supported by FWF project I 5943-N

Automated Analysis of Logically Constrained Programs

Jonas Schöpf

IFI Lunchtime Seminar
12 December 2024

Table of Contents

Table of Contents

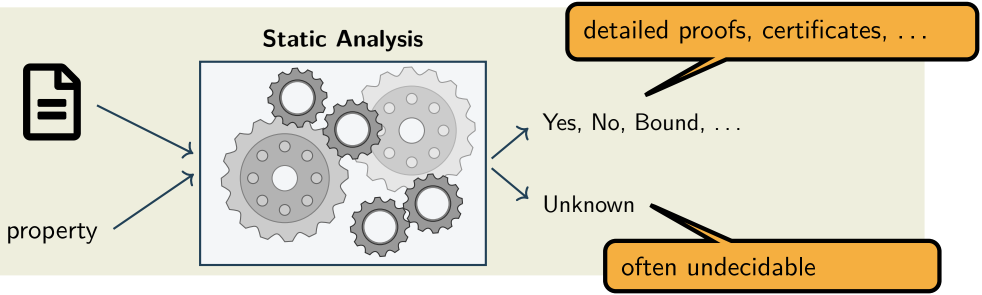
Program Properties

- does it terminate? should it terminate?
- does it follow the specification?
- are there any (critical) bugs?
- ...

Program Analysis/Verification

- absence of bugs by formal verification
- analyze specific properties
- show program equivalence
- static vs. dynamic program analysis

Static Program Analysis



- integral part of formal verification
- improving the quality of complex software
- medical software, aviation software, nuclear software, compiler optimizations, . . .

- term rewriting
- type systems
- model checking
- . . .

$\max$ computes the maximum of two integers:

$$\max(x, y) = \begin{cases} x & x \geqslant y \\ y & \text{otherwise} \end{cases}$$

```
int max (int x, int y) {
  if (x >= y) {
    return x;
  }
  else if (y >= x) {
    return y;
  }
  else {
    return (max(y,x));
  }
}
```

- maximum of two integers
- 3 different cases
- correct?
- unique result?
- terminating?

Term Rewrite Systems (TRSs)

set of function symbols	$\{\mathsf{max}, \mathsf{0}\}$
set of variables	$\{x, y, z, \ldots\}$
terms	$\mathsf{max}(x, y), \mathsf{max}(\mathsf{max}(x, \mathsf{0}), z), \ldots$
rules	$\mathsf{max}(\mathsf{s}(\mathsf{0}), \mathsf{0}) \to \mathsf{s}(\mathsf{0})$
set of rules	$\{\mathsf{max}(\mathsf{s}(\mathsf{0}), \mathsf{0}) \to \mathsf{s}(\mathsf{0}), \ldots\}$

- terms represent program states
- rewriting represents computation
- term rewriting is Turing-complete

Term Rewrite System

signature $\{\mathsf{max}, \mathsf{ite}, \mathsf{s}, \mathsf{p}, \mathsf{geq}, \mathsf{geq2}, \mathsf{0}, \mathsf{true}, \mathsf{false}\}$ and rules

$$\begin{aligned}
\mathsf{max}(x,y) &\to \mathsf{ite}(\mathsf{geq}(x,y),x,y) & \mathsf{ite}(\mathsf{true},x,y) &\to x \\
\mathsf{max}(x,y) &\to \mathsf{max}(y,x) & \mathsf{ite}(\mathsf{false},x,y) &\to y \\
\mathsf{s}(\mathsf{p}(x)) &\to x & \mathsf{p}(\mathsf{s}(x)) &\to x \\
\mathsf{geq}(x,y) &\to \mathsf{geq2}(x,y,\mathsf{0},\mathsf{0}) & \mathsf{geq2}(\mathsf{s}(x),y,z,u) &\to \mathsf{geq2}(x,y,\mathsf{s}(z),u) \\
\mathsf{geq2}(\mathsf{p}(x),y,z,u) &\to \mathsf{geq2}(x,y,z,\mathsf{s}(u)) & \mathsf{geq2}(\mathsf{0},\mathsf{s}(x),y,z) &\to \mathsf{geq2}(\mathsf{0},x,y,\mathsf{s}(z)) \\
\mathsf{geq2}(\mathsf{0},\mathsf{p}(x),y,z) &\to \mathsf{geq2}(\mathsf{0},x,\mathsf{s}(y),z) & \mathsf{geq2}(\mathsf{0},\mathsf{0},\mathsf{s}(x),\mathsf{s}(y)) &\to \mathsf{geq2}(\mathsf{0},\mathsf{0},x,y) \\
\mathsf{geq2}(\mathsf{0},\mathsf{0},x,\mathsf{0}) &\to \mathsf{true} & \mathsf{geq2}(\mathsf{0},\mathsf{0},\mathsf{0},\mathsf{s}(x)) &\to \mathsf{false}
\end{aligned}$$

