Interpolation-based height analysis for improving a recurrence solver

Manuel Montenegro[1], Olha Shkaravska[2], Ricardo Peña[1] and Marko van Eekelen[2 3]

[1]Universidad Complutense de Madrid
[2]Radboud University Nijmegen
[3]Open University of the Netherlands

2nd International Workshop on Foundational and Practical Aspects of Resource Analysis, FOPARA 2011
May 19, 2011 - Madrid

Outline

Outline

The COSTA System

COSTA = COSt and Termination Analyzer for Java Bytecode

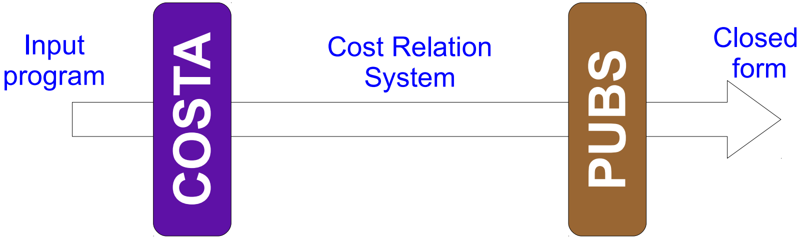
Universidad Complutense de Madrid (UCM)

Elvira Albert
Puri Arenas
Samir Genaim
Diego Alonso
Jesús Correas
Miguel Gómez-Zamalloa

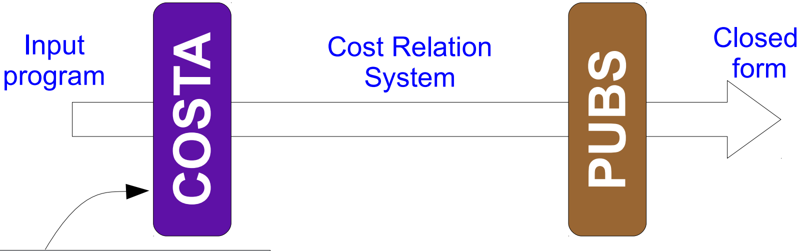
Universidad Politécnica de Madrid (UPM)

Germán Puebla
Damiano Zanardini
Abu Naser Masud
Diana Ramírez
José Miguel Rojas
Guillermo Román

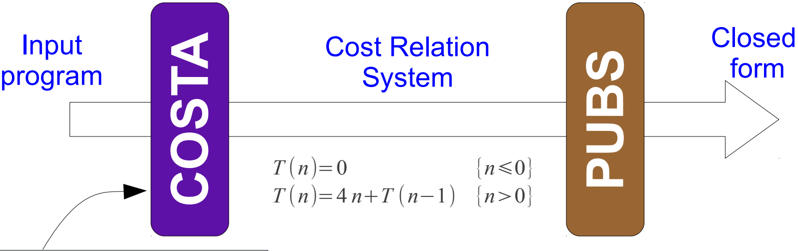
The COSTA System



The COSTA System



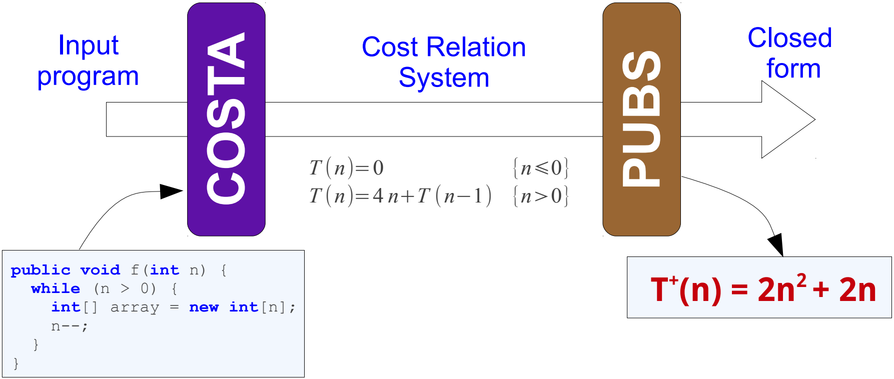

```
public void f(int n) {
  while (n > 0) {
    int[] array = new int[n];
    n--;
  }
}
```



$$
\begin{array}{ll}
T(n)=0 & \{n\leqslant 0\}\\
T(n)=4n+T(n-1) & \{n>0\}
\end{array}
$$

```java
public void f(int n) {
  while (n > 0) {
    int[] array = new int[n];
    n--;
  }
}
```

The COSTA System

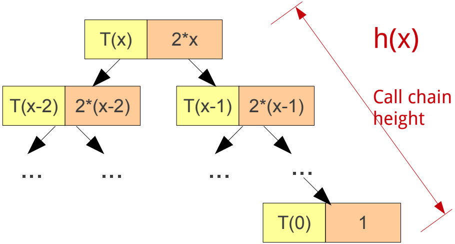


PUBS = Practical Upper Bounds Solver

- Cost Relation Systems
  - Nondeterministic (several equations appliable, inexact constraints).
  - Multivariate.
- Computation of a closed-form upper bound (expression without recursion)
  - It is based on the notion of evaluation trees.

