
2022

# 11th Confluence Competition

Raúl Gutiérrez  Aart Middeldorp  Kiraku Shintani  Naoki Nishida

http://project-coco.uibk.ac.at/2022

# Outline

# Acknowledgements


- CoCo 2022 tool authors

# Acknowledgements


- CoCo 2022 tool authors
- Fabian Mitterwallner and Jonas Schöpf

# Acknowledgements


- CoCo 2022 tool authors
- Fabian Mitterwallner and Jonas Schöpf
- Nao Hirokawa

# Acknowledgements


- CoCo 2022 tool authors
- Fabian Mitterwallner and Jonas Schöpf
- Nao Hirokawa
- IWC 2022 chairs

# Acknowledgements


- CoCo 2022 tool authors
- Fabian Mitterwallner and Jonas Schöpf
- Nao Hirokawa
- IWC 2022 chairs
- CoCo 2022 panel
  - Jörg Endrullis
  - Mizuhito Ogawa
  - Sarah Winkler

# Acknowledgements


- CoCo 2022 tool authors
- Fabian Mitterwallner and Jonas Schöpf
- Nao Hirokawa
- IWC 2022 chairs
- CoCo 2022 panel
  - Jörg Endrullis
  - Mizuhito Ogawa
  - Sarah Winkler
- Aaron Stump

# Outline

| 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 | | | | ✓ | | |
| CSIˆho | | | | ✓ | | |
| AGCP | | | | | ✓ | |
| NoCo | | | | | | ✓ |

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

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

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

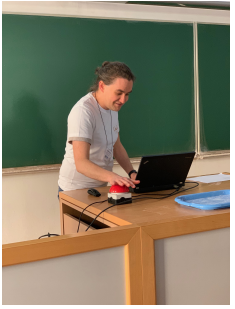
| 2019 | TRS | CPF-TRS | CTRS | HRS | GCR | NFP | UNR | CPF-CTRS | UNC | COM | INF | SRS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACP | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ |
| CSI | ✓ | ✓ | | | | ✓ | ✓ | | ✓ | | | ✓ |
| CoLL | | | | | | | | | | ✓ | | |
| CeTA | | ✓ | | | | | | ✓ | | |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

| 2020 | TRS | CPF-TRS | CTRS | HRS | GCR | NFP | UNR | CPF-CTRS | UNC | COM | INF | SRS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACP | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ |
| CSI | ✓ | ✓ | | | | ✓ | ✓ | | ✓ | | | ✓ |
| CoLL | | | | | | | | | | ✓ | | |
| CeTA | | ✓ | | | | | | ✓ | | | | |
| CoLL-Saigawa | ✓ | | | | | | | | | | | ✓ |
| ConCon | | | ✓ | | | | | ✓ | | | ✓ | |
| CO3 | | | ✓ | | | | | | | | ✓ | |
| infChecker | | | | | | | | | | | ✓ | |
| | | | | | | | | | | | | |
| CSIˆho | | | | ✓ | | | | | | | | |
| AGCP | | | | | ✓ | | | | | | | |
| Moca | | | | | | | | | | | ✓ | |
| FORT-h | | | | | ✓ | ✓ | ✓ | | ✓ | ✓ | | |
| SOL | | | | ✓ | | | | | | | | |
| nonreach | | | | | | | | | | | ✓ | |

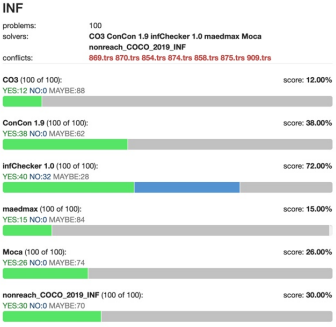
| 2021 | TRS | CPF-TRS | CTRS | | GCR | NFP | UNR | | UNC | COM | INF | SRS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACP | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ |
| CSI | ✓ | ✓ | | | | ✓ | ✓ | | ✓ | | | ✓ |
| CoLL | | | | | | | | | | ✓ | | |
| CeTA | | ✓ | | | | | | | | | | |
| CoLL-Saigawa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

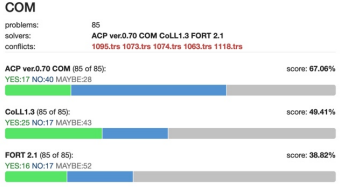
- CoCo is powered by StarExec 

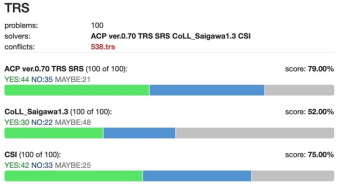
- CoCo is powered by StarExec 
- exciting to watch

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



# Outline

CoCo 2022

CoCo 2022

http://cocograph.uibk.ac.at/2022.html

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

| 2022 | TRS | CPF-TRS | CTRS | | GCR | NFP | UNR | CSR | UNC | COM | INF | SRS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ACP | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ |
| CSI | ✓ | ✓ | | | | ✓ | ✓ | | ✓ | | | ✓ |
| CoLL | | | | | | | | | | ✓ | | |
| CeTA | | ✓ | | | | | | | | | | |
| Hakusan | ✓ | | | | | | | | | | | ✓ |
| Toma | | | | | | | | | | | ✓ | |
| CO3 | | | ✓ | | | | | | | | ✓ | |
| infChecker | | | | | | | | | | | ✓ | |
| FORTify | | ✓ | | | ✓ | ✓ | ✓ | | ✓ | ✓ | | |
| CONFident | ✓ | | ✓ | | | | | ✓ | | | | ✓ |
| AGCP | | | | | ✓ | | | | | | | |
| NaTT | | | | | | | | | | | ✓ | |
| FORT-h | | ✓ | | | ✓ | ✓ | ✓ | | ✓ | ✓ | | |

# Outline

# Categories

| | |
|---|---|
| TRS | confluence of first-order term rewrite systems |
| CPF-TRS | 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 |

# Categories

| | |
|---|---|
| TRS | confluence of first-order term rewrite systems |
| CPF-TRS | 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 |
| CSR | confluence of context-sensitive rewriting |

# Outline

# Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES, NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2021 participate as demonstration tools in 2022

# Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES, NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2021 participate as demonstration tools in 2022
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool

# Secret Problems

4 secret problems submitted to CoCo 2022

# Secret Problems

4 secret problems submitted to CoCo 2022:

- 2 by Christina Kohl (CeTA)

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{h}(x,x) \qquad \mathsf{f}(x) \to \mathsf{h}(x,x) \qquad \mathsf{g}(\mathsf{a}) \to \mathsf{b} \qquad \mathsf{b} \to \mathsf{a}$$
$$\mathsf{h}(x,y) \to \mathsf{h}(\mathsf{g}(x),\mathsf{g}(y)) \qquad\qquad \mathsf{g}(x) \to x \qquad \mathsf{a} \to \mathsf{b}$$

$$\mathsf{f}(x_1,\mathsf{g}(x_2)) \to \mathsf{f}(x_1,\mathsf{g}(x_1)) \qquad \mathsf{f}(\mathsf{g}(y_1),y_2) \to \mathsf{f}(\mathsf{g}(y_1),\mathsf{g}(y_1)) \qquad \mathsf{g}(\mathsf{a}) \to \mathsf{g}(\mathsf{b}) \qquad \mathsf{b} \to \mathsf{a}$$

# Secret Problems

4 secret problems submitted to CoCo 2022:

- 2 by Christina Kohl (CeTA)

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{h}(x,x) \qquad \mathsf{f}(x) \to \mathsf{h}(x,x) \qquad \mathsf{g}(\mathsf{a}) \to \mathsf{b} \qquad \mathsf{b} \to \mathsf{a}$$
$$\mathsf{h}(x,y) \to \mathsf{h}(\mathsf{g}(x),\mathsf{g}(y)) \qquad\qquad\qquad\quad \mathsf{g}(x) \to x \qquad \mathsf{a} \to \mathsf{b}$$

$$\mathsf{f}(x_1,\mathsf{g}(x_2)) \to \mathsf{f}(x_1,\mathsf{g}(x_1)) \qquad \mathsf{f}(\mathsf{g}(y_1),y_2) \to \mathsf{f}(\mathsf{g}(y_1),\mathsf{g}(y_1)) \qquad \mathsf{g}(\mathsf{a}) \to \mathsf{g}(\mathsf{b}) \qquad \mathsf{b} \to \mathsf{a}$$

- 2 by Miguel Vítores and Salvador Lucas (CONFident)

$$\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{c},\mathsf{d}),\mathsf{g}(\mathsf{b},\mathsf{e})),\mathsf{b}) \to \mathsf{a} \qquad \mathsf{g}(\mathsf{a},y) \to \mathsf{g}(\mathsf{b},\mathsf{b}) \qquad \mathsf{g}(x,y) \to \mathsf{g}(y,x)$$

$$\mathsf{f}(\mathsf{h}(\mathsf{f}(\mathsf{f}(\mathsf{a},\mathsf{a}),\mathsf{h}(\mathsf{a}))),\mathsf{g}(\mathsf{f}(x,\mathsf{g}(\mathsf{b})))) \to \mathsf{c} \qquad \mathsf{g}(\mathsf{g}(\mathsf{a})) \to \mathsf{f}(\mathsf{h}(\mathsf{g}(\mathsf{f}(\mathsf{c},\mathsf{c}))),\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{c}),\mathsf{a}),\mathsf{g}(\mathsf{f}(\mathsf{a},\mathsf{a}))))$$
$$\mathsf{c} \to \mathsf{c} \qquad\qquad\qquad\qquad\qquad\quad \mathsf{c} \to \mathsf{f}(\mathsf{a},\mathsf{h}(\mathsf{b})) \qquad \mathsf{f}(x,y) \to \mathsf{f}(y,x)$$

# Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES, NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2021 participate as demonstration tools in 2022
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members
- exclude SRSs

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members
- exclude SRSs, duplicates

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members
- exclude SRSs, duplicates and TRSs whose confluence status is known

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
   returns 27 problems whose status is unknown
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`

- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members
- exclude SRSs, duplicates and TRSs whose confluence status is known

# Problem Selection

Example: COPS queries for TRS category

1. `1652..1655`
   returns 4 secret problems
2. `limit:96,870 1..1651 trs !srs !duplicate {!confluent !non_confluent}`
   returns 27 problems whose status is unknown
3. `limit:69,870 1..1651 trs !srs !duplicate !{!confluent !non_confluent}`
   returns 69 random problems whose status is known

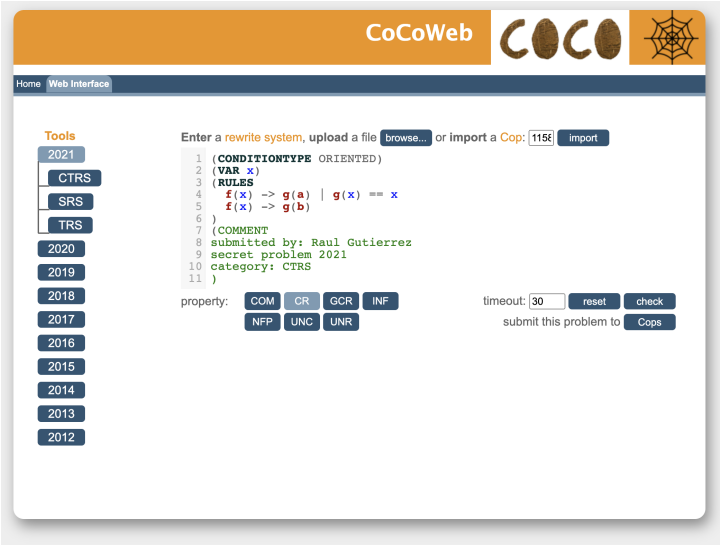
- secret problems have numbers 1652 1653 1654 1655
- seed digits of panel members
- exclude SRSs, duplicates and TRSs whose confluence status is known

