Linear Ranking with Reachability

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

Computer Science Department

Stanford University

Overview

Goal: Automatic synthesis of ranking functions

- linear
- lexicographic
- over linear loops

Outline of presentation:

- Example
- Complete synthesis technique
- Results on C source code

Example: McCarthy 91

```
int f(int x)
  int s = 1
  while true do
    if x > 100
    then  if s = 1
          then return x − 10
          else  x := x − 10
                s := s − 1
    else  x := x + 11
          s := s + 1
  done
```

$$f(x) = \begin{cases} f(f(x+11)) & \text{if } x \leq 100 \\ x - 10 & \text{if } x > 100 \end{cases}$$

Abstract loop:

$$\begin{aligned} \Theta: &\quad s = 1 \\ \tau_1: &\quad x \geq 101 \ \wedge \ s \neq 1 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\ \tau_2: &\quad x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1 \end{aligned}$$

for $x, s \in \mathbb{R}$

Example: McCarthy 91

$$\begin{aligned}
\Theta: &\quad s = 1 \\
\tau_1: &\quad x \geq 101 \ \wedge \ s \neq 1 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_2: &\quad x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1
\end{aligned}$$

For every $1 \leq n \leq 92$, $f(n) = 91$, if it terminates.

Example:

$$\begin{aligned}
f(89) &= f(f(100)) = f(f(f(111))) = f(f(101)) \\
&= f(91) = f(f(102)) = \cdots = 91
\end{aligned}$$

We prove termination for all $n \in \mathbb{Z}^+$.

Example: McCarthy 91

$$
\begin{aligned}
\Theta: &\quad s = 1 \\
\tau_1: &\quad x \geq 101 \ \wedge \ s \neq 1 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_2: &\quad x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1
\end{aligned}
$$

Invariant: $s \geq 1$

Choose **lexicographic ranking function**

$$\delta: \ \langle 10s - x + 90, \ x \rangle$$

Bounded:

- τ_1: $x \geq 101 \ \rightarrow \ x \geq 0$
- τ_2: $x \leq 100 \wedge \boxed{s \geq 1} \ \rightarrow \ 10s - x + 90 \geq 0$

$\Rightarrow$ If $10s - x + 90 < 0 \wedge x < 0$, then the loop exited.

Example: McCarthy 91

$$\begin{aligned}
\Theta: &\quad s = 1 \\
\tau_1: &\quad x \geq 101 \ \wedge \ s \neq 1 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_2: &\quad x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1
\end{aligned}$$

$$\delta: \ \langle 10s - x + 90, \ x \rangle$$

Ranking:

- τ_1:

$$10s' - x' + 90 = 10(s-1) - (x - 10) + 90 \ = \ 10s - x + 90$$

$$x' = x - 10 \ < \ x$$

- τ_2:

$$10s' - x' + 90 = 10(s+1) - (x + 11) + 90 \ < \ 10s - x + 90$$

$\Rightarrow \delta$ decreases (lexicographically) on each iteration.

Example: McCarthy 91

$$\begin{aligned}
\Theta: &\quad s = 1 \\
\tau_1: &\quad x \geq 101 \ \wedge \ s \neq 1 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_2: &\quad x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1
\end{aligned}$$

So

$$\delta: \ \langle 10s - x + 90, \ x \rangle$$

is bounded from below, yet decreases on each iteration.

$\Rightarrow$ McCarthy 91 terminates on all input.

Outline

1. **Loop abstraction & definitions**
2. Related work
3. Synthesis
4. Synthesis of lexicographic functions
5. Empirical data
6. Conclusion

Loops

Loop Abstraction: McCarthy 91

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

- **variables** $\mathcal{V}$ range over $\mathbb{R}$ $\{x, s\}$
- **initial condition** Θ is assertion over $\mathcal{V}$ $s = 1$
- **transitions** $\tau \in \mathcal{T}$ are assertions $\{\tau_1, \tau_2\}$
  $\tau(\mathcal{V}, \mathcal{V}')$ over $\mathcal{V} \cup \mathcal{V}'$

