features

# New Features

- improved performance of AC-related methods

# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness

# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness

## Theorem (Oyamaguchi and Ohta 1997)

*left-linear TRS is confluent if*

$$\blacksquare \leftarrow^p \times \xrightarrow{\epsilon} \subseteq \leftarrow^{\Delta p} \cup \xrightarrow{\epsilon} = \text{ for all } p > \epsilon \text{ and}$$

$$\blacksquare \leftarrow^{\epsilon} \times \xrightarrow{\epsilon} \subseteq (\leftarrow^{>\epsilon} \cup \xrightarrow{\epsilon}) \cdot (\leftarrow^{\epsilon} \cup \xrightarrow{\epsilon})$$

*where,  $s \xrightarrow{\Delta p} t$  if  $s \xrightarrow{Q} t$  for some  $Q \subseteq \{q \mid |q| \leq |p|\}$*

# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness

## Theorem (Oyamaguchi and Ohta 2004)

*left-linear TRS is confluent if*

- $\overleftarrow{p} \times \overrightarrow{\epsilon} \subseteq \overleftarrow{>p} \cup \overrightarrow{\epsilon} =$  for all  $p > \epsilon$  and
- $\overleftarrow{\epsilon} \times \overrightarrow{\epsilon} \subseteq (\overleftarrow{>\epsilon} \cup \overrightarrow{\epsilon}) \cdot (\overleftarrow{+} \cup \overrightarrow{\epsilon})$

# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness
- supported criteria based on parallel/simultaneous critical pairs

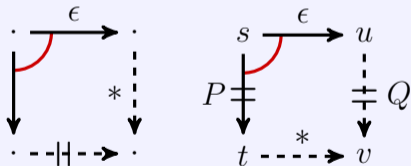


# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness
- supported criteria based on parallel/simultaneous critical pairs

## Theorem (Toyama 1981)

*left-linear TRS is confluent if*



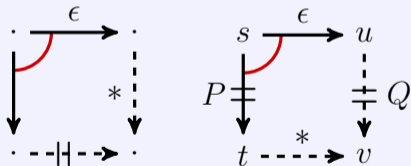
*for some  $v$  with  $\text{Var}(v, Q) \subseteq \text{Var}(s, P)$*

# New Features

- improved performance of AC-related methods
- supported upside-parallel/outside closedness
- supported criteria based on parallel/simultaneous critical pairs

## Theorem (Toyama 1981)

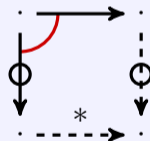
left-linear TRS is confluent if



for some  $v$  with  $\text{Var}(v, Q) \subseteq \text{Var}(s, P)$

## Theorem (Okui 1998)

left-linear TRS is confluent if





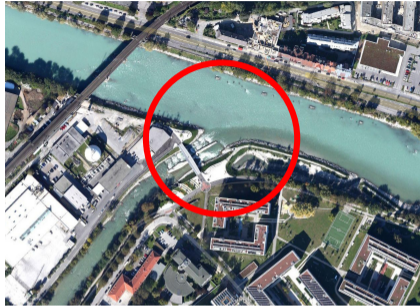
CoCo 2020 Participant: **CSI** 1.2.4

Fabian Mitterwallner

Aart Middeldorp

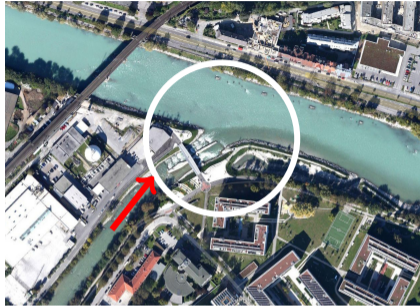


[Google Maps]



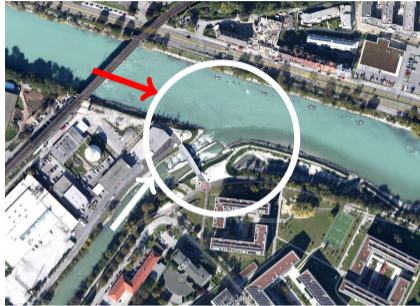
[Google Maps]

C



[Google Maps]

CS



[Google Maps]

CSI



[Google Maps]

## CSI 1.2.4



- 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 2019 Categories

- CPF-TRS    NFP    TRS    UNC    UNR    SRS

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

## CoCo 2019 Categories

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

- 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

## New Features

- criterion for **confluence** of upside-parallel-closed rewrite systems (Oyamaguchi and Ohta, RTA 1997)