$T(x) = 1 \quad \{ x \leq 1 \}$

$T(x) = 2x + T(x-1) + T(x-2) \quad \{ x > 1 \}$



- Call-chain height $h(x)$: Upper-bound to the maximum number of unfoldings that may be undergone to reach a base case.
- It is closely related to the concept of ranking function.

$$\underbrace{T(x) \longrightarrow T(x-1) \longrightarrow T(x-2) \longrightarrow \cdots \longrightarrow T(0)}_{\mathbf{h(x) = x}}$$

The PUBS System

Ranking functions

- Call-chain height $h(x)$: Upper-bound to the maximum number of unfoldings that may be undergone to reach a base case.
- It is closely related to the concept of ranking function.

$$\underbrace{\boxed{T(x)} \longrightarrow \boxed{T(x - 1)} \longrightarrow \boxed{T(x - 2)} \longrightarrow \cdots \longrightarrow \boxed{T(0)}}_{\mathbf{h(x) = x}}$$

A. Podelski, A. Rybalchenko.
A Complete Method for the Synthesis of Linear Ranking Functions.
5th International Conference on Verification, Model Checking and Abstract Interpretation, VMCAI 2004

- The length of the maximal call chain associated with a CRS may not linearly depend on the sizes of the arguments.

Example

$$
\begin{array}{rcll}
T(x,y) & = & \mathrm{nat}(x) & \{x=0, y=0\} \\
T(x,y) & = & \mathrm{nat}(x) + T(x-1, x-1) & \{x>0, y=0\} \\
T(x,y) & = & \mathrm{nat}(x) + T(x, y-1) & \{x\geq 0, y>0\}
\end{array}
$$

The PUBS System

Polynomial ranking functions

- The length of the maximal call chain associated with a CRS may not linearly depend on the sizes of the arguments.

Example

$$
\begin{array}{rcll}
T(x,y) & = & \mathsf{nat}(x) & \{x = 0, y = 0\} \\
T(x,y) & = & \mathsf{nat}(x) + T(x-1, x-1) & \{x > 0, y = 0\} \\
T(x,y) & = & \mathsf{nat}(x) + T(x, y-1) & \{x \geq 0, y > 0\}
\end{array}
$$

$$
\underbrace{\boxed{T(x,y)} \longrightarrow \boxed{T(x,y\text{-}1)} \longrightarrow \boxed{T(x,y\text{-}2)} \longrightarrow \ \dots \ \longrightarrow \boxed{T(0,0)}}
$$

$$\mathbf{h(x) = 0.5x^2 + 0.5x + y}$$

Example (Taken from Shkaravska et al. PPPJ 2010)

```
while (i < x && j <= x) {
    if (j == x) { i++; j = 0; }
    j++;
}
```

Example (Taken from Shkaravska et al. PPPJ 2010)

```
while (i < x && j <= x) {
    if (j == x) { i++; j = 0; }
    j++;
}
```

$$
\begin{array}{ll}
T(x,i,j) = 0 & \{0 \geq x\} \\
T(x,i,j) = 0 & \{0 \geq i\} \\
T(x,i,j) = 0 & \{i \geq x\} \\
T(x,i,j) = 0 & \{0 \geq j\} \\
T(x,i,j) = 0 & \{j \geq x+1\} \\
T(x,i,j) = 1 + T(x,i+1,j) & \{x \geq 1, i \geq 1, x \geq y+1, j \geq 0, x = j\} \\
T(x,i,j) = 1 + T(x,i,j+1) & \{x \geq 1, i \geq 1, x \geq y+1, j \geq 0, x \geq j+1\}
\end{array}
$$

$$h(x,i,j) = x^2 - xi - j + 1$$

Outline

1. Introduction
   - The COSTA System
   - Polynomial interpolation-based techniques
2. Integrating PI into PUBS
3. Nondeterministic CRS
4. Conclusions

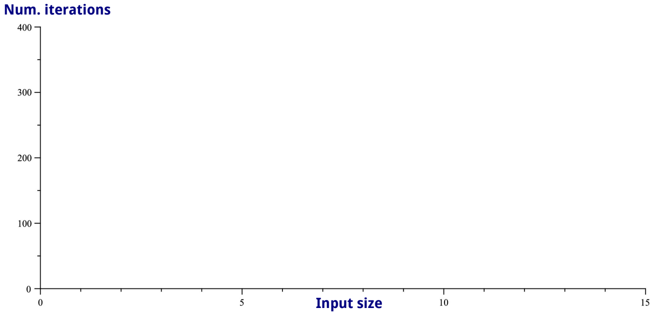
Radboud University Nijmegen

Marko van Eekelen
Olha Shkaravska
Rody Kersten
Alejandro Tamalet
Ron van Kesteren

- AHA Project: Polynomial size analysis for a functional language.
- CHARTER Project: Polynomial loop-bound analysis.