Loop Validity:

Assertion φ is **valid over loop** L

$$L \models \varphi$$

if φ holds on all **reachable states** $\mathcal{S}_L$ of L. *values of* (x, s)

In practice, replace "$L \models$" with **loop invariants**. $s \geq 1$

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:

$$\mathbf{Ax} \geq 0$$

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,m} & a_{1,m+1} \\ & & \vdots & \\ a_{k,1} & \cdots & a_{k,m} & a_{k,m+1} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \\ 1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\bigwedge_{i \in \{1,\ldots,k\}} (a_{i,1}x_1 + \cdots + a_{i,m}x_m + a_{i,m+1} \geq 0)$$

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: $\tau_i(\mathbf{x}\mathbf{x}') \geq 0$

Linear Ranking Function

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

$\mathbf{r}^{\mathrm{T}}\mathbf{x}$ is a **linear ranking function** if

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

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}\mathbf{x} \geq 0$$

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

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}^{\mathrm{T}}\mathbf{x} - \mathbf{r}^{\mathrm{T}}\mathbf{x}' \geq \epsilon$$

Lexicographic Linear Ranking Function

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

$\langle \mathbf{r_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{r_k}^{\mathrm{T}}\mathbf{x} \rangle$ is a **k-lexicographic linear ranking function** if there exists map $\pi : \mathcal{T} \to \{1, \ldots, k\}$ such that

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

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}_{\pi(\tau)}^{\mathrm{T}}\mathbf{x} \geq 0$$

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

$$L \models \quad \tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}_{\pi(\tau)}^{\mathrm{T}}\mathbf{x} - \mathbf{r}_{\pi(\tau)}^{\mathrm{T}}\mathbf{x}' \geq \epsilon$$

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

$$L \models \quad (\forall j < \pi(\tau))[\tau(\mathbf{x}\mathbf{x}') \geq 0 \ \rightarrow \ \mathbf{r}_j^{\mathrm{T}}\mathbf{x} - \mathbf{r}_j^{\mathrm{T}}\mathbf{x}' \geq 0]$$

Linear Supporting Invariant

Ranking functions often require **supporting invariants**.

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

$\mathbf{Ix} \geq 0$ is an inductive **linear invariant** if

(Initiation)

$$\Theta \mathbf{x} \geq 0 \;\rightarrow\; \mathbf{Ix} \geq \mathbf{0}$$

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

$$\mathbf{Ix} \geq 0 \;\wedge\; \tau(\mathbf{xx'}) \geq 0 \;\rightarrow\; \mathbf{Ix'} \geq \mathbf{0}$$

Outline

1. Loop abstraction & definitions
2. **Related work**
3. Synthesis
4. Synthesis of lexicographic functions
5. Empirical data
6. Conclusion

Related Work: Synthesis of Measures

Katz & Manna 1975

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

Colón & Sipma 2001, 2002

Polyhedra-based synthesis of linear ranking functions for linear loops.

Colón, Sankaranarayanan & Sipma 2003

Constraint-based linear invariant generation.

Podelski & Rybalchenko 2004

Complete method, over loops with one transition and without an initial condition.

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**]

Contributions

1. Complete synthesis method for
   lexicographic linear ranking functions over
   assertional multi-path linear loops with
   supporting linear invariants

   Simultaneous synthesis of invariants $\Rightarrow$ nonlinear constraints

2. Constraint solver for
   nonlinear (parametric linear) **constraint systems**
   that arise

Outline

1. Loop abstraction & definitions
2. Related work
3. **Synthesis**
4. Synthesis of lexicographic functions
5. Empirical data
6. Conclusion

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

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

$$\boxed{\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x},\ \ldots, \mathbf{c_k}^{\mathrm{T}}\mathbf{x}\}}$$

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

Synthesize π.