compute $\max(4,5)$: $\mathsf{max}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))),\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0}))))))$

$$\begin{aligned}
\mathsf{max}(\mathsf{s}^4(\mathsf{0}),\mathsf{s}^5(\mathsf{0})) &\to \mathsf{ite}(\mathsf{geq}(\mathsf{s}^4(\mathsf{0}),\mathsf{s}^5(\mathsf{0})),\mathsf{s}^4(\mathsf{0}),\mathsf{s}^5(\mathsf{0})) \\
&\to \mathsf{ite}(\mathsf{geq2}(\mathsf{s}^4(\mathsf{0}),\mathsf{s}^5(\mathsf{0})),\mathsf{s}^4(\mathsf{0}),\mathsf{s}^5(\mathsf{0}),\mathsf{0},\mathsf{0}) \\
&\to^{12} \mathsf{s}^5(\mathsf{0}) = \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{0})))))
\end{aligned}$$

Rewriting Formalisms

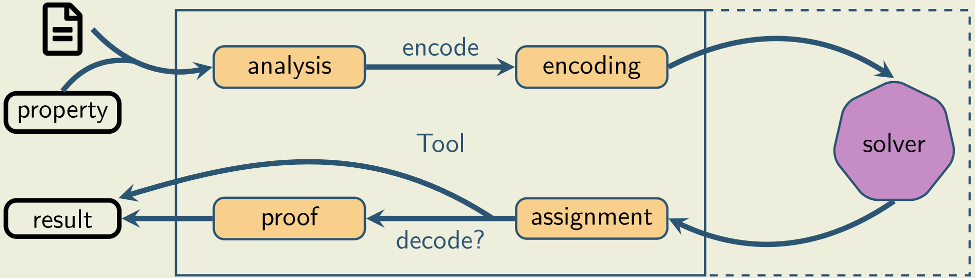
MSTRS
SRS
HRS
AFS
CTRS
LCTRS
CRS
CSTRS
TRS
LCSTRS
LCTRS (Ciobâcă et al.)
CSCTRS

Properties

- **termination** does the program finish in a finite amount of time?
- **confluence** does the program compute unique solutions?
- **equivalence** do the programs produce the same output for equal inputs?
- **complexity** which runtime complexity does the program has?
- ...

Automation

- tedious & error-prone by hand
- large & complex systems
- properties involve non-trivial checks



