The Polyranking Principle

Aaron R. Bradley, Zohar Manna, and Henny B. Sipma

Computer Science Department

Stanford University

Overview

Goal:

1. Definition of **polyranking** functions
2. Complete **synthesis** method for polyranking functions
    - linear, over linear loops
    - lexicographic

Outline:

1. Example
2. Polyranking functions
3. Synthesis

Example

```
int x, y, k, N
Θ :  k > 0
while (x ≤ N) do
    if  (...)
    then   x := (x + y)/2          // integer division
           y := y + k
    else   x := x + 1
           y := y + consume()      // consume() ≥ 0
done
```

Abstract program:

$$\Theta : \; k \geq 1$$
$$\tau_1 : \; x \leq N \;\wedge\; 2x' \geq x + y - 1 \;\wedge\; y' = y + k \;\wedge\; k' = k \;\wedge\; N' = N$$
$$\tau_2 : \; x \leq N \;\wedge\; x' = x + 1 \;\wedge\; y' \geq y \;\wedge\; k' = k \;\wedge\; N' = N$$

for $x, y, k, N \in \mathbb{R}$

Example

$\texttt{int}\ x, y, k, N$

$\Theta:\ k > 0$

while $(x \leq N)$ **do**

  if $(\ldots)$

  then $x := (x + y)/2$

    $y := y + k$

  else $x := x + 1$

    $y := y + consume()$

done

x	y	k	N	$N - x$
6	0	1	7	1
3	1	1	7	4
2	2	1	7	5
2	3	1	7	5
2	4	1	7	5
3	5	1	7	4
4	6	1	7	3
5	7	1	7	2
6	8	1	7	1
7	9	1	7	0
8	10	1	7	−1

$\Theta:\ k \geq 1$

$\tau_1:\ x \leq N\ \wedge\ 2x' \geq x + y - 1\ \wedge\ y' = y + k\ \wedge\ k' = k\ \wedge\ N' = N$

$\tau_2:\ x \leq N\ \wedge\ x' = x + 1\ \wedge\ y' \geq y\ \wedge\ k' = k\ \wedge\ N' = N$

Example: Termination

$\Theta: \; k \geq 1$
$\tau_1: \; x \leq N \;\wedge\; 2x' \geq x + y - 1 \;\wedge\; y' = y + k \;\wedge\; k' = k \;\wedge\; N' = N$
$\tau_2: \; x \leq N \;\wedge\; x' = x + 1 \;\wedge\; y' \geq y \;\wedge\; k' = k \;\wedge\; N' = N$

Invariant: $k \geq 1$

Choose **polyranking function**

$$\delta: \; N - x$$

Bounded $(N - x)$:

- $\tau_1: \; x \leq N \;\rightarrow\; N - x \geq 0$
- $\tau_2: \; x \leq N \;\rightarrow\; N - x \geq 0$

$\Rightarrow$ If $N - x < 0$, then the loop exited.

Example: Termination

Polyranking $(N - x)$:

- τ_2 decreases $N - x$:

$$\Delta_{\tau_2}(N - x) = (N' - x') - (N - x) = (N - x - 1) - (N - x)$$
$$= \boxed{-1 \ < \ 0}$$

- τ_1 **eventually** decreases $N - x$:

$$\Delta_{\tau_1}(N - x) = (N' - x') - (N - x) = (2N - 2x') - (2N - 2x)$$
$$\leq (2N - (x + y - 1)) - (2N - 2x) = x - y + 1$$
$$\leq \boxed{N - y + 1}$$

$$\Delta_{\tau_1}(N - y + 1) = (N' - y' + 1) - (N - y + 1) = \boxed{-1 \ < \ 0}$$

