Certifying polynomial time and linear/polynomial space for imperative programs: improvements and optimality

Karl-Heinz Niggl

(Joint work with Henning Wunderlich and Jan Mehler)

Geocal'06: Implicit Computational Complexity

Marseille-Luminy

February 13–17, 2006

Starting point

Earlier research [Kristiansen & N.]:

- Stack programs of μ-measure $0 = \textsc{fptime}$
- Loop programs of μ-measure $0 = \textsc{flinspace}$

From a programming perspective, this is not practically appealing:

- no user-friendly basic instructions (BI),
  except for nonsize-increasing BI's
- no vital instructions such as assignment statements $\mathtt{X}_i = \mathtt{X}_j$
- no mixed datatypes

General outline – 1

Programs in this talk are built from arbitrary BI's $\mathtt{imp}(\mathtt{X}_1, \ldots, \mathtt{X}_n)$ by

- sequences $\mathtt{P}_1; \mathtt{P}_2$
- conditionals $\mathtt{if}\ (\mathtt{cond})\ \mathtt{then}\ \mathtt{P}_1$ and $\mathtt{if}\ (\mathtt{cond})\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2$
- loop statements $\mathtt{loopI}\ \mathtt{X}_\mathtt{h}\ [\mathtt{Q}]$ and $\mathtt{loopII}\ \mathtt{X}_\mathtt{h}\ [\mathtt{Q}]$.

Each variable $\mathtt{X}_i$ may represent any datatype, e.g.

— stacks, registers, trees, graphs, arrays, ...

and is implicitly equipped with a notion of **size** $|\mathtt{X}_i|$, e.g.

— $\mathtt{X}_i$ serves as a register $\Longrightarrow$ $|\mathtt{X}_i|$ might be the unary or binary length of the number stored in $\mathtt{X}_i$.

General outline – 2

No specification on $(\mathtt{cond})$ or loop statements, except that

- each instance of $(\mathtt{cond})$ be evaluated in polynomial time
- for each instance of a loop statement
  - $\mathtt{loopI}\ \mathtt{X_h}\,[\mathtt{Q}]$, the body $\mathtt{Q}$ is executed $|\mathtt{X}_h|$ times
  - $\mathtt{loopII}\ \mathtt{X_h}\,[\mathtt{Q}]$, the body $\mathtt{Q}$ is executed $2^{|\mathtt{X}_h|}-1$ times
  - during its execution, the contents of $\mathtt{X}_h$ remain unchanged.

Novelty: A new method of certifying polynomial size boundedness (psb) for such imperative programs, provided that all BI's are psb, too.

General outline – 3

Thm (Characterisations).

- FPTIME = Certified string programs (stack programs built from polynomial-time computable BI's).
- FLINSPACE = Certified general loop programs (loop programs built from linear-space computable BI's)
- FPSPACE = Certified power string programs (string programs built from poly-space computable BI's and extended by loopII statements)

Thm (Optimality).

The new method is optimal on core programs, i.e., programs built from honestly certified BI's by sequencing and loops.

Method – 1

Def. A program $\mathtt{P}$ in variables $\mathtt{X}_1, \ldots, \mathtt{X}_n$ is **polynomially size bounded (psb)** iff there are polynomials $p_1, \ldots, p_n \in \mathbb{N}[\mathtt{X}_1, \ldots, \mathtt{X}_n]$ such that

$$\{|\mathtt{X}_1| = s_1, \ldots, |\mathtt{X}_n| = s_n\} \ \mathtt{P} \ \{|\mathtt{X}_i| \le p_i(s_1, \ldots, s_n)\}$$

Call $p_1, \ldots, p_n$ a **polynomial bound (pb) on** $\mathtt{P}$.