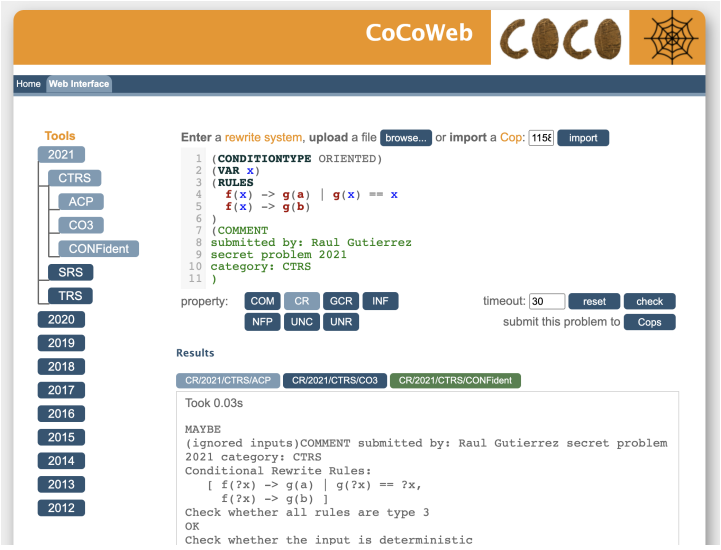
# Competition Rules

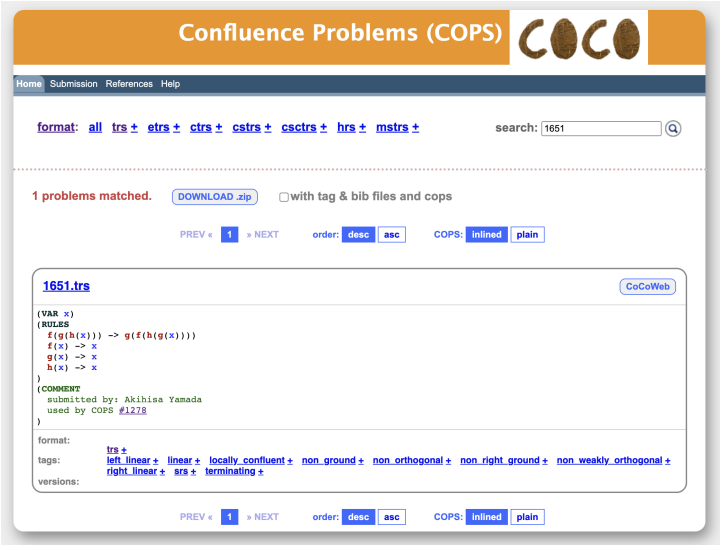
- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES, NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2021 participate as demonstration tools in 2022
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
- 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

# Competition Rules

- Scoring
  - 100 random problems per category, using seed digits provided by panel members
  - tools output YES, NO or MAYBE on first line followed by proof
  - separate rankings for YES and NO and combined YES / NO answers
  - winning tools of 2021 participate as demonstration tools in 2022
- Secret Problems
  - guaranteed to be selected
  - at most two problems per category per tool
- 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


CoCoWeb
COCO
Home
Web Interface
Tools
2021
CTRS
SRS
TRS
2020
2019
2018
2017
2016
2015
2014
2013
2012
Enter a rewrite system, upload a file browse... or import a Cop: 1158 import
1 (CONDITIONTYPE ORIENTED)
2 (VAR x)
3 (RULES
4 f(x) -> g(a) | g(x) == x
5 f(x) -> g(b)
6 )
7 (COMMENT
8 submitted by: Raul Gutierrez
9 secret problem 2021
10 category: CTRS
11 )
property: COM CR GCR INF
NFP UNC UNR
timeout: 30 reset check
submit this problem to Cops


CoCoWeb
COCO
Home
Web Interface
Tools
2021
CTRS
ACP
CO3
CONFident
SRS
TRS
2020
2019
2018
2017
2016
2015
2014
2013
2012
Enter a rewrite system, upload a file browse... or import a Cop: 1158 import
1 (CONDITIONTYPE ORIENTED)
2 (VAR x)
3 (RULES
4 f(x) -> g(a) | g(x) == x
5 f(x) -> g(b)
6 )
7 (COMMENT
8 submitted by: Raul Gutierrez
9 secret problem 2021
10 category: CTRS
11 )
property: COM CR GCR INF
NFP UNC UNR
timeout: 30 reset check
submit this problem to Cops
Results
CR/2021/CTRS/ACP
CR/2021/CTRS/CO3
CR/2021/CTRS/CONFident
Took 0.03s
MAYBE
(ignored inputs)COMMENT submitted by: Raul Gutierrez secret problem
2021 category: CTRS
Conditional Rewrite Rules:
[ f(?x) -> g(a) | g(?x) == ?x,
f(?x) -> g(b) ]
Check whether all rules are type 3
OK
Check whether the input is deterministic




confluence problems database (COPS) consists of 1651 problems (up from 1155)

# CoCo 2022 Live View

http://cocograph.uibk.ac.at/2022.html

# 2021 Results


**TRS** category

# 2021 Results


**TRS** category

- previous winners: ACP (2012 2013 2014 2015 2019) CSI (2015 2016 2017 2018 2020)

# 2021 Results


**TRS** category

- previous winners: ACP (2012 2013 2014 2015 2019) CSI (2015 2016 2017 2018 2020)
- 2022 participants: ACP CONFident CSI Hakusan

# 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
- Our latest effort is to incorporate a decision procedure of UNC 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.

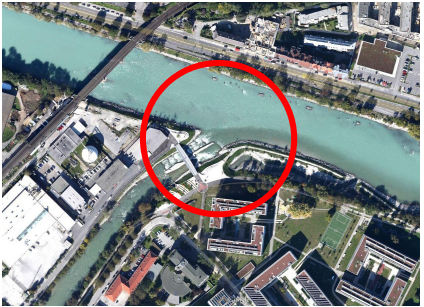
universität innsbruck

# CoCo 2022 Participant: CSI 1.2.6

Fabian Mitterwallner Aart Middeldorp

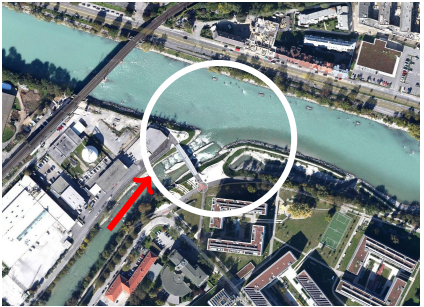
University of Innsbruck

[Google Maps]



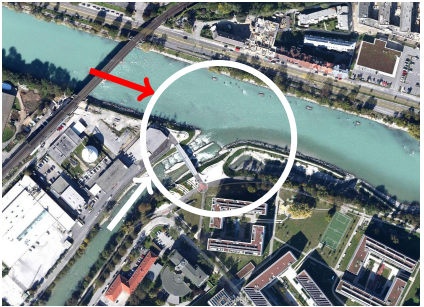
[Google Maps]

