Confluence Tools

Nao Hirokawa

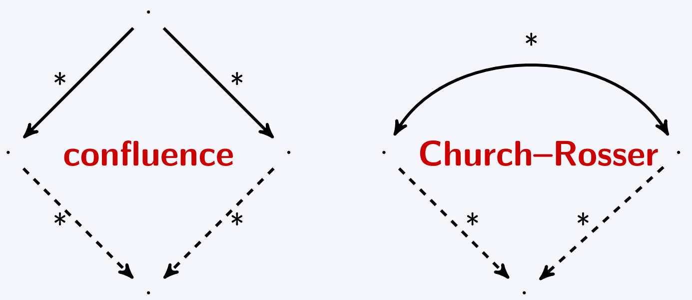
JAISTs

14th ISR, August 28, 2024

`https://www.jaist.ac.jp/~hirokawa/24isr/`

Confluence

Definition



Fact

confluence and Church–Rosser property are equivalent

ARI Format

```
(format TRS)
(fun 0 0)
(fun s 1)
(fun + 2)
(rule (+ 0 x) x)
(rule (+ (s x) y) (s (+ x y)))
(rule (+ x y) (+ y x))
```


Contents: Proving (Non-)Confluence Automatically

full run of TRS category in CoCo 2024

	tool/method	YES	NO
	CSI	272	205
	ACP	257	168
	Hakusan	139	74
	CONFident	108	142
	FORT-h	34	89
1	**rule labeling**	133	—
2	**persistency decomposition (RL+KB)**	$\geqslant$ 150	—
3	**generating convertible pairs**	—	—
4	**non-joinability test**	—	207

Rule Labeling

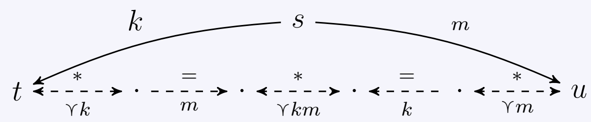
Rule Labeling

given rule labeling function $\phi : \mathcal{R} \to \mathbb{N}$

- $s \to_k t$ if $s \to_\alpha t$ and $\phi(\ell \to r) \leqslant k$ for some $\ell \to r \in \mathcal{R}$
- $s \to_{\curlyvee km} t$ if $s \to_i t$ for some $i < \max\{k, m\}$

Theorem (van Oostrom 2008)

linear TRS is confluent if every critical peak $t \leftarrow s \xrightarrow{\epsilon} u$ is decreasing wrt ϕ, i.e.:



Proof.

immediate from decreasing diagram techniques (van Oostrom 1995) □

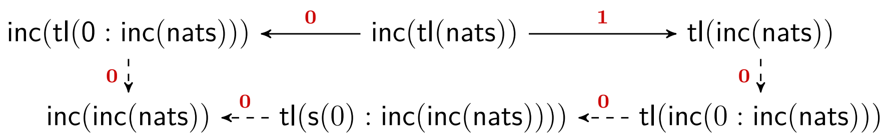
Example for Rule Labeling (COPS #20)

consider linear TRS and labeling for rules

$$\mathsf{hd}(x : y) \xrightarrow{0} x \qquad \mathsf{inc}(x : y) \xrightarrow{0} \mathsf{s}(x) : \mathsf{inc}(y) \qquad \mathsf{inc}(\mathsf{tl}(\mathsf{nats})) \xrightarrow{1} \mathsf{tl}(\mathsf{inc}(\mathsf{nats}))$$

$$\mathsf{tl}(x : y) \xrightarrow{0} y \qquad \mathsf{nats} \xrightarrow{0} 0 : \mathsf{inc}(\mathsf{nats})$$

confluence follows from decreasingness of critical peak:



Note: given join sequences, suitable labeling can be found by SAT/SMT

Parallel Critical Pairs

Definition

$(\ell\sigma)[r_p\sigma]_{p\in P} \twoheadleftarrow\!\!\!\!\leftarrow \ell\sigma \xrightarrow{\epsilon} r\sigma$ is parallel critical peak if

- P is non-empty set of parallel function positions in ℓ
- none of rules $\ell \to r$ and $\ell_p \to r_p$ for $p \in P$ shares variable with other rules
- σ is most general unifier of $\{\ell_p \approx (\ell|_p)\}_{p\in P}$, and
- if $P = \{\epsilon\}$ then $\ell_\epsilon \to r_\epsilon$ is not variant of $\ell \to r$