Farkas Lemma Rules: Invariant

(Initiation)

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

(Consecution)

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

(Disabled)

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

Farkas Lemma Rules: Ranking Function

(Bounded)

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

(Ranking)

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

(Nonincreasing)

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

Example: $\mathbb{R}_{2,1}$ for McCarthy 91

$\tau_2 : x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1$

$$
\begin{array}{c|rcrcrcrcrcl}
\lambda_1 & i_1 x & + & i_2 s & & & & & + & i_3 & \geq & 0 \\
\lambda_2 & -x & & & & & & & + & 100 & \geq & 0 \\
\lambda_3 & x & & & - & x' & & & - & 11 & = & 0 \\
\lambda_4 & & & s & & & - & s' & - & 1 & = & 0 \\
\hline
 & c_{1,1} x & + & c_{1,2} s & - & c_{1,1} x' & - & c_{1,2} s' & - & \epsilon & \geq & 0
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{rcl}
\lambda_1 i_1 - \lambda_2 + \lambda_3 & = & c_{1,1} \\
\lambda_1 i_2 + \lambda_4 & = & c_{1,2} \\
-\lambda_3 & = & -c_{1,1} \\
-\lambda_4 & = & -c_{1,2}
\end{array}
\qquad
\begin{array}{rcl}
\lambda_1 i_3 + 100\lambda_2 - 11\lambda_3 - \lambda_4 & \leq & -\epsilon \\
\lambda_1, \lambda_2 & \geq & 0 \\
\epsilon & > & 0
\end{array}
$$

Constraints are over $\{c_{1,1}, c_{1,2}, \lambda_1, \lambda_2, \lambda_3, \lambda_4, i_1, i_2, i_3, \epsilon\}$.

Synthesis: Soundness and Completeness

Theorem (Special Case)

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

iff

the constraint system generated by

$$\mathbb{I} \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T}} (\mathbb{D}_i \vee (\mathbb{C}_i \wedge \mathbb{B}_i \wedge \mathbb{R}_i))$$

is **satisfiable**.

Synthesis: Soundness and Completeness

Theorem (General Case)

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

iff

the constraint system generated by

$$\mathbb{I} \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T}} \left(\mathbb{D}_i \vee \left(\mathbb{C}_i \wedge \mathbb{B}_{i,\pi(i)} \wedge \mathbb{R}_{i,\pi(i)} \right) \right) \ \wedge \ \bigwedge_{\tau_i \in \mathcal{T},\ j < \pi(i)} \left(\mathbb{D}_i \vee \mathbb{N}_{ij} \right)$$

is **satisfiable** for some $\pi : \mathcal{T} \to \{1, \ldots, k\}$.

One Constraint System per Component

Given lexicographic template components

$$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_n}^{\mathrm{T}}\mathbf{x}\}$$

and **partial order** $\prec$ over $\mathcal{T}$, solve n constraint systems induced by

$$\underbrace{\mathbb{I} \wedge \bigwedge_{\tau_i} (\mathbb{D}_i \vee \mathbb{C}_i)}_{\text{Invariant}} \quad \wedge \quad \underbrace{(\mathbb{D}_j \vee (\mathbb{B}_j \wedge \mathbb{R}_j))}_{\tau_j \text{ ranked by } \mathbf{c_j}^{\mathrm{T}}}$$

$$\wedge \quad \underbrace{\bigwedge_{\tau_i \text{ s.t. } \tau_j \prec \tau_i} (\mathbb{D}_i \vee \mathbb{N}_{ij})}_{\tau_i \text{ does not increase } \mathbf{c_j}^{\mathrm{T}} \text{ if } \tau_j \prec \tau_i}$$

$$\begin{array}{ccccccc} \langle \ldots, & \mathbf{c_j}^{\mathrm{T}}\mathbf{x}, & \ldots, & \mathbf{c_i}^{\mathrm{T}}\mathbf{x}, & \ldots \rangle \\ & \uparrow & & \uparrow & \\ & \tau_j & & \tau_i & \end{array}$$

for $j \in \{1, \ldots, n\}$.

