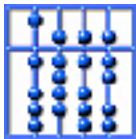


Programm-Analyse mittels linearer Algebra

Helmut Seidl, München

Markus Müller-Olm, Hagen



Tag der Informatik, 2003



Lineare Algebra:

- Vektoren;
- Vektorräume;

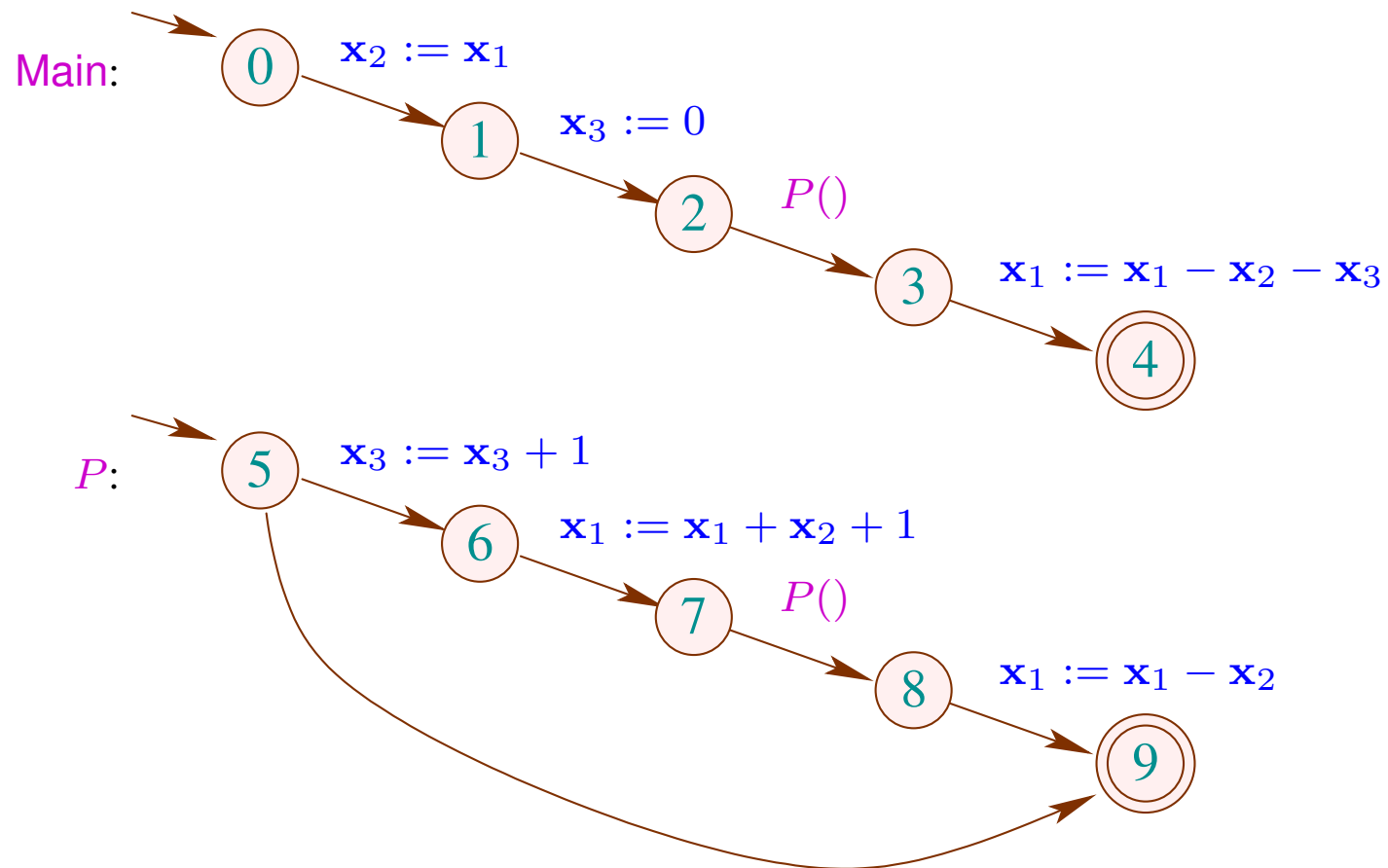
Lineare Algebra:

- Vektoren;
- Vektorräume;
- Matrizen;
- Vektorräume von Matrizen;

Lineare Algebra:

- Vektoren;
- Vektorräume;
- Matrizen;
- Vektorräume von Matrizen;
- lineare Transformationen von Vektorräumen von Matrizen;
- ... :-)

Programmanalyse:



Fragen:

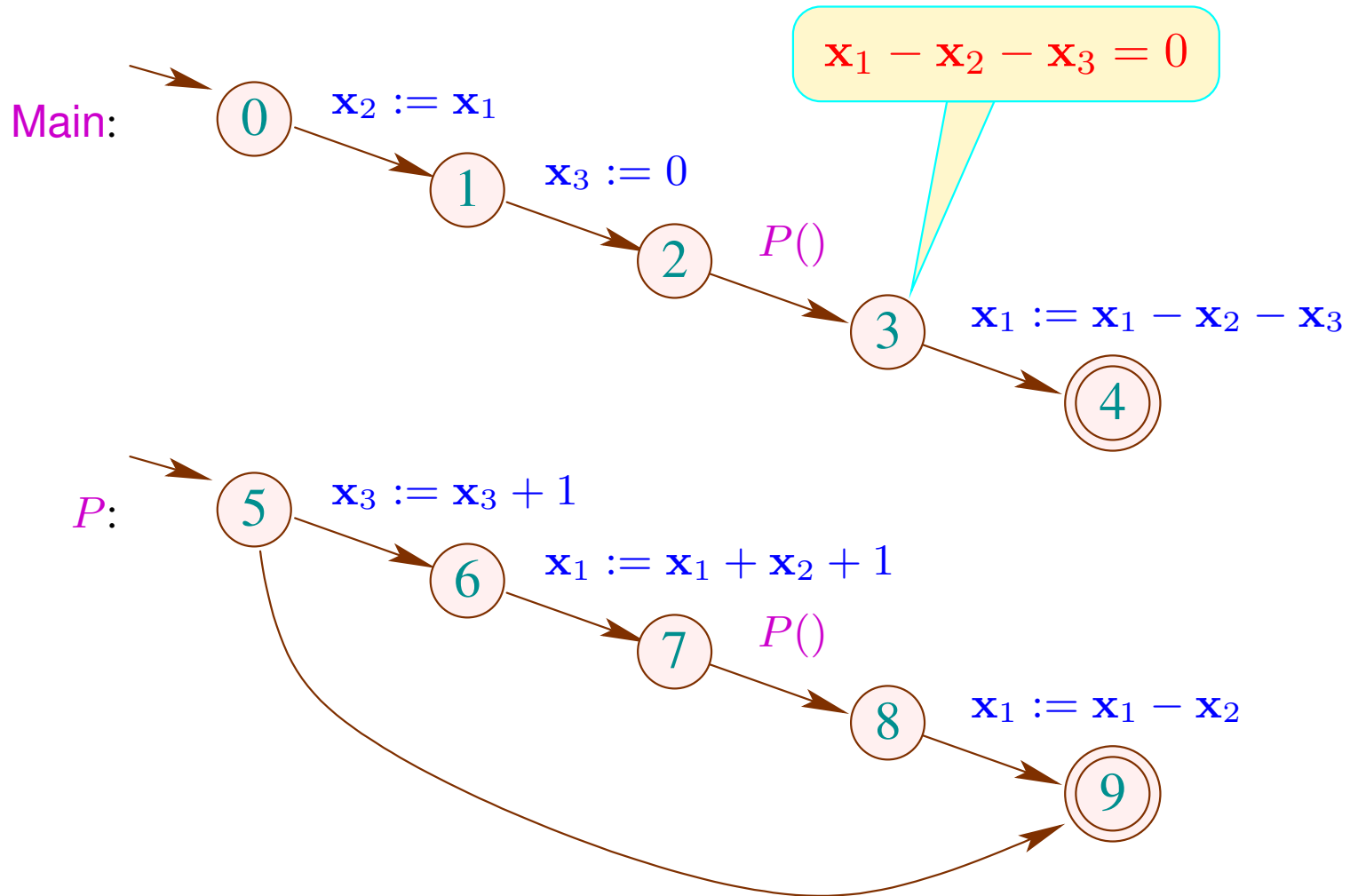
- Welchen Wert hat x_1 am Ende des Programms?
- Wo gilt $x_1 - x_2 = 0$?
- Welche Beziehung besteht zwischen x_1, x_2 und x_3 am Programmpunkt 3 ?

Fragen:

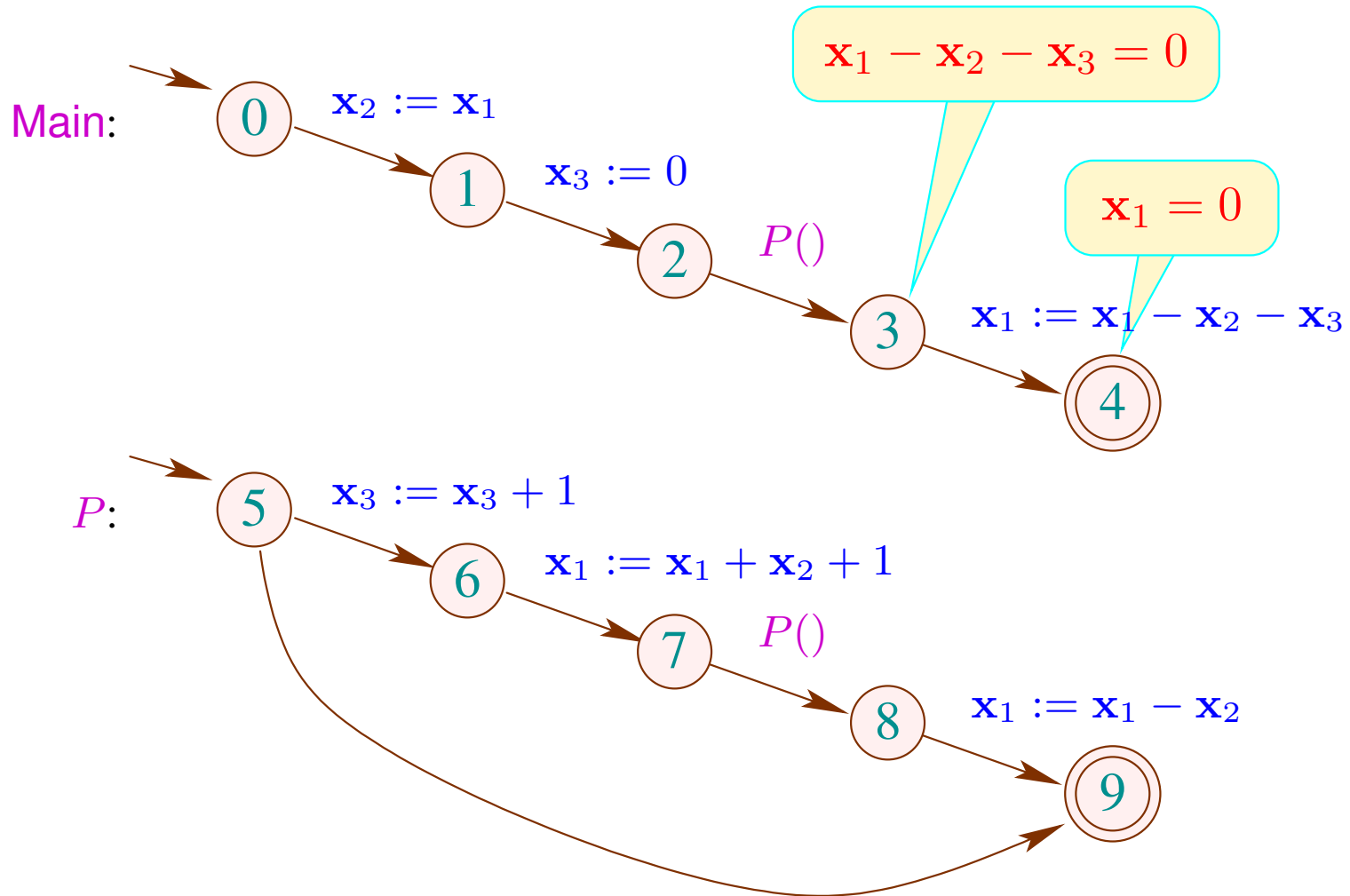
- Welchen Wert hat x_1 am Ende des Programms?
- Wo gilt $x_1 - x_2 = 0$?
- Welche Beziehung besteht zwischen x_1, x_2 und x_3 am Programmpunkt 3 ?