$$\Delta_{\tau_2}(N - y + 1) = (N' - y' + 1) - (N - y + 1) \ \boxed{\leq \ 0}$$

Example: Summary

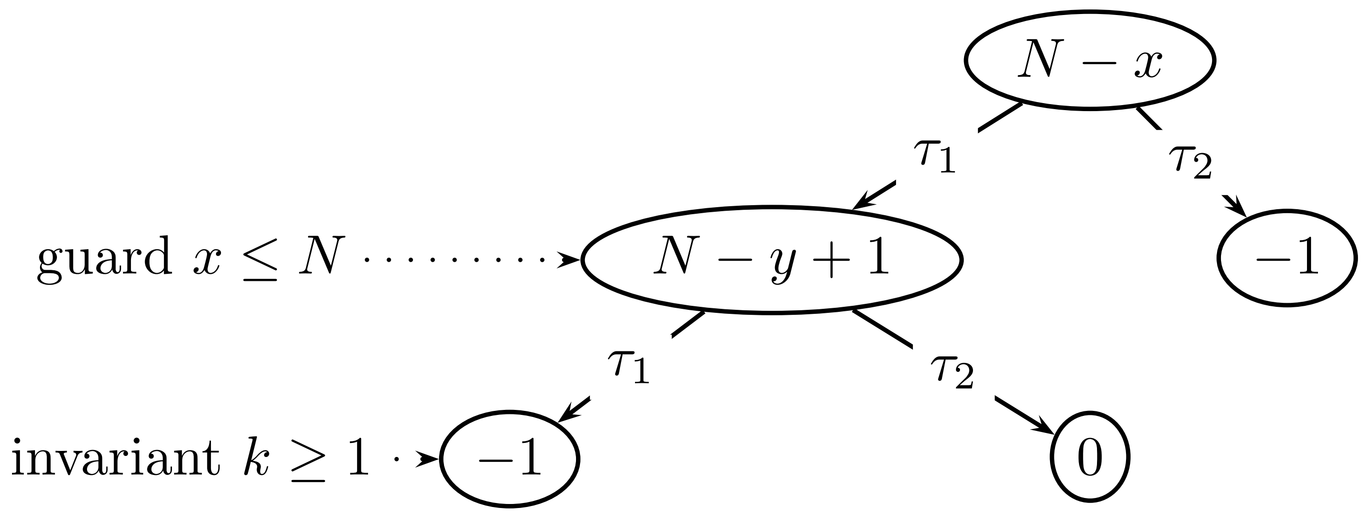
$\Theta : \; k \geq 1$

$\tau_1 : \; x \leq N \;\wedge\; 2x' \geq x + y - 1 \;\wedge\; y' = y + k \;\wedge\; k' = k \;\wedge\; N' = N$

$\tau_2 : \; x \leq N \;\wedge\; x' = x + 1 \;\wedge\; y' \geq y \;\wedge\; k' = k \;\wedge\; N' = N$

Bounded $x \leq N \;\rightarrow\; N - x \geq 0$

Polyranking



$\Rightarrow$ Terminates on all input.

Outline

1. **Loop abstraction & definitions**
2. Related work
3. Polyranking
4. Synthesis
5. Conclusion

Loops

Loop Abstraction: $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$

- **variables** $\mathcal{V}$ range over $\mathbb{R}$
- **initial condition** Θ is polynomial assertion over $\mathcal{V}$
- **transitions** $\tau \in \mathcal{T}$ are polynomial assertions $\tau(\mathcal{V}, \mathcal{V}')$ over $\mathcal{V} \cup \mathcal{V}'$

$$\begin{aligned}
\Theta &: \; k \geq 1 \\
\tau_1 &: \; x \leq N \;\wedge\; 2x' \geq x + y - 1 \;\wedge\; y' = y + k \;\wedge\; k' = k \;\wedge\; N' = N \\
\tau_2 &: \; x \leq N \;\wedge\; x' = x + 1 \;\wedge\; y' \geq y \;\wedge\; k' = k \;\wedge\; N' = N
\end{aligned}$$

Loop Invariant

Loop Validity:

Assertion φ is **valid over loop** $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$

$$L \models \varphi$$

if φ holds on all **reachable states** $\mathcal{S}_L$ of L.

In practice, replace "$L \models$" with **loop invariants**.

