Scalable Analysis of Linear Systems using Mathematical Programming

wslee @ ropas

Reference

- **S. Sankaranarayanan, H. Sipma, and Z. Manna.**
- ***Scalable analysis of linear systems using mathematical programming.*** **VMCAI'05.**

Contents

- Motivation
- Previous related work
- Concrete domain
- Abstract domain
- Experiment
- Conclusion

Motivation

- Discover invariant relationships between the variables of a system.

Previous related work

⦿Polyhedral analysis.

- All the linear inequalities over all the variables.
- Precise.
- Time & Space complexity : $O(x^n)$

⦿Interval domain or DBM based approach.

- $a \le x_i \le b$, $x_i - x_j \le c$

Octagon domain-based approach.

- $\pm x_i \pm x_j \leq c$
- Scalable

Octahedral analysis.

- $a_1 x_1 + \ldots a_n x_n \leq c \quad (a_i = \{0,1\})$

- Computing invariants on an abstract domain less powerful than polyhedra.
- But more general than intervals, octagons and octahedra.
- (By means of LP solver and chosen template constraint matrices.)

Preliminaries

- Linear assertions
- Farkas Lemma
- Linear Programming
- Linear Transition Systems
- Inductive Assertion Maps

Linear Assertions

- A finite conjunction of linear inequalities.

$$\varphi : \begin{bmatrix} a_{11}x_1 + \ldots + a_{1n}x_n + b_1 \geq 0 \wedge \\ \ldots \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n + b_m \geq 0 \end{bmatrix}$$

- The assertion can be written in matrix form as

$$\mathbf{Ax} + \mathbf{b} \geq \mathbf{0} \quad (\mathbf{A} : \mathrm{m} \times \mathrm{n},\ \mathbf{x} : \mathrm{n} \times 1,\ \mathbf{b} : \mathrm{m} \times 1)$$

Farkas Lemma

- Consider the linear assertion

$$\boxed{\varphi:\ \mathbf{Ax} + \mathbf{b} \geq \mathbf{0}}$$

- If φ is satisfiable, then

$$\boxed{\begin{array}{l}\mathbf{c}^T\mathbf{x}+\mathbf{d}\geq 0,\ \text{there exists } \boldsymbol{\lambda}\geq\mathbf{0} \text{ such that} \\ \qquad\qquad \mathbf{A}^T\boldsymbol{\lambda}=\mathbf{c}\ \text{ and }\ \mathbf{b}^T\boldsymbol{\lambda}\leq\mathbf{d}.\end{array}}$$

- If φ is unsatisfiable, then

$$\boxed{\begin{array}{l}\text{there exists } \boldsymbol{\lambda}\geq\mathbf{0} \text{ such that} \\ \qquad\qquad \mathbf{A}^T\boldsymbol{\lambda}=\mathbf{0}\ \text{ and }\ \mathbf{b}^T\boldsymbol{\lambda}\leq -1.\end{array}}$$

Linear Programming

- **To determine the solution of φ for which *objective function f* is minimal.**

$$f : \mathbf{b}^T \mathbf{x}$$

- **Possible three results:**
  - An optimal solution.
  - Non-optimal solutions

    (f is unbounded in φ.)
  - φ has no solutions.

Linear Transition Systems

- $S : \langle L, \Gamma, l_0, \Theta \rangle$
  - L : a set of locations.
  - Γ : a set of transitions. Transition $\tau : \langle l_i, l_j, \rho_\tau \rangle$
    - li : pre-location
    - lj : post-location
    - ρ_τ : a linear assertion over $V \cup V'$
  - $l_0 \in L$: the initial location.
  - Θ : a linear assertion specifying the initial condition.

integer i,j

(where i = 2 ∧ j = 0)

l_0 : while true do

i := i + 4

l_1 : or

(i,j) := (i + 2, j + 1)

$$L = \{l_o, l_1\},\ V = \{i, j\},$$

$$\Theta : (i = 2 \wedge j = 0),\ \mathrm{T} = \{\tau_0, \tau_1, \tau_2\},$$

$$\tau_0 : \langle l_0, l_1, true \rangle$$

$$\tau_1 : \langle l_1, l_0, (i' = i + 4 \wedge j' = j) \rangle$$

$$\tau_2 : \langle l_1, l_0, (i' = i + 2 \wedge j' = j + 1) \rangle$$

Inductive Assertion Maps

⦿Inductive assertion

- An assertion at a program location if it holds the first time the location is reached and is preserved under every cycle back to the location.

⦿Inductive assertion maps (η)

- Initial : $\Theta \vDash \eta(l_0)$

- Consecution :

$$\text{For each transition } \tau : \langle l_i, l_j, \rho_\tau \rangle, \eta(l_i) \wedge \rho_\tau \vDash \eta(l_j)'$$

- **Any inductive assertion is also an invariant assertion.**
- **Any inductive assertion map is also an invariant map.**
- **Therefore, our purpose is finding an inductive assertion map.**

Propagation-based analysis

- **Assertion map η : loc → assertion**

$$F(X) = \Theta \vee X \vee \bigvee_{\tau \in \mathrm{T}} post(\tau, X)$$

$$post(\tau, \varphi) : \exists V_0.(\varphi(V_0) \wedge \rho_\tau(V_0, V))$$

$$\begin{pmatrix} post & : & transition \times \eta & \rightarrow & 2^{\Sigma} \\ F & : & \eta & \rightarrow & \eta \end{pmatrix}$$

- **Objective : to find *fix F* starting from *F(false)***

Need to abstract

- 1) *F(false), F^2(false)* … may not converge in finite number of steps.