- **polynomials**: $c_0 + \ldots + c_i \cdot \mathtt{X}_1^{i_1} \cdot \ldots \cdot \mathtt{X}_n^{i_n} + \ldots$
- **maximum** $p \sqcup q$:

$$\begin{aligned} p &= c_0 + \ldots + c_i \cdot \mathtt{X}_1^{i_1} \cdot \ldots \cdot \mathtt{X}_n^{i_n} + \ldots \\ q &= d_0 + \ldots + d_i \cdot \mathtt{X}_1^{i_1} \cdot \ldots \cdot \mathtt{X}_n^{i_n} + \ldots \end{aligned}$$

$$\Longrightarrow \ p \sqcup q := \max(c_0, d_0) + \max(c_i, d_i) \cdot \mathtt{X}_1^{i_1} \cdot \ldots \cdot \mathtt{X}_n^{i_n} + \ldots$$

Method – 2

Big Question: Given a pb $\vec{q}$ on the body `Q` of a loop `P`,

- what criterion on $\vec{q}$ guarantees the existence of a pb on `P` ?
- how can one construct from $\vec{q}$ a pb on `P` ?

Central to the certification method:

We store and process only a <u>finite</u> amount of information on the <u>class</u> of possible polynomial size bounds for programs.

Let $\vec{p}$ be a pb on a program `P`. Then we store and process only for each

$$p_i(\vec{\mathtt{X}}) = c_0 + \ldots + c_j \cdot \mathtt{X}_1^{j_1} \cdot \ldots \cdot \mathtt{X}_n^{j_n} + \ldots$$

an $(n+1)$-tuple $\langle p_i \rangle$ (the **representation of p_i**) over the **forgetting set** $A := \{0, 1, \infty\}$ ordered by $0 < 1 < \infty$:

Method – 3

$$\text{for } 1 \le j \le n, \quad \langle p_i \rangle[j] := \begin{cases} 0 & \text{if } \mathtt{X}_j \notin p_i \\ 1 & \text{if } p_i = \mathtt{X}_j + q(\vec{\mathtt{X}} \setminus \mathtt{X}_j) \\ \infty & \text{else} \end{cases}$$

$$\langle p_i \rangle[n+1] := \begin{cases} c_0 & \text{if } \kappa(p_i) = c_0 \le 1 \\ \infty & \text{else} \end{cases}$$

For $\vec{a} \in A^{n+1}$, the class of **polynomials of bound** $\vec{a}$ is defined by:

$$\textsc{poly}(\vec{a}) := \{ p(\mathtt{X}_1, \ldots, \mathtt{X}_n) \mid \langle p \rangle \le \vec{a} \}$$

A **certificate for** $\mathtt{P}$ is any $(n+1) \times (n+1)$ matrix Y over A with last row $0^n 1$ such that $\exists$ pb $q_1 \in \textsc{poly}(Y[1]), \ldots, q_n \in \textsc{poly}(Y[n])$ on $\mathtt{P}$.

Example: $\langle \mathtt{P}, \vec{p} \rangle$ with rows $\langle p_1 \rangle, \ldots, \langle p_n \rangle, 0^n 1$ is a certificate for $\mathtt{P}$.

Method – 4 (Examples for $n = 2$)

imperative	bounding polynomial	certificate
$\texttt{push}(\texttt{a}, \texttt{X}_2)$, $\texttt{inc}(\texttt{X}_2)$	$\texttt{X}_2 + 1$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$
$\texttt{pop}(\texttt{X}_\texttt{i})$, $\texttt{reverse}(\texttt{X}_\texttt{i})$	$\texttt{X}_i$	1_3
$\texttt{X}_2 = \texttt{X}_1$	$\texttt{X}_1$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
$\texttt{concat}(\texttt{X}_2, \texttt{X}_1)$, $\texttt{X}_2 \mathrel{+}= \texttt{X}_1$	$\texttt{X}_2 + \texttt{X}_1$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Method – 5

Operations on $A = \{0, 1, \infty\}$.

$+$	0	1	∞
0	0	1	∞
1	1	∞	∞
∞	∞	∞	∞

$\bullet$	0	1	∞
0	0	0	0
1	0	1	∞
∞	0	∞	∞

$\sqcup$	0	1	∞
0	0	1	∞
1	1	1	∞
∞	∞	∞	∞

$+, \bullet, \sqcup$ are commutate, associative, with neutral elements $0, 1, 0$

$\mathcal{M}_n[A]$:= all $(n+1) \times (n+1)$ matrices over A with last row $0^n 1$.