$\mathrm{PCP}(\mathcal{R}) = \{(t, u) \mid t \twoheadleftarrow\!\!\!\!\leftarrow s \xrightarrow{\epsilon} u$ is parallel critical peak$\}$

Exercise

compute $\mathrm{PCP}(\mathcal{R})$ for $\mathcal{R} = \{\mathsf{f}(\mathsf{g}(x), \mathsf{g}(y)) \to x,\ \mathsf{g}(\mathsf{a}) \to \mathsf{a}\}$

Parallel-Critical-Pairs Closing Systems

Theorem (Shintani and Hirokawa 2024)

$\mathsf{CR}(\mathcal{S}) \implies \mathsf{CR}(\mathcal{R})$ *if $\mathcal{R}$ is left-linear, $\mathcal{S} \subseteq \mathcal{R}$, and* $\mathsf{PCP}(\mathcal{R}) \subseteq \leftrightarrow_{\mathcal{S}}^*$,

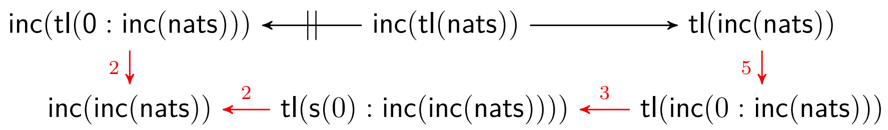
Proof.

$\dashrightarrow\!\!\!\!\!\!+\!\!\!\!\rightarrow_{\mathcal{R}} \cdot \rightarrow_{\mathcal{S}}^*$ has diamond property □

we show confluence of left-linear TRS $\mathcal{R}$:

$$\mathsf{hd}(x : y) \xrightarrow{1} x \qquad \mathsf{inc}(x : y) \xrightarrow{3} \mathsf{s}(x) : \mathsf{inc}(y) \qquad \mathsf{inc}(\mathsf{tl}(\mathsf{nats})) \xrightarrow{5} \mathsf{tl}(\mathsf{inc}(\mathsf{nats}))$$
$$\mathsf{tl}(x : y) \xrightarrow{2} y \qquad \mathsf{nats} \xrightarrow{4} 0 : \mathsf{inc}(\mathsf{nats})$$

1. parallel critical peak is closed by $\mathcal{S} = \{2, 3, 5\}$:



2. $\mathcal{S}$ is confluent as it is orthogonal
3. hence $\mathcal{R}$ is confluent

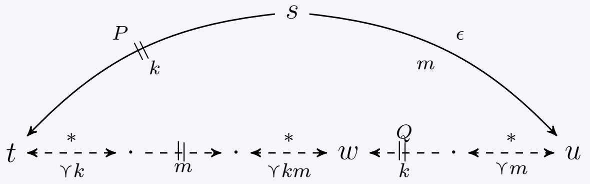
Rule Labeling

Definition

rule labeling for $\mathcal{R}$ is function $\phi : \mathcal{R} \to \mathbb{N}$

Theorem (Zankl et al. 2015)

left-linear TRS $\mathcal{R}$ is confluent if every parallel critical peak is decreasing wrt ϕ



with $\mathcal{V}\mathsf{ar}(w, Q) \subseteq \mathcal{V}\mathsf{ar}(s, P)$

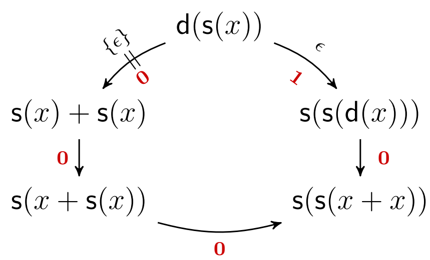
here $\to_k$ is rewrite step of $\{\ell \to r \in \mathcal{R} \mid \phi(\ell \to r) \leqslant k\}$