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

## CoCo 2020 Categories

- CPF-TRS    NFP    TRS    UNC    UNR    SRS

## New Features

- criterion for confluence of upside-parallel-closed rewrite systems (Oyamaguchi and Ohta, RTA 1997)
- minor changes to the strategy for **(certified) confluence** for performance improvements

# 2019 results





CoCo 2020 Participant: CeTA 2.39

Jonas Schöpf, Christian Sternagel,  
René Thiemann and Akihisa Yamada

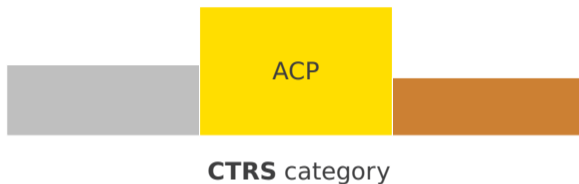


## CeTA 2.39



- CeTA: certifier of confluence proofs, verified in Isabelle/HOL
- several confluence techniques supported, see a complete list at:  
<http://cl-informatik.uibk.ac.at/software/ceta/>
- no new confluence techniques in CeTA 2.39
- **improved support for** confluence techniques based on (relative) **termination**
  - added support for **weighted path order** (WPO)
  - added support for **max-polynomial interpretations**, with WPO or stand-alone
  - certifiable proofs available via TTT2 and, for the first time, via NaTT
- new potential to **optimise your certified confluence strategy**

# 2019 results



- surprise win by ACP
- incorrect answers by ConCon and CO3

# CO3 (Ver. 2.1)

a COnverter for proving COnfluence of COnditional 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]

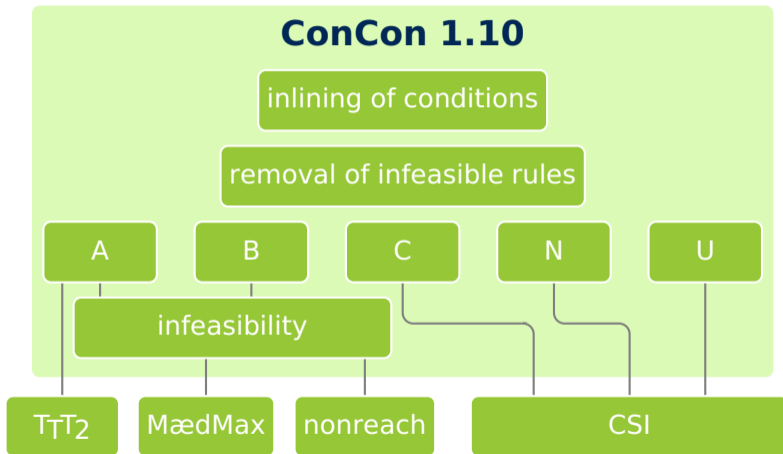
## Infeasibility Criterion for Condition $c$ w.r.t. DCTRSs

$\mathbb{U}_{conf}(\mathcal{R})$  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]

# Christian Sternagel



- open source
- <http://cl-informatik.uibk.ac.at/software/concon/>

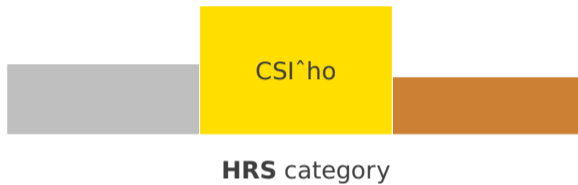
## New Features

- external infeasibility tool: nonreach by Florian Meßner

## CoCo 2020 Categories

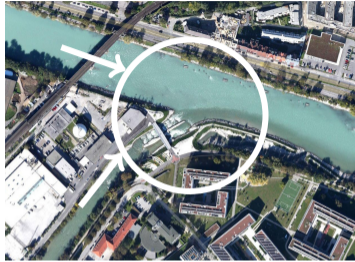
- CTRS
- CPF-CTRS
- INF

# 2019 results



- more participation welcome!