$\rightsquigarrow$ **matrix multiplication** $\otimes$ on $\mathcal{M}_n[A]$, which is associative, preserves the last row property, and with neutral element 1_n

$+, \bullet, \sqcup, <$ denote the componentwise extensions to A^m and $\mathcal{M}_n[A]$

Method – 6

Lemma (Base). If $\texttt{imp}(\vec{\texttt{X}})$ is a basic instruction with pb $p_1, \ldots, p_n$, then $\langle \texttt{imp}(\vec{\texttt{X}}), \vec{p} \rangle$ with rows $\langle p_1 \rangle, \ldots, \langle p_n \rangle, 0^n 1$ is a certificate for $\texttt{imp}(\vec{\texttt{X}})$.

Lemma (Structure). Let u, v be in A^{n+1}.

(a) $u \leq v \implies \mathrm{POLY}(u) \subseteq \mathrm{POLY}(v)$

(b) $p \in \mathrm{POLY}(u)$, $q \in \mathrm{POLY}(v) \implies p \sqcup q \in \mathrm{POLY}(u \sqcup v)$

Corollary (Conditional).

Z_1, Z_2 are certificates for $\texttt{P}_1$, $\texttt{P}_2$ $\implies$

$Z_1 \sqcup Z_2$ is a certificate for $\texttt{if (cond) then P}_1\ \texttt{else P}_2$

$Z_1 \sqcup 1_{n+1}$ is a certificate of $\texttt{if (cond) then P}_1$

Method – 7

Lemma (Composition and Sequence).

(a) $q \in \mathrm{POLY}(v), p_1 \in \mathrm{POLY}(w_1), \ldots, p_n \in \mathrm{POLY}(w_n) \implies$

$$q(p_1, \ldots, p_n) \in \mathrm{POLY}(v \otimes M) \text{ with } M := \begin{pmatrix} w_1 \\ \vdots \\ w_n \\ 0^n 1 \end{pmatrix}.$$

(b) Z_1, Z_2 certificates for $\mathtt{P}_1$, $\mathtt{P}_2$ $\implies$ $Z_2 \otimes Z_1$ certificate for $\mathtt{P}_1; \mathtt{P}_2$

Clearly, (a) implies (b). **Idea for (a)**: for each $i = 1, \ldots n$,

$$(v \otimes M)[i] = v \otimes M[\cdot][i] = \sum_{j=1}^{n} v[j] \bullet M[j][i]$$

gives full information on the usage of $\mathtt{X}_i$ in $q(p_1, \ldots, p_n)$ w.r.t. A.

Method – 8

Big Question. Given a certificate Y for the body `Q` of a loop `P` with variables among $\mathtt{X}_1, \ldots, \mathtt{X}_n$,

- what criterion on Y guarantees a certificate Z for `P` ?
- how can one construct Z from Y ?

Know: For each $m \geq 0$, Y^m is a certificate for $\mathtt{Q}^m$ ($=$`Q`; $\ldots$; `Q`, m times)

$\rightsquigarrow$ It is natural to investigate the following two (finite) **limit forms**:

- $Y^+ := \bigsqcup_{k=1}^{\infty} Y^k$ **non-monotonic**
- $\hat{Y}^* := \bigsqcup_{k=0}^{\infty} \hat{Y}^k$ where $\hat{Y} := 1_{n+1} \sqcup Y$ **monotonic** ($\hat{Y}^i \leq \hat{Y}^{i+1}$)

Control graphs and Certificates – 1

Intuition. An entry $Y[i][j] > 0$ with $i \in \{1, \ldots, n\}$ represents

- a possible data-flow in $\mathtt{P}$ from $\mathtt{X}_j$ to $\mathtt{X}_i$ denoted by $j \rightarrow_Y i$ (read j controls i in Y), for $j = 1, \ldots, n$, or
- a possible influence of a constant on $\mathtt{X}_i$, $(* \rightarrow_Y i)$, for $j = n+1$.

Let $\xrightarrow{+}_Y$ ($\xrightarrow{*}_Y$) denote the transitive (transitive, reflexive) closure of $\rightarrow_Y$. The **control graph $\mathcal{C}_Y$ of Y** is then defined by

$$\mathcal{C}_Y := (\{1, \ldots, n, *\}, \rightarrow_Y).$$

Lemma (Control Graph).

(a) $\mathcal{C}_{\hat{Y}}$ is the reflexive closure of $\mathcal{C}_Y$.

(b) $\mathcal{C}_{Y+}$ is the transitive closure of $\mathcal{C}_Y$.