⇒ affine Relationen

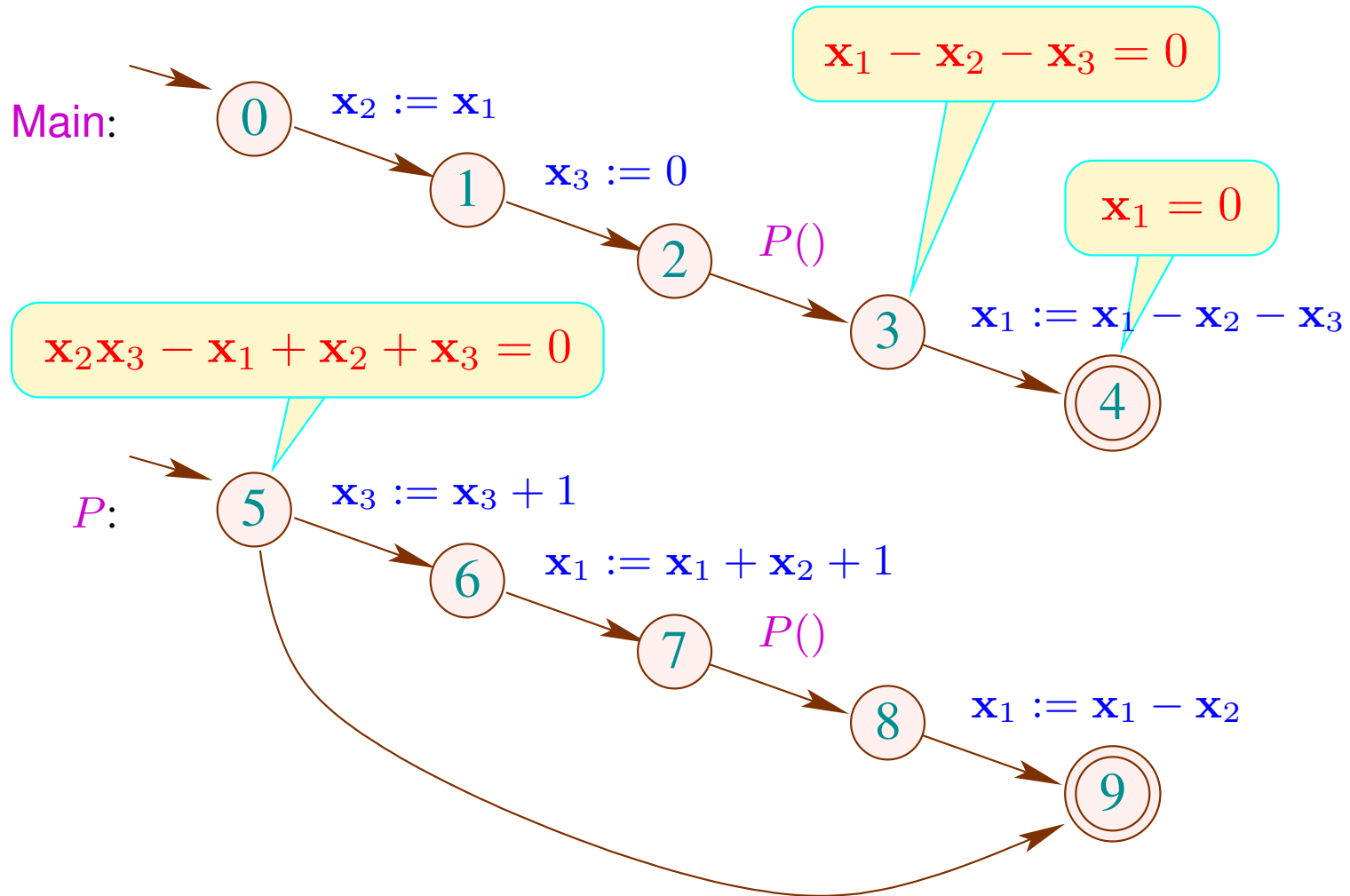
Beispiel (Forts.):



Beispiel (Forts.):



Beispiel (Forts.):



Abstraktion:

Affine Programme ...