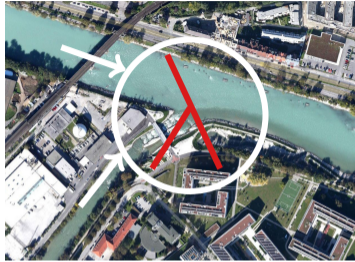
Julian Nagele



[Google Maps]

C S I

Julian Nagele



[Google Maps]

CSI ^ h o 0.3.2



# Julian Nagele



[Google Maps]

CSI<sup>h</sup>o 0.3.2

- extension of CSI for higher-order (pattern) rewrite systems
- <http://cl-informatik.uibk.ac.at/software/csi/ho/>

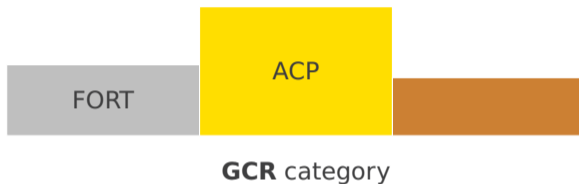
2020

- no new features

## The System SOL version 2020

- ▷ **Haskell-based tool** for analysing **confluence** and **termination**
- ▷ Authors: Makoto Hamana (Main developer, Gunma University)  
Kentarō Kikuchi (Tohoku University)  
Date Yao Faustin Dieudonne, Kazuki Fujū (Gunma University)
- ▷ References:
  - **Polymorphic computation systems: Theory and practice of confluence with call-by-value**, Hamana. Science of Computer Programming, Vo.187, 2020.
  - **How to prove decidability of equational theories with second-order computation analyser SOL**, Hamana, Abe, Kikuchi. Journal of Functional Programming, Vol. 29, e20, 2019.
- ▷ CR: Knuth and Bendix's critical pair checking  
using **Functions-as-Constructors unification (FCU)** [Libal,Miller'16]
- ▷ SN: - the General Schema criterion [Blanqui, RTA'00,TCS'16]  
- Modular termination of second-order computation systems
- ▷ Contact **Hamana** (hamana@gunma-u.ac.jp)  
if you are interested in/want to use SOL