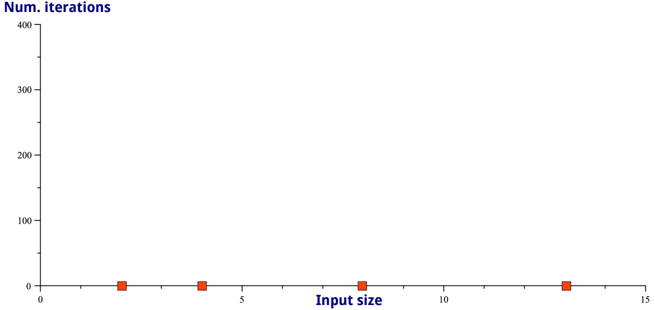
Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.



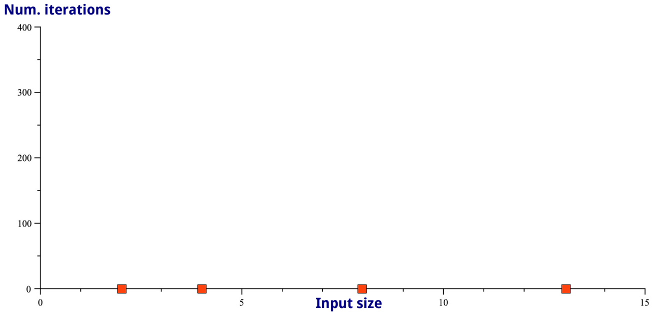
Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.



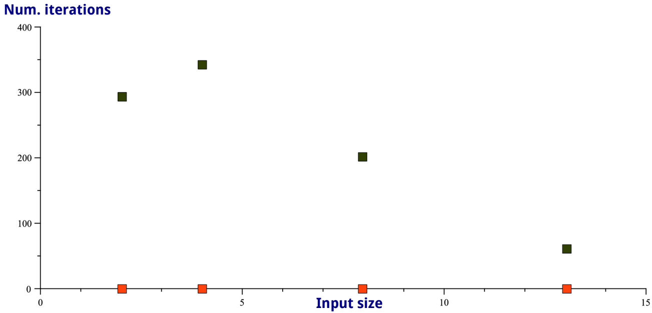
Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Run the program for the given input size and measure results.



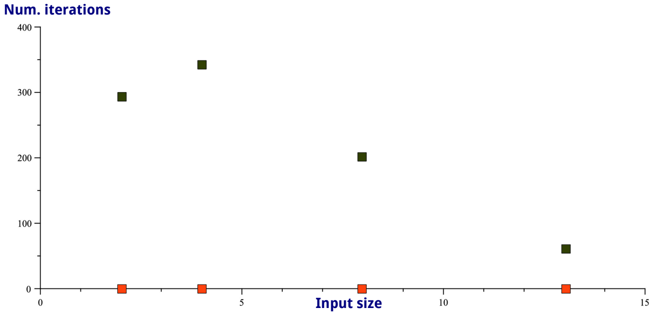
Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Run the program for the given input size and measure results.



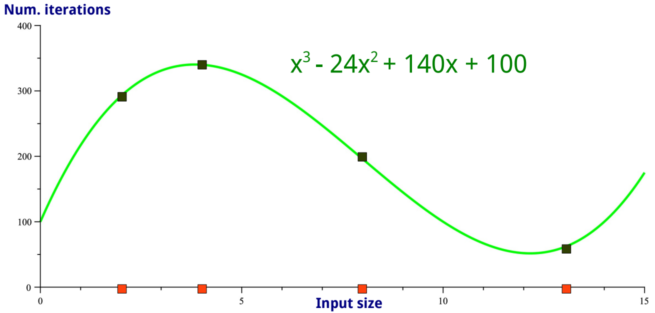
Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Run the program for the given input size and measure results.
- **Step 3:** Polynomial interpolation.



Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Run the program for the given input size and measure results.
- **Step 3:** Polynomial interpolation.



Testing & interpolation-based approach

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Run the program for the given input size and measure results.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Type system.
- External tool (KeY).
- If checking fails $\Rightarrow$ try a higher degree.