φ is an inductive **polynomial invariant** if

(Initiation)

$$(\forall \mathcal{V})[\Theta(\mathcal{V}) \;\rightarrow\; \varphi(\mathcal{V})]$$

(Consecution) $(\forall \tau \in \mathcal{T})$

$$(\forall \mathcal{V}, \mathcal{V}')[\varphi(\mathcal{V}) \;\wedge\; \tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \varphi(\mathcal{V}')]$$

Ranking Function

Consider loop $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$.

$\delta : \mathcal{S}_L \rightarrow \mathbb{R}$ is a **ranking function** of L if

(Bounded) $(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \delta(\mathcal{V}) \geq 0$$

(Ranking) $(\exists \epsilon > 0)(\forall \tau \in \mathcal{T})$

$$L \models \quad \tau(\mathcal{V}, \mathcal{V}') \;\rightarrow\; \delta(\mathcal{V}') \leq \delta(\mathcal{V}) - \epsilon$$

Outline

1. Loop abstraction & definitions
2. **Related work**
3. Polyranking
4. Synthesis
5. Conclusion

Related Work: Synthesis of Measures

Katz & Manna 1975

Generate and solve constraint systems for linear ranking functions, over linear loops with affine assignments.

Colón & Sipma 2001, 2002

Synthesis of linear ranking functions for linear loops.

Colón, Sankaranarayanan & Sipma 2003

Constraint-based linear invariant generation.

Podelski & Rybalchenko 2004

Complete method for linear ranking functions over linear loops with one transition and without an initial condition.

Bradley, Manna & Sipma 2005

Complete method for lexicographic linear ranking functions over linear loops.

Related Work: Synthesis of Measures

Cousot 2005

Incomplete but efficient method for synthesis of polynomial ranking functions over polynomial loops.

Bradley, Manna & Sipma 2005

Incomplete method for synthesis of polynomial polyranking functions over polynomial loops with assignments.

Our contribution:

- Complete formulation of polynomial polyranking functions over assertional polynomial loops.
- Complete and efficient method for synthesis of lexicographic linear polyranking functions over linear loops.

Related Work: Verification Frameworks

Verification Diagrams

- State-based abstraction of state-space.
- Syntactic loops can result in multiple SCCs in diagram.
- Each SCC requires termination proof.
- [**Manna, Browne, Sipma & Uribe 1998**]

Transition Invariants

- Relation-based abstraction of transition relation.
- Syntactic loops can result in disjunctive transition invariant.
- Each disjunct requires termination proof.
- [**Podelski & Rybalchenko 2004**], [**Dershowitz, Lindenstrauss, Sagiv & Serebrenik 2001**], [**Lee, Jones & Ben-Amram 2001**], [**Codish, Genaim, Bruynooghe, Gallagher & Vanhoof 2003**]

Outline

1. Loop abstraction & definitions
2. Related work
3. **Polyranking**
4. Synthesis
5. Conclusion

Eventually Negative

Consider $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$.

$E(\mathcal{V})$ is **eventually negative by** $\mathcal{A} \subseteq \overline{\mathcal{T}}$ (for $\mathcal{A} \neq \emptyset$, $\overline{\mathcal{T}} \subseteq \mathcal{T}$) if

(Negative) $(\exists \epsilon > 0)[L \models\ E(\mathcal{V}) \leq -\epsilon]$

(Eventually Negative) $(\forall\ \tau \in \overline{\mathcal{T}})$

(Nonincreasing)

(*not active, not increasing*)

1. $\tau \in \overline{\mathcal{T}} - \mathcal{A}$
2. $L \models\ \tau(\mathcal{V}, \mathcal{V}') \rightarrow E(\mathcal{V}') - E(\mathcal{V}) \leq 0$

(Eventually Decreasing)

(*difference is bounded by* $F(\mathcal{V})$)

1. $L \models\ \tau(\mathcal{V}, \mathcal{V}') \rightarrow E(\mathcal{V}') - E(\mathcal{V}) \leq F(\mathcal{V})$
2. $F(\mathcal{V})$ is **eventually negative by** $\{\tau\} \subseteq \overline{\mathcal{T}}$