# 2019 results



- only previously unsolved problems due to incorrect selection query

# 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.**

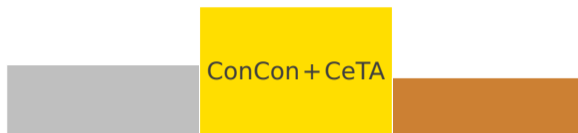
# 2019 results



# 2019 results



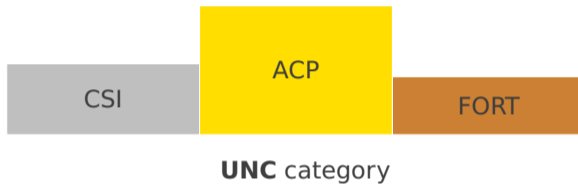
# 2019 results



**CPF-CTRS** category

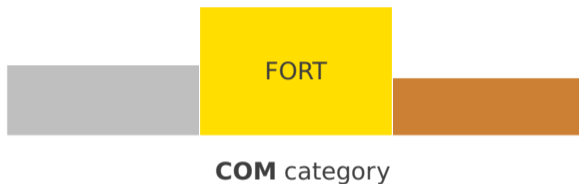


# 2019 results



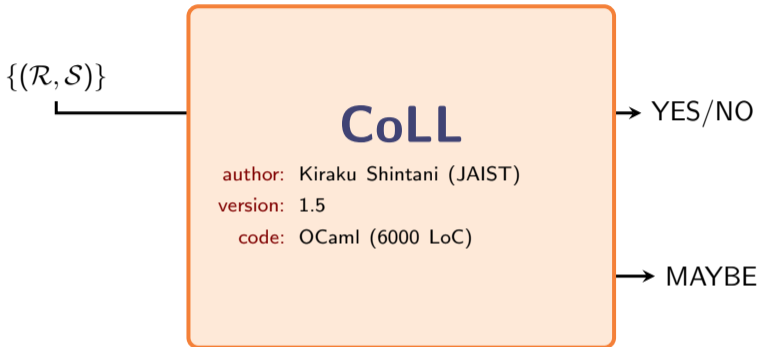


# 2019 results

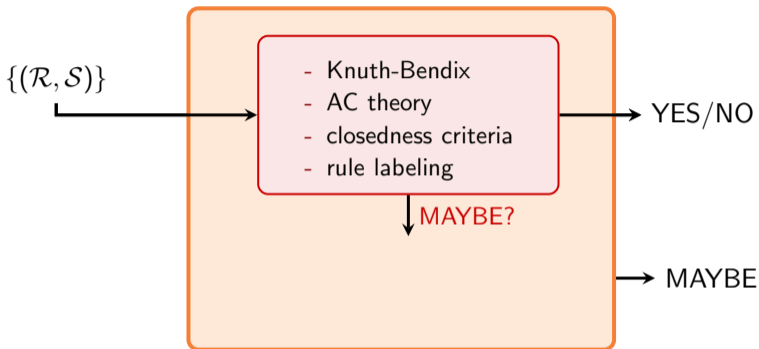


- first ever win for FORT in any category
- incorrect answers by ACP and CoLL

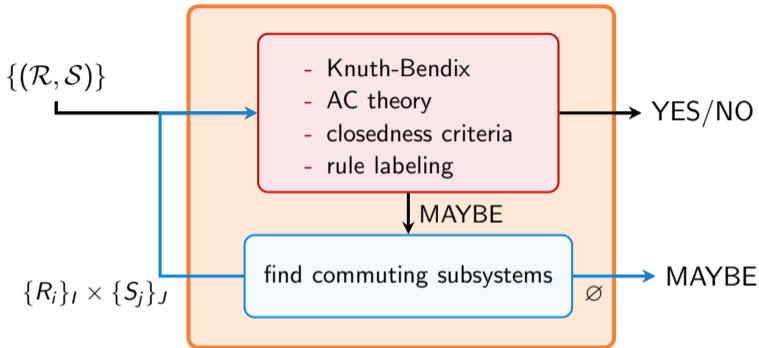
# CoLL: Commutation Tool for Left-Linear TRSs



# CoLL: Commutation Tool for Left-Linear TRSs

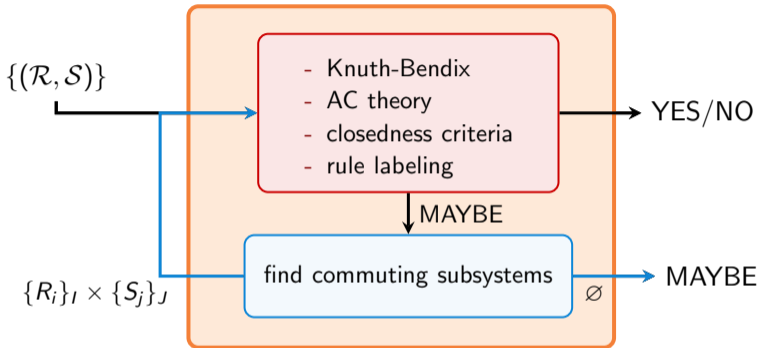


# CoLL: Commutation Tool for Left-Linear TRSs



- 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)

and also improved AC-related feature



CoCo 2020 Participant: **FORT-h** 0.9

Fabian Mitterwallner

Aart Middeldorp

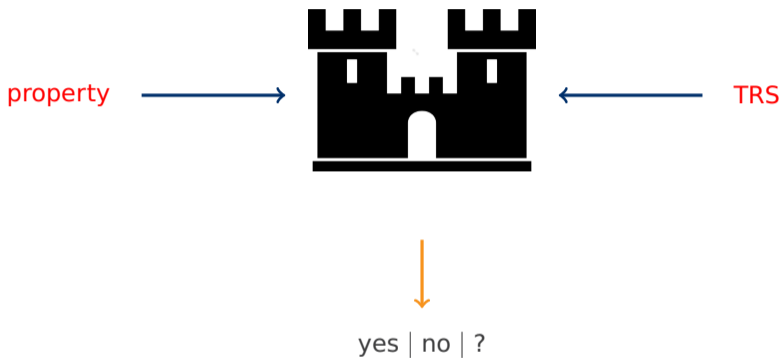
Bertram Felgenhauer

# FORT-h





# FORT-h



# FORT-h



property



linear variable-separated



TRS

$$\forall s \exists t (s \rightarrow^* t \wedge \neg \exists u (t \rightarrow u)) \\ \Rightarrow \exists v (s \twoheadrightarrow v \vee v \xrightarrow{\epsilon} t)$$



yes | no | ?