C



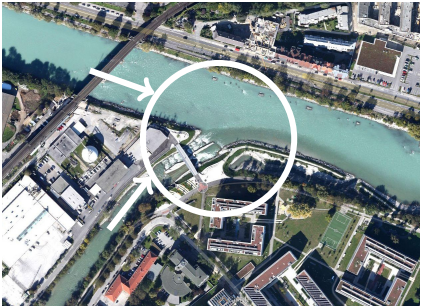
[Google Maps]

CS



[Google Maps]

# CSI



[Google Maps]

# CSI 1.2.6

- 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 2021 Categories

- CPF-TRS (with CeTA)  NFP  TRS  UNC  UNR  SRS

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

## CoCo 2021 Categories

- CPF-TRS (with CeTA) NFP TRS UNC UNR SRS 1st place

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

## CoCo 2022 Categories

- CPF-TRS (with CeTA) NFP TRS UNC UNR SRS

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

## CoCo 2022 Categories

- CPF-TRS (with CeTA) NFP TRS UNC UNR SRS

## New Features

- CSI can produce certificates for left-linear development-closed rewrite systems which are validated by CeTA 2.42 (Kohl and Middeldorp, IWC 2022)

# 2021 Results


**SRS** category

# 2021 Results


- previous winner: CSI (2019 2020)

# 2021 Results


**SRS** category

- previous winner: CSI (2019 2020)
- 2022 participants: ACP CONFident CSI Hakusan

# 2021 Results


- previous winner: CSI (2019 2020)
- 2022 participants: ACP CONFident CSI Hakusan
- noko-leipzig produced most NO answers in 2019




# CONFident **at CoCo 2022**

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

HAIFA, AUGUST 1ST, 2022

[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.
- CONFident can deal with:
  - Term Rewriting Systems.
  - Conditional Term Rewriting Systems (oriented, join and semi-equational).
  - **Context-Sensitive** Term Rewriting Systems. NEW!
  - **Conditional Context-Sensitive** Term Rewriting Systems. NEW!

## Implementation

- The tool is available here:
  `http://zenon.dsic.upv.es/confident/`
- It is written in Haskell and its based on a divide-and-conquer scheme where different techniques are recursively applied using a pre-defined strategy. [GVL22]
- Different techniques are defined based on the logical representation of the systems to check **(non-)joinability** of (conditional) critical pairs [GLV21,LVG22]:
  - Proving **conditional joinability** as a logical consequence using Theorem Provers.
  - Disproving **conditional joinability** as a feasibility problem using infChecker. [GL20]

## Strategy

Our **proof strategy** is:

1. We apply unification and narrowing so simplify conditional rules and pairs. We also use modularity results;
2. We compute the set of extended conditional critical pairs [Luc22];
3. We analyze the system to extract good properties. External tools like MU-TERM can be used to check *operational termination* or *termination* of CTRSs. Prover9 is used to check joinability of conditional pairs. infChecker is used to check infeasibility of rules and conditional pairs;
4. We apply graph-based or feasibility techniques to prove or disprove joinability of the extended conditional critical pairs.

# Bibliography


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

**GLV21** R. Gutiérrez and S. Lucas and M. Vítores. *Confluence Of Conditional Rewriting In Logic Form*. 41th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, LIPIcs 213, pages 44:1-44:18, December 2021.

**LVG22** S. Lucas and M. Vítores and R. Gutiérrez. *Proving and disproving confluence of context-sensitive rewriting*. Journal of Logical and Algebraic Methods in Programming, 126, paper 100749, March 2022.

**GVL22** R. Gutiérrez, M. Vítores and S. Lucas. *Proving Confluence with* CONFident *(System Description)*. LOPSTR22 (to appear, 2022).

**Luc22** S. Lucas. *On local confluence of conditional rewrite systems*. In Proc. of IWC 2022, pages 7-12, 2022.

# Hakusan v0.5 (Shintani & Hirokawa, JAIST)

photo: © 白山市観光連盟

# Hakusan v0.5 (Shintani & Hirokawa, JAIST)

**Definition**

$\mathbf{P}$ is compositional confluence criterion if for all TRSs $\mathcal{R}$ and $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{R}$:

$$\mathbf{P}(\mathcal{R}, \mathcal{C}) \text{ and } \mathsf{CR}(\mathcal{C}) \implies \mathsf{CR}(\mathcal{R})$$

Hakusan repeatedly uses compositional variants of two criteria:

# Hakusan v0.5 (Shintani & Hirokawa, JAIST)

**Definition**

$\mathbf{P}$ is compositional confluence criterion if for all TRSs $\mathcal{R}$ and $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{R}$:

$$\mathbf{P}(\mathcal{R}, \mathcal{C}) \text{ and } \mathrm{CR}(\mathcal{C}) \implies \mathrm{CR}(\mathcal{R})$$

Hakusan repeatedly uses compositional variants of two criteria:

- rule labeling (van Oostrom 2008; Zankl et al. 2015)

# Hakusan v0.5 (Shintani & Hirokawa, JAIST)

**Definition**

$\mathbf{P}$ is compositional confluence criterion if for all TRSs $\mathcal{R}$ and $\mathcal{C}$ with $\mathcal{C} \subseteq \mathcal{R}$:

$$\mathbf{P}(\mathcal{R}, \mathcal{C}) \text{ and } \mathsf{CR}(\mathcal{C}) \implies \mathsf{CR}(\mathcal{R})$$

Hakusan repeatedly uses compositional variants of two criteria:

- rule labeling (van Oostrom 2008; Zankl et al. 2015)
- critical pair systems (next slide) (Hirokawa and Middeldorp 2011)

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

- *all parallel critical pairs are joinable and*

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

- *all parallel critical pairs are joinable and*
- *$\mathcal{P}/\mathcal{R}$ is terminating*

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

- *all parallel critical pairs are joinable and*
- *$\mathcal{P}/\mathcal{R}$ is terminating*

*for some confluent subsystem $\mathcal{C} \subseteq \mathcal{R}$, where*

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

- *all parallel critical pairs are joinable and*
- *$\mathcal{P}/\mathcal{R}$ is terminating*