Testing & interpolation-based approach

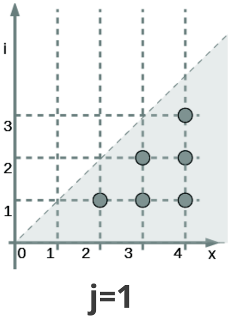
An example

Example (Taken from Shkaravska et al. PPPJ 2010)

```
while (i < x && j <= x) {
    if (j == x) { i++; j = 0; }
    j++;
}
```

Example (Taken from Shkaravska et al. PPPJ 2010)

```
while (i < x && j <= x) {
    if (j == x) { i++; j = 0; }
    j++;
}
```



x	i	j
2	1	1
3	1	1
4	1	1
3	2	1
4	2	1
4	3	1
3	1	2
4	1	2
3	2	2

$\Rightarrow$ $x^2 - xi - j + 1$

Our Aim

Compute ranking functions in PUBS by using polynomial interpolation-based techniques, as an alternative to the Podelski-Rybalchenko approach

Our Aim

Compute ranking functions in PUBS by using polynomial interpolation-based techniques, as an alternative to the Podelski-Rybalchenko approach

Consequence

PUBS can deal with a broader set of recurrences

Outline

Interpolation-based call chain height analysis

- Generate a CRS T_h capturing the height of the call chain.

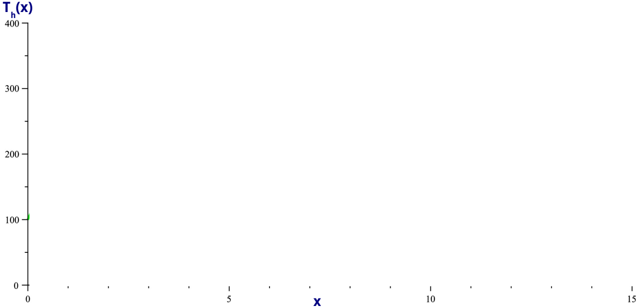
$$T(\overline{x}) = \mathrm{exp} \quad \psi \qquad \Longrightarrow \qquad T_h(\overline{x}) = 0 \quad \psi$$

$$T(\overline{x}) = \mathrm{exp} + \sum_{i=1}^{m} T(\overline{y_i}) \quad \psi \qquad \Longrightarrow \qquad T_h(\overline{x}) = 1 + \max_{i=1\dots m}\{T_h(\overline{y_i})\} \quad \psi$$

- Apply testing & interpolation-based analysis to T_h.

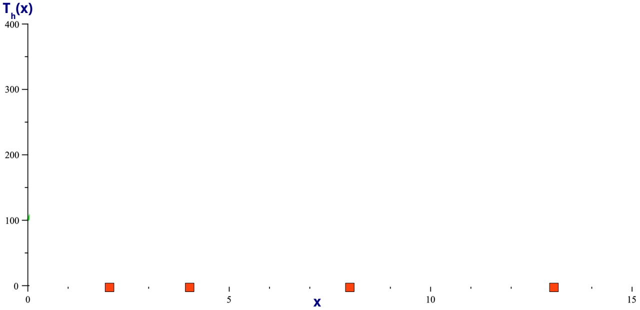
Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.



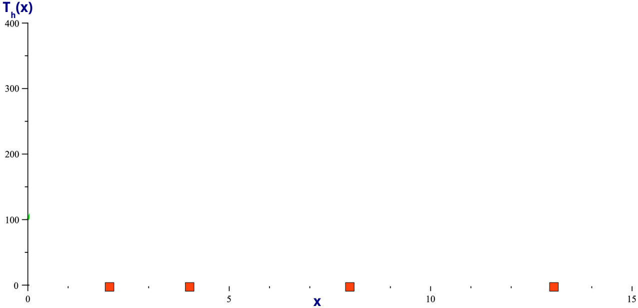
Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.



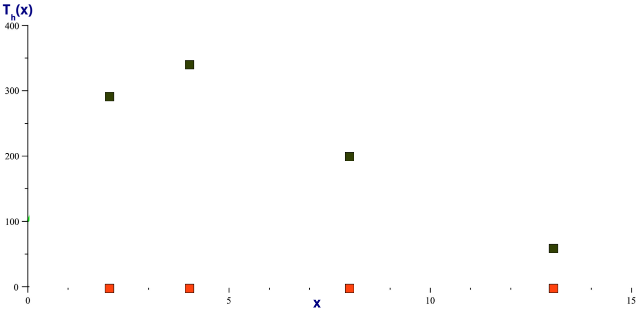
Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.

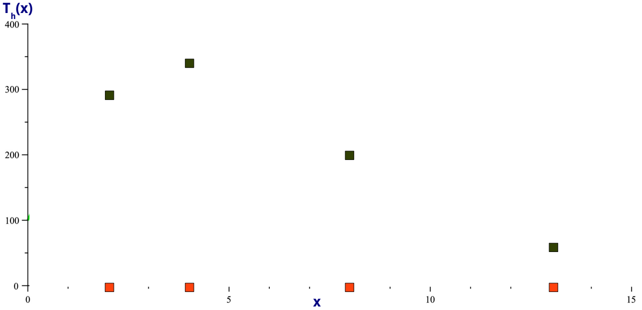


Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.