Control graphs and Certificates – 2

Example. Consider the following stack program $\mathtt{Q}$.

$\mathtt{X}_5 = \mathtt{X}_1;$

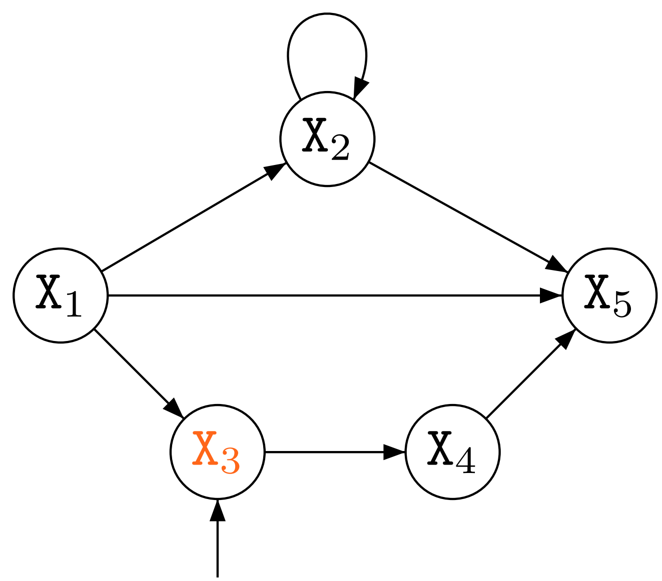
$\mathtt{concat}(\mathtt{X}_5, \mathtt{X}_2);\ \mathtt{concat}(\mathtt{X}_5, \mathtt{X}_4);$

$\mathtt{concat}(\mathtt{X}_2, \mathtt{X}_1);$

$\mathtt{X}_4 = \mathtt{X}_3;\ \mathtt{X}_3 = \mathtt{X}_1;$

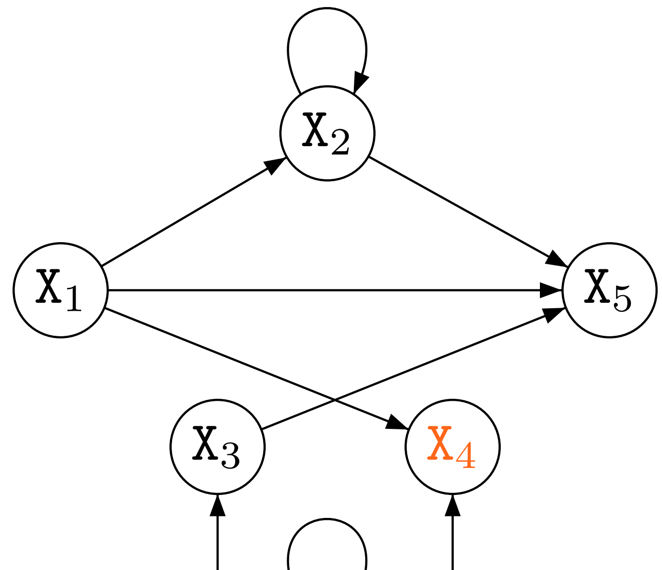
$\mathtt{push}(\mathtt{a}, \mathtt{X}_3);\ \mathtt{nil}(\mathtt{X}_1)$