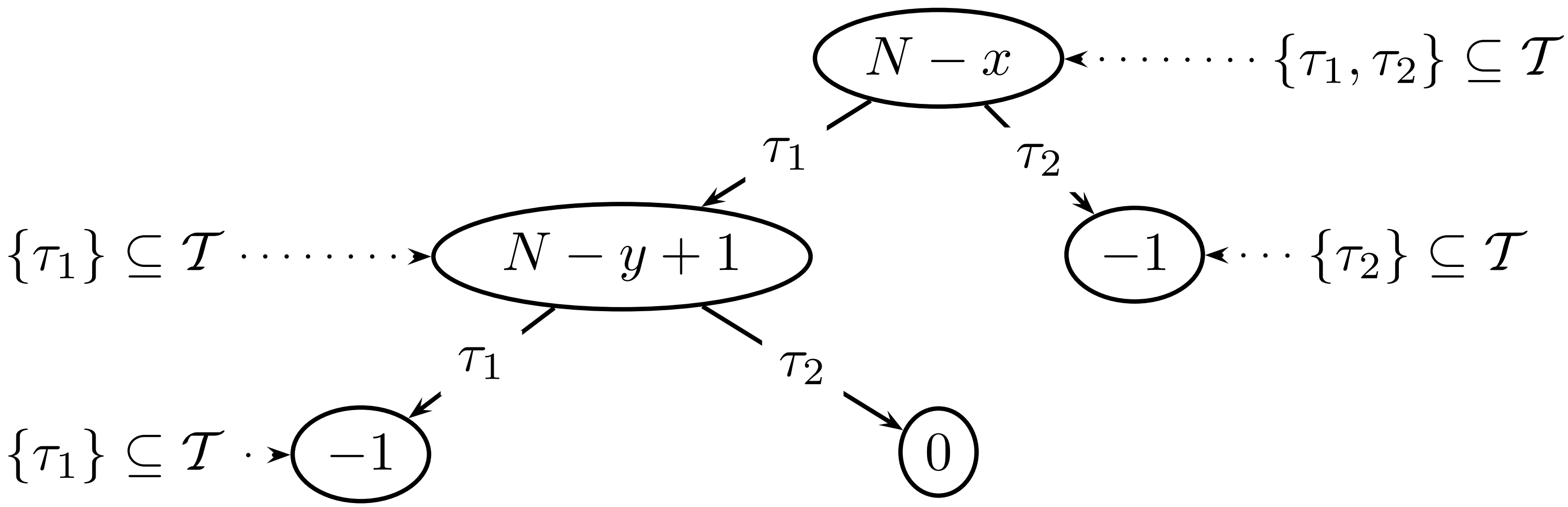
Active transitions $\mathcal{A}$ make progress. $\overline{\mathcal{T}} - \mathcal{A}$ do not interfere.

EN-Tree

Graphical representation of **eventually negative**.

$$\Theta: \; k \geq 1$$
$$\tau_1: \; x \leq N \;\wedge\; 2x' \geq x + y - 1 \;\wedge\; y' = y + k \;\wedge\; k' = k \;\wedge\; N' = N$$
$$\tau_2: \; x \leq N \;\wedge\; x' = x + 1 \;\wedge\; y' \geq y \;\wedge\; k' = k \;\wedge\; N' = N$$

$N - x$ is **eventually negative by** $\{\tau_1, \tau_2\} \subseteq \{\tau_1, \tau_2\}$



Polyranking Function

Consider $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$.

$$\delta(\mathcal{V})$$

is a **polyranking function** if

Bounded $(\forall \tau \in \mathcal{T})$

$$L \models \tau(\mathcal{V}, \mathcal{V}') \rightarrow \delta(\mathcal{V}) \geq 0$$

Polyranking

$$\delta(\mathcal{V}) \text{ is eventually negative by } \mathcal{T} \subseteq \mathcal{T}$$

Theorem If $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$ has a polyranking function, then it always terminates.