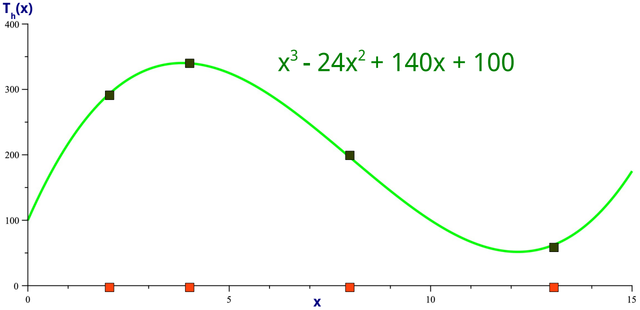


- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.



Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.



Interpolation-based call chain height analysis

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Sufficient conditions:

$$T_h^+(\overline{x}) \geq 0$$

$$T_h^+(\overline{x}) \geq 1 + T_h^+(\overline{x}') \text{ whenever the recursive guard } \psi_r(\overline{x}, \overline{x}') \text{ holds}$$

- Existing tools: QEPCAD

Interpolation-based call chain height analysis

Example (I)

$$
\begin{array}{ll}
T(x,y,z) = 0 & \{x = 0, y = 0, z \geq 1\} \\
T(x,y,z) = 2 & \{x \geq 0, y \geq 0, z = 0\} \\
T(x,y,z) = 1 + T(z, y-1, z) + T(x, y, z-1) & \{x = 0, y \geq 0, z \geq 1\} \\
T(x,y,z) = 1 + T(x-1, y, z) + T(x, y, z-1) & \{x > 0, y \geq 0, z \geq 1\}
\end{array}
$$

Interpolation-based call chain height analysis

Example (I)

$$
\begin{array}{ll}
T(x,y,z) = 0 & \{x = 0, y = 0, z \geq 1\} \\
T(x,y,z) = 2 & \{x \geq 0, y \geq 0, z = 0\} \\
T(x,y,z) = 1 + T(z, y-1, z) + T(x, y, z-1) & \{x = 0, y \geq 0, z \geq 1\} \\
T(x,y,z) = 1 + T(x-1, y, z) + T(x, y, z-1) & \{x > 0, y \geq 0, z \geq 1\}
\end{array}
$$

Derived CRS

$$
\begin{array}{ll}
T_h(x,y,z) = 0 & \{x = 0, y = 0, z \geq 1\} \\
T_h(x,y,z) = 0 & \{x \geq 0, y \geq 0, z = 0\} \\
T_h(x,y,z) = 1 + \max\{T_h(z, y-1, z), T_h(x, y, z-1)\} & \{x = 0, y \geq 0, z \geq 1\} \\
T_h(x,y,z) = 1 + \max\{T_h(x-1, y, z), T_h(x, y, z-1)\} & \{x > 0, y \geq 0, z \geq 1\}
\end{array}
$$

x	y	z	$T_h(x,y,z)$
1	1	1	3
1	2	1	5
2	1	1	4
3	1	1	5
2	2	1	6
1	3	1	7
1	1	2	5
1	2	2	8
2	1	2	6
1	1	3	7

Interpolation returns:

$$T_h^+(x,y,z) = yz + x + y + z - 1$$

$$T^+(x,y,z) = 2^{nat(y*z+x+y+z-1)} - 1 + 2 * 2^{nat(y*z+x+y+z-1)}$$

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Evaluating a CRS is a difficult task because of nondeterminism.

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Evaluating a CRS is a difficult task because of nondeterminism.
  - Choice between several equations.

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Evaluating a CRS is a difficult task because of nondeterminism.
  - Choice between several equations.
  - Choice between several values for the recursive calls.

$$T_h(x) = 1 + T_h(x') \qquad \{0 \leq x' \leq 5\}$$

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Evaluating a CRS is a difficult task because of nondeterminism.
  - Choice between several equations.
  - Choice between several values for the recursive calls.

$$T_h(x) = 1 + T_h(x') \qquad \{0 \leq x' \leq 5\}$$

  - Choice between an infinite amount of values!

Problem

- **Step 1:** Choose test points.
  - Degree d, dimension s: $\binom{d+s}{s}$ points in a NCA configuration.
- **Step 2:** Evaluate T_h for the given test points.
- **Step 3:** Polynomial interpolation.
- **Step 4:** Check the obtained bound.

- Evaluating a CRS is a difficult task because of nondeterminism.
  - Choice between several equations.
  - Choice between several values for the recursive calls.

$$T_h(x) = 1 + T_h(x') \qquad \{0 \le x' \le 5\}$$

  - Choice between an infinite amount of values!