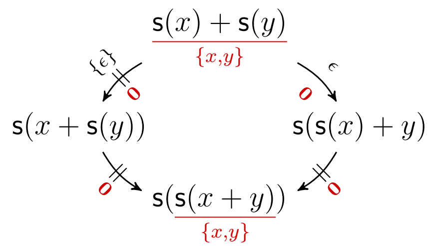
Confluence Proof by Rule Labeling

left-linear TRS

$$\mathsf{d}(x) \xrightarrow{0} x + x \qquad \mathsf{s}(x) + y \xrightarrow{0} \mathsf{s}(x + y) \qquad \infty \xrightarrow{0} \mathsf{s}(\infty)$$
$$\mathsf{d}(\mathsf{s}(x)) \xrightarrow{1} \mathsf{s}(\mathsf{s}(\mathsf{d}(x))) \qquad x + \mathsf{s}(y) \xrightarrow{0} \mathsf{s}(x + y)$$

all parallel critical peaks are decreasing; for instance,



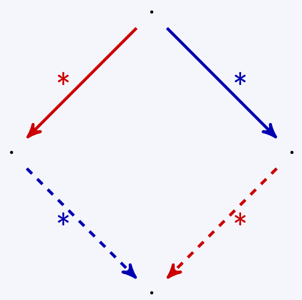


hence TRS is confluent

Commutation

Definition

$\rightarrow$ and $\rightarrow$ commute if



Fact

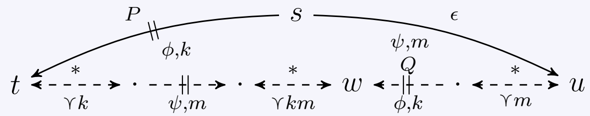
confluence of $\mathcal{R}$ is equivalent to commutation of $\rightarrow_{\mathcal{R}}$ and $\rightarrow_{\mathcal{R}}$

given rule labeling functions $\phi, \psi : \mathcal{R} \rightarrow \mathbb{N}$

- $s \rightarrow_{\phi,k} t$ if $s \rightarrow_{\alpha} t$ and $k \geqslant \phi(\alpha)$ for some $\alpha \in \mathcal{R}$
- $s \leftrightarrow_{\curlyvee km} t$ if $s \rightarrow_{\psi,i} t$ or $s\ {}_{\phi,i}\!\leftarrow t$ for some $i < k$ or $i < m$

Definition (parallel version of rule labeling)

parallel critical peak $t\ {}_{\phi,k}\overset{P}{\leftarrow\!\!\shortmid\!\!\shortmid} s \overset{\epsilon}{\rightarrow}_{\psi,m} u$ is (ψ,ϕ)-decreasing if



with $\mathcal{V}\mathsf{ar}(w,Q) \subseteq \mathcal{V}\mathsf{ar}(s,P)$

Theorem

left-linear TRS is confluent if
every parallel critical peak is (ϕ,ψ)- and (ψ,ϕ)-decreasing

Example for Rule Labeling with Two Labelings

TRS

$$x + y \xrightarrow{1,0} y + x \qquad (x+y)+z \xrightarrow{1,0} x + (y+z)$$

all parallel critical peaks $t\ {}_1\!\leftarrow s \rightarrow_0 u$ satisfies

$$t \rightarrow_0^* u$$

hence TRS is confluent

Remark on Related Criteria

rule labeling based on PCPs subsumes confluence criteria for left-linear TRSs:

1. orthogonality (Rozen 1973)
2. parallel closedness (Huet 1980)
3. almost parallel closedness (Toyama 1988)
4. Gramlich's PCP-based parallel closedness (Gramlich 1996)
5. Toyama's PCP-based parallel closedness (Toyama 1981)
6. repeated application of criterion based on PCP-closing systems (Shintani and Hirokawa 2024)

Contents: Proving (Non-)Confluence Automatically

full run of TRS category in CoCo 2024

tool/method	YES	NO
CSI	272	205
ACP	257	168
Hakusan	139	74
CONFident	108	142
FORT-h	34	89
1 **rule labeling**	133	—
2 **persistency decomposition (RL+KB)**	$\geqslant$ 150	—
3 **generating convertible pairs**	—	—
4 **non-joinability test**	—	207

Persistency Decomposition

Example

TRS

$$x - 0 \to x \qquad \mathsf{s}(x) - \mathsf{s}(y) \to x - y \qquad x - x \to 0 \qquad \mathsf{c} \to \mathsf{f}(\mathsf{c})$$

is non-left-linear and non-terminating

how to prove confluence? ☞ split it into two subsystems

Modularity

Theorem (Toyama 1987)