property is arbitrary formula in first-order theory of rewriting

## CoCo 2019 Categories (FORT)

GCR

NFP

UNC

UNR

COM

## CoCo 2019 Categories (FORT)

GCR

NFP

UNC

UNR

COM

1st place

## CoCo 2020 Categories (FORT-h)

GCR

NFP

UNC

UNR

COM

## CoCo 2020 Categories (FORT-h)

GCR

NFP

UNC

UNR

COM

## Differences FORT and FORT-h

- modified decision procedure

## CoCo 2020 Categories (FORT-h)

GCR

NFP

UNC

UNR

COM

## Differences FORT and FORT-h

- modified decision procedure
- supports linear variable-separated TRSs

## CoCo 2020 Categories (FORT-h)

GCR    NFP    UNC    UNR    COM

## Differences FORT and FORT-h

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



## CoCo 2020 Categories (FORT-h)

GCR    NFP    UNC    UNR    COM

## Differences FORT and FORT-h

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

## CoCo 2020 Categories (FORT-h)

GCR    NFP    UNC    UNR    COM

## Differences FORT and FORT-h

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

<http://cl-informatik.uibk.ac.at/software/FORT/>

# 2019 results



- most contested category with 6 participating tools
- incorrect answers by ConCon
- infChecker is only tool with NO answers



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSIDAD  
POLITÈCNICA  
DE MADRID

## infChecker at CoCo 2020

---

Raúl Gutiérrez<sup>1</sup> Salvador Lucas<sup>2</sup>

PARIS, JUNE 30, 2020

<sup>1</sup>Universidad Politécnica de Madrid  
Spain

<sup>2</sup>Valencian Research Institute for Artificial Intelligence  
Universitat Politècnica de València  
Spain

- 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*  $\text{Th}_{\bowtie_{ij}}$ .
- $\bowtie_{ij}$  can be one of the following predicates:
  - One (CS-)rewriting step ( $\rightarrow$ ,  $\rightarrow$ ).
  - Zero or more (CS-)rewriting steps ( $\rightarrow^*$ ,  $\rightarrow^*$ ).
  - One or more (CS-)rewriting steps ( $\rightarrow^+$ ,  $\rightarrow^+$ ).
  - Subterm ( $\mid \triangleright =$ ) and strict subterm ( $\mid \triangleright$ ).
  - (CS-)Joinability ( $\rightarrow^* \leftarrow$ ,  $\rightarrow^* \leftarrow$ ).
  - One (CS-)convertibility step ( $\leftarrow \rightarrow$ ,  $\leftarrow / \rightarrow$ ).
  - Zero or more (CS-)convertibility steps ( $\leftarrow \rightarrow^*$ ,  $\leftarrow / \rightarrow^*$ ).

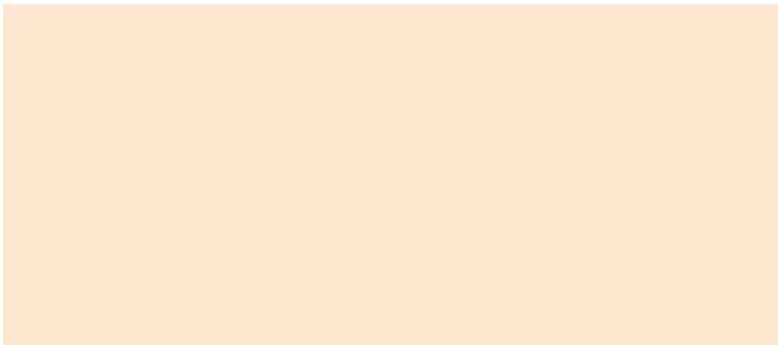
## Implementation

- 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:
  - $P^{\text{Sat}}$  integrates a satisfiability approach to **prove infeasibility using model generators** as AGES and Mace4 to find a proof.
  - $P^{\text{UR}}$  **simplifies** problems by removing non-usable rules.
  - $P^{\text{Prov}}$  integrates a logic-based approach to program analysis to **prove feasibility by theorem proving**. In infChecker, we use the theorem prover Prover9.
  - $P^{\text{NC}}$  adapt the processor that **narrow conditions** in the 2D DP framework for proving operational termination of CTRs to be used with feasibility sequences.

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

# Theorem Prover **Moca 0.2** (Oi and Hirokawa, JAIST)

**Infeasibility Problem:**  $\bigvee_i (s_i \sigma \not\stackrel{*}{\rightarrow}_{\mathcal{R}} t_i \sigma)$  for every  $\sigma$ ?