Tools

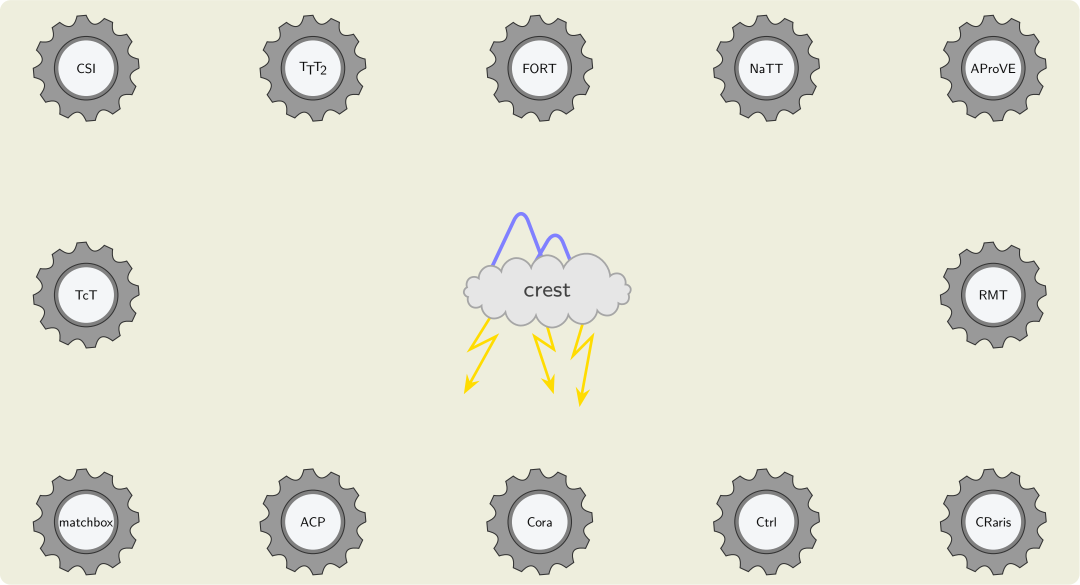

CSI
TTT2
FORT
NaTT
AProVE
TcT
crest
RMT
matchbox
ACP
Cora
Ctrl
CRaris

Table of Contents

- Overview
- LCTRSs
- Confluence Analysis
- Automation
- Experiments

Example

LCTRS $\mathcal{M}$

$$\mathcal{I}_{\mathsf{Bool}} = \mathbb{B} \qquad \mathcal{I}_{\mathsf{Int}} = \mathbb{Z}$$

$$\mathcal{F}_{\mathsf{te}} = \quad \ldots, -1, 0, 1, \ldots : \mathsf{Int} \qquad \mathsf{max} : [\mathsf{Int}] \Rightarrow \mathsf{Int}$$

$$\mathcal{F}_{\mathsf{th}} = \quad \ldots, -1, 0, 1, \ldots : \mathsf{Int} \qquad \wedge : [\mathsf{Bool} \times \mathsf{Bool}] \Rightarrow \mathsf{Bool}$$

$$\mathsf{true}, \mathsf{false} : \mathsf{Bool} \qquad +, - : [\mathsf{Int} \times \mathsf{Int}] \Rightarrow \mathsf{Int}$$

$$\neg : [\mathsf{Bool}] \Rightarrow \mathsf{Bool} \qquad \leqslant, \geqslant, = : [\mathsf{Int} \times \mathsf{Int}] \Rightarrow \mathsf{Bool}$$

$$\mathcal{M} = \qquad \mathsf{max}(x, y) \rightarrow x \ [x \geqslant y] \qquad \mathsf{max}(x, y) \rightarrow y \ [y \geqslant x] \qquad \mathsf{max}(x, y) \rightarrow \mathsf{max}(y, x)$$

$$\mathsf{max}(\underline{2 + 1}, 1 + 3) \rightarrow \mathsf{max}(3, \underline{1 + 3}) \rightarrow \underline{\mathsf{max}(3, 4)} \rightarrow \underline{\mathsf{max}(4, 3)} \rightarrow 4$$

Example

LCTRS $\mathcal{M}$

$$\mathcal{I}_{\mathsf{Bool}} = \mathbb{B} \qquad \mathcal{I}_{\mathsf{Int}} = \mathbb{Z}$$

$$\mathcal{F}_{\mathsf{te}} = \quad \ldots, -1, 0, 1, \ldots : \mathsf{Int} \qquad \mathsf{max} : [\mathsf{Int}] \Rightarrow \mathsf{Int}$$

$$\mathcal{F}_{\mathsf{th}} = \quad \ldots, -1, 0, 1, \ldots : \mathsf{Int} \qquad \wedge : [\mathsf{Bool} \times \mathsf{Bool}] \Rightarrow \mathsf{Bool}$$

$$\mathsf{true}, \mathsf{false} : \mathsf{Bool} \qquad +, - : [\mathsf{Int} \times \mathsf{Int}] \Rightarrow \mathsf{Int}$$

$$\neg : [\mathsf{Bool}] \Rightarrow \mathsf{Bool} \qquad \leqslant, \geqslant, = : [\mathsf{Int} \times \mathsf{Int}] \Rightarrow \mathsf{Bool}$$

$$\mathcal{M} = \qquad \mathsf{max}(x,y) \to x \; [x \geqslant y] \qquad \mathsf{max}(x,y) \to y \; [y \geqslant x] \qquad \mathsf{max}(x,y) \to \mathsf{max}(y,x)$$

$$\begin{aligned}
\mathsf{max}(x, 1+3) \; [x > 4] &\sim \mathsf{max}(x, \underline{1+3}) \; [x > 4 \wedge y = 1 + 3] \\
&\to_{\mathcal{M}} \mathsf{max}(x, \underline{y}) \; [x > 4 \wedge y = 1 + 3] \sim \underline{\mathsf{max}(x, 4)} \; [x > 4] \\
&\to_{\mathcal{M}} x \; [x > 4]
\end{aligned}$$

Example

$\mathsf{max}(x, y) \to x\ [x \geqslant y]$ $\mathsf{max}(x, y) \to y\ [y \geqslant x]$ $\mathsf{max}(x, y) \to \mathsf{max}(y, x)$

compute $\max(4, 5)$: $\mathsf{max}(4, 5) \to 5$

$\mathsf{max}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))), \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))))) \to^{14} \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0)))))$

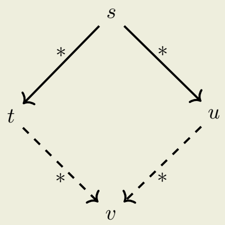
Utilize SMT Solver

- solving formula with special interpretations
- built-in structures (e.g. integers)
- split into term (syntax) and theory (semantics)
- automation via SMT-solvers
  - recently more powerful
  - real numbers, integers, bit-vectors, arrays, ...
  - Z3, CVC5, ...