$\mathsf{CR}(\mathcal{R} \cup \mathcal{S}) \iff \mathsf{CR}(\mathcal{R}) \wedge \mathsf{CR}(\mathcal{S})$ *if* $\mathcal{F}\mathsf{un}(\mathcal{R}) \cap \mathcal{F}\mathsf{un}(\mathcal{S}) = \varnothing$

Proof.

difficult (induction on rank) □

Example

non-left-linear and non-terminating TRS

$$x - 0 \xrightarrow{1} x \qquad \mathsf{s}(x) - \mathsf{s}(y) \xrightarrow{2} x - y \qquad x - x \xrightarrow{3} 0 \qquad \mathsf{c} \xrightarrow{4} \mathsf{f}(\mathsf{c})$$

is confluent because $\underbrace{\{\mathbf{1}, \mathbf{2}, \mathbf{3}\}}_{\text{complete}}$ and $\underbrace{\{\mathbf{4}\}}_{\text{orthogonal}}$ are confluent

Limitation of Modularity

consider TRS

$$\mathsf{s}(x) + y \xrightarrow{\mathbf{1}} \mathsf{s}(x+y) \qquad \mathsf{eq}(x,x) \xrightarrow{\mathbf{2}} \mathsf{T} \qquad \mathsf{from}(x) \xrightarrow{\mathbf{4}} x : \mathsf{from}(\mathsf{s}(x))$$
$$\mathsf{eq}(0,\mathsf{s}(x)) \xrightarrow{\mathbf{3}} \mathsf{F}$$

last modularity result is not applicable because

$$\mathsf{s} \in \mathcal{F}\mathsf{un}(\{\mathbf{1},\mathbf{2},\mathbf{3}\}) \cap \mathcal{F}\mathsf{un}(\{\mathbf{4}\})$$

Persistency

Theorem (Aoto 1997)

many-sorted TRS and its untyped version are equi-confluent

Proof.

very similar to proof of modularity □

Definition

$\mathcal{R}_\alpha$ is smallest subsystem of many-sorted TRS $\mathcal{R}$ such that

- $\ell \to r \in \mathcal{R}_\alpha$ whenever ℓ is of sort α, and
- $\mathcal{R}_\beta \subseteq \mathcal{R}_\alpha$ whenever subterm in $\mathcal{R}_\alpha$ is of sort β

Theorem (van de Pol 1995)

$\mathsf{CR}(\mathcal{R})$ *if and only if* $\mathsf{CR}(\mathcal{R}_\alpha)$ *for all sorts* α

Example of Persistency Decomposition

TRS

$$\mathsf{s}(x) + y \xrightarrow{\mathbf{1}} \mathsf{s}(x+y) \qquad \mathsf{eq}(x,x) \xrightarrow{\mathbf{2}} \mathsf{T} \qquad \mathsf{from}(x) \xrightarrow{\mathbf{4}} x : \mathsf{from}(\mathsf{s}(x))$$
$$\mathsf{eq}(0,\mathsf{s}(x)) \xrightarrow{\mathbf{3}} \mathsf{F}$$

admits many-sorted signature:

$$0 : \mathsf{N} \qquad \mathsf{T} : \mathsf{B} \qquad (+) : \mathsf{N} \times \mathsf{N} \to \mathsf{N} \qquad \mathsf{eq} : \mathsf{N} \times \mathsf{N} \to \mathsf{B}$$
$$\mathsf{s} : \mathsf{N} \to \mathsf{N} \qquad \mathsf{F} : \mathsf{B} \qquad (:) : \mathsf{N} \times \mathsf{L} \to \mathsf{L} \qquad \mathsf{from} : \mathsf{N} \to \mathsf{L}$$

$\mathcal{R}$ is confluent because $(\mathcal{R}_\mathsf{N} = \underbrace{\{\mathbf{1}\}}_{\text{orthogonal}},)$ $\mathcal{R}_\mathsf{B} = \underbrace{\{\mathbf{1},\mathbf{2},\mathbf{3}\}}_{\text{complete}}$, and $\mathcal{R}_\mathsf{L} = \underbrace{\{\mathbf{1},\mathbf{4}\}}_{\text{orthogonal}}$