# Theorem Prover **Moca 0.2** (Oi and Hirokawa, JAIST)

**Infeasibility Problem:**  $\bigvee_i (s_i \sigma \not\stackrel{*}{\vdash}_{\mathcal{R}} t_i \sigma)$  for every  $\sigma$ ?

transformation

**Horn Satisfiability Problem:**  $\mathcal{M} \models \Phi$  for some  $\mathcal{M}$ ?

# Theorem Prover **Moca 0.2** (Oi and Hirokawa, JAIST)

**Infeasibility Problem:**  $\bigvee_i (s_i \sigma \not\stackrel{*}{\rightarrow}_{\mathcal{R}} t_i \sigma)$  for every  $\sigma$ ?

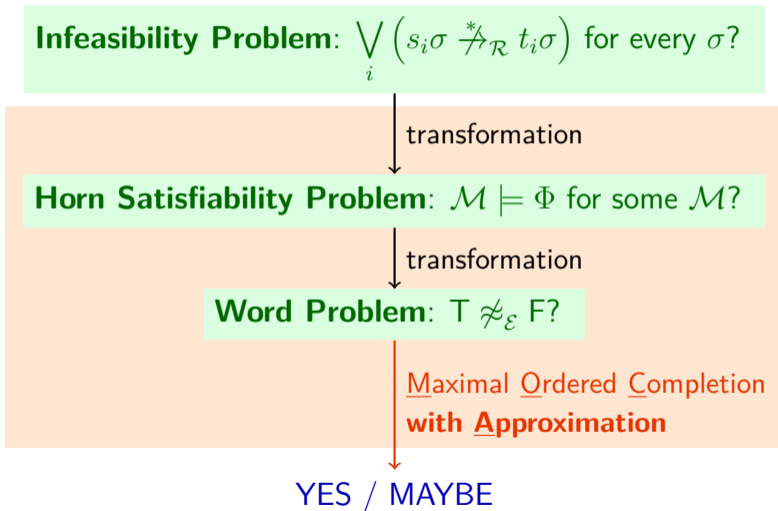
transformation

**Horn Satisfiability Problem:**  $\mathcal{M} \models \Phi$  for some  $\mathcal{M}$ ?

transformation

**Word Problem:**  $T \not\stackrel{\varepsilon}{=} F$ ?

# Theorem Prover **Moca 0.2** (Oi and Hirokawa, JAIST)



# Transformations to Word Problem

## Infeasibility Problem

### ■ CTRS:

$$x - 0 \rightarrow x$$

$$0 - x \rightarrow 0$$

$$s(x) - s(y) \rightarrow x - y$$

### ■ Condition:

$$x - x \xrightarrow{*} \mathcal{R} s(x)$$

# Transformations to Word Problem

## Infeasibility Problem

### ■ CTRS:

$$x - 0 \rightarrow x$$

$$0 - x \rightarrow 0$$

$$s(x) - s(y) \rightarrow x - y$$

### ■ Condition:

$$x - x \xrightarrow{*} \mathcal{R} s(x)$$



## Horn Satisfiability Problem

### ■ Formulas:

$$\forall x. \quad x - 0 = x$$

$$\forall x. \quad 0 - x = 0$$

$$\forall x, y. \quad s(x) - s(y) = x - y$$

$$\forall x. \quad x - x \neq s(x)$$

# Transformations to Word Problem

## Infeasibility Problem

### ■ CTRS:

$$x - 0 \rightarrow x$$

$$0 - x \rightarrow 0$$

$$s(x) - s(y) \rightarrow x - y$$

### ■ Condition:

$$x - x \xrightarrow{*} \mathcal{R} s(x)$$



## Horn Satisfiability Problem

### ■ Formulas:

$$\forall x. \quad x - 0 = x$$

$$\forall x. \quad 0 - x = 0$$

$$\forall x, y. \quad s(x) - s(y) = x - y$$

$$\forall x. \quad x - x \neq s(x)$$



## Word Problem

### ■ ES:

$$x - 0 = x$$

$$0 - x = 0$$

$$s(x) - s(y) = x - y$$

$$f(x - x, x) = T$$

$$f(s(x), x) = F$$

### ■ Goal:

$$T = F$$

# Transformations to Word Problem

## Infeasibility Problem

### ■ CTRS:

$$x - 0 \rightarrow x$$

$$0 - x \rightarrow 0$$

$$s(x) - s(y) \rightarrow x - y$$

### ■ Condition:

$$x - x \xrightarrow{*} \mathcal{R} s(x)$$

## Horn Satisfiability Problem

### ■ Formulas:

$$\forall x. \quad x - 0 = x$$