*for some confluent subsystem $\mathcal{C} \subseteq \mathcal{R}$, where*

$$\mathcal{P} = \{s \to t, s \to u \mid t \mathrel{{}_{\mathcal{R}}\!\leftarrow\!\!\!+} s \xrightarrow{\epsilon}_{\mathcal{R}} u \text{ is parallel critical peak but not } t \leftrightarrow_{\mathcal{C}}^{*} u\}$$

# Compositional Criterion based on Critical Pair System

**Theorem (Shintani and Hirokawa, FSCD 2022)**

*left-linear TRS $\mathcal{R}$ is confluent if*

- *all parallel critical pairs are joinable and*
- *$\mathcal{P}/\mathcal{R}$ is terminating*

*for some confluent subsystem $\mathcal{C} \subseteq \mathcal{R}$, where*

$$\mathcal{P} = \{s \to t, s \to u \mid t \ {}_{\mathcal{R}}\!\Leftarrow\!\!\Vdash s \xrightarrow{\epsilon}_{\mathcal{R}} u \text{ is parallel critical peak but not } t \leftrightarrow^*_{\mathcal{C}} u\}$$

**Note**

- Hakusan only deals with **left-linear** TRSs
- it only supports two methods but it is quite powerful

# 2021 Results


**CPF-TRS** category

# 2021 Results


**CPF-TRS** category

- previous winners: CSI + CeTA (2012 2013 2014 2015 2016 2017 2018 2019 2020)

# 2021 Results


**CPF-TRS** category

- previous winners: CSI + CeTA (2012 2013 2014 2015 2016 2017 2018 2019 2020)
- 2022 participants: ACP + CeTA CSI + CeTA FORT-h + FORTify

universität
innsbruck


# CoCo 2022 Participant: CeTA 2.42

Christina Kohl, René Thiemann, and Aart Middeldorp

# CeTA 2.42


- CeTA: certifier of various properties, verified in Isabelle/HOL
- developed by Computational Logic Group in Innsbruck
- several confluence techniques supported, see a complete list at:

  `http://cl-informatik.uibk.ac.at/software/ceta/`

- new confluence techniques in CeTA 2.42 in comparison to 2021
  - development closed TRSs are confluent (Christina Kohl, Aart Middeldorp)
- further differences
  - improved speed of parser
  - upcoming version: support of decision procedure for right-ground termination (Ismael Kira, René Thiemann)

# 2021 Results


**GCR** category

# 2021 Results


**GCR** category

- previous winner: AGCP (2015 2016 2017 2018 2019 2020)

# 2021 Results


**GCR** category

- previous winner: AGCP (2015 2016 2017 2018 2019 2020)
- 2022 participants: AGCP FORT-h (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.**

# 2021 Results


**NFP** category

# 2021 Results


- previous winner: CSI (2017 2018 2019 2020)

# 2021 Results


- previous winner: CSI (2017 2018 2019 2020)
- 2022 participants: CSI FORT-h (FORT-h +) FORTify

universität innsbruck

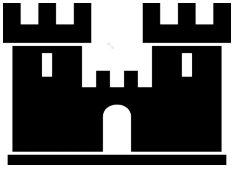
supported by FWF project P30301

# CoCo 2022 Participant: FORT-h 2.0

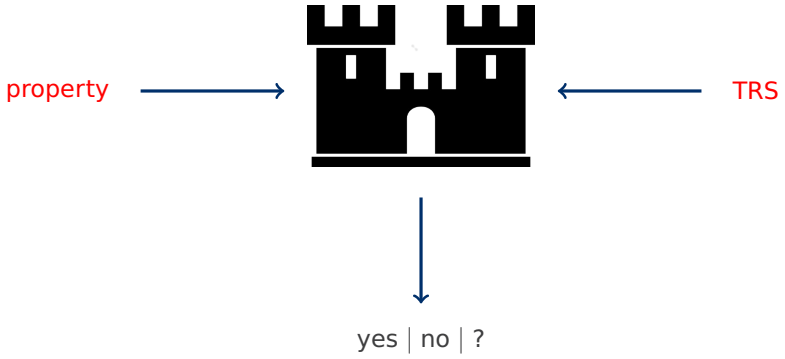
Fabian Mitterwallner Aart Middeldorp

University of Innsbruck

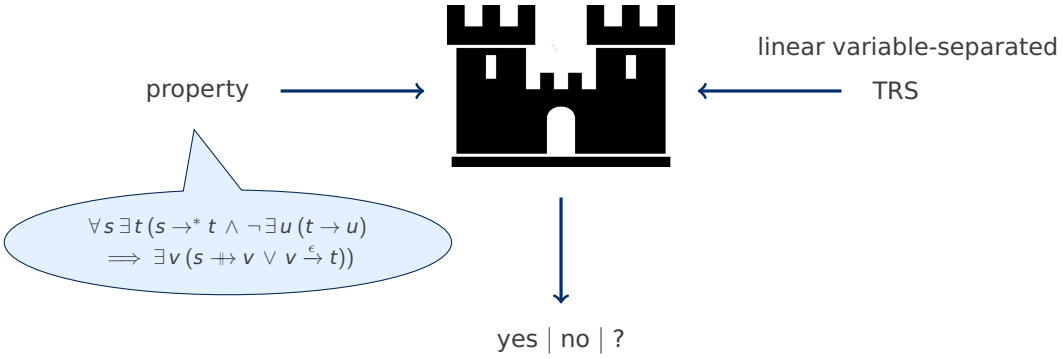
# FORT-h



# FORT-h



# FORT-h




property is arbitrary formula in first-order theory of rewriting

## CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM

## CoCo 2021 Categories (FORT-h)

GCR NFP UNC UNR COM most YES results

## CoCo 2022 Categories (FORT-h)

GCR NFP UNC UNR COM CPF-TRS (with FORTify)

## CoCo 2022 Categories (FORT-h)

GCR NFP UNC UNR COM CPF-TRS (with FORTify)

## Differences 2021 and 2022

- optimized signature extension results (IWC 2022)

## CoCo 2022 Categories (FORT-h)

GCR NFP UNC UNR COM CPF-TRS (with FORTify)

## Differences 2021 and 2022

- optimized signature extension results (IWC 2022)
- more expressive input language

## CoCo 2022 Categories (FORT-h)

GCR NFP UNC UNR COM CPF-TRS (with FORTify)

## Differences 2021 and 2022

- optimized signature extension results (IWC 2022)
- more expressive input language

