universität innsbruck

supported by FWF project I 5943-N

Automated Analysis of LCTRSs using crest

Jonas Schöpf and Aart Middeldorp

31st TACAS – Hamilton
5 May 2025

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

compute $\sum_{i=1}^{n} i$ for natural number n

Example

- term rewrite system (TRS)

$$\mathsf{sum}(0) \to 0 \qquad \mathsf{add}(0, y) \to y$$
$$\mathsf{sum}(\mathsf{s}(x)) \to \mathsf{add}(\mathsf{s}(x), \mathsf{sum}(x)) \qquad \mathsf{add}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{add}(x, y))$$

- rewriting

$$\mathsf{sum}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))) \to \mathsf{add}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))), \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)))) \to \mathsf{s}(\mathsf{add}(\mathsf{s}(\mathsf{s}(0)), \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)))))$$
$$\to \cdots \to \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))))))$$

- logically constrained term rewrite system (LCTRS)

$$\mathsf{sum}(x) \to 0\ [x \leqslant 0] \qquad \mathsf{sum}(x) \to x + \mathsf{sum}(x - 1)\ [x > 0]$$

- rewriting

$$\mathsf{sum}(3) \to 3 + \mathsf{sum}(3 - 1) \to 3 + \mathsf{sum}(2) \to 3 + (2 + \mathsf{sum}(2 - 1)) \to \cdots \to 6$$

Example

- LCTRS

$$\mathsf{sum}(x) \to 0 \; [x \leqslant 0] \qquad\qquad \mathsf{sum}(x) \to x + \mathsf{sum}(x-1) \; [x > 0]$$

- two sorts Bool and Int with $\mathcal{V}\mathsf{al}_{\mathsf{Bool}} = \{\bot, \top\}$ and $\mathcal{V}\mathsf{al}_{\mathsf{Int}} = \mathbb{Z}$
- signature $\mathcal{F}_{\mathsf{th}}$ $\quad +, -\colon \mathsf{Int} \times \mathsf{Int} \to \mathsf{Int} \quad \leqslant, >\colon \mathsf{Int} \times \mathsf{Int} \to \mathsf{Bool} \quad \ldots, -1, 0, 1, \ldots\colon \mathsf{Int}$
- signature $\mathcal{F}_{\mathsf{te}}$ $\quad \mathsf{sum}\colon \mathsf{Int} \to \mathsf{Int}$

standard rule

$$\mathsf{sum}(3-1) \to_{\mathcal{R}} \mathsf{sum}(2) \to_{\mathcal{R}} 2 + \mathsf{sum}(2-1)$$

calculation rule

Constrained Term

$\mathsf{sum}(x) \; [x > 3]$ represents the set of instances $\{\mathsf{sum}(4), \mathsf{sum}(5), \mathsf{sum}(6), \ldots\}$

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

Termination

LCTRS $\mathcal{R}$ does not admit an infinite rewrite sequence

Theorem

LCTRS $\mathcal{R}$ is terminating $\iff$ there exists well-founded order $\succ$ on terms and $\to_{\mathcal{R}} \subseteq \succ$

Confluence

- undecidable in general
- (parallel) critical pairs
- extensive use of rewriting constrained terms
- closing criteria by rewriting constrained terms

$$\begin{tikzcd} & s \arrow[dl, "*"'] \arrow[dr, "*"] & \\ t \arrow[dr, dashed, "*"'] & & u \arrow[dl, dashed, "*"] \\ & v & \end{tikzcd}$$

Applications of Confluence

order of computations/operations is irrelevant

- (term) rewrite systems
- programming languages semantics
- formal verification & theorem proving
- computer algebra systems
- ...

Theorems

- linear strongly closed LCTRSs are confluent — CADE 2023
- left-linear almost parallel closed LCTRSs are confluent — CADE 2023
- left-linear almost development closed LCTRSs are confluent — IJCAR 2024
- left-linear LCTRSs with parallel closed parallel critical pairs are confluent — IJCAR 2024

Example

LCTRS $\mathcal{R}$

$$\mathsf{f}(x) \to \mathsf{g}(x)\ [x \geqslant 1] \qquad \mathsf{g}(1) \to \mathsf{a} \qquad \mathsf{h}(x) \to \mathsf{a}\ [x \leqslant 1]$$
$$\mathsf{f}(x) \to \mathsf{h}(x)\ [x \leqslant 2] \qquad \mathsf{g}(x) \to \mathsf{b}\ [x \geqslant 2] \qquad \mathsf{h}(x) \to \mathsf{b}\ [x > 1]$$
$$\mathsf{g}(x) \to \mathsf{c}\ [x < 1]$$