$$\forall x. \quad 0 - x = 0$$

$$\forall x, y. \quad s(x) - s(y) = x - y$$

$$\forall x. \quad x - x \neq s(x)$$

## Word Problem

### ■ ES:

$$x - 0 = x$$

$$0 - x = 0$$

$$s(x) - s(y) = x - y$$

$$f(x - x, x) = T$$

$$f(s(x), x) = F$$

### ■ Goal:

$$T = F$$

Infeasible



Satisfiable



$T \neq_{\varepsilon} F$

## Approximation for Showing $T \not\approx_{\mathcal{E}} F$

### Fact

if  $\approx_{\mathcal{E}} \subseteq \approx_{\mathcal{E}'}$  ( $\mathcal{E}$  is approximated as  $\mathcal{E}'$ ) then  $T \not\approx_{\mathcal{E}'} F$  implies  $T \not\approx_{\mathcal{E}} F$



## Approximation for Showing $T \not\approx_{\mathcal{E}} F$

### Fact

if  $\approx_{\mathcal{E}} \subseteq \approx_{\mathcal{E}'}$  ( $\mathcal{E}$  is approximated as  $\mathcal{E}'$ ) then  $T \not\approx_{\mathcal{E}'} F$  implies  $T \not\approx_{\mathcal{E}} F$

### Original System $\mathcal{E}$

$$x - 0 = x$$

$$0 - x = 0$$

$$s(x) - s(y) = x - y$$

$$f(x - x, x) = T$$

$$f(s(x), x) = F$$

admits **no** complete TRS

## Approximation for Showing $T \not\approx_{\mathcal{E}} F$

### Fact

if  $\approx_{\mathcal{E}} \subseteq \approx_{\mathcal{E}'}$  ( $\mathcal{E}$  is approximated as  $\mathcal{E}'$ ) then  $T \not\approx_{\mathcal{E}'} F$  implies  $T \not\approx_{\mathcal{E}} F$

### Original System $\mathcal{E}$

$$x - 0 = x$$

$$0 - x = 0$$

$$s(x) - s(y) = x - y$$

$$f(x - x, x) = T$$

$$f(s(x), x) = F$$

admits **no** complete TRS

### Approximation $\mathcal{E}'$

$$x - 0 = x$$

$$0 - x = 0$$

$$s(x) - s(y) = x - y$$

$$f(x - x, y) = T$$

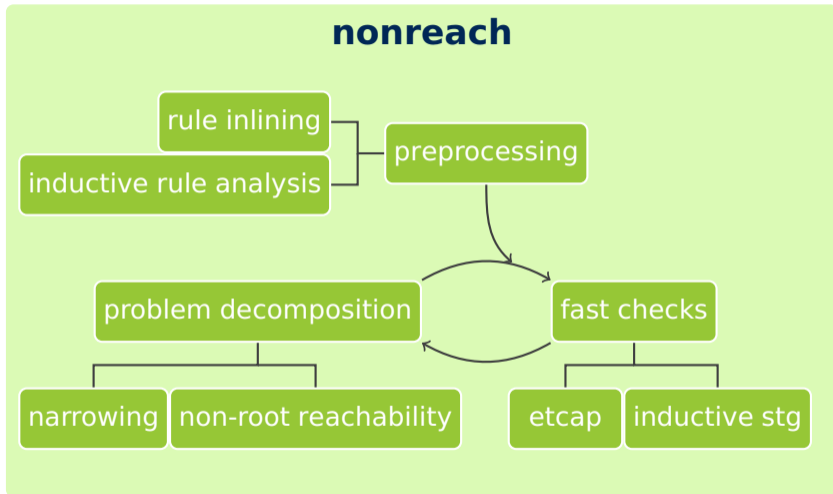
$$f(s(x), x) = F$$

admits **complete TRS!**

## New Features

Moca version 0.2 supports:

- **generalized split-if encoding** (Oi 2019)
- **inlining for conditional rewrite rules** (Sternagel & Sternagel 2017)



- open source
- <https://bitbucket.org/fmessner/nonreach/>

## Features

- implements fast infeasibility checks based on tcap and the inductive symbol transition graph
- combined with decomposition into easier subproblems
- easy interface
- Yes and No results
- can output certificates

## CoCo 2020 Category

- INF

# 2019 results



- noko-leipzig produces most NO answers

and the 2020 winners are ...



<http://cocograph.uibk.ac.at/2020.html>