Contents: Proving (Non-)Confluence Automatically

full run of TRS category in CoCo 2024

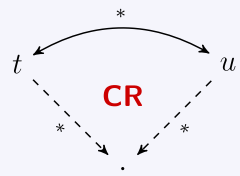
	tool/method	YES	NO
	CSI	272	205
	ACP	257	168
	Hakusan	139	74
	CONFident	108	142
	FORT-h	34	89
1	**rule labeling**	133	—
2	**persistency decomposition (RL+KB)**	⩾ 150	—
3	**generating convertible pairs**	—	—
4	**non-joinability test**	—	207

Non-Confluence Analysis

Proving Non-Confluence Automatically

Fact

confluence $\iff$ $\forall t, u.$



How To Prove Non-Confluence

1. generate convertible pairs $t \leftrightarrow^* u$
2. test non-joinability $t \not\downarrow u$

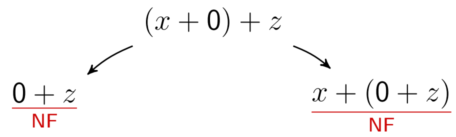
Generating Convertible Pairs

Associativity — Critical Pairs

TRS

$$x + 0 \to x \qquad\qquad (x + y) + z \to x + (y + z)$$

admits non-joinable critical pair:



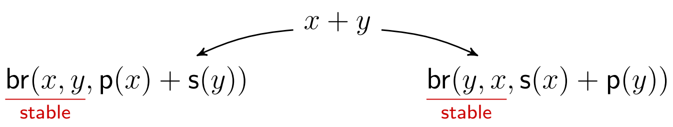
hence $\mathcal{R}$ is not confluent

ARI-COPS #493 — Critical Pairs

TRS

$$\mathsf{br}(0, y, z) \to y \qquad \mathsf{p}(0) \to 0 \qquad x + y \to \mathsf{br}(x, y, \mathsf{p}(x) + s(y))$$
$$\mathsf{br}(\mathsf{s}(x), y, z) \to z \qquad \mathsf{p}(\mathsf{s}(x)) \to x \qquad x + y \to \mathsf{br}(y, x, \mathsf{s}(x) + p(y))$$

admits non-joinable critical pair:



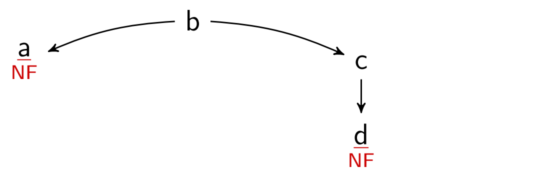
hence $\mathcal{R}$ is not confluent

Kleene's Example — Critical Pairs, Normalized

TRS

$$\mathsf{b} \to \mathsf{a} \qquad \mathsf{b} \to \mathsf{c} \qquad \mathsf{c} \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{d}$$

admits non-joinable pair originating from critical pair:



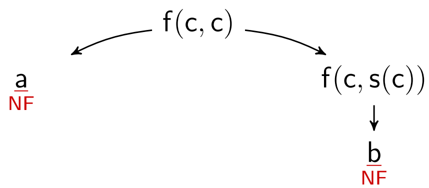
hence $\mathcal{R}$ is not confluent

Huet's TRS — No Critical Pairs...

TRS $\mathcal{R}$

$$\mathsf{f}(x, x) \to \mathsf{a} \qquad \mathsf{f}(x, \mathsf{s}(x)) \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{s}(\mathsf{c})$$

has admits non-joinable conversion:



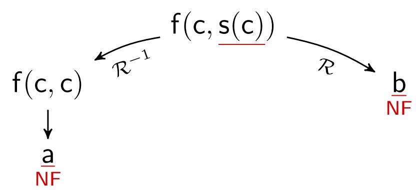
how to find such peak automatically? ☞ peak between $\mathcal{R}$ and $\mathcal{R}^{-1}$

Huet's Example — Critical Pairs of $\mathcal{R} \cup \mathcal{R}^{-1}$

TRS $\mathcal{R}$

$$\mathsf{f}(x, x) \to \mathsf{a} \qquad \mathsf{f}(x, \mathsf{s}(x)) \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{s}(\mathsf{c})$$

has admits non-joinable critical peak between $\mathcal{R}$ and $\mathcal{R}^{-1}$:



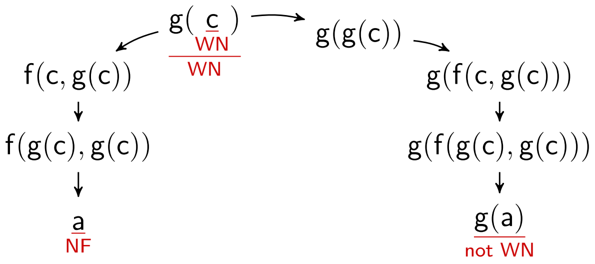
hence $\mathcal{R}$ is not confluent

Klop's Example — Reducing Right-Hand Sides

Klop's TRS

$$\mathsf{f}(x,x) \to \mathsf{a} \qquad \mathsf{g}(x) \to \mathsf{f}(x,\mathsf{g}(x)) \qquad \mathsf{c} \to \underline{\mathsf{g}(\mathsf{c})}$$

admits non-joinable conversion:



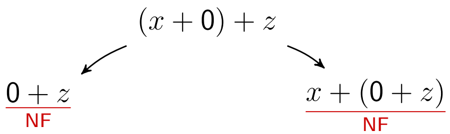
Testing Non-Joinability by tcap

Associativity — Distinct Normal Forms

TRS

$$x + 0 \to x \qquad (x+y)+z \to x+(y+z)$$

admits non-joinable critical pair:



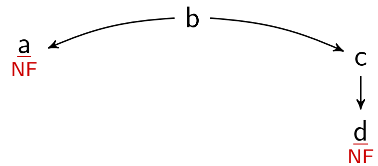
hence $\mathcal{R}$ is not confluent

Kleene's Example — Distinct Normal Forms

TRS

$$\mathsf{b} \to \mathsf{a} \qquad \mathsf{b} \to \mathsf{c} \qquad \mathsf{c} \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{d}$$

admits non-joinable pair originating from critical pair:



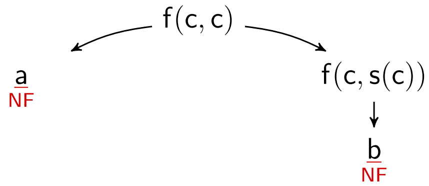
hence $\mathcal{R}$ is not confluent

Huet's TRS — Distinct Normal Forms

TRS $\mathcal{R}$

$$\mathsf{f}(x,x) \to \mathsf{a} \qquad \mathsf{f}(x,\mathsf{s}(x)) \to \mathsf{b} \qquad \mathsf{c} \to \mathsf{s}(\mathsf{c})$$

has admits non-joinable conversion:

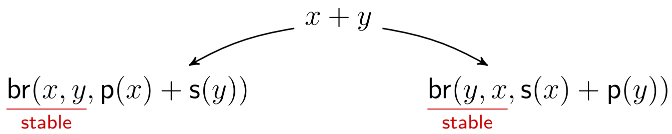


ARI-COPS #493 — No Normal Forms...

TRS

$$\begin{array}{lll} \mathsf{br}(0,y,z) \to y & \mathsf{p}(0) \to 0 & x+y \to \mathsf{br}(x,y,\mathsf{p}(x)+s(y)) \\ \mathsf{br}(\mathsf{s}(x),y,z) \to z & \mathsf{p}(\mathsf{s}(x)) \to x & x+y \to \mathsf{br}(y,x,\mathsf{s}(x)+p(y)) \end{array}$$

admits non-joinable critical pair:



how to identify stable part? ☞ cap function

Non-Reachability by tcap

Definition

$$\mathsf{tcap}_{\mathcal{R}}(t) = \begin{cases} u & \text{if } t = f(t_1,\ldots,t_n),\ u = f(\mathsf{tcap}_{\mathcal{R}}(t_1),\ldots,\mathsf{tcap}_{\mathcal{R}}(t_n)), \text{ and (*)} \\ y & \text{otherwise} \end{cases}$$

- (*): u is not unifiable with ℓ for any $\ell \to r \in \mathcal{R}$, and
- y is fresh variable

Example

$$\mathsf{tcap}_{\mathcal{R}}(\mathsf{s}(\underbrace{\underbrace{x}_{y_1} + \underbrace{x}_{y_2}}_{y_3}) + \mathsf{a}) = \mathsf{s}(y_3) + \mathsf{a} \qquad \text{for } \mathcal{R} = \{x + \mathsf{s}(y) \to \mathsf{s}(x+y)\}$$