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

# 2021 Results


**UNR** category

- FORT-h and (FORT+)FORTIFY had most YES answers

# 2021 Results


**UNR** category

- FORT-h and (FORT+)FORTIFY had most YES answers
- previous winner: CSI (2017 2018 2019 2020)

# 2021 Results


**UNR** category

- FORT-h and (FORT+)FORTIFY had most YES answers
- previous winner: CSI (2017 2018 2019 2020)
- 2022 participants: CSI FORT-h (FORT-h+)FORTify

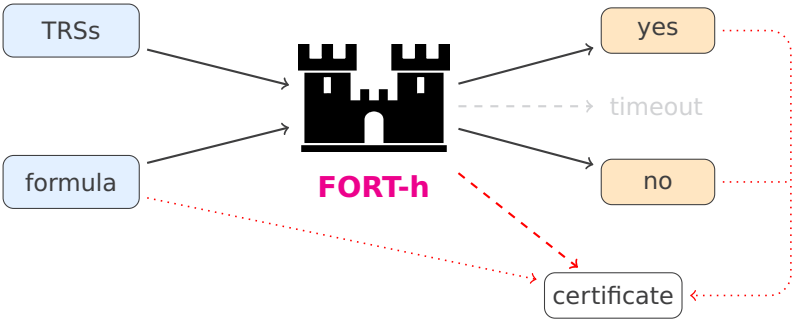
universität innsbruck

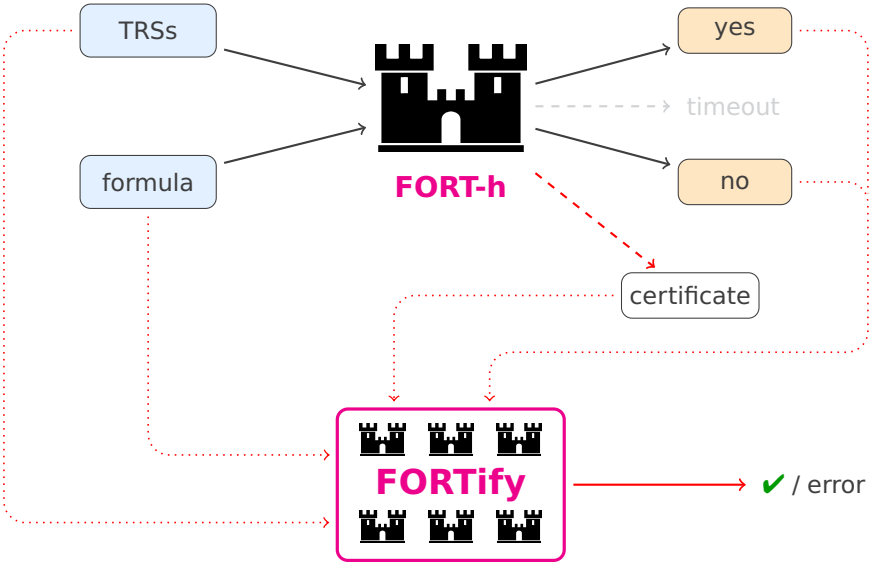
supported by FWF project P30301


# CoCo 2022 Participant: 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

- code generated from formalization of decision procedure in Isabelle/HOL
- `https://fortissimo.uibk.ac.at/fort(ify)/`

- code generated from formalization of decision procedure in Isabelle/HOL
- `https://fortissimo.uibk.ac.at/fort(ify)/`

**CoCo 2021 Categories (FORT-h + FORTify)**

| COM | TRS | GCR | UNC | UNR |
|---|---|---|---|---|

- code generated from formalization of decision procedure in Isabelle/HOL
- `https://fortissimo.uibk.ac.at/fort(ify)/`

**CoCo 2022 Categories (FORT-h + FORTify)**

COM CPF-TRS GCR UNC UNR NFP

- code generated from formalization of decision procedure in Isabelle/HOL
- `https://fortissimo.uibk.ac.at/fort(ify)/`

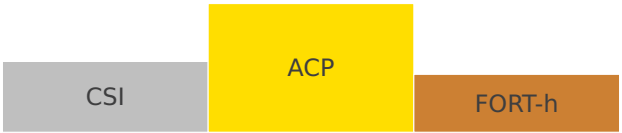
## CoCo 2022 Categories (FORT-h + FORTify)

COM CPF-TRS GCR UNC UNR NFP

## New Features

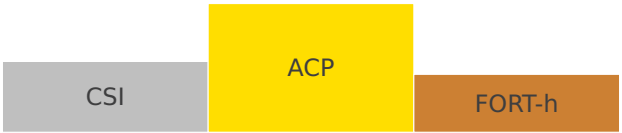
- improved efficiency
- certificates for NFP can be validated
- more signature extension results (IWC 2022)

# 2021 Results




**UNC** category

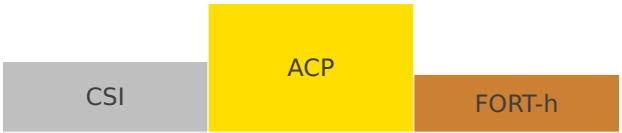
# 2021 Results




**UNC** category

- previous winners: CSI (2017) ACP (2018 2019 2020)

# 2021 Results




**UNC** category

- previous winners: CSI (2017) ACP (2018 2019 2020)
- 2022 participants: ACP CSI FORT-h (FORT-h+)FORTify

# 2021 Results


**CTRS** category

# 2021 Results


**CTRS** category

- previous winners: CO3 (2014) ConCon (2015 2016 2017 2018 2020) ACP (2019)

# 2021 Results


**CTRS** category

- previous winners: CO3 (2014) ConCon (2015 2016 2017 2018 2020) ACP (2019)
- 2022 participants: ACP CO3 CONFident

# CO3 (Ver. 2.3)

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

Naoki Nishida Misaki Kojima
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],
- narrowing trees [Nishida & Maeda, 18], and
- reduction of confluence of join or semi-equational CTRSs to that of oriented ones

## Infeasibility and Confluence Criterion

- Condition $c$ is **infeasible** w.r.t. DCTRS $\mathcal{R}$ if $\mathbb{U}_{conf}(\mathcal{R})$ is right-linear and a narrowing tree for $c$ defines $\emptyset$ [Maeda et al, 19]
- Syntactically deterministic 3-CTRSs $\mathcal{R}$ is **confluent** if either
  - $\mathcal{R}$ 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]

