Experimental Evaluation of Numerical Domains for Inferring Ranges

Gianluca Amato   Marco Rubino

Università di Chieti–Pescara

7th International Workshop on
Numerical and Symbolic Abstract Domains (NSAD 2017)

New York University, 2017-08-29

Comparing analyses performed with different

- abstract domains
- widening delays
- narrowing delays

to evaluate precision on

- interval constraints
- (octagonal constraints)

on a selection of linear transition systems.

Aim of this work

Comparing analyses performed with different

- abstract domains
- widening delays
- narrowing delays

to evaluate precision on

- interval constraints
- (octagonal constraints)

on a selection of linear transition systems.

Comparing analyses performed with different

- abstract domains
- widening delays
- narrowing delays

to evaluate precision on

- interval constraints
- (octagonal constraints)

on a selection of linear transition systems.

Outline

Benchmarks setting

Abstract domains

- Intervals [Cousot & Cousot '76]
  - $\pm x \leq b$
- Octagons [Miné '06]
  - $\pm x \pm y \leq b$
- Polyhedra H79 [Cousot & Halbwachs '78]
  - $\mathbf{a} \cdot \mathbf{x} \leq b$
  - standard widening in [Halbwachs '79]
- Polyhedra BHRZ03 [Cousot & Halbwachs '78]
  - $\mathbf{a} \cdot \mathbf{x} \leq b$
  - widening in [Bagnara, Hill, Ricci, Zaffanella '05]
- Parallelotopes [Amato & Scozzari '12]
  - $\mathbf{a} \cdot \mathbf{x} \leq b$
  - but all the **a**'s are linearly independent
- Par $\sqcap$ Int [Amato, Rubino, Scozzari '17]
  - (reduced) product of Parallelotopes and Intervals

Benchmark suite

- 108 linear transition systems
  - 102 from the ALICe benchmarks: `http://alice.cri.ensmp.fr/`
  - 6 from our previous works
- up to
  - 11 different locations
  - 4 loop heads
  - 10 variables
- quite different from Static Single Assignment form often generated by some program analyzers

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - https://github.com/jandom-devel/Jandom
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - https://github.com/jandom-devel/Jandom
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicitly represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

The analysis process

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

- all tests performed with the Jandom static analyzer
  - `https://github.com/jandom-devel/Jandom`
  - no particular optimization for any abstract domain
- analysis steps
  - l.t.s. are first transformed into equation systems
  - equations are solved using classic widening/narrowing based analysis
  - widening/narrowing on all loop heads
  - native implementation for Intervals and Parallelotopes
  - PPL for Octagons and Polyhedra
- assessing precision on intervals: we count the number of non-trivial bounds for each variable
  - bounds of the form $\pm x \leq 4$ and $\pm x \leq -\infty$ (while $\pm x \leq +\infty$ is trivial)
  - (obviously) not only explicity represented bounds, also entailed ones

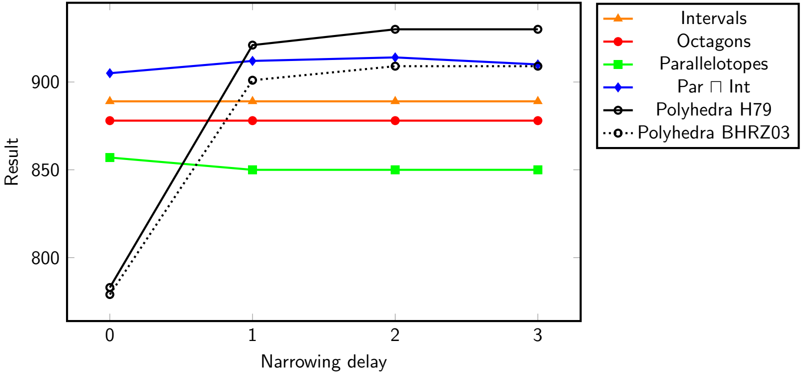
Benchmark results

Number of non-trivial bounds for variables

Narrowing delay \ Widening delay	Domains	0	1	2	3	4	5	6
0	Intervals	889	890	907	919	919	920	923
	Octagons	878	878	880	899	922	920	920
	Parallelotopes	847	848	876	883	884	884	884
	Par ⊓ Int	905	921	935	946	962	961	980
	Polyhedra H79	783	771	729	752	778	794	800
	Polyhedra BHRZ03	779	785	791	807	826	838	846
1	Intervals	889	890	907	919	919	920	923
	Octagons	878	878	880	899	922	920	920
	Parallelotopes	850	851	881	886	886	887	887
	Par ⊓ Int	912	926	940	953	963	966	983
	Polyhedra H79	921	909	863	879	885	889	893
	Polyhedra BHRZ03	901	907	909	912	921	923	927
2	Intervals	889	890	907	919	919	920	923
	Octagons	878	878	880	899	922	920	920
	Parallelotopes	850	851	881	886	886	887	887
	Par ⊓ Int	914	928	939	955	965	968	985
	Polyhedra H79	930	918	870	886	888	892	896
	Polyhedra BHRZ03	909	912	914	920	925	927	931
3	Intervals	889	890	907	919	919	920	923
	Octagons	878	878	880	899	922	920	920
	Parallelotopes	850	851	881	889	889	890	890
	Par ⊓ Int	910	925	942	952	962	965	982
	Polyhedra H79	930	918	870	886	888	892	896
	Polyhedra BHRZ03	909	912	914	920	925	927	931