Lexicographic Polyranking Function

Consider $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$.

$$\delta : \ \langle r_1(\mathcal{V}), \ldots, r_\ell(\mathcal{V}) \rangle$$

is a **lexicographic polyranking function** if for some

$$\pi : \ \mathcal{T} \rightarrow \{1, \ldots, \ell\}$$

Bounded $(\forall \tau \in \mathcal{T})$

$$L \models \ \tau(\mathcal{V}, \mathcal{V}') \ \rightarrow \ r_{\pi(\tau)}(\mathcal{V}) \geq 0$$

Polyranking $(\forall i \in \{1, \ldots, \ell\})$

$r_i(\mathcal{V})$ is eventually negative by $\{\tau : \pi(\tau) = i\} \subseteq \{\tau : \pi(\tau) \geq i\}$

Theorem If $L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$ has a lexicographic polyranking function, then it always terminates.

Outline

1. Loop abstraction & definitions
2. Related work
3. Polyranking
4. **Synthesis**
5. Conclusion

Linear Loops

Consider variables $\mathcal{V} = \{x_1, x_2, \ldots, x_m\}$.

homogenous vector:

$$\mathbf{x} = (x_1, \ldots, x_m, 1)^{\mathrm{T}}$$

linear assertion:

$$\begin{array}{rcl}
\mathbf{A}\mathbf{x} & \geq & 0 \\
\begin{bmatrix} a_{1,1} & \cdots & a_{1,m} & a_{1,m+1} \\ & & \vdots & \\ a_{k,1} & \cdots & a_{k,m} & a_{k,m+1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 1 \end{bmatrix} & \geq & \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\
\bigwedge_{i \in \{1,\ldots,k\}} (a_{i,1}x_1 + \cdots + a_{i,m}x_m + a_{i,m+1} & \geq & 0)
\end{array}$$

Linear Loops

$L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$ in which all assertions are affine.

Notation:

- $\mathcal{V} = \{x_1, x_2, \ldots, x_m\}$
- let $(\mathbf{x}\mathbf{x}') = (x_1, \ldots, x_m, x'_1, \ldots, x'_m, 1)^{\mathrm{T}}$
- initial condition: $\Theta\mathbf{x} \geq 0$
- transitions: $\tau_i(\mathbf{x}\mathbf{x}') \geq 0$

Farkas Lemma (1894)

System of linear inequalities over real variables $\mathbf{x} = \{x_1, \ldots, x_n\}$:

$$S: \left[\begin{array}{ccccccccc} a_{1,1}x_1 & + & \cdots & + & a_{1,n}x_n & + & b_1 & \geq & 0 \\ \vdots & & & & \vdots & & \vdots & & \\ a_{m,1}x_1 & + & \cdots & + & a_{m,n}x_n & + & b_m & \geq & 0 \end{array}\right]$$

S entails linear inequality

$$\psi: \ c_1x_1 + \cdots + c_nx_n + d \geq 0$$

if and only if there exist real numbers $\lambda_1, \ldots, \lambda_m \geq 0$ such that

$$c_1 = \sum_{i=1}^{m} \lambda_i a_{i,1} \quad \cdots \quad c_n = \sum_{i=1}^{m} \lambda_i a_{i,n} \quad d \geq \left(\sum_{i=1}^{m} \lambda_i b_i\right)$$

Synthesis Overview

Templates:

- **expression** $\mathbf{c}^{\mathrm{T}}\mathbf{x}$, unknown coefficient vector $\mathbf{c}$
- **assertion** $\mathbf{Cx} \geq 0$, unknown coefficient matrix $\mathbf{C}$

Given templates

- n-conjunct invariant template $\boxed{\mathbf{Ix} \geq \mathbf{0}}$ ($\mathbf{I}$ has n rows)
- ℓ-component lexicographic polyranking function templates

$$\boxed{\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x},\ \ldots, \mathbf{c}_\ell{}^{\mathrm{T}}\mathbf{x}\}}$$

Apply Farkas Lemma **rules** to encode **polyranking function** and **supporting invariant** conditions.