$$Y := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

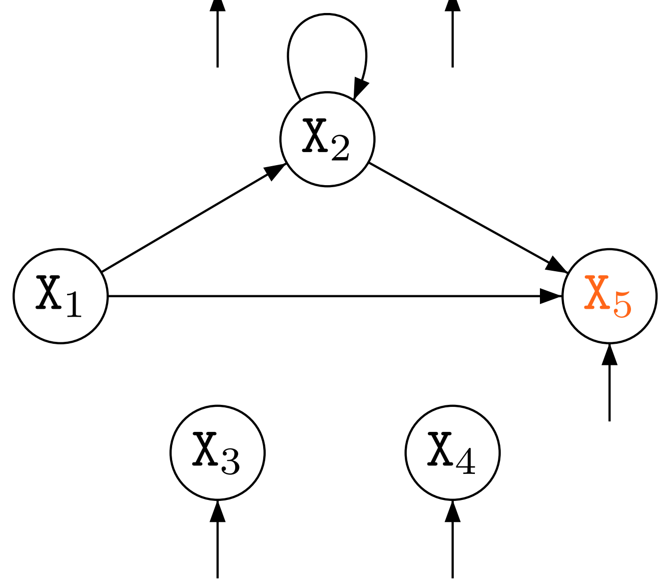


Control graphs and Certificates – 3

$$Y^2 := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



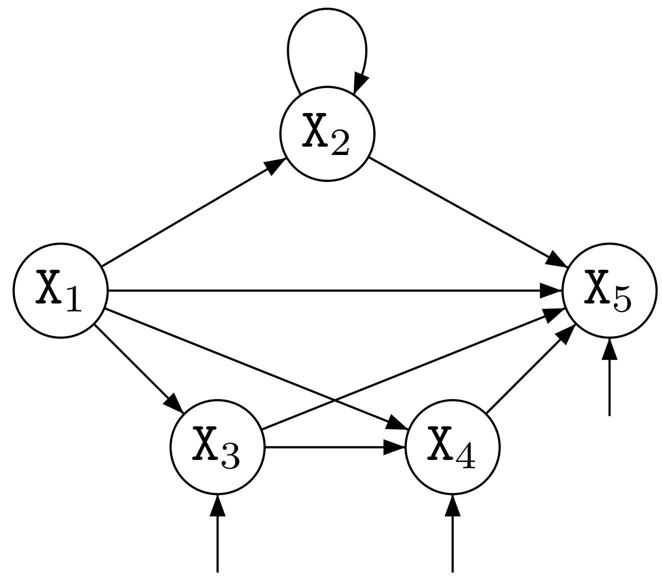
$$Y^3 := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \infty & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



Control graphs and Certificates – 4

Since $Y^3 = Y^4 = Y^5 = \ldots$, we obtain $Y^+ = Y \sqcup Y^2 \sqcup Y^3$, that is:

$$Y^+ = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ \infty & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$



[Kristiansen & N.] If the body `Q` of a loop `P` has no "control circle", then `P` causes no blow up in complexity.

As $\mathcal{C}_{\hat{Y}^*}$ is the reflexive, transitive closure of $\mathcal{C}_Y$, that suggests the following criterion on Y: $\infty \notin \mathrm{Diag}(\hat{Y}^*)$

Method – 9

Lemma (Certificate for loopI).

Let $\mathtt{P} :\equiv \mathtt{loopI}\ \mathtt{X_h}\ [\mathtt{Q}]$ be a program in $\mathtt{X}_1, \ldots, \mathtt{X}_n$.

If Y is a certificate for $\mathtt{Q}$ such that $\infty \notin \mathrm{Diag}(\hat{Y}^*)$,

then $\mathtt{P}$ has a certificate Z of the form

$$Z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \\ 0^n 1 \end{pmatrix} \sqcup 1_{n+1}$$

where $z_1, \ldots, z_n$ fall into two groups:

$$z_i := \begin{cases} \mathrm{add}(z_i) & \text{if } i \in \mathrm{ADD}(Y) \\ \mathrm{else}(z_i) & \text{else} \end{cases}$$

The ADD-case – 1

Def. An $i \in \{1, \ldots, n\}$ is additive in Y, $i \in \mathrm{ADD}(Y)$, if $Y^+[i] \leq 1^n\infty$.

For $i \in \mathrm{ADD}(Y)$ and $j \in \{1, \ldots, n\}$, one has $j \rightarrow_Y i \iff Y[i][j] = 1$, and each $p \in \textsc{poly}(Y^+[i])$ satisfies $p \leq c_i + \sum_{j \rightarrow_Y i} \mathtt{X}_j$.

$\rightsquigarrow$ For $i \in \mathrm{ADD}(Y)$, one can read from $Y^+[i]$ a bp on $\mathtt{X}_i$ w.r.t. `loopI Xh [Q]`, except for the coefficient for $\mathtt{X}_h$, and the constant c_i.

examples for $Y^+[i] \leq 1^n\infty$	bounding polynomial
$\infty \notin Y^+[i]$ (e.g. variable assign.)	$Y^+[i][n+1] + \sum_j Y^+[i][j] \cdot \mathtt{X}_j$
$Y^+[i] = 0^n\infty$ (e.g. const. assign.)	$c_i \geq 0$
$Y^+[i] = 0^{i-1}10^{n-i}\infty$ (e.g. push/inc)	$\mathtt{X}_i + c_i \cdot \mathtt{X}_h$

The ADD-case – 2

In the general situation of an additive i in Y,
$\mathtt{X}_i$ will depend on other variables $\mathtt{X}_j$ for which j is also additive in Y.