Impact of delayed narrowing

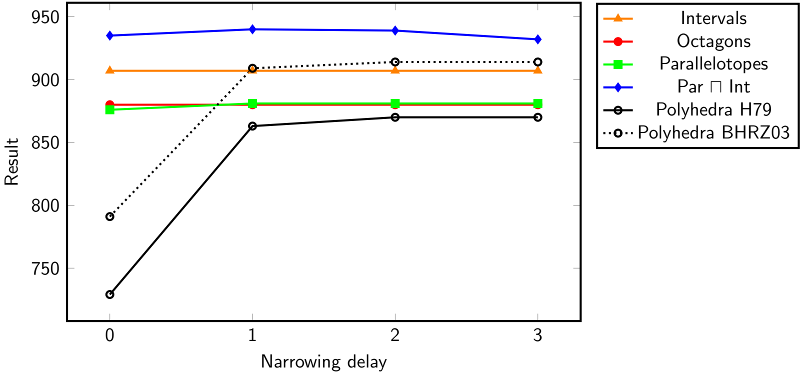
- All results with widening delay: 0



- Delayed narrowing has a limited impact on precision, with the exception of the Polyhedra domain

Impact of delayed narrowing

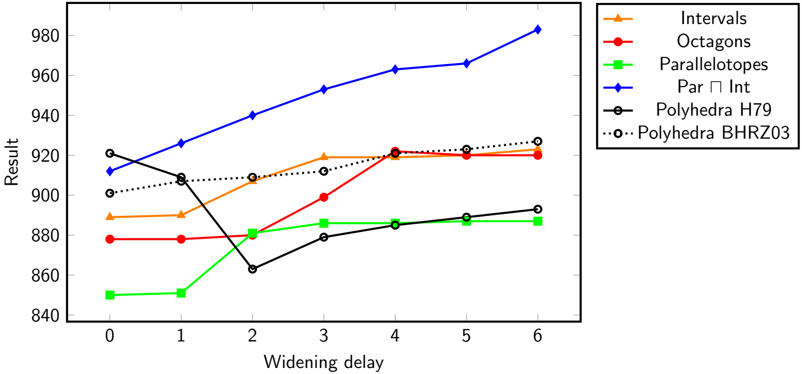
- All results with widening delay: 2



- Delayed narrowing has a limited impact on precision, with the exception of the Polyhedra domain

Impact of delayed widening

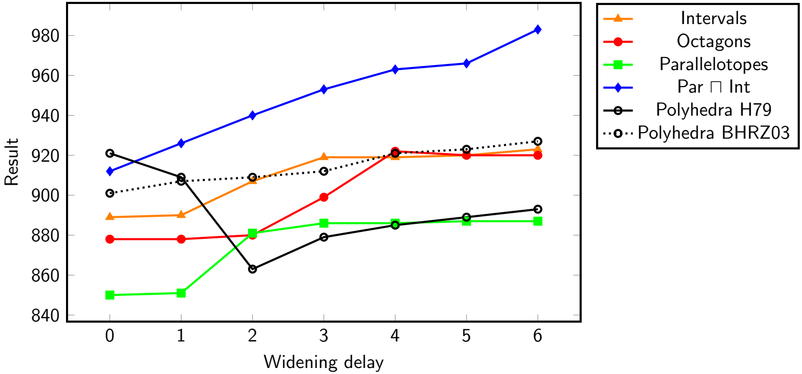
- All results with narrowing delay: 1



- Polyhedra H79 really suffers delayed widening
- Parallelotopes are the ones which benefit most from delay

Impact of the abstract domain

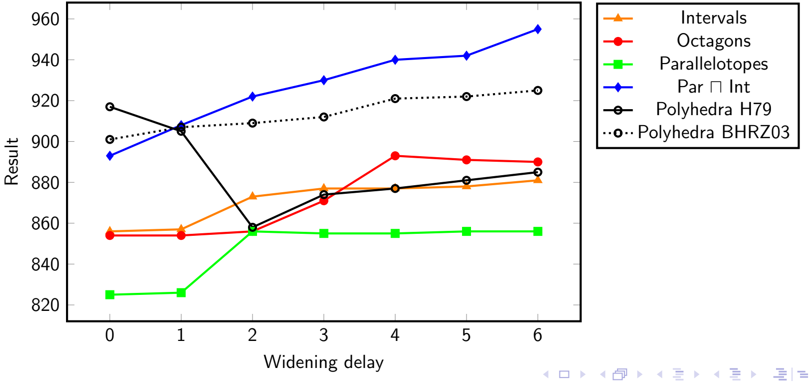
- All results with narrowing delay: 1



- with a low delay, Polyhedra H79 is the most precise
- overall, Par ⊓ Int is the most precise