Polynomielle Programme ...

Abstraktion:

Affine Programme ... bestehen aus

- affinen Zuweisungen: $\mathbf{x}_1 := \mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3;$
- unbekannten Zuweisungen: $\mathbf{x}_i := ?$
für zu komplexe Zuweisungen :-)
- nicht-deterministischer statt bedingter Auswahl :-)

Polynomielle Programme ...

Abstraktion:

Affine Programme ... bestehen aus

- affinen Zuweisungen: $\mathbf{x}_1 := \mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3;$
- unbekanntem Zuweisungen: $\mathbf{x}_i := ?$
für zu komplexe Zuweisungen :-)
- nicht-deterministischer statt bedingter Auswahl :-)

Polynomielle Programme ... bestehen aus

- polynomiellen Zuweisungen: $\mathbf{x}_1 := \mathbf{x}_1 \mathbf{x}_2 - \mathbf{x}_3$
- ... sonst wie bei affinen Programmen :-))

Ziel:

Bestimme unter dieser Abstraktion,

- ... alle gültigen affinen Identitäten zwischen Variablen;
- ... alle gültigen polynomiellen Identitäten zwischen Variablen ...

Ziel:

Bestimme unter dieser Abstraktion,

- ... alle gültigen affinen Identitäten zwischen Variablen;
- ... alle gültigen polynomiellen Identitäten zwischen Variablen ...

und alles natürlich blitzschnell :-))

Hintergrund:

intraprozedural:

Karr	affine Programme + affine Relationen	1976
MMO., Rüthing	affine Programme + affine Rel.*)	2001
MMO., S.	polynomielle Programme + polynomielle Rel.*)	2002

*) nur testen ...

Hintergrund:

intraprozedural:

Karr	affine Programme + affine Relationen	1976
MMO., Rüthing	affine Programme + affine Rel.	2001
MMO., S.	polynomielle Programme + polynomielle Rel.*)	2003

interprozedural:

Horwitz et al.	lineare Konstanten	1996
MMO., S.	affine Programme + polynomielle Rel.	2003

*) auch inferieren ...

Anwendungen:

- aggressive Programm-Optimierungen;
- Programm-Verifikation :-))

Teil 1

Affine Programme

Die konkrete Semantik

Jeder Ausführungspfad π bewirkt eine **affine Transformation** des Programm-Zustands:

$$\llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket(\underline{v}) =$$

$$\llbracket \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket(\llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1 \rrbracket(\underline{v})) =$$

$$\llbracket \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket \left(\left(\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) \right)$$

Die konkrete Semantik

Jeder Ausführungspfad π bewirkt eine **affine Transformation** des Programm-Zustands:

$$\llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket(\underline{v}) =$$

$$\llbracket \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket(\llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1 \rrbracket(\underline{v})) =$$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Eine affine Relation: $\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 = 0$ kann als Vektor aufgefasst werden:

$$\begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \end{pmatrix}$$

Jeder Ausführungspfad π bewirkt eine **lineare Transformation** der Nach-Bedingung in die Vor-Bedingung:

$$\begin{aligned} \llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket^\top(\underline{a}) &= \\ \llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1 \rrbracket^\top(\llbracket \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket^\top(\underline{a})) &= \end{aligned}$$

$$\llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1 \rrbracket^\top \left(\left(\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \right) \right)$$

$$\begin{aligned} \llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket^T(\underline{a}) &= \\ \llbracket \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1 \rrbracket^T(\llbracket \mathbf{x}_3 := \mathbf{x}_3 + 1 \rrbracket^T(\underline{a})) &= \end{aligned}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

Darum,

$$\left\{ \begin{array}{l} \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \\ \mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 = 0 \end{array} \right\}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}$$

Darum,

$$\{\mathbf{x}_1 - \mathbf{x}_3 = 0\} \quad \mathbf{x}_1 := \mathbf{x}_1 + \mathbf{x}_2 + 1; \mathbf{x}_3 := \mathbf{x}_3 + 1 \quad \{\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{x}_3 = 0\}$$

Beobachtung I:

→ Die einzige gültige Relation bei Programm-Start ist:

$$0 = 0$$

→ Die Relation $a_0 + a_1\mathbf{x}_1 + \dots + a_k\mathbf{x}_k = 0$ ist gültig am Programm-Punkt v gdw.

$$0 = \llbracket \pi \rrbracket^T(\underline{a})$$

für jedes π , das v erreicht.