Table of Contents

Confluence

- undecidable in general
- test this for all terms?
- test this for all rules?
- analyze all peaks of this form?
- analyze all **critical** peaks of this form?



Example

6 critical pairs of $\mathcal{M}$

$x \approx y \; [y \geqslant x \wedge x \geqslant y]$ $\qquad$ $x \approx \mathsf{max}(y, x) \; [x \geqslant y]$ $\qquad$ $\ldots$

TRS Confluence Methods

- (weak) orthogonality, strong closedness, (almost) parallel closedness, ...
- rule labeling, critical pair closing systems, ...
- parallel closed parallel critical pairs, Okui's criterion, ...
- redundant rules, order-sorted decomposition, reduction method, ...

How to Obtain LCTRS Confluence Methods?

- LCTRSs subsume TRSs
- reuse existing methods?
- decades of research
- **difficult** to adapt TRS proofs
- proofs via special transformation

Example

$$\mathsf{max}(x, y) \to x\ [x \geqslant y] \qquad \mathsf{max}(x, y) \to y\ [y \geqslant x] \qquad \mathsf{max}(x, y) \to \mathsf{max}(y, x)$$

single-step rewriting:

$$\begin{aligned}
\mathsf{max}(3 + \underline{(5 + 6)}, 3 + (y + 0))\ [\underline{y = 2}] &\xrightarrow{\sim} \mathsf{max}(\underline{3 + z_1}, 3 + (y + 0))\ [\varphi] \\
&\to \mathsf{max}(z_2, 3 + \underline{(y + 0)})\ [\varphi] \\
&\to \mathsf{max}(z_2, \underline{3 + y_1})\ [\varphi] \\
&\to \underline{\mathsf{max}(z_2, y_2)}\ [\varphi] \to \underline{\mathsf{max}(y_2, z_2)}\ [\underline{\varphi}] \xrightarrow{\sim} 14\ [\mathsf{true}]
\end{aligned}$$

multi-step rewriting:

$$\begin{aligned}
\mathsf{max}(3 + \underline{(5 + 6)}, 3 + \underline{(y + 0)})\ [\underline{y = 2}] &\tilde{\twoheadrightarrow} \mathsf{max}(\underline{3 + z_1}, \underline{3 + y_1})\ [\varphi] \\
&\twoheadrightarrow \mathsf{max}(\underline{y_2}, \underline{z_2})\ [\underline{\varphi}] \tilde{\twoheadrightarrow} 14\ [\mathsf{true}]
\end{aligned}$$

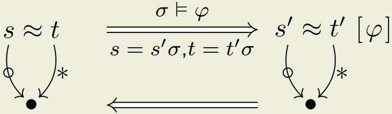
General

- almost development closedness by Vincent van Oostrom
- based on multi-step
- closing critical peaks/pairs
- recently formalized in Isabelle/HOL by Christina Kirk

Theorem TRSs

left-linear almost development closed TRSs are confluent

Proof Overview



Theorem

left-linear almost development closed LCTRSs are confluent

Example

LCTRS $\mathcal{M}$

$$\mathsf{max}(x,y) \to x\ [x \geqslant y] \qquad \mathsf{max}(x,y) \to y\ [y \geqslant x] \qquad \mathsf{max}(x,y) \to \mathsf{max}(y,x)$$

$x \approx y\ [y \geqslant x \wedge x \geqslant y]$:

$$x \approx y\ [y \geqslant x \wedge x \geqslant y] \xrightarrow{\tilde{\ }}\!\!\!\!\!\!\circ\ _{\geqslant 1} \cdot \xrightarrow{\sim}{}^{*}_{\geqslant 2} x \approx y\ [x = y] \qquad \text{is trivial}$$