Lemma

if $t \to^* u$ then $u = \mathsf{tcap}_{\mathcal{R}}(t)\sigma$ for some σ

Non-Joinability by tcap

Lemma

$t \not\downarrow u$ if $\mathsf{tcap}_{\mathcal{R}}(t)$ and $\mathsf{tcap}_{\mathcal{R}}(u)$ are not unifiable

Proof.

since $t' = \mathsf{tcap}_{\mathcal{R}}(t)$ and $u' = \mathsf{tcap}_{\mathcal{R}}(u)$ have no common variables,

$$t \to^* v \;{}^*\!\!\leftarrow u \implies t'\sigma = v = u'\sigma \text{ for some } \sigma$$

hence contraposition holds □

ARI-COPS #493 — tcap ?

TRS

$$\begin{array}{lll} \mathsf{br}(0,y,z) \to y & \mathsf{p}(0) \to 0 & x + y \to \mathsf{br}(x,y,\mathsf{p}(x)+s(y)) \\ \mathsf{br}(\mathsf{s}(x),y,z) \to z & \mathsf{p}(\mathsf{s}(x)) \to x & x + y \to \mathsf{br}(y,x,\mathsf{s}(x)+p(y)) \end{array}$$

by applying tcap to peak

$$t = \underbrace{\mathsf{br}(\underbrace{x}_{x_1}, \underbrace{y}_{x_2}, \underbrace{\underbrace{\mathsf{p}(\underbrace{x}_{x_3})}_{x_4} + \mathsf{s}(\underbrace{y}_{x_5})}_{x_6})}_{x_7} \leftarrow x + y \rightarrow \underbrace{\mathsf{br}(\underbrace{y}_{y_1}, \underbrace{x}_{y_2}, \underbrace{\mathsf{s}(\underbrace{x}_{y_3}) + \underbrace{\mathsf{p}(\underbrace{y}_{y_4})}_{y_5}}_{y_6})}_{y_7} = u$$

$\mathsf{tcap}_{\mathcal{R}}(t) = x_7$ and $\mathsf{tcap}_{\mathcal{R}}(u) = y_7$ are unifiable ...

ARI-COPS #493 — tcap and Grounding

TRS

$$\begin{array}{lll} \mathsf{br}(0,y,z) \to y & \mathsf{p}(0) \to 0 & x + y \to \mathsf{br}(x,y,\mathsf{p}(x)+s(y)) \\ \mathsf{br}(\mathsf{s}(x),y,z) \to z & \mathsf{p}(\mathsf{s}(x)) \to x & x + y \to \mathsf{br}(y,x,\mathsf{s}(x)+p(y)) \end{array}$$

by applying tcap to peak

$$t = \mathsf{br}(\mathsf{c}_x, \mathsf{c}_y, \underbrace{\mathsf{p}(\mathsf{c}_x) + \mathsf{s}(\mathsf{c}_y)}_{x_1}) \leftarrow \mathsf{c}_x + \mathsf{c}_y \rightarrow \mathsf{br}(\mathsf{c}_y, \mathsf{c}_x, \underbrace{\mathsf{s}(\mathsf{c}_x) + \mathsf{p}(\mathsf{c}_y)}_{y_1}) = u$$

$t \not\downarrow u$ follows from non-unifiability of $\begin{cases} \mathsf{tcap}_{\mathcal{R}}(t) = \mathsf{br}(\mathsf{c}_x, \mathsf{c}_y, x_1) \\ \mathsf{tcap}_{\mathcal{R}}(u) = \mathsf{br}(\mathsf{c}_y, \mathsf{c}_x, y_1) \end{cases}$

Summary of tcap

Definition

$\hat{t} = t\{x \mapsto \mathsf{c}_x \mid x \in \mathcal{V}\mathsf{ar}(t)\}$ where c_x is fresh constant

Theorem (Zankl et al. 2011)

$t \not\downarrow u$ if $\mathsf{tcap}_{\mathcal{R}}(\hat{t})$ and $\mathsf{tcap}_{\mathcal{R}}(\hat{u})$ are not unifiable

Fact

$\mathsf{tcap}_{\mathcal{R}}(\hat{t})$ and $\mathsf{tcap}_{\mathcal{R}}(\hat{u})$ are not unifiable if t and u are distinct normal form

Proof.