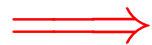
Beobachtung II:

Die folgenden Aussagen sind für \underline{a} äquivalent:

$$0 = W\underline{a} \quad \text{für alle } W \in \mathcal{W} \quad (1)$$

$$0 = W\underline{a} \quad \text{für alle } W \in \text{Span } \mathcal{W} \quad (2)$$

$$0 = W\underline{a} \quad \text{für alle } W \text{ einer Basis von Span } \mathcal{W} \quad (3)$$



Beobachtung III:

Die Menge aller am Programm-Punkt v gültigen affinen Relationen \underline{a} ist gleich der Menge der Lösungen des linearen Gleichungssystems:

$$0 = W\underline{a} \quad , \quad W \in \mathcal{B}$$

für eine Basis \mathcal{B} von:

$$\text{Span} \{ [\pi]^\top \mid \pi \text{ erreicht } v \} \quad (4)$$



Es genügt, die Vektorräume (4) zu berechnen :-)

Beobachtung IV:

Die Menge aller Unterräume von $\mathbb{Q}^{k \times k}$ bilden einen
vollständigen Verband:

Ordnung:

$$\sqsubseteq = \subseteq$$

kleinstes Element:

$$\{0\}$$

kleinste obere Schranke: $M_1 \sqcup M_2 = \text{Span}(M_1 \cup M_2)$

Höhe:

$$k^2$$

Ein Ungleichungs-System für Pfade

Balancierte Pfade:

$S(v) \supseteq \{\epsilon\}$	v Eintrittspunkt
$S(p) \supseteq S(v)$	v Endpunkt von p
$S(v) \supseteq S(u) ; S(u, v)$	(u, v) Zuweisungs-Kante
$S(v) \supseteq S(u) ; S(p)$	(u, v) ruft Prozedur p auf

Die Menge $S(u, v)$ für Zuweisungen ist gegeben durch:

$\{\mathbf{x}_i := t\}$ falls die Zuweisung $\mathbf{x}_i := t$ ist

$\{\mathbf{x}_i := d \mid d \in \mathbb{Q}\}$ falls die Zuweisung $\mathbf{x}_i := ?$ ist

Die Menge $S(u, v)$ für Zuweisungen ist gegeben durch:

$\{\mathbf{x}_i := t\}$ falls die Zuweisung $\mathbf{x}_i := t$ ist

$\{\mathbf{x}_i := d \mid d \in \mathbb{Q}\}$ falls die Zuweisung $\mathbf{x}_i := ?$ ist

Letztere **unendliche** Menge ist (bzgl. affiner Relationen)
äquivalent zu der **endlichen** Menge:

$$\{\mathbf{x}_i := 0, \mathbf{x}_i := 1\}$$

:-)

Ankommende Pfade:

$$\begin{array}{lll} R(v) \supseteq S(v) & & v \text{ in Main} \\ R(v) \supseteq R(p); S(v) & & v \text{ in } p \\ R(p) \supseteq R(u) & & (u, _) \text{ calls } p \end{array}$$

Fakt

Span commutiert mit “ \cup ” und “ \circ ”:

$$\mathbf{Span} (M_1 \cup M_2) = \mathbf{Span} (\mathbf{Span} M_1 \cup \mathbf{Span} M_2)$$

$$\mathbf{Span} (M_1 \circ M_2) = \mathbf{Span} (\mathbf{Span} M_1 \circ \mathbf{Span} M_2)$$

für beliebige Mengen M_i von Matrizen :-)

Theorem

- Der Vektorraum der Matrizen

$$\alpha(R(v)) = \mathbf{Span} \{ \llbracket \pi \rrbracket^T \mid \pi \in R(v) \}$$

kann exakt berechnet werden.

- Die Menge aller gültigen affinen Relationen kann exakt berechnet werden.
- Die Laufzeit ist **linear** in der Programm-Größe und **polynomiell** in der Anzahl der Variablen.

Erweiterungen

- lokale Variablen, Parameter, Rückgabe-Werte :-)
- Inferenz aller gültigen **polynomiellen** Relationen etwa:

$$\mathbf{x}_2\mathbf{x}_3 - \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 = 0$$

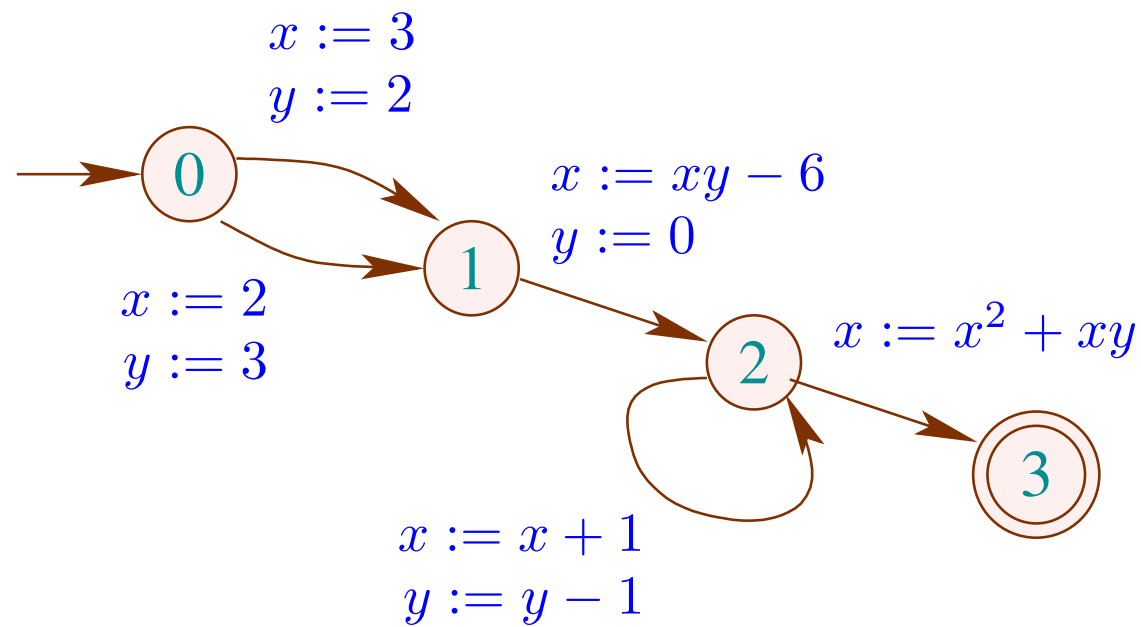
... des Grads höchstens d :-)

- **polynomielle** Programme ???