One Constraint System per Component

Advantages:

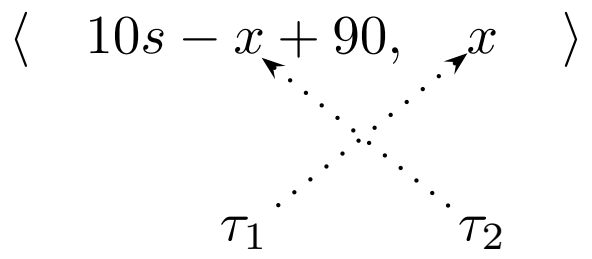
- Multiple smaller constraint systems are easier to solve in practice.
- Invariant template may be instantiated differently for each constraint system.

Outline

1. Loop abstraction & definitions
2. Related work
3. Synthesis
4. **Synthesis of lexicographic functions**
5. Empirical data
6. Conclusion

Synthesizing Lexicographic Ranking Functions

Goal: Find map π.



- τ_1 decreases component 2, does not increase component 1
- τ_2 decreases component 1

Synthesizing Lexicographic Ranking Functions

Overview:

- Initially propose one template component per transition

$$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_n}^{\mathrm{T}}\mathbf{x}\}$$

  Inexpensive because template coefficients $\mathbf{c_i}$ appear only **linearly** in generated constraints.

- Incrementally build linear order $\prec$ over $\mathcal{T}$.
  Linear order gives π.

- Use intermediate partial orders to generate and solve constraint systems to guide search.

Simplified Farkas Lemma Rules

One component per transition:

(Bounded)

$$\mathbb{B}_i : \quad \begin{array}{ll} \mathbf{Ix} & \geq \mathbf{0} \\ \tau_i(\mathbf{xx'}) & \geq \mathbf{0} \\ \hline \mathbf{c_i}^{\mathrm{T}}\mathbf{x} & \geq \mathbf{0} \end{array}$$

(Ranking)

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

Synthesizing Lexicographic Ranking Functions

Incrementally build $\prec$ over $\mathcal{T}$:

1. Guess $\tau_i \prec \tau_j$:

$$\langle \ldots, \mathbf{c_i}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_j}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

2. Generate constraint system and solve.
   - $\mathbb{I}$, $\mathbb{C}_i$, $\mathbb{D}_i$, $\mathbb{B}_i$, $\mathbb{R}_i$ are fixed.
   - $\tau_i \prec \tau_j$ induces $\mathbb{N}_{ji}$: τ_j should not increase $\mathbf{c_i}^{\mathrm{T}}\mathbf{x}$

   Satisfiable $\Rightarrow$ Continue search

   Unsatisfiable $\Rightarrow$ Try $\tau_i \succ \tau_j$

   Satisfiable $\Rightarrow$ Continue search

   Unsatisfiable $\Rightarrow$ Backtrack

3. Finished when order is **linear**
   and generated constraint system is **satisfiable**.

Example: McCarthy 91

$$
\begin{array}{ll}
\Theta: & s = 1 \\
\tau_1: & x \geq 101 \ \wedge \ s \leq 0 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_2: & x \geq 101 \ \wedge \ s \geq 2 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 \\
\tau_3: & x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1
\end{array}
\Bigg\} \text{ split of } s \neq 1
$$

Find

- a 3-component lexicographic linear ranking function

$$\langle \mathbf{r_1}^{\mathrm{T}}\mathbf{x},\ \mathbf{r_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{r_3}^{\mathrm{T}}\mathbf{x} \rangle$$

  and a map $\pi : \mathcal{T} \to \{1, 2, 3\}$

- with a 1-conjunct supporting invariant $\mathbf{Ix} \geq 0$

Example: McCarthy 91

Iteration 1:

- One template per transition.

$$\{\mathbf{c_1}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_3}^{\mathrm{T}}\mathbf{x}\}$$

- No order assumed between transitions.
- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 2:

- Guess $\tau_3 \prec \tau_2$.

$$\langle \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_2}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_2 should not increase $\mathbf{c_3}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 3:

- Guess $\tau_1 \prec \tau_3$.

$$\langle \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_3 should not increase $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \wedge \mathbb{N}_{3,1} \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3}
\end{aligned}$$

- Induced constraint system for $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$ is **unsatisfiable**.