$\mathsf{tcap}_{\mathcal{R}}(\hat{t}) = \hat{t}$ if t is ground normal form □

Testing Non-Joinability by Tree Automata

Klop's Example — No Stable Part...

TRS

$$\mathsf{f}(x,x) \to \mathsf{a} \qquad \mathsf{g}(x) \to \mathsf{f}(x,\mathsf{g}(x)) \qquad \mathsf{c} \to \mathsf{g}(\mathsf{c})$$

admits conversion:

$$\mathsf{a} \overset{*}{\longleftrightarrow} \underbrace{\mathsf{g}(\mathsf{a})}_{y_1}$$

$\mathsf{tcap}_{\mathcal{R}}(\mathsf{a}) = \mathsf{a}$ and $\mathsf{tcap}_{\mathcal{R}}(\mathsf{g}(\mathsf{a})) = y_1$ are unifiable ...

Klop's Example — Disjoint Closed Sets

TRS $\mathcal{R}$

$$\mathsf{f}(x,x) \to \mathsf{a} \qquad \mathsf{g}(x) \to \mathsf{f}(x,\mathsf{g}(x)) \qquad \mathsf{c} \to \mathsf{g}(\mathsf{c})$$

admits conversion:

$$\mathsf{a} \overset{*}{\longleftrightarrow} \mathsf{g}(\mathsf{a})$$

define

$$T = \{\mathsf{a}\} \qquad U = \{\mathsf{g}(\mathsf{a}), \mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a})), \mathsf{f}(\mathsf{a},\mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a}))), \ldots\}$$

because T and U are disjoint and closed under rewriting, $\mathsf{a} \not\downarrow \mathsf{g}(\mathsf{a})$

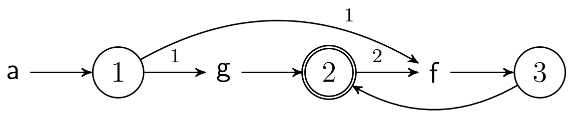
☞ automatable if T and U are represented by tree automata

Tree Automata Construction

TRS $\mathcal{R}$:

$$\mathsf{f}(x,x) \to \mathsf{a} \qquad \mathsf{g}(x) \to \mathsf{f}(x,\mathsf{g}(x)) \qquad \mathsf{c} \to \mathsf{g}(\mathsf{c})$$

tree automaton M:



1 $\mathsf{g}(\mathsf{a}) \in \mathsf{L}(M)$

2 $\mathsf{g}(1) \to_M^* 2$ demands $\underbrace{\mathsf{f}(1, \underbrace{\mathsf{g}(1)}_{2})}_{3} \to_M^* 2$

3 completed; $L(M) = \{\mathsf{g}(\mathsf{a}), \mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a})), \mathsf{f}(\mathsf{a},\mathsf{f}(\mathsf{a},\mathsf{g}(\mathsf{a}))), \ldots\}$

Summary of Tree Automata Techniques

Theorem

$t \not\downarrow u$ if there exist tree automata M and N such that

- $t \in L(M)$, $u \in L(N)$,
- T and U are closed under rewriting, and — decidable
- $T \cap U = \varnothing$ — decidable

Fact (simulation of tcap by tree automata)

for every ground term t there exists tree automaton M such that

$$\mathsf{L}(M) = \{u \mid u \text{ is ground instance of } \mathsf{tcap}_{\mathcal{R}}(t)\}$$

and $\mathsf{L}(M)$ is closed under $\mathcal{R}$-rewriting

Contents: Proving (Non-)Confluence Automatically

full run of TRS category in CoCo 2024

	tool/method	YES	NO
	CSI	272	205
	ACP	257	168
	Hakusan	139	74
	CONFident	108	142
	FORT-h	34	89
1	**rule labeling**	133	—
2	**persistency decomposition (RL+KB)**	$\geqslant$ 150	—
3	**generating convertible pairs**	—	—
4	**non-joinability test**	—	207

References for Tree Automata Techniques

- T. Genet.
  Decidable Approximations of Sets of Descendants and Sets of Normal Forms.
  9th RTA, LNCS 1379, pp. 151–165, 1998.

- B. Felgenhauer and R. Thiemann.
  Reachability, Confluence, and Termination Analysis with State-Compatible Automata. I&C 253:467–483, 2017.