$\{\mathsf{g}(1) \approx \mathsf{h}(1), \mathsf{g}(2) \approx \mathsf{h}(2)\}$

constrained critical pair

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ [x \geqslant 1 \wedge x \leqslant 2]$$

in normal form? Yes. Not confluent? No, because

$$\{x \mapsto 1\}: \qquad \mathsf{g}(1) \approx \mathsf{h}(1)\ [\varphi] \to \mathsf{a} \approx \mathsf{h}(1)\ [\varphi] \to \mathsf{a} \approx \mathsf{a}\ [\varphi]$$
$$\{x \mapsto 2\}: \qquad \mathsf{g}(2) \approx \mathsf{h}(2)\ [\varphi] \to \mathsf{b} \approx \mathsf{h}(2)\ [\varphi] \to \mathsf{b} \approx \mathsf{b}\ [\varphi]$$

Normal Form

$s\ [\varphi]$ is in normal form $\iff$ for all $\sigma \vDash \varphi$ exists no t with $s\sigma \to_{\mathcal{R}} t$ in $\mathcal{R}$

Example

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ \underbrace{[x \geqslant 1 \wedge x \leqslant 2]}_{\varphi}$$

is not a normal form!

$$\begin{aligned} \{x \mapsto 1\}: &\qquad \mathsf{g}(1) \approx \mathsf{h}(1)\ [\varphi] \to \mathsf{a} \approx \mathsf{h}(1)\ [\varphi] \to \mathsf{a} \approx \mathsf{a}\ [\varphi] \\ \{x \mapsto 2\}: &\qquad \mathsf{g}(2) \approx \mathsf{h}(2)\ [\varphi] \to \mathsf{b} \approx \mathsf{h}(2)\ [\varphi] \to \mathsf{b} \approx \mathsf{b}\ [\varphi] \end{aligned}$$

How to show confluence?

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

Transformations

- LCTRS to TRS for proofs $\quad \mathsf{f}(x) \to \mathsf{g}(x)\ [x \geqslant 1] \rightsquigarrow \mathsf{f}(1) \to \mathsf{g}(1), \mathsf{f}(2) \to \mathsf{g}(2), \ldots$
- move values to constraint $\quad \mathsf{f}(1) \to \mathsf{g}(x)\ [x \geqslant 1] \rightsquigarrow \mathsf{f}(y) \to \mathsf{g}(x)\ [x \geqslant 1 \land y = 1]$
- how to proceed?
  split problem?

Example

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ [x \geqslant 1 \land x \leqslant 2]$$

- admits rewrite step? ✗
- is normal form? (✓) ✗
- is non-confluent? ✗
- joinable instances? ✓
- can be shown joinable? ✗

Example

LCTRS $\mathcal{R}$

$$\mathsf{f}(x) \to \mathsf{g}(x)\ [x \geqslant 1] \qquad \mathsf{g}(1) \to \mathsf{a} \qquad \mathsf{h}(x) \to \mathsf{a}\ [x \leqslant 1]$$

$$\mathsf{f}(x) \to \mathsf{h}(x)\ [x \leqslant 2] \qquad \mathsf{g}(x) \to \mathsf{b}\ [x \geqslant 2] \qquad \mathsf{h}(x) \to \mathsf{b}\ [x > 1]$$

$$\mathsf{g}(x) \to \mathsf{c}\ [x < 1]$$

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ \underbrace{[x \geqslant 1 \land x \leqslant 2]}_{\varphi} \qquad \mathsf{g}(1) \to \mathsf{a} \rightsquigarrow \mathsf{g}(y) \to \mathsf{a}\ [y = 1]$$

given

$$\mathsf{g}(y)\{y \mapsto x\} = \mathsf{g}(x) \qquad \varphi \land x = 1 \text{ is satisfiable}$$

we split

$$\mathsf{g}(x) \approx \mathsf{h}(x)[\varphi \land x = 1] \qquad \mathsf{g}(x) \approx \mathsf{h}(x)[\varphi \land x \neq 1]$$

Example

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ [\varphi \wedge x = 1] \qquad\qquad \mathsf{g}(x) \approx \mathsf{h}(x)\ [\varphi \wedge x \neq 1]$$

using $\mathsf{g}(y) \to \mathsf{a}\ [y = 1]$ and $\mathsf{h}(x) \to \mathsf{a}\ [x \leqslant 1]$

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ [\varphi \wedge x = 1] \xrightarrow{\sim}{}^{*} \mathsf{a} \approx \mathsf{a}\ [\varphi \wedge x = 1]$$

using $\mathsf{g}(y) \to \mathsf{b}\ [y \geqslant 2]$ and $\mathsf{h}(x) \to \mathsf{b}\ [x > 1]$

$$\mathsf{g}(x) \approx \mathsf{h}(x)\ [\varphi \wedge x \neq 1] \xrightarrow{\sim}{}^{*} \mathsf{b} \approx \mathsf{b}\ [\varphi \wedge x \neq 1]$$

$\Longrightarrow$ locally confluent

using RPO $\mathsf{f} > \mathsf{g} > \mathsf{h} > \mathsf{a} > \mathsf{b} > \mathsf{c}$ $\Longrightarrow$ terminating