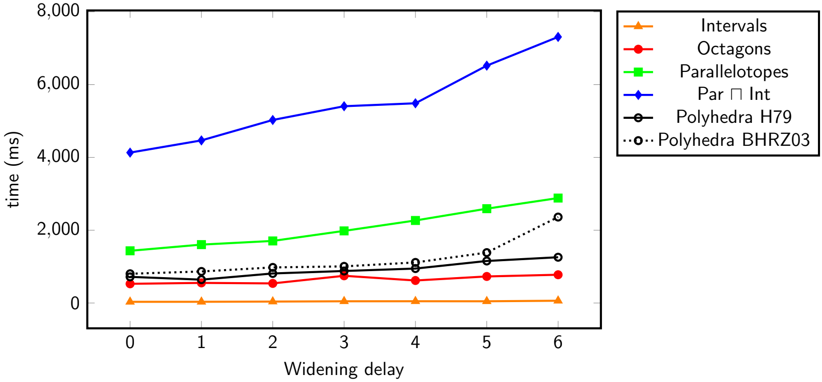
Impact of the abstract domain 2

- We count the number of non-trivial bounds which are no worse than those found by the other abstract domains.
  - example: if with Intervals and Octagons we get $x \leq 4$ while for all the other domains we get $x \leq 6$, we count this bound as one for intervals and octagons, zero for the others.



Performance

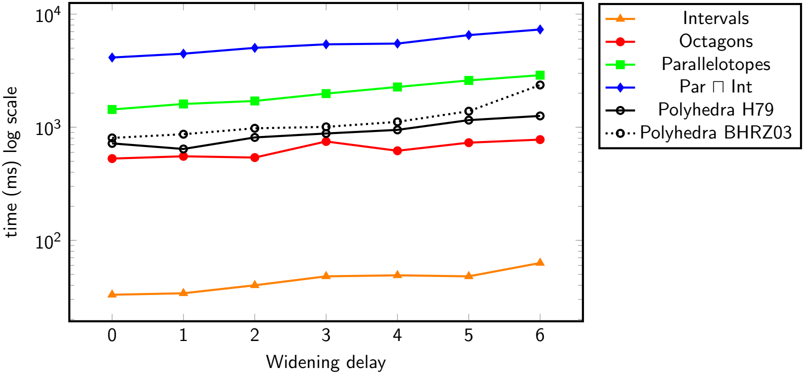
- All results with narrowing delay: 1



- Intervals are fast
- Parallelotopes are slow

Performance

- All results with narrowing delay: 1



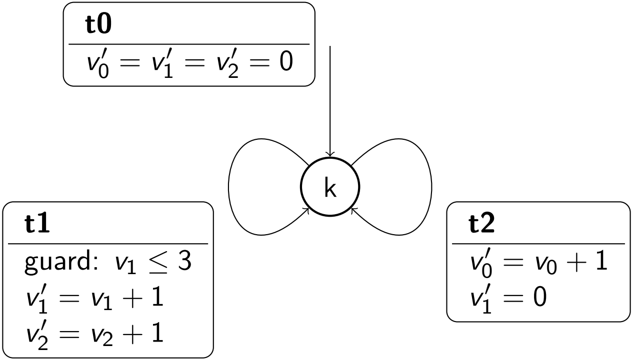
- Intervals are fast
- Parallelotopes are slow

Summary of results

1. delayed narrowing has small or no impact, with the exception of one delay step for polyhedra
2. delayed widening has generally a positive impact, but for Polyhedra H79
3. parallelotopes and intervals work well together, especially with high values for widening delay
4. parallelotopes are slow, but this is probably due to their implementation. A faster implementation is on the work.

A couple of examples in detail

- `halbwachs7.fst` from the ALICe benchmarks.



- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

Comparing the cases for delay 0 and 1

- Ascending chain only with Polyhedra H79

Delay 0

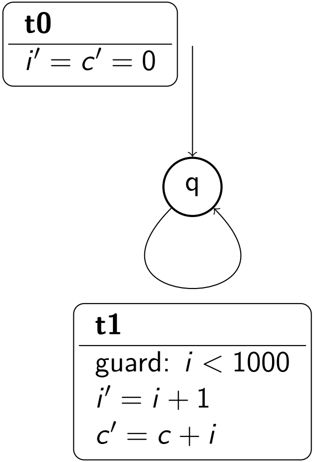
#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$
2	$v_0 \geq 0, v_1 \geq 0, v_2 \geq v_1$

Delay 1

#	prop.
0	$v_0 = v_1 = v_2 = 0$
1	$v_0 \geq 0, v_1 \geq 0, v_2 = v_1$ $v_0 + v_2 \leq 1$
2	$v_1 \geq 0, v_2 \geq v_1$ $v_0 + v_1 - v_2 \geq 0$ ($v_0 \geq 0$ is implied)
3	$v_1 \geq 0, v_2 \geq v_1$

A bad linear transition system

- `slam_bad.fst` from the ALICe benchmarks.



Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

Comparing intervals and octagons

Intervals

#	prop.
0	$i = c = 0$
1	$0 \leq i, c = 0$
2	$0 \leq i, 0 \leq c$

Octagons

#	prop.
0	$i = c = 0$
1	$0 \leq i \leq 1, c = 0$
2	$0 \leq c \leq 1$ $i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
3	$0 \leq c, i - 1 \leq c \leq i$ ($0 \leq i$ is implied)
4	$0 \leq c, i - 1 \leq c$
5	$i - 1 \leq c$
6	no constraints

APRON

- The same problem does not happen in APRON.
- Tested with the following Interproc program

```
var i:int, c:int;
begin
  i = 0;
  c = 0;
  while true do
    if (i < 1000) then
      c = c + i;
      i = i + 1;
    endif;
  done;
end
```

- Interproc with octagons finds the invariant $0 \leq i, 0 \leq c$.

PPL vs APRON

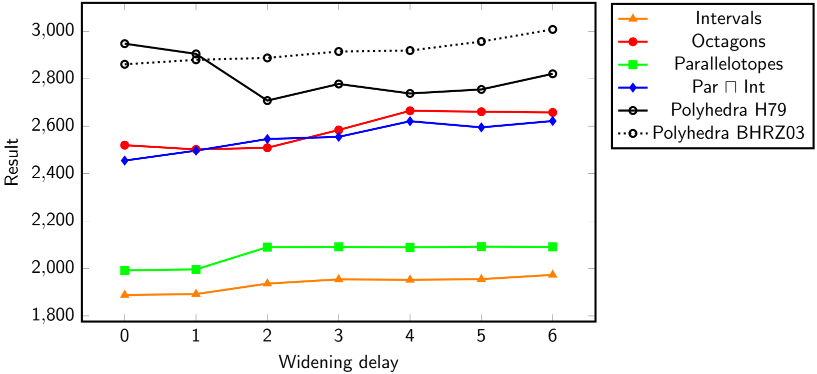
PPL and APRON use different implementations for octagons.

- APRON use (mostly) closed sets of octagonal constrains:
  all entailed constraints are made explicit
- PPL use reduced sets of octagonal constrains:
  there are no entailed constraints

Actually, this problem in the precision of widening is discussed in [Bagnara, Hill, Mazzi, Zaffanella '05], which suggests to use a variant of widening with threshold.

Conclusions

- Counting "no worse than other" bounds for octagonal constrains
- All results with narrowing delay: 1



- Poyhedra BHRZ03 is the most precise almost always.
- Par ⊓ Int has lost its top spot, but what about Par ⊓ Oct ?

Conclusions and future work

- no revolutionary results here but
  - confirmation of expected results
  - some surprise (bad effect of delayed widening on H79)
  - if we want to be precise on intervals, we need to adapt our domains to better propagate ranges
- future work
  - bigger test suite
    - Java programs using the bytecode analyzer of Jandom
  - varying other parameters
    - widening with threshold
    - guided abstract interpretation
    - localized widening/narrowing
    - warrowing

Bibliography

Bibliography

- G. Amato and F. Scozzari, *The abstract domain of parallelotopes*, NSAD 2012.
- G. Amato, M. Rubino and F. Scozzari, *Inferring linear invariants with parallelotopes*, Science of Computer Programming, to appear (2017).
- R. Bagnara, P. M. Hill, E. Ricci and E. Zaffanella, *Precise widening operators for convex polyhedra*, Science of Computer Programming 58 (2005).
- R. Bagnara, P. M. Hill, E. Mazzi, E. Zaffanella, *Widening Operators for Weakly-Relational Numeric Abstractions*, SAS 2005.
- P. Cousot and R. Cousot, *Static determination of dynamic properties of programs*, in: *Proceedings of the Second International Symposium on Programming* (1976).
- P. Cousot and N. Halbwachs, *Automatic discovery of linear restraints among variables of a program*, POPL 1978.
- A. Miné, *The octagon abstract domain*, Higher-Order and Symbolic Computation 19 (2006).