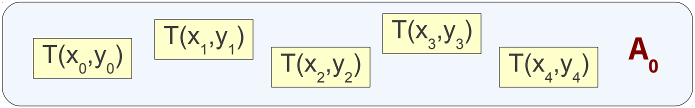
Example

$$\begin{array}{rcl l} T_h(x,y) & = & 0 & \{x \ge y\} \\ T_h(x,y) & = & 1 + T_h(x',y) & \{x < y, x' > x\} \end{array}$$

Outline

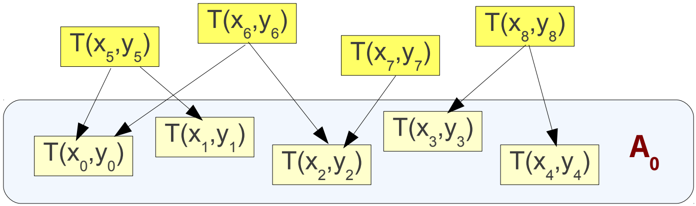
Evaluation of CRS in presence of nondeterminism

- Solution: evaluate a CRS in a bottom-up fashion.



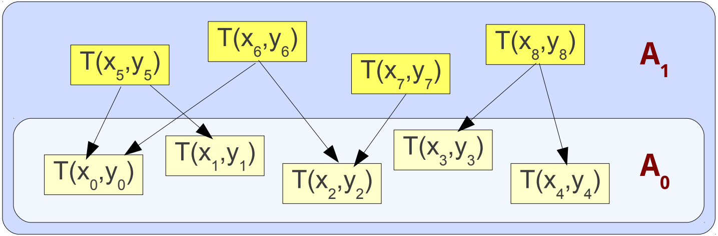
Evaluation of CRS in presence of nondeterminism

- Solution: evaluate a CRS in a bottom-up fashion.

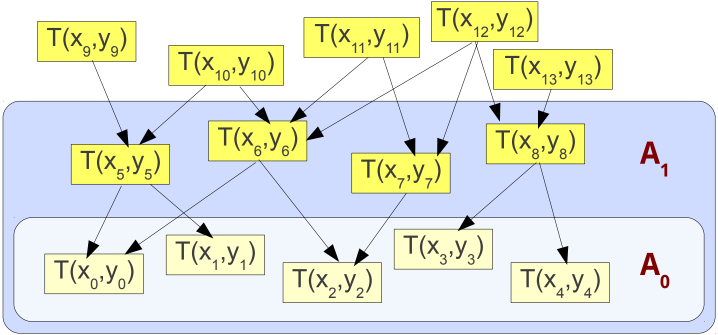


Evaluation of CRS in presence of nondeterminism

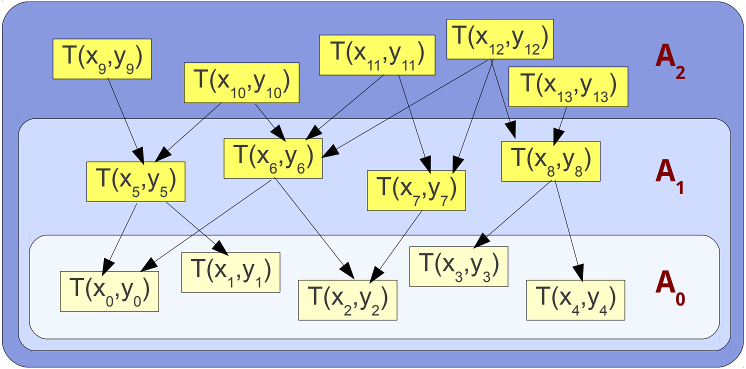
- Solution: evaluate a CRS in a bottom-up fashion.



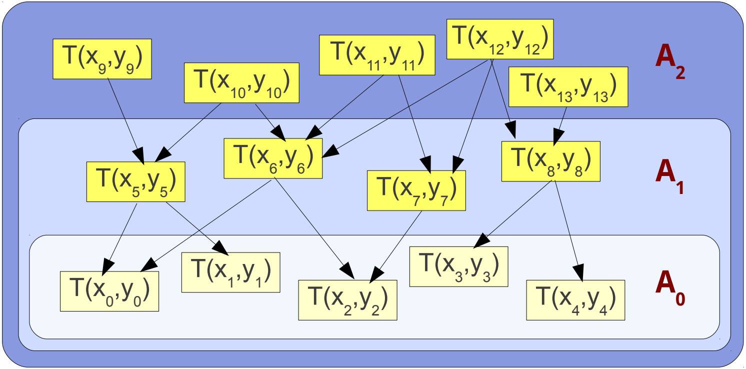
- Solution: evaluate a CRS in a bottom-up fashion.



- Solution: evaluate a CRS in a bottom-up fashion.