Teil 2

Polynomielle Programme

Beispiel:



Idee:

MMO., S. 2002

- Verallgemeinere Vektorräume zu **Idealen**:

Ein Ideal $I \subseteq \mathbb{Q}[x_1, \dots, x_k]$ ist eine Menge von Polynomen mit:

$$q_1, q_2 \in I \text{ impliziert } q_1 + q_2 \in I;$$

$$q \in I \text{ impliziert } r \cdot q \in I \text{ für alle Polynome } r.$$

Idee:

MMO., S. 2002

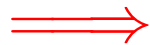
- Verallgemeinere Vektorräume zu **Idealen**:

Ein Ideal $I \subseteq \mathbb{Q}[x_1, \dots, x_k]$ ist eine Menge von Polynomen mit:

$$q_1, q_2 \in I \text{ impliziert } q_1 + q_2 \in I;$$

$$q \in I \text{ impliziert } r \cdot q \in I \text{ für alle Polynome } r.$$

- Jedes Polynom-Ideal ist **endlich erzeugt**.



Jede aufsteigende Kette von Polynom-Idealen wird stabil :-)

Ein Ungleichungs-System für WP:

$$\begin{aligned} I(v_0) &\supseteq \langle p \rangle && p \text{ zu testende Relation} \\ I(u) &\supseteq \llbracket \text{lab}(u, v) \rrbracket^T(I(v)) && (u, v) \text{ Kante} \end{aligned}$$

wobei:

$$\begin{aligned} \llbracket x := t \rrbracket^T(p) &= \langle p[x \mapsto t] \rangle \\ \llbracket x := ? \rrbracket^T(p) &= \langle p_0, \dots, p_k \rangle \\ \text{if } p &= p_0 + p_1 \cdot x + \dots + p_k \cdot x^k \end{aligned}$$

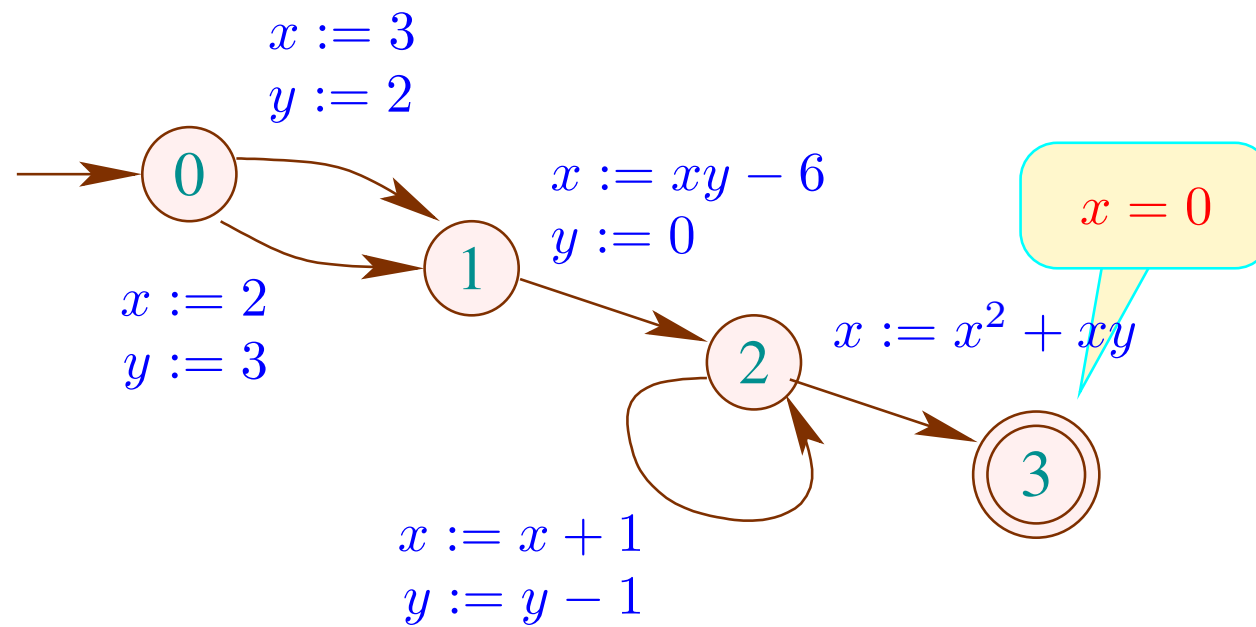
Theorem

- Relation $p = 0$ gilt am Punkt v_0 gdw.

