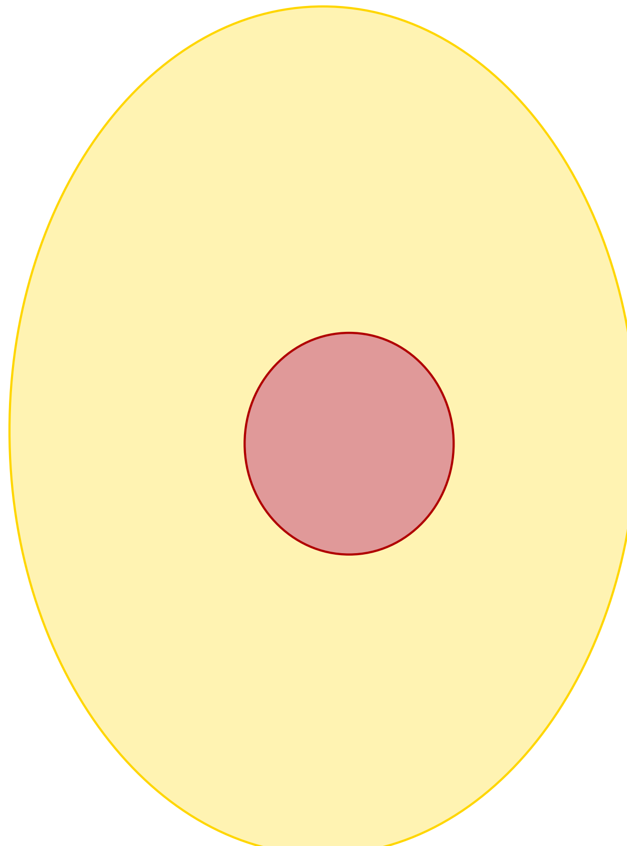