- Solution: evaluate a CRS in a bottom-up fashion.



$$A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_i \subseteq \ldots$$

$$A_i = \{\overline{x} \mid \max\, T_h(\overline{x}) \leq i\}$$

- We characterize the A_i sets in terms of the guards of the CRS.
  - $\psi_b(\overline{x})$: base guard.
  - $\psi_r(\overline{x}, \overline{x}')$: recursive guard

Definition

$$\varphi_0(\overline{x}) \stackrel{\text{def}}{=} \psi_b(\overline{x}) \wedge \forall \overline{x}'. \neg \psi_r(\overline{x}, \overline{x}')$$

$$\varphi_i(\overline{x}) \stackrel{\text{def}}{=} \varphi_0(\overline{x}) \vee \left[(\exists \overline{x}'. \psi_r(\overline{x}, \overline{x}')) \wedge \forall \overline{x}'. (\psi_r(\overline{x}, \overline{x}') \Rightarrow \varphi_{i-1}(\overline{x}'))\right] \qquad (i > 0)$$

Theorem

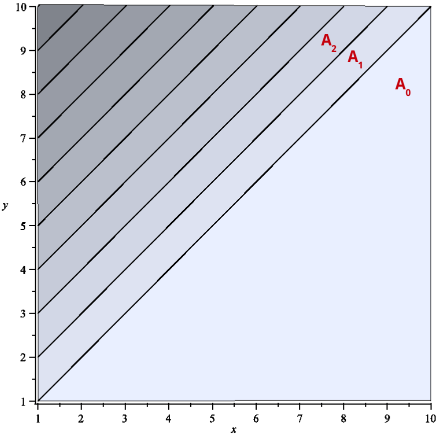
$$A_i = \{\overline{x} \mid \varphi_i(\overline{x})\}$$

- Quantifier elimination methods: Cooper (1972)

Evaluation of CRS in presence of nondeterminism

Example

$$
\begin{array}{lcll}
T_h(x,y) & = & 0 & \{x \geq y\} \\
T_h(x,y) & = & 1 + T_h(x',y) & \{x < y, x' > x\}
\end{array}
$$



$\varphi_0(x,y) \equiv x \geq y$

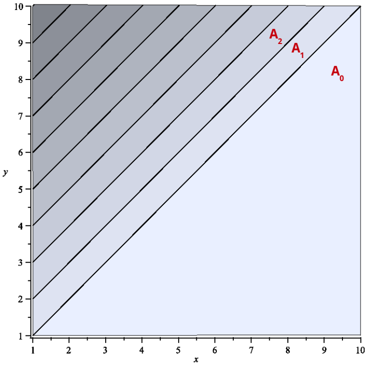
$\varphi_1(x,y) \equiv x \geq y - 1$

$\varphi_2(x,y) \equiv x \geq y - 2$

$\vdots$

Searching for test nodes
Gradient-based approach

- The graph of $h(\overline{x})$ in $s+1$ dimensional space is the union of sets $A'_i = \{(\overline{x}, i) \mid \overline{x} \in A_i \backslash A_{i-1}\}$
- It may be viewed as a collection of upside-down terraces.

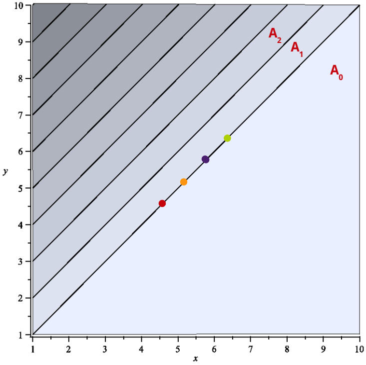


- The gradient $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right)$ shows the direction of the greatest increase rate of the function.
  - h is not a continuous function.
- Gradient $\approx$ direction of closest-point in the next-level terrace.

Searching for test nodes

Gradient-based approach

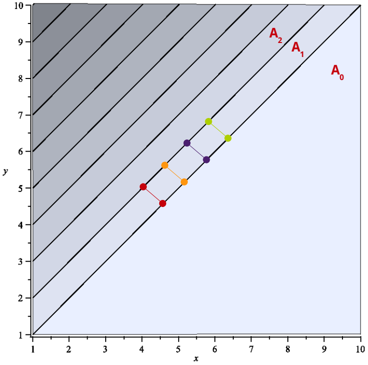
- The graph of $h(\overline{x})$ in $s+1$ dimensional space is the union of sets $A_i' = \{(\overline{x}, i) \mid \overline{x} \in A_i \backslash A_{i-1}\}$
- It may be viewed as a collection of upside-down terraces.



- The gradient $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right)$ shows the direction of the greatest increase rate of the function.
  - h is not a continuous function.
- Gradient $\approx$ direction of closest-point in the next-level terrace.