- 2) Detection of convergence may be undecidable.

$$F^{n+1}(false) \subseteq F^{n}(false)$$

Abstract domain

- Using Galois connection.

$$2^{\Sigma} \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \Sigma_A$$

-

- Objective : to find fix F_A s.t.

$$F_A(X) = \Theta_A \sqcup X \sqcup \bigsqcup_{\tau \in T} post_A(\tau, X)$$

$$fix\ F \vDash \gamma(fix\ F_A)$$

- Σ_A consists of polyhedra of a fixed shape for a given set of variables x.
- The shape is fixed by an m*n template constraint matrix (TCM) T.
- Σ_T contains c = $\langle c_1, \ldots, c_m \rangle$ ($c_i \in R \cup \{\infty, -\infty\}$)
- c in Σ_T represents the set of states described by the set of constraints

$$T\mathbf{x} + \mathbf{c} \geq 0$$

⦿Concretization function

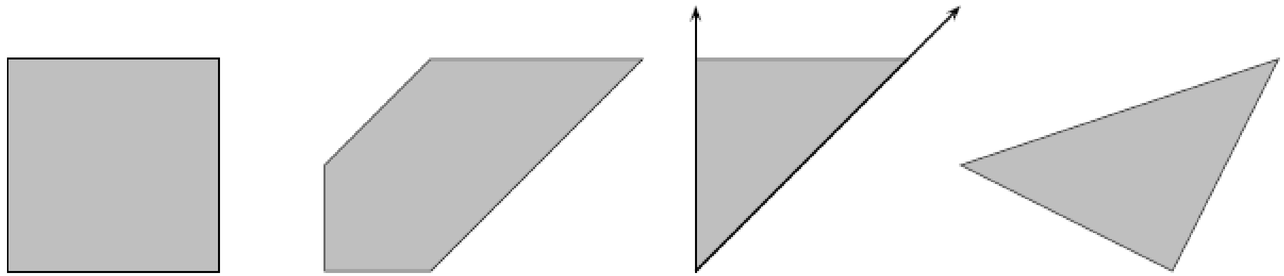
$$\gamma_T(c) \equiv \begin{cases} \mathit{false} & \text{if } \exists c_i = -\infty \text{ or } \mathrm{c} = \mathrm{c}_\perp, \\ \mathit{true} & \text{if } \mathrm{c} = \mathrm{c}_\top \\ \bigwedge_{i \;\; s.t.\; c_i \neq \infty} (T_i\mathbf{x} + c_i \geq 0) & \text{otherwise.} \end{cases}$$

Ex).

$$\mathbf{c} \; : \; \langle \infty, 2, 3, \infty, 5, 1 \rangle$$

$$T = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \text{ representing the template assertions } \begin{bmatrix} x + c_1 \geq 0 \\ -x + c_2 \geq 0 \\ y + c_3 \geq 0 \\ -y + c_4 \geq 0 \\ -x + y + c_5 \geq 0 \\ x - y + c_6 \geq 0 \end{bmatrix}$$

$$\gamma_T(\mathbf{c}) \; : \; [-x + 2 \geq 0 \wedge y + 3 \geq 0 \wedge -x + y + 5 \geq 0 \wedge x - y + 1 \geq 0]$$



$\langle 1,1,1,1,\infty,\infty\rangle$ (a) $\langle 1,\infty,1,4,3,3\rangle$ (b) $\langle 1,\infty,\infty,\infty,3,\infty\rangle$ (c) $\langle ?\rangle$ (d)

- **All the vectors can be concretized but (d).**

Abstraction function

For a linear assertion describing sets of states

$$\varphi : \mathbf{Ax} + \mathbf{b} \geq \mathbf{0}$$

$$\alpha_T : 2^{\Sigma}(= \varphi) \;\rightarrow\; \mathbf{c}\;(\in\; \Sigma_T)$$

$$\mathbf{Ax} + \mathbf{b} \geq \mathbf{0} \;\vDash\; T\mathbf{x} + \mathbf{c} \geq \mathbf{0}$$

$$\mathbf{Ax} + \mathbf{b} \geq \mathbf{0} \;\vDash\; T_i\mathbf{x} + c_i \geq \mathbf{0}$$

$$\Leftrightarrow (\exists \boldsymbol{\lambda} \geq 0)\mathbf{A}^T\boldsymbol{\lambda} = T_i \wedge \mathbf{b}^T\boldsymbol{\lambda} \leq c_i$$

$\psi : \boldsymbol{\lambda} \geq 0 \wedge A^T\boldsymbol{\lambda} = T_i$ with objective function $\mathbf{b}^T\boldsymbol{\lambda}$