Synthesize π.

Farkas Lemma Rules: [Bradley, Manna & Sipma 2005]

(Initiation)

$$\mathbb{I} : \frac{\Theta\mathbf{x} \geq \mathbf{0}}{\mathbf{I}\mathbf{x} \geq \mathbf{0}}$$

(Consecution)

$$\mathbb{C}_i : \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\begin{array}{ll} \mathbf{I}\mathbf{x}' & \geq \mathbf{0} \end{array}}$$

(Disabled)

$$\mathbb{D}_i : \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\begin{array}{ll} -1 & \geq \mathbf{0} \end{array}} \quad \leftarrow \text{false}$$

(Bounded)

$$\mathbb{B}_{ij} : \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\begin{array}{ll} \mathbf{c_j}^{\mathrm{T}}\mathbf{x} & \geq \mathbf{0} \end{array}}$$

(Ranking)

$$\mathbb{R}_{ij} : \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\begin{array}{ll} \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}' - \epsilon & \geq \mathbf{0} \end{array}}$$

(Nonincreasing)

$$\mathbb{N}_{ij} : \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\begin{array}{ll} \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}' & \geq \mathbf{0} \end{array}}$$

Example: $\mathbb{R}_{11}$

$\tau_1 : x \leq N \ \wedge \ 2x' \geq x + y - 1 \ \wedge \ y' = y + k \ \wedge \ k' = k \ \wedge \ N' = N$

$$
\begin{array}{c|l}
\lambda_1 & i_1 x + i_2 y + i_3 k + i_4 N + i_5 \geq 0 \\
\lambda_2 & -x + N \geq 0 \\
\lambda_3 & -x - y + 2x' + 1 \geq 0 \\
\lambda_4 & -y - k - y' = 0 \\
\lambda_5 & -k + k' = 0 \\
\lambda_6 & -N + N' = 0 \\
\hline
 & c_1 x + c_2 y + c_3 k + c_4 N - c_1 x' - c_2 y' - c_3 k' - c_4 N' - \epsilon \geq 0
\end{array}
$$

$$
\Rightarrow \quad
\begin{array}{rclrcl}
\lambda_1 i_1 - \lambda_2 - \lambda_3 & = & c_1 & 2\lambda_3 & = & -c_1 \\
\lambda_1 i_2 - \lambda_3 - \lambda_4 & = & c_2 & -\lambda_4 & = & -c_2 \\
\lambda_1 i_3 - \lambda_4 - \lambda_5 & = & c_3 & \lambda_5 & = & -c_3 \\
\lambda_1 i_4 + \lambda_2 - \lambda_6 & = & c_4 & \lambda_6 & = & -c_4
\end{array}
$$

$$\lambda_1 i_5 + \lambda_3 \leq -\epsilon \qquad \lambda_1, \lambda_2, \lambda_3 \geq 0 \qquad \epsilon > 0$$

Constraints over $\{\lambda_1, \ldots, \lambda_6, i_1, \ldots, i_5, c_1, \ldots, c_4, \epsilon\}$.

Farkas Lemma Rule: Decreasing

$$\mathbb{D}^{\star}_{ijk}: \quad \frac{\begin{array}{ll} \mathbf{I}\mathbf{x} & \geq \mathbf{0} \\ \tau_i(\mathbf{x}\mathbf{x}') & \geq \mathbf{0} \end{array}}{\mathbf{c_k}^{\mathrm{T}}\mathbf{x} + \mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}' \ \geq \mathbf{0}}$$