Searching for test nodes
Gradient-based approach

- The graph of $h(\overline{x})$ in $s+1$ dimensional space is the union of sets $A'_i = \{(\overline{x}, i) \mid \overline{x} \in A_i \backslash A_{i-1}\}$
- It may be viewed as a collection of upside-down terraces.

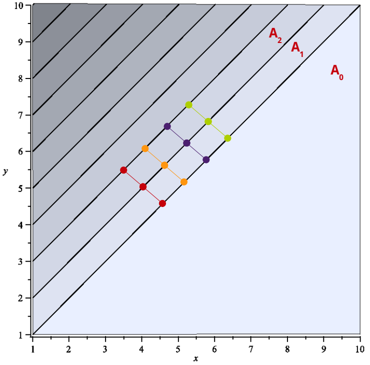


- The gradient $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right)$ shows the direction of the greatest increase rate of the function.
  - h is not a continuous function.
- Gradient $\approx$ direction of closest-point in the next-level terrace.

Searching for test nodes

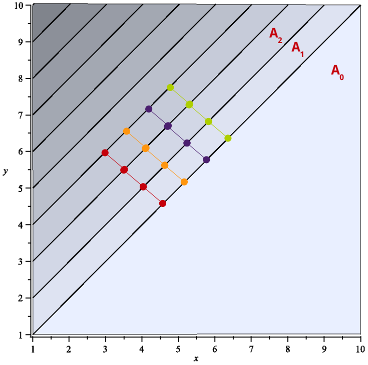
Gradient-based approach

- The graph of $h(\overline{x})$ in $s+1$ dimensional space is the union of sets $A'_i = \{(\overline{x}, i) \mid \overline{x} \in A_i \backslash A_{i-1}\}$
- It may be viewed as a collection of upside-down terraces.



- The gradient $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right)$ shows the direction of the greatest increase rate of the function.
  - h is not a continuous function.
- Gradient $\approx$ direction of closest-point in the next-level terrace.

- The graph of $h(\overline{x})$ in $s+1$ dimensional space is the union of sets $A'_i = \{(\overline{x}, i) \mid \overline{x} \in A_i \backslash A_{i-1}\}$
- It may be viewed as a collection of upside-down terraces.



- The gradient $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y}\right)$ shows the direction of the greatest increase rate of the function.
  - h is not a continuous function.
- Gradient $\approx$ direction of closest-point in the next-level terrace.

Searching for test nodes

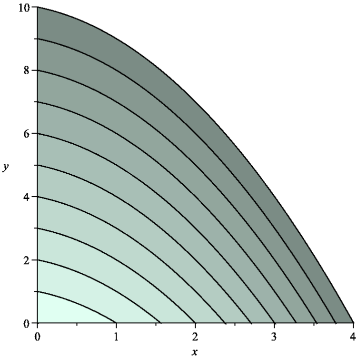
Gradient-based approach

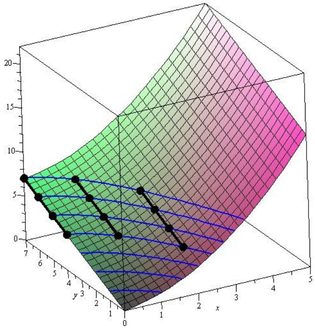
- Given:
  - Initial level i_0.
  - Number of routes r.
  - Length of routes l.
    - such that $r * l \geq \binom{d+s}{s}$
  - Initial points $x_{0,j}$, $j \in \{1..r\}$ (vertices or edges of A_{i_0})

1. For each $i = i_0 + 1 \ldots l$
   - For each $j = 1..r$
     - The next i-th point on the j-th route $x_{i,j}$ is computed as the closest to $x_{i-1,j}$ point on $A_i \backslash A_{i-1}$.
2. Check whether the routes satisfy NCA condition
3. Solve the corresponding linear equation system (interpolation)

Example

$$
\begin{array}{rcll}
T(x,y) & = & \mathrm{nat}(x) & \{x=0, y=0\} \\
T(x,y) & = & \mathrm{nat}(x) + T(x-1, x-1) & \{x>0, y=0\} \\
T(x,y) & = & \mathrm{nat}(x) + T(x, y-1) & \{x \geq 0, y>0\}
\end{array}
$$





$$T_h^+(x,y) = \frac{1}{2}x^2 + \frac{1}{2}x + y$$

Outline

Conclusions

- Applied polynomial interpolation-based techniques in order to extend the PUBS recurrence solver.
    - linear ranking functions $\Longrightarrow$ polynomial ranking functions.
- Complement the existing technique (Podelski-Rybalchenko: complete for linear RFs)

Future work

- Integration within PUBS.
- Logarithmic bounds: $\log p(x)$.
- Adjustment of standard interpolation: approximation (upper bounds).