$x \approx \mathsf{max}(y,x)\ [x \geqslant y]$:

$$x \approx \underline{\mathsf{max}(y,x)}\ [x \geqslant y] \xrightarrow{\tilde{\ }}\!\!\!\!\!\!\circ\ _{\geqslant 1} \cdot \xrightarrow{\sim}{}^{*}_{\geqslant 2} x \approx x\ [x \geqslant y] \qquad \text{is trivial}$$

$\mathsf{max}(y,x) \approx y\ [y \geqslant x]$:

$$\underline{\mathsf{max}(y,x)} \approx y\ [y \geqslant x] \xrightarrow{\tilde{\ }}\!\!\!\!\!\!\circ\ _{\geqslant 1} \cdot \xrightarrow{\sim}{}^{*}_{\geqslant 2} y \approx y\ [y \geqslant x] \qquad \text{is trivial}$$

. . . 3 CCPs remaining

Table of Contents

Observation

- no official input format
- no official database
- unmaintained tool Ctrl
- weak confluence methods
- ...

ARI format

Automation of Rewriting Infrastructure

```
(format LCTRS)
(theory Ints)

(fun max (-> Int Int Int))

(rule (max x y) x :guard (>= x y))
(rule (max x y) y :guard (>= y x))
(rule (max x y) (max y x))
```

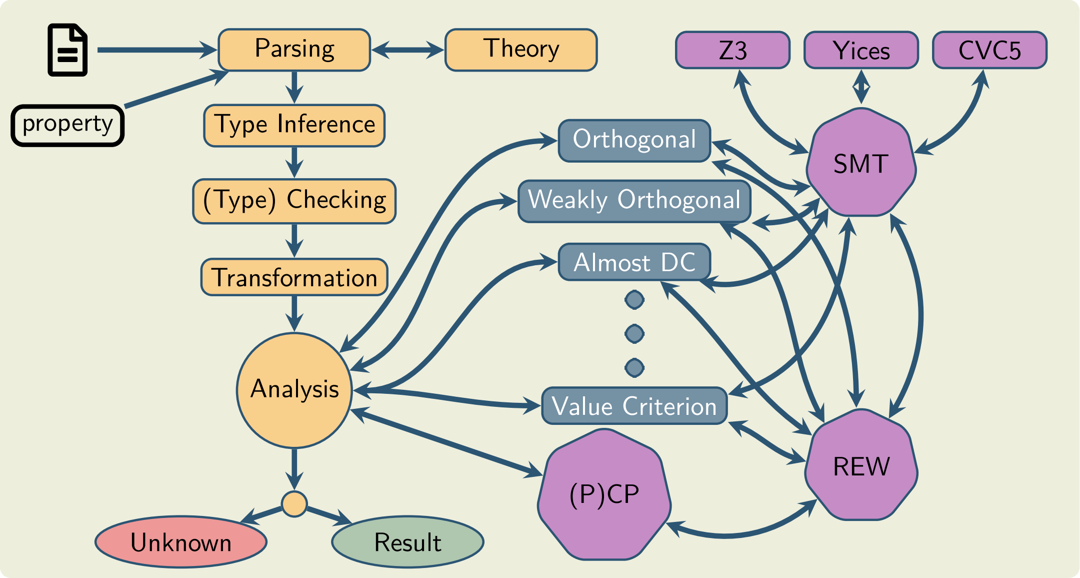
ARI database

- official format for LCTRSs
- 107 problems from the literature
- LCTRS competition category in CoCo

Constrained REwriting Software Tool (crest)

- implemented in Haskell (≈ 12000 LOC)
- open source
- confluence and termination analysis
- winner of LCTRS category in CoCo 2024

Simplified Overview of crest



Live Demo

Table of Contents

Confluence Experiments on 107 Problems

tool	✓	✗	?	solved	time (AVG)	time (total)
CRaris	58	0	49	54 %	0.13 s	14 s
crest	72	26	9	92 %	1.84 s	197 s
Ctrl	54	0	53	50 %	0.17 s	18 s
total solved	72	26	—	92 %	—	—

Termination

- program terminates on all inputs
- well-founded orders
- rules are terminating
- recursive-path order, value criterion, subterm criterion, ...

Termination Experiments on 107 Problems

tool	✓	?	solved	time (AVG)	time (total)
Cora	71	36	66 %	2.47 s	264 s
crest	74	33	69 %	0.15 s	16 s
Ctrl	74	33	69 %	0.96 s	103 s
total solved	78	—	73 %	—	—

Summary

- static program analysis via LCTRSs
- LCTRS confluence methods from TRSs
- automation of confluence and termination in crest
- experiments