In those cases, there will be an accumulation of values for c_h or c_i.

Example. Consider the following program

$$\mathtt{Q} :\equiv \mathtt{push}(\mathtt{a}, \mathtt{X}_\mathtt{j}); \mathtt{X}_i = \mathtt{X}_j; \mathtt{push}(\mathtt{a}, \mathtt{X}_\mathtt{i}).$$

Then $\mathtt{X}_i$ w.r.t. $\mathtt{Q}$ is polynomially bounded by $(\mathtt{X}_j + 1) + 1$.

Hence $\mathtt{X}_i$ w.r.t. $\mathtt{loopI}\ \mathtt{X}_\mathtt{h}\,[\mathtt{Q}]$ is polynomially bounded by
$\mathtt{X}_i + (\mathtt{X}_j + 1) + 1$.

The ADD-case – 3

Def. Let $Y \in \mathcal{M}_n[A]$ be any possible certificate.

$V_Y(i) := \{j \in \{1,\dots,n\} \setminus \{i\} \mid j \rightarrow_Y i\}$ (imme. predecessors of i in Y)

$V_Y^+(i) := \{j \in \{1,\dots,n\} \setminus \{i\} \mid j \overset{+}{\rightarrow}_Y i\}$ (predecessors of i in Y)

Lemma (Partial Ordering I and ADD-Closure).
If $\infty \notin \mathrm{Diag}(\hat{Y}^*)$, then the following holds:

- $\overset{*}{\rightarrow}_Y$ is a partial ordering (po) of $\{1,\dots,n\}$
- $i \in \mathrm{ADD}(Y)$ and $j \overset{+}{\rightarrow}_Y i \implies j \in \mathrm{ADD}(Y)$
- $i \in \mathrm{ADD}(Y)$ and $i \rightarrow_Y i \implies \forall j \in V_Y^+(i) \colon j \not\rightarrow_Y j$

Thus, one can proceed by induction on $\overset{*}{\rightarrow}_Y$, and the ADD-case can be treated seperately.

The ADD-case – 4

For $i \in \mathrm{ADD}(Y)$ one defines

$$\mathrm{add}(z_i) := (y_1, \ldots, y_{h-1}, Y^*[i][h] + K_i, y_{h+1}, \ldots, y_n, C_i)$$

where $y_j := Y^*[i][j]$ for $j = \{1, \ldots, n\} \setminus \{h\}$, and

$$C_i := \begin{cases} 0 & \text{if } Y[i][i] = 1 \\ Y[i][n+1] + \sum\limits_{j \in V_Y(i)} C_j & \text{if } Y[i][i] = 0 \end{cases}$$

$$K_i := \begin{cases} Y[i][n+1] + \sum\limits_{j \in V_Y(i)} C_j & \text{if } Y[i][i] = 1 \\ \sum\limits_{j \in V_Y(i)} K_j & \text{if } Y[i][i] = 0 \end{cases}$$

The ADD-case – 5

Proof sketch. For $i \in \mathrm{ADD}(Y)$, and for each round $m \geq 0$ define

$$q_{i,m}(\vec{\mathtt{X}}) := \sum_{j=1}^{n} Y^m[i][j] \cdot \mathtt{X}_j + K_i \cdot m + C_i.$$

By induction on m, simultaneously for all $i \in \mathrm{ADD}(Y)$, one obtains:

$$\{|\vec{\mathtt{X}}| = \vec{s}\}\ \mathtt{Q}^m\ \{|\mathtt{X}_i| \leq q_{i,m}(\vec{s})\}$$