Example: McCarthy 91

Iteration 3:

- Try $\tau_3 \prec \tau_1$ instead.

$$\langle \ldots, \mathbf{c_3}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_1 should not increase $\mathbf{c_3}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3} \wedge \mathbb{N}_{1,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 4:

- Guess $\tau_2 \prec \tau_1$ (but $\tau_1 \prec \tau_2$ works, too).

$$\langle \ldots, \mathbf{c_2}^{\mathrm{T}}\mathbf{x}, \ldots, \mathbf{c_1}^{\mathrm{T}}\mathbf{x}, \ldots \rangle$$

  τ_1 should not increase $\mathbf{c_2}^{\mathrm{T}}\mathbf{x}$.

- Three sets of conditions:

$$\begin{aligned}
\mathbf{c_1}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_1 \wedge \mathbb{R}_1 \\
\mathbf{c_2}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_2 \wedge \mathbb{R}_2 \wedge \mathbb{N}_{1,2} \\
\mathbf{c_3}^{\mathrm{T}}\mathbf{x}: &\quad \mathbb{I} \wedge \mathbb{C}_1 \wedge \mathbb{C}_2 \wedge \mathbb{C}_3 \wedge \mathbb{B}_3 \wedge \mathbb{R}_3 \wedge \mathbb{N}_{2,3} \wedge \mathbb{N}_{1,3}
\end{aligned}$$

- Induced constraint systems are **satisfiable**.

Example: McCarthy 91

Iteration 4:

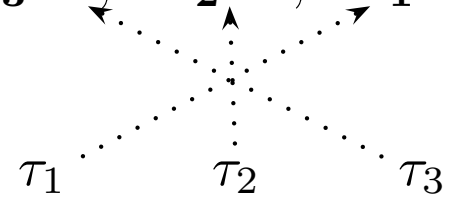
- $\prec$ is a linear order:

$$\langle \mathbf{c_3}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_2}^{\mathrm{T}}\mathbf{x},\ \mathbf{c_1}^{\mathrm{T}}\mathbf{x} \rangle$$

$$\tau_3 \prec \tau_2 \prec \tau_1$$

- $\prec$ gives π:

$$\langle \quad \mathbf{c_3}^{\mathrm{T}}\mathbf{x},\quad \mathbf{c_2}^{\mathrm{T}}\mathbf{x},\quad \mathbf{c_1}^{\mathrm{T}}\mathbf{x} \quad \rangle$$



$$\pi(\tau_1) = 3,\ \pi(\tau_2) = 2,\ \pi(\tau_3) = 1$$

Example: McCarthy 91

$$
\begin{array}{lll}
\Theta: & s = 1 & \\
\tau_1: & x \geq 101 \ \wedge \ s \leq 0 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 & \left.\begin{array}{l}\\\\\end{array}\right\} \text{split of} \\
\tau_2: & x \geq 101 \ \wedge \ s \geq 2 \ \wedge \ x' = x - 10 \ \wedge \ s' = s - 1 & \left.\begin{array}{l}\\\\\end{array}\right\} s \neq 1 \\
\tau_3: & x \leq 100 \ \wedge \ x' = x + 11 \ \wedge \ s' = s + 1 &
\end{array}
$$

Solving the final constraint systems reveals ranking function

$$\langle 10s - x + 90,\ x,\ x \rangle$$

$$\pi(\tau_1) = 3,\ \pi(\tau_2) = 2,\ \pi(\tau_3) = 1$$

supported by the invariant

$$s \geq 1$$

which proves that McCarthy 91 always terminates.

Outline

1. Loop abstraction & definitions
2. Related work
3. Synthesis
4. Synthesis of lexicographic functions
5. **Empirical data**
6. Conclusion

In Practice

- Implementation of analysis of abstract loops in O'Caml.
  - Solves constraints using method discussed in paper.
  - Proves termination of loops like McCarthy 91 in seconds.
  - Available for download
    `http://theory.stanford.edu/~arbrad`
- Prototype **unsound** abstraction of C loops, using CIL.
  Why unsound?
  - Overflows, `unsigned`, `double`s abstracted as $\mathbb{R}$s, *etc.*
  - Aliasing, globals changed by function calls, *etc.*

  In principle, can be sound — an engineering task.