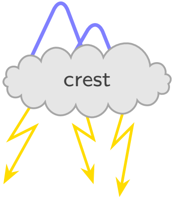
using Newman's lemma $\Longrightarrow$ confluent

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

Constrained REwriting Software Tool (crest)

- Haskell
- open source
- Z3 and CVC5
- (non-)confluence and termination analysis



Rewrite Engine

- single-step rewriting
- parallel-step rewriting
- multi-step rewriting
- (parallel) critical pairs

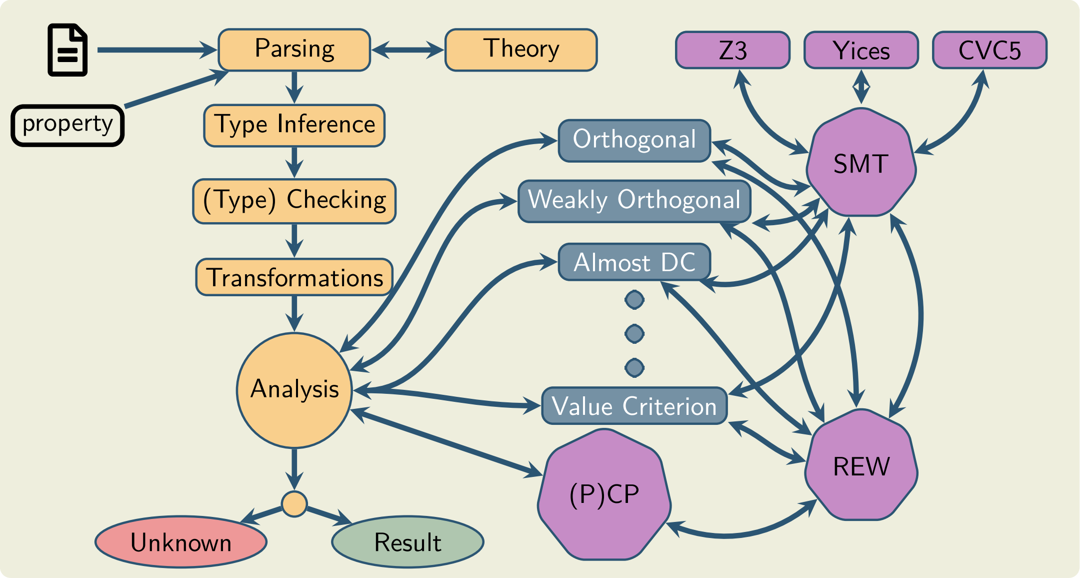
SMT Backend

- SMT-theories
- SMT-formulas
- checking constraints
- link to SMT solver

Analysis

- select methods
- concurrent execution
- evaluating results
- preparing proofs

Simplified Overview of crest



Ctrl (Kop & Nishida)

- first tool on LCTRSs (not maintained)
- OCaml (based on $\mathsf{T_TT_2}$)
- termination, confluence, completion, rewriting induction, . . .

prototype (Schöpf & Middeldorp)

- prototype of CADE 2023
- Haskell
- confluence

Cora (Vale et al.)

- higher-order LCTRSs
- Java
- termination, . . .

CRaris (Nishida & Kojima)

- OCaml
- basic confluence criteria

Confluence Competition

- annual competition since 2012
- LCTRS category (since 2024)
- 1st place for crest

Confluence Experiments on 107 Benchmarks

official ARI database

criterion in crest	solved	time (total)
termination and joinable critical pairs	50	487 s
strongly closed critical pairs	56	129 s
almost parallel closed critical pairs	70	1180 s
almost development closed critical pairs	71	220 s
parallel closed parallel critical pairs	71	1490 s
all confluence methods	72	899 s
non-confluence	26	210 s
all methods	98	197 s
total solved	98	—

Confluence Experiments on 107 Benchmarks

tool	✓	✗	solved	time (total)
CRaris	58	0	54 %	14 s
crest	72	26	92 %	197 s
Ctrl	54	0	50 %	18 s
prototype	67	0	63 %	122 s
total solved	72	26	92 %	—

Termination Experiments on 107 Benchmarks

method in crest	solved	time (total)
DP graph	9	9 s
recursive path order	27	12 s
recursive path order (DP)	28	12 s
subterm criterion	12	13 s
value criterion	34	14 s
special value criterion	70	13 s
reduction pairs no SVC	37	15 s
default	74	16 s
total solved	74	—

Termination Experiments on 107 Benchmarks

tool	✓	solved	time (total)
Cora	71	66 %	264 s
crest	74	69 %	16 s
Ctrl	74	69 %	103 s
total solved	78	73 %	—

Table of Contents

- Logically Constrained Rewrite Systems
- Confluence & Termination
- Transformations
- Automation & Experiments
- Final Remarks

Concluding Remarks

- new transformations for automation
- automated confluence and termination analysis
- push-button analysis

Future Work

- non-termination
- completion
- reachability analysis
- formalize theoretical results in a proof assistant
- ...