As $Y^{s_h}[i][j] \leq Y^*[i][j]$, a polynomial bound $p_i \in \mathrm{POLY}(\mathrm{add}(z_i))$ on $|\mathtt{X}_i|$ w.r.t. `loopI` $\mathtt{X_h}$ [`Q`] is obtained by setting

$$p_i(\vec{\mathtt{X}}) := \sum_{j=1}^{n} Y^*[i][j] \cdot \mathtt{X}_j + K_i \cdot \mathtt{X}_h + C_i$$

recalling $\mathrm{add}(z_i) = (y_1, \ldots, y_{h-1}, Y^*[i][h] + K_i, y_{h+1}, \ldots, y_n, C_i)$. $\square$

The Else-case

For $i \notin \mathrm{ADD}(Y)$, we know $i \neq h$ (since $h \in \mathrm{ADD}(Y)$), and we construct (in the style of [Kristiansen & N.] a polynomial bound p_i on $|\mathtt{X}_i|$ w.r.t. `loopI X_h [Q]` of the form

$$p_i = \mathtt{X}_i + \mathtt{X}_h \cdot q_i$$

for some polynomial q_i in $\{\mathtt{X}_j \mid j \in V_Y(i)\} \cup \{\mathtt{X}_h\}$.

Therefore, we define $\mathrm{else}(z_i)$ componentwise as follows:

$$\mathrm{else}(z_i)[j] := \begin{cases} 1 & \text{if } j = i \\ \infty & \text{if } j = h \\ 0 & \text{if } j = n+1 \\ \hat{Y}^*[i][j] & \text{else.} \end{cases}$$

For the “else” case, $\hat{Y}^*[i][j]$ is provably in $\{\infty, 0\}$. $\square$

Method – 10

Recall that for $\texttt{loopII}\ \texttt{X}_\texttt{h}\ [\texttt{Q}]$, the body $\texttt{Q}$ is executed $2^{|\texttt{X}_h|}-1$ times.

Lemma (Partial Ordering II).
For any $Y \in \mathcal{M}_n[A]$, if

$$\text{(II)} \qquad Y^+[i] = 1_{n+1}[i] \text{ or } Y^+[i][i] = 0 \text{ for i=1, \dots, n}$$

then $\rightarrow_{Y^*}$ is a partial ordering of $\{1,\dots,n\}$. $\qquad (Y^* := 1_n \sqcup M^+)$

Lemma (Certificate for loopII).
Let $\texttt{P} :\equiv \texttt{loopII}\ \texttt{X}_\texttt{h}\ [\texttt{Q}]$ be a program in $\texttt{X}_1,\dots,\texttt{X}_n$.
If Y is a certificate for $\texttt{Q}$ with (II), then Y^* is a certificate for $\texttt{P}$.

Method – 11

Proof. For $m \geq 0$ define $q_{1,m}(\vec{\mathtt{X}}), \ldots, q_{n,m}(\vec{\mathtt{X}})$ such that for $i = 1, \ldots, n$:

(A) $\{|\vec{\mathtt{X}}| = \vec{s}\}\ \mathtt{Q}^m\ \{|\mathtt{X}_i| \leq q_{i,m}(\vec{s})\}$ for all $m \geq 0$

(B) $q_{i,m} \in \text{POLY}(Y^m[i])$ for all $m > 0$

(C) $\exists\, m_i \geq 0$ such that $q_{i,m} = q_{i,m_i}$ for all $m \geq m_i$.

One obtains a pb $p_1 \in \text{POLY}(Y^*[1]), \ldots, p_n \in \text{POLY}(Y^*[n])$ on $\mathtt{P}$ by

$$p_i := \bigsqcup_{m \leq m_i} q_{i,m}$$

By (B) and Structure: $p_i \in \text{POLY}(\bigsqcup_{m \leq m_i} Y^m[i]) \subseteq \text{POLY}(Y^*[i])$

By (A) and (C): $\{|\vec{\mathtt{X}}| = \vec{s}\}\ \mathtt{P}\ \{|\mathtt{X}_i| \leq q_{i,2^{s_h}-1}(\vec{s}) \leq p_i(\vec{s})\}$.