- Remove valid conditions, improve removal of infeasible conditions, etc [Ver. 2.3]

# 2021 Results


**INF** category

# 2021 Results


**INF** category

- previous winner: infChecker (2019 2020)

# 2021 Results


**INF** category

- previous winner: infChecker (2019 2020)
- 2022 participants: CO3 infChecker NaTT Toma

# infChecker **at CoCo 2022**

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

HAIFA, AUGUST 1TH, 2022

[1]Universidad Politécnica de Madrid
Spain
[2]DSIC & VRAIN
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 $\mathsf{P}^{\mathsf{UR}}$ whenever it is sound and complete;
  2. we try to prove feasibility using $\mathsf{P}^{\mathsf{Prov}}$;
  3. if $\mathsf{P}^{\mathsf{Prov}}$ fails, we apply $\mathsf{P}^{\mathsf{Sat}}$;
  4. if $\mathsf{P}^{\mathsf{Sat}}$ fails, we apply $\mathsf{P}^{\mathsf{NC}}$;
  5. if $\mathsf{P}^{\mathsf{NC}}$ 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 2022

Akihisa Yamada @ AIST

# NaTT in CoCo

- NaTT a **termination prover** for **plain** term rewriting
- In CoCo 2021:
  - participated **INF** category with an easy checker [Sternagel & Yamada '19]
  - ignores conditions
  - weak but fast
- In CoCo 2022:
  - Generalized reduction pairs for INF [IJCAR '22]
  - don't ignore conditions
  - not too weak, but slow

# Orderings for non-reachability

**reduction pair** [*ver.* Y, CADE '21]:

- $\succcurlyeq$ and $\succ$ are closed under substitutions
- $\succcurlyeq$ is closed under contexts
- $\succcurlyeq$ is a quasi-order
- $\succ$ is well-founded (order)
- $\succcurlyeq \cdot \succ \subseteq \succ$ **and** $\succ \cdot \succcurlyeq \subseteq \succ$
- $\succ \subseteq \succcurlyeq$

# Orderings for non-reachability

**rewrite pair** [Y, IJCAR '22]:

- $\succcurlyeq$ and $\succ$ are closed under substitutions
- $\succcurlyeq$ is closed under contexts
- $\succcurlyeq$ is a quasi-order
- $\succ$ **is irreflexive** (order)
- $\succcurlyeq \cdot \succ \subseteq \succ$ and $\succ \cdot \succcurlyeq \subseteq \succ$
- $\succ \subseteq \succcurlyeq$

# Orderings for non-reachability

**rewrite pair** [Y, IJCAR '22]:

- $\succcurlyeq$ and $\succ$ are closed under substitutions
- $\succcurlyeq$ is closed under contexts
- $\succcurlyeq$ is a quasi-order
- $\succ$ **is irreflexive** (order)
- $\succcurlyeq \cdot \succ \subseteq \succ$ and $\succ \cdot \succcurlyeq \subseteq \succ$
- $\succ \subseteq \succcurlyeq$

**Theorem**:

$s \twoheadrightarrow t$ is $\mathcal{R}$-unsat **if** there's a rewrite pair $\langle \succcurlyeq, \succ \rangle$ s.t. $\mathcal{R} \subseteq \succcurlyeq$ and $s \prec t$

# Orderings for non-reachability

**co-rewrite pair** [Y, IJCAR '22]:

- $\succcurlyeq$ and $\prec$ are closed under substitutions
- $\succcurlyeq$ is closed under contexts
- $\succcurlyeq$ is a quasi-order
- ($\prec$ is irreflexive)
- $\succcurlyeq \cap \prec = \emptyset$

**Theorem**:

$s \twoheadrightarrow t$ is $\mathcal{R}$-unsat **iff** there's a **co-**rewrite pair $\langle \succcurlyeq, \prec \rangle$ s.t. $\mathcal{R} \subseteq \succcurlyeq$ and $s \prec t$

**Corollary**:

$s \twoheadrightarrow t$ is $\mathcal{R}$-unsat **iff** $\mathcal{R}$ has a quasi-model $\langle \mathcal{A}, \geq \rangle$ s.t. $\mathcal{A} \vDash s \not\geq t$

# Co-WPO

$s \sqsupseteq_{\mathrm{WPO}} t$
- **1.** $\mathcal{A} \vDash s > t$ or
- **2.** $\mathcal{A} \vDash s \geq t$ and
  - **a.** $\exists i \in \pi(f).\, s_i \sqsupseteq_{\mathrm{WPO}} t$; or
  - **b.** $\forall j \in \pi(g).\, s \sqsupset_{\mathrm{WPO}} t_j$ and
    - **i.** $f \succ g$ or
    - **ii.** $f \succcurlyeq g$ and $\pi_f(s_1, \ldots, s_n) \sqsupseteq^{\mathrm{lex}}_{\mathrm{WPO}} \pi_g(t_1, \ldots, t_m)$;
  - **c.** $s = t \in \mathcal{V}$

$t \sqsupset_{\overline{\mathrm{WPO}}} s$
- **1.** $\mathcal{A} \vDash t \not\trianglelefteq s$ or
- **2.** $\mathcal{A} \vDash t \prec s$ and
  - **a.** $\exists j \in \pi(g).\, t_j \sqsupseteq_{\overline{\mathrm{WPO}}} s$; or
  - **b.** $\forall i \in \pi(f).\, t \sqsupset_{\overline{\mathrm{WPO}}} s_i$ and
    - **i.** $g \not\preccurlyeq f$ or
    - **ii.** $g \prec f$ and $\pi_g(t_1, \ldots, t_m) \sqsupset^{\mathrm{lex}}_{\overline{\mathrm{WPO}}} \pi_f(s_1, \ldots, s_n)$;

**Theorem** [Y, IJCAR '22]:
$\langle \sqsupseteq_{\mathrm{WPO}}, \sqsubset_{\overline{\mathrm{WPO}}} \rangle$ is a co-rewrite pair (under mild condition).