Abstraction function

$$\varphi : \mathbf{Ax} + \mathbf{b} \geq \mathbf{0}$$

$$\text{given TCM T },\ \alpha(\varphi) = \mathbf{c} = \langle c_1, \ldots, c_m \rangle$$

$$c_i = \begin{cases} -\infty & \text{if } \varphi \text{ is unsatisfiable} \\ \min.\ \mathbf{b}^T \boldsymbol{\lambda},\ s.t.\ \underbrace{\boldsymbol{\lambda} \geq 0 \ \wedge\ \mathbf{A}^T \boldsymbol{\lambda} = T}_{\Psi_i} & \text{if } \Psi_i \text{ is feasible.} \\ \infty & \text{if } \Psi_i \text{ is infeasible.} \end{cases}$$

- Canonicalization (Eliminating redundancy)
  - $-1 \le x, y \le 1 \wedge -2 \le x - y \le 2$

    $\Leftrightarrow -1 \le x, y \le 1 \wedge -3 \le x - y \le 3$

    ...

    $\Leftrightarrow -1 \le x, y \le 1 \wedge a \le x - y \le b$

    $\therefore\ [\langle 1,1,1,1,2,2 \rangle] = \{\langle 1,1,1,1,a,b \rangle \mid a, b \ge 2\}$

  - Given an equivalence class [c],

$$\mathrm{can}(\mathbf{c}) = \alpha_T(\gamma_T(\mathbf{c}))$$

Post condition operator

Given $\tau : \langle l_i, l_j, \rho_\tau \rangle$,

$$post(\eta(l_i), \tau) = \begin{cases} \bot & \eta(l_i) = \bot \\ \alpha_j(\gamma_i(\eta(l_i) \wedge \rho_\tau)) & \text{otherwise} \end{cases}$$

Using the postcondition the map at step i > 0 is updated as follows:

$$\eta^{i+1}(l_n) = \eta^i(l_n) \sqcup \left(\bigsqcup_{\tau:\langle l_m, l_n, rho \rangle} post(\eta^i(l_m), \tau) \right)$$

Template formation

- User defined patterns
  - "%i + 2*%j + 3*%k" generates all constraints of the form

$$x_i + 2x_j + 3x_k + b_{ijk} \geq 0$$

- Automatically derived
  - From condition expressions in program.

This corresponds to shape-corpus in our project.

Experiment

Program			Template		Statistics				
name	$\|L\|$	$\|T\|$	#t	#s	t(sec)	t_{lp}(sec)	# LPS	#avg.	#dim.
MCC91 (3)	1	2	11	0	0.05	0.01	227	1.5	15 (20)
TRAINHPR97(3)	4	12	58	3	0.1	0.02	673	0.9	18(25)
BERKELEY(4)	1	3	63	16	0.23	0.11	1,632	1.36	64(96)
DRAGON(5)	1	12	129	157	3.94	2.38	11,426	3.23	202 (298)
HEAPSORT(5)	1	4	33	24	0.34	0.13	1,751	2.45	75(90)
EFM(6)	1	5	506	461	7.65	2.36	10,872	0.69	359(981)
LIFO(7)	1	10	85	79	1.87	0.91	5,401	3.37	141 (174)
CARS-MIDPT(7)	1	2	101	324	3.72	2.21	4,641	6.23	154(329)
BARBER(8)	1	12	128	0	1.97	0.83	9,210	1.96	124(141)
SWIM-POOL(9)	1	6	104	0	0.56	0.27	2,710	2.11	97(118)
TTP(9)	4	20	3,555	127	62.8	40.9	61,263	4.41	574(1032)
REQ-GRANT(11)	1	8	221	18	2.96	1.41	8,635	2.10	241(255)
CONSPROT(12)	2	14	533	40	4.88	2.00	12,487	1.83	266(286)
CSM(13)	1	8	313	73	9.65	5.21	14,890	3.69	380(414)
C-PJAVA(16)	1	14	453	93	35.16	15.19	33,288	5.00	433(567)
CONSPROD(18)	1	14	529	96	38.72	19.43	35,797	5.17	468(663)
INCDEC(32)	1	28	961	267	287.54	110.27	103,841	6.57	877(1294)
MESH2X2(32)	1	32	438	0	43.9	17.5	52,622	4.53	390(506)
BIGJAVA(44)	1	37	864	376	331.98	117.68	122,643	5.25	1018 (1280)
MESH3X2(52)	1	54	1133	0	432.85	192.15	216,600	6.70	930(1241)

- **Complexity:**

$$O(km^2|L||T|)$$

Conclusion

- This work is less powerful than that of polyhedra, but more general than intervals, octagons, and octahedra.
- The power of LP solver makes this work time and space-efficient alternative to polyhedra.
- A wiser choice of templates(TCM) improves scalability & precision.

Lessons

- **A wiser choice of templates(TCM) improves scalability & precision.**
  - Shows the possibility of success with corpus-based approach.

- **A choice of templates is conducted both statically & dynamically.**
  - We should consider this.