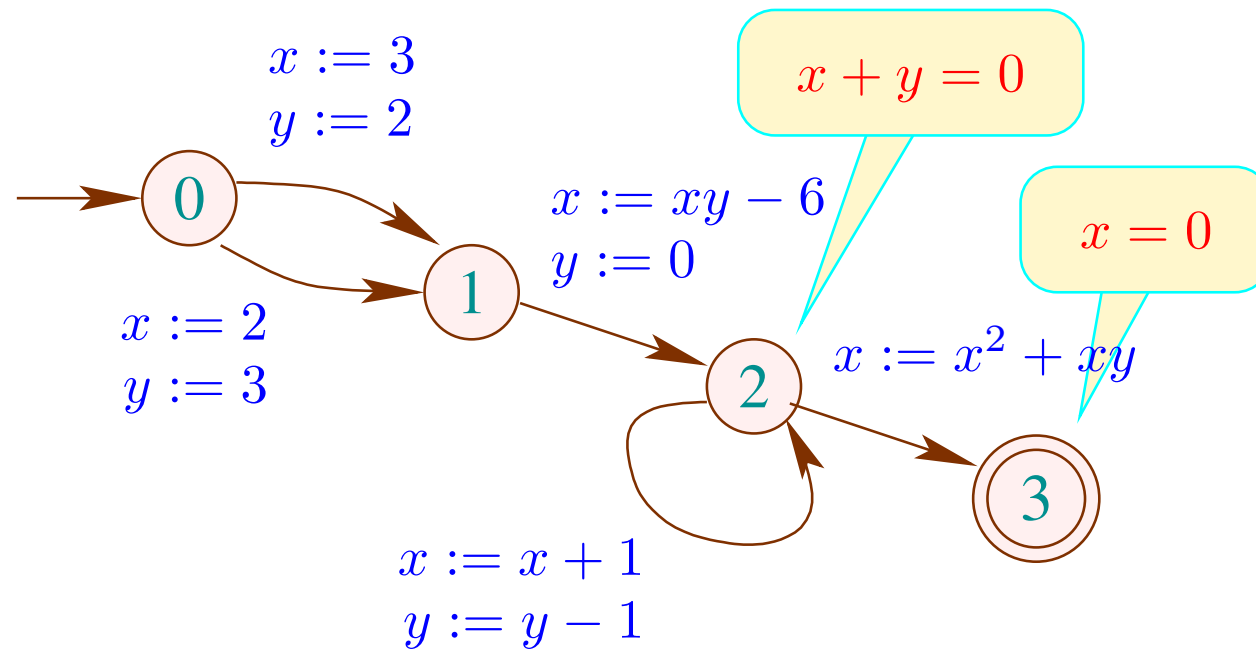
$$\langle 0 \rangle \supseteq I(\text{start})$$

- Die Ideale $I(v)$ sind berechenbar.

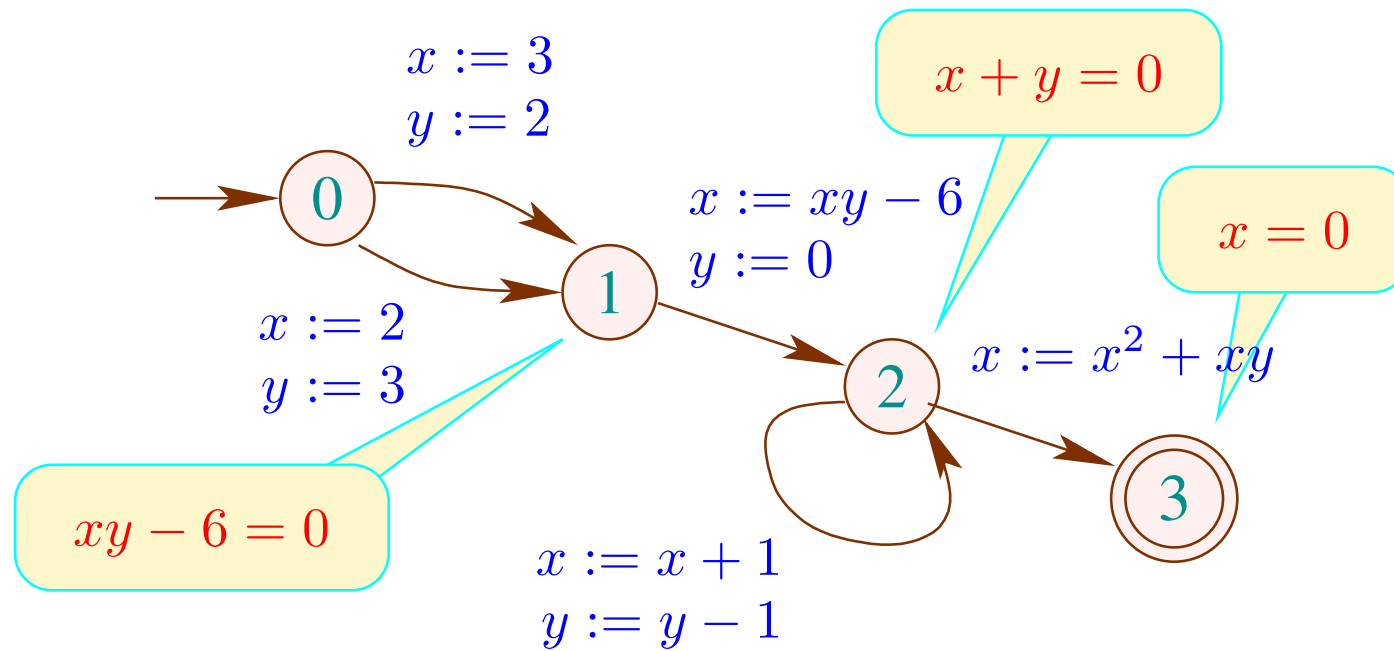
Beispiel:



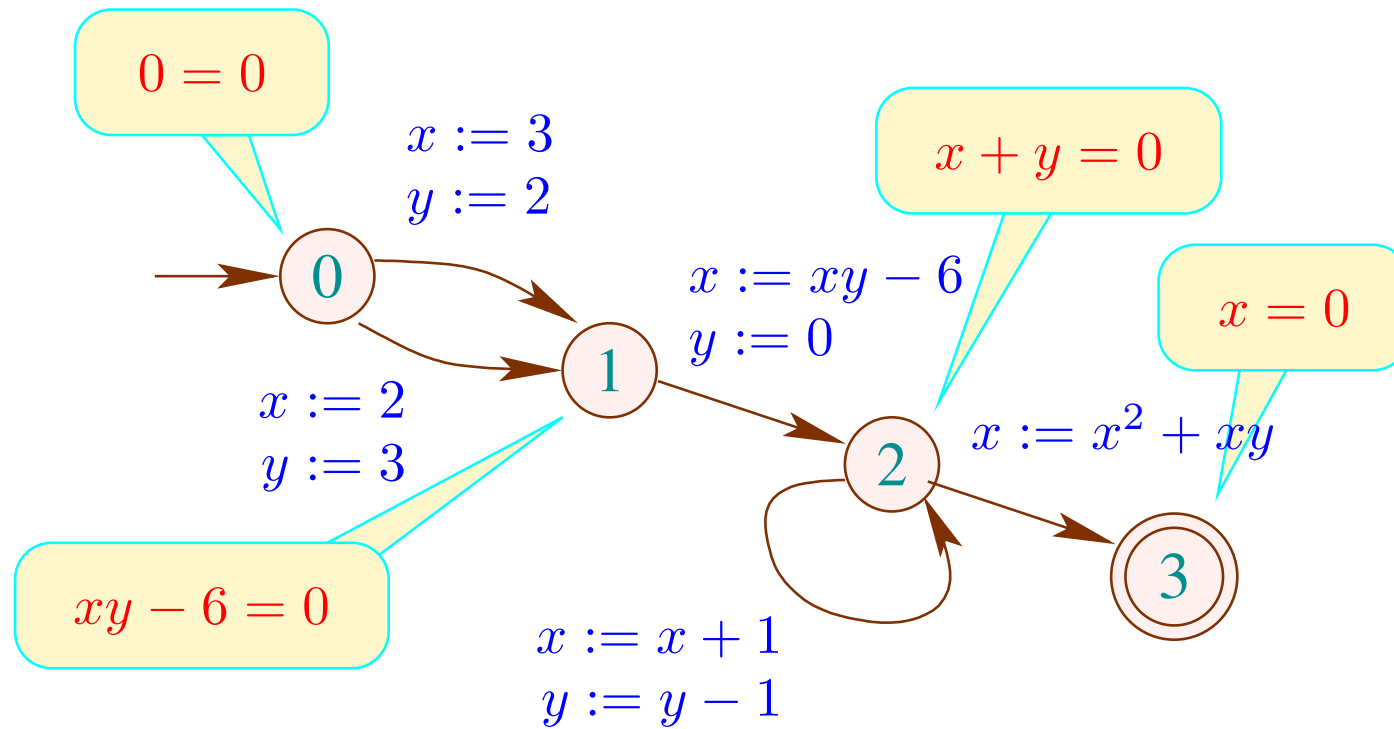
Beispiel:



Beispiel:



Beispiel:



Frage:

Wie inferiert man **unbekannte** Relationen??

Frage:

Wie inferiert man unbekannte Relationen??

Idee:

Betrachte das generische Polynom vom Grad d :

$$p_d = \sum_{j_1 + \dots + j_k \leq d} a_{j_1, \dots, j_k} \cdot x_1^{j_1} \cdots x_k^{j_k}$$

$(a_{j_1, \dots, j_k}$ neue Variablen)

Beobachtung:

Die generische schwächste Vorbedingung an start ist ein Ideal $\langle p_1, \dots, p_n \rangle$ wobei

$$p_i = \sum_{j_1, \dots, j_k} b_{j_1, \dots, j_k}^{(i)} \cdot x_1^{j_1} \cdot \dots \cdot x_k^{j_k}$$

für lineare Kombinationen $b_{j_1, \dots, j_k}^{(i)}$ der $a_{j'_1, \dots, j'_k} \text{ :-}$)

Theorem

- Die folgenden zwei Aussagen sind äquivalent:
 - (1) $p_d = 0$ gilt an v_0 ;
 - (2) $b_{j_1, \dots, j_k}^{(i)} = 0$ für alle i, j_1, \dots, j_k .
- Die Menge aller gültigen polynomiellen Relationen (vom Grad höchstens d) an v_0 ist berechenbar :-)

Offene Probleme:

- Kann die Grad-Beschränkung beseitigt werden?
- Kann die Analyse polynomieller Programme interprozedural verallgemeinert werden?
- ... wenigstens in einigen (nützlichen) Spezialfällen?
- Kann die Analyse auf parallele Programme erweitert werden?
- Was ist die genaue Komplexität?
- ... zumindest in (nützlichen) Spezialfällen?