# Toma


## Teppei Saito and Nao Hirokawa

## JAIST


August 1, 2022

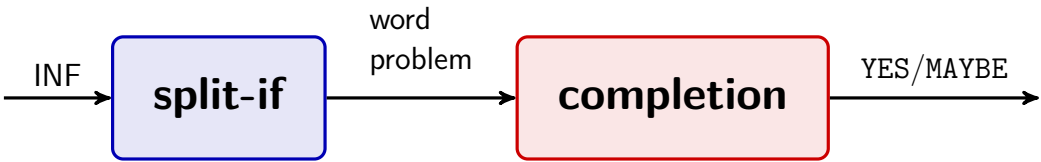
# Toma: Theorem prover based on ordered maximal completion

Equational theorem prover Toma

https://www.jaist.ac.jp/project/maxcomp/

# Architecture

Toma solves CoCo INF problem as follows:




1. INF problem is transformed into word problem by *split-if* encoding (Claessen and Smallbone, 2018).
2. Word problem is solved by new variant of maximal ordered completion:
   - maximal ordered completion (Winkler and Moser, 2018) plus
   - maximal completion with inter-reduction (Hirokawa, 2021)

# Example

Toma outputs ground-complete ordered rewrite system as witness:

```
$ toma --inf 831.trs
YES
...
f4(x201,x112) -> false__
f3(pin(x12)) -> false__
pout(b) -> pin(a)
pout(c) -> pin(b)
...
Since true__ and false__ are not joinable,
the original problem is infeasible.
```

# 2021 Results


**COM** category

- ACP had most NO answers

# 2021 Results


**COM** category

- ACP had most NO answers
- previous winners: FORT (2019) ACP (2020)

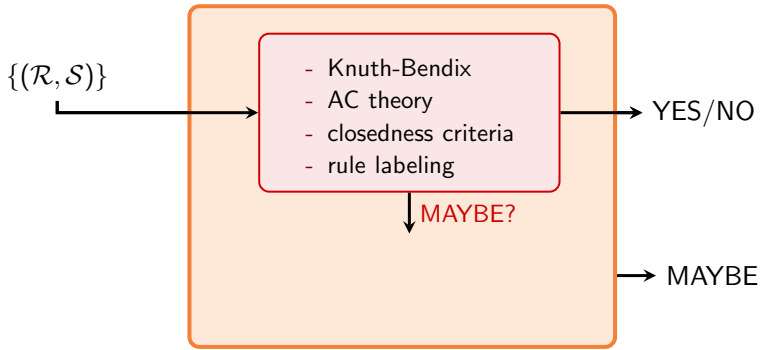
# 2021 Results


**COM** category

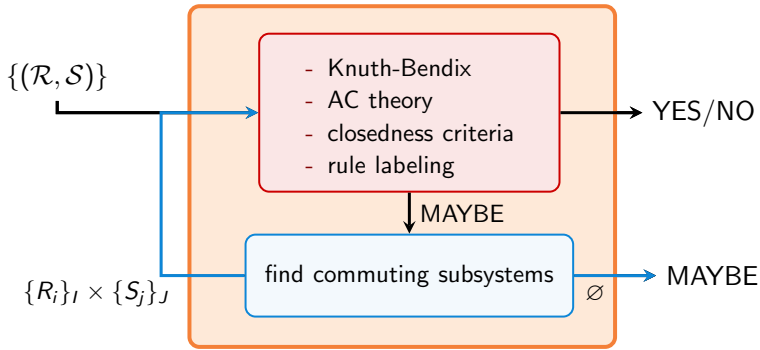
- ACP had most NO answers
- previous winners: FORT (2019) ACP (2020)
- 2022 participants: ACP CoLL FORT-h (FORT-h+)FORTify

# CoLL: Commutation Tool for Left-Linear TRSs

# CoLL: Commutation Tool for Left-Linear TRSs

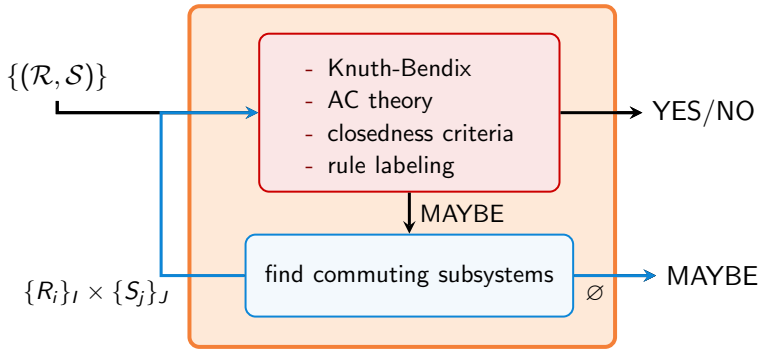


# CoLL: Commutation Tool for Left-Linear TRSs




- 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?

- fixed minor bug for application of asymmetric criterion
  - $\times$ $\mathcal{R}$ and $\mathcal{S}$ commute if $P(\mathcal{R}, \mathcal{S})$ holds
  - ✓ $\mathcal{R}$ and $\mathcal{S}$ commute if $P(\mathcal{R}, \mathcal{S})$ or $P(\mathcal{S}, \mathcal{R})$ hold

# and the 2022 winners are . . .

http://cocograph.uibk.ac.at/2022.html

# Outline

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

- combination of TRS and SRS categories

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

- combination of TRS and SRS categories
- certification: most certified answers

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

- combination of TRS and SRS categories
- certification: most certified answers
- INF category

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

- combination of TRS and SRS categories
- certification: most certified answers
- INF category
- COM category

# Olympic Games

FLoC Olympic Games: August 3 16:00 – 17:30 Churchill Auditorium

- combination of TRS and SRS categories
- certification: most certified answers
- INF category
- COM category
- CTRS category

# Outline

# Outlook

- CoCo 2023 during IWC 2023 in Obergurgl

# Outlook

- CoCo 2023 during IWC 2023 in Obergurgl
- no separate CPF categories: tool + certifier runs in categories in which tool participates

# Outlook

- CoCo 2023 during IWC 2023 in Obergurgl
- no separate CPF categories: tool + certifier runs in categories in which tool participates
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS

# Outlook

- CoCo 2023 during IWC 2023 in Obergurgl
- no separate CPF categories: tool + certifier runs in categories in which tool participates
- revive previous categories?
  - HRS  
  - CPF-CTRS  
  - NRS  

# Outlook

- CoCo 2023 during IWC 2023 in Obergurgl
- no separate CPF categories: tool + certifier runs in categories in which tool participates
- revive previous categories?
  - HRS
  - CPF-CTRS
  - NRS