Method – 12

Construction of $q_{1,m}(\vec{\mathtt{X}}), \ldots, q_{n,m}(\vec{\mathtt{X}})$ satisfying (A),(B),(C): Since Y is a certificate for Q, there exist $q'_1 \in \mathrm{POLY}(Y[1]), \ldots, q'_n \in \mathrm{POLY}(Y[n])$ s.t.:

$$(*) \qquad \{|\vec{\mathtt{X}}| = \vec{s}\}\ \mathtt{Q}\ \{|\mathtt{X}_j| \leq q'_j(\vec{s})\}$$

We define the required polynomials $q_{i,0}, q_{i,1}, q_{i,2}, \ldots$ inductively by:

$$q_{i,0} := \mathtt{X}_i$$

$$q_{i,m+1} := q'_i(r_1, \ldots, r_n), \text{ where } r_j := \begin{cases} q_{j,m} & \text{if } j \in V_Y(i) \cup \{i\} \\ 0 & \text{else.} \end{cases}$$

(A), (B) follow by an easy induction on $m \geq 0$, using $(*)$ and Sequence:
$q'_i \in \mathrm{POLY}(Y[i]), r_1, \ldots, r_n \in \mathrm{POLY}(Y^m[i]) \implies q_{i,m+1} \in \mathrm{POLY}(Y \otimes Y^m[i])$

(C) follows by easy induction on $\xrightarrow{*}_Y$. $\square$

Embedding of μ-measure 0 programs

Lemma (Embedding of Control).
Let $\mathtt{P}$ be any loop/stack program, and let $Z \in \mathcal{M}_n[A]$ be a certificate for $\mathtt{P}$. Then for all $i, j \in \{1, \ldots, n\}$, one has:

(a) $Z[i][j] \geq 1$ and $i \neq j \implies \mathtt{X}_j \twoheadrightarrow_{\mathtt{P}} \mathtt{X}_i$

(b) $Z[i][i] = \infty \implies \mathtt{X}_i \twoheadrightarrow_{\mathtt{P}} \mathtt{X}_i$ (top circle!)

Note. (a), (b) are distinct cases, e.g. $\mathtt{P} :\equiv \mathtt{foreach}\ \mathtt{X_h}\ [\mathtt{pop}(\mathtt{X_i})]$. As $M(\mathtt{pop}(\mathtt{X_i})) = 1_{n+1}$, we obtain $M(\mathtt{P}) = 1_{n+1}$; hence $M(\mathtt{P})[i][i] = 1$, but $\mathtt{X}_i \not\twoheadrightarrow_{\mathtt{P}} \mathtt{X}_i$.

Theorem (Embedding of μ-measure 0).
Every loop/stack program of μ-measure 0 has a certificate.

Optimality

Fact ([Kristiansen & N.]). There is no certification method for the psb-property that certifies all psb programs.

Def. A certificate $Z \in \mathcal{M}_n[A]$ for a program $\mathtt{P}$ is **honest**, if there exist constants $\vec{m} \geq \vec{1}$ such that for all $i \neq j$ in $\{1, \dots, n\}$,

$$\begin{aligned}
Z[i][i] \geq 1 &\implies \{|\vec{\mathtt{X}}| = \vec{s} \geq \vec{m}\}\ \mathtt{P}\ \{|\mathtt{X}_i| \geq s_i\}\\
Z[i][j] \geq 1 &\implies \{|\vec{\mathtt{X}}| = \vec{s} \geq \vec{m}\}\ \mathtt{P}\ \{|\mathtt{X}_i| \geq s_i + s_j\}\\
Z[i][i] = \infty &\implies \{|\vec{\mathtt{X}}| = \vec{s} \geq \vec{m}\}\ \mathtt{P}\ \{|\mathtt{X}_i| \geq 2 \cdot s_i\}\\
Z[i][n+1] \geq 1 &\implies \{|\vec{\mathtt{X}}| = \vec{s} \geq \vec{m}\}\ \mathtt{P}\ \{|\mathtt{X}_i| \geq s_i + 1\}.
\end{aligned}$$

Core programs are built from honestly certified basic instructions by sequencing and loop statements.

Thm (Optimality). For core programs $\mathtt{P}$, the following holds true:

$$\mathtt{P} \text{ has a certificate} \iff \mathtt{P} \text{ is psb}$$

Examples of certified algorithms

- elementary school method algorithms like

  BINARY ADDITION

  BINARY MULTIPLICATION

- benchmark algorithms like INSERTION SORT

There exists a Java applet of the present certification method by

Jan Mehler (TU-Ilmenau)

Everybody is invited to a demonstration
at any time !