Experimental Results

Name	LOC	#L	#A	#P	P/A	P/L	Tm
small1	310	8	3	2	66	25	2
vector	361	13	13	12	92	92	2
serv	457	9	6	5	83	55	2
dcg	1K	55	53	53	100	96	4
bcc	4K	70	18	18	100	25	5
sarg	7K	110	25	25	100	22	77
spin	19K	652	124	119	95	18	76
meschach	28K	911	778	758	97	83	64
f2c	30K	436	100	98	98	22	47
ffmpeg1	33K	451	230	217	94	48	55
gnuplot	50K	826	312	301	96	36	88
gaim	57K	594	54	52	96	8	94
ffmpeg2	75K	2223	1885	1864	98	83	143

Example of Failed Abstraction

```
List * iter = items;
while (iter != NULL) {
  ...
  iter = iter->next;
}
```

Proving termination requires:

- proving that `items` is noncircular (**nontrivial**);
- abstracting iteration to counting down (**trivial**).

Example of Failed Abstraction

```
char * ptr = input;
while (*ptr != '\0') {
  ...
  ptr++;
}
```

Proving termination requires:

- proving that `input` is a well-formed C string (**nontrivial**);
- abstracting pointer arithmetic to counting down (**trivial**).

Reasons for Failed Proofs

1. **Prototype** abstracter!
2. Need for invariants. Examples:

   `i = 2 * i;` $i \geq 0$ to deduce increase in `i`

   `i = i + k;` $k > 0$ to deduce increase in `i`
3. Need for **summarizing** embedded loops. Example:

```
while (i < n) {
  while (...) { i++; }    ← summarize with i' ≥ i
  i++;
}
```

4. Need for function invariants. Example:

   `i = i + strlen(str);` knowledge about `str` and `strlen`
5. Loop does not terminate.

Outline

1. Loop abstraction & definitions
2. Related work
3. Synthesis
4. Synthesis of lexicographic functions
5. Empirical data
6. **Conclusion**

Conclusion & Future Work

- Complete method of synthesizing
  lexicographic linear ranking functions
  with **supporting invariants**
  of linear loops.
- Empirical evidence shows lexicographic technique is fast.
- Future work:
  - Strengthen abstraction.
  - 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 $\mathcal{F}$.

Related work:

- For fixed class of loops and fixed class of functions, **polyranking** generalizes **ranking**. **(ICALP 2005)**
- Abstract loops with variables ranging over $\mathbb{Z}$ (rather than $\mathbb{R}$). **(CONCUR 2005)**

Expanding a Farkas Lemma Rule

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

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

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

$\Downarrow$

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

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

Example: McCarthy 91

Constraints on $\mathbf{c_1}^{\mathrm{T}}\mathbf{x}$:

Initiation

$$\mathbb{I}: \quad \frac{s = 1}{i_1 s + i_2 x + i_3 \geq 0}$$

Consecution

$$\mathbb{C}_1: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

$$\mathbb{C}_2: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \geq 2 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

$$\mathbb{C}_3: \quad \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \leq 100 \\ x' = x + 11 \\ s' = s + 1 \end{array}}{i_1 s' + i_2 x' + i_3 \geq 0}$$

Example: McCarthy 91

Bounded

$$\mathbb{B}_1 : \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \end{array}}{c_{1,1} s + c_{1,2} x + c_{1,3} \geq 0}$$

Ranking

$$\mathbb{R}_1 : \frac{\begin{array}{l} i_1 s + i_2 x + i_3 \geq 0 \\ x \geq 101 \\ s \leq 0 \\ x' = x - 10 \\ s' = s - 1 \end{array}}{c_{1,1} s + c_{1,2} x \geq c_{1,1} s' + c_{1,2} x' + \epsilon}$$