Difference

$$\mathbf{c_j}^{\mathrm{T}}\mathbf{x}' - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}$$

is upper bounded by

$$\mathbf{c_k}^{\mathrm{T}}\mathbf{x}$$

Combining the Rules

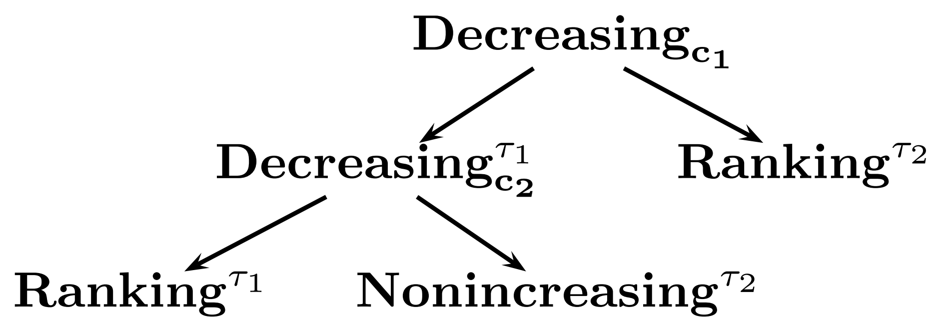
Combining the rules produces a numerical constraint system.

How are rules combined?

- According to definitions of
  - inductive invariant
  - polyranking function
- A **template tree** describes combination of rules for polyranking.

Template Tree & Tree Constraints

T:



$$\{\tau_1, \tau_2\} \subseteq \{\tau_1, \tau_2\}$$

tree_constraints(T): conjunction of

$$\begin{aligned}
\mathbf{Decreasing_{c_1}} &\Rightarrow (\mathbb{D}_1 \vee \mathbb{B}_{1,1}) \wedge (\mathbb{D}_2 \vee \mathbb{B}_{2,1}) \\
\mathbf{Decreasing^{\tau_1}_{c_2}} &\Rightarrow \mathbb{D}_1 \vee \mathbb{D}^{\star}_{1,1,2} \\
\mathbf{Ranking}^{\tau_1} &\Rightarrow \mathbb{D}_1 \vee \mathbb{R}_{1,2} \\
\mathbf{Nonincreasing}^{\tau_2} &\Rightarrow \mathbb{D}_2 \vee \mathbb{N}_{2,2} \\
\mathbf{Ranking}^{\tau_2} &\Rightarrow \mathbb{D}_2 \vee \mathbb{R}_{2,1}
\end{aligned}$$

Theorem: Synthesis (Special Case)

$L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$ has a
linear polyranking function
supported by an n-conjunct linear inductive invariant

iff

there exists template tree T for $\mathcal{T} \subseteq \mathcal{T}$ such that
the numeric constraint system induced by

$$\mathbb{I} \;\wedge\; \bigwedge_{\tau_i \in \mathcal{T}} (\mathbb{D}_i \vee \mathbb{C}_i) \;\wedge\; \mathit{tree\_constraints}(T)$$

is satisfiable.

Theorem: Synthesis (General Case)

$L : \langle \mathcal{V}, \Theta, \mathcal{T} \rangle$ has an
ℓ-lexicographic linear polyranking function
supported by an n-conjunct linear inductive invariant

iff

there exists $\pi : \mathcal{T} \rightarrow \{1, \ldots, \ell\}$
and tuple of template trees $\langle T_1,\ T_2,\ \ldots,\ T_\ell \rangle$
such that

1. the i^{th} template tree is for $\{\tau : \pi(\tau) = i\} \subseteq \{\tau' : \pi(\tau') \geq i\}$;
2. the ℓ numeric constraint systems induced by

$$\mathbb{I} \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T}} (\mathbb{D}_i \vee \mathbb{C}_i) \ \wedge \ \mathit{tree\_constraints}(T_j)$$

   for $j \in \{1, \ldots, \ell\}$ are satisfiable.

Synthesis In Practice

Search for

- $\langle T_1,\ T_2,\ \ldots,\ T_{|\mathcal{T}|}\rangle$
- $\pi : \mathcal{T} \rightarrow \{1, \ldots, |\mathcal{T}|\}$

such that induced constraint system is satisfiable.

- Enumerate **partial** template trees, starting small.
- **partial** template trees $\Rightarrow$ **partial** constraint system
  - **Unsatisfiable** $\Rightarrow$ no completion of trees is solution
    $\Rightarrow$ backtrack
  - **Satisfiable** $\Rightarrow$ continue search
- Solving constraint systems guides incremental construction.
- Solution if template trees are **complete**
  and constraint system is **satisfiable**.

Synthesis In Practice

Finding π is similar to search for **lexicographic ranking functions**. See [**Bradley, Manna & Sipma 2005**].

Implementation available for download at

`http://theory.stanford.edu/~arbrad`

Outline

1. Loop abstraction & definitions
2. Related work
3. Polyranking
4. Synthesis
5. **Conclusion**

Conclusion & Future Work

- Complete formulation of
  lexicographic polynomial polyranking functions
  over assertional polynomial loops.
- Complete method for synthesis of
  lexicographic linear polyranking functions
  with **supporting linear invariants**
  over assertional linear loops.
- Future work:
  - Methods of [**Cousot 2005**] may make (incomplete) polynomial synthesis tractable.
  - Integrate with verification diagram and transition invariant methods.

Thank you!

Appendix

The Big Picture

Theorem: For linear loops with only assignment, there does not exist a class of functions F that is

- **termination complete**: if L terminates, then $\exists f \in F$ s.t. f is a ranking function for L;
- and **synthesis complete**: there exists procedure P s.t. if L has a ranking function $f \in F$, then P finds one.

$\Rightarrow$ Find loops $\mathcal{L}$ and functions $\mathcal{F}$ s.t. synthesis problem is complete/decidable for $\mathcal{L}$ and $\mathcal{F}$.

Abstract loops with variables ranging over $\mathbb{Z}$ (rather than $\mathbb{R}$). **(CONCUR 2005)**

Expanding a Farkas Lemma Rule

$$\mathbb{R}_{ij}: \quad \frac{\begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx}') & \geq \mathbf{0} \end{array}}{\mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}' - \epsilon \geq \mathbf{0}}$$

$\mathbf{x} = (x_1, \ldots, x_m, 1)^{\mathrm{T}}$
$(\mathbf{xx}') = (x_1, \ldots, x_m, x'_1, \ldots, x'_m, 1)^{\mathrm{T}}$

$$\Downarrow$$

$$\frac{\begin{array}{llllll} \lambda_I & \mathbf{Ix} & & + & \mathbf{i} & \geq \mathbf{0} \\ \lambda_G & \mathbf{G_i x} & & + & \mathbf{g_i} & \geq \mathbf{0} \\ \lambda_U & \mathbf{U_i x} + & \mathbf{V_i x}' & + & \mathbf{u_i} & \geq \mathbf{0} \end{array}}{\mathbf{c_j}^{\mathrm{T}}\mathbf{x} - \mathbf{c_j}^{\mathrm{T}}\mathbf{x}' - \epsilon \geq \mathbf{0}}$$

Expand assertions:
$\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}}$
$\mathbf{x}' = (x'_1, \ldots, x'_m)^{\mathrm{T}}$
$\mathbf{I}, \mathbf{i}$ define Θ
$\mathbf{G_i}, \mathbf{g_i}, \mathbf{U_i}, \mathbf{V_i}, \mathbf{u_i}$ define τ_i

$$\Downarrow$$

$$\begin{array}{rcl} \lambda_I^{\mathrm{T}}\mathbf{I} + \lambda_G^{\mathrm{T}}\mathbf{G_i} + \lambda_U^{\mathrm{T}}\mathbf{U_i} & = & \mathbf{c_j} \\ \lambda_U^{\mathrm{T}}\mathbf{V_i} & = & -\mathbf{c_j} \\ \lambda_I^{\mathrm{T}}\mathbf{i} + \lambda_G^{\mathrm{T}}\mathbf{g_i} + \lambda_U^{\mathrm{T}}\mathbf{u_i} & \leq & -\epsilon \\ \lambda_I, \lambda_G, \lambda_U & \geq & 0 \\ \epsilon & > & 0 \end{array}$$

Constraints over $\{\lambda_I, \lambda_G, \lambda_U, \mathbf{c_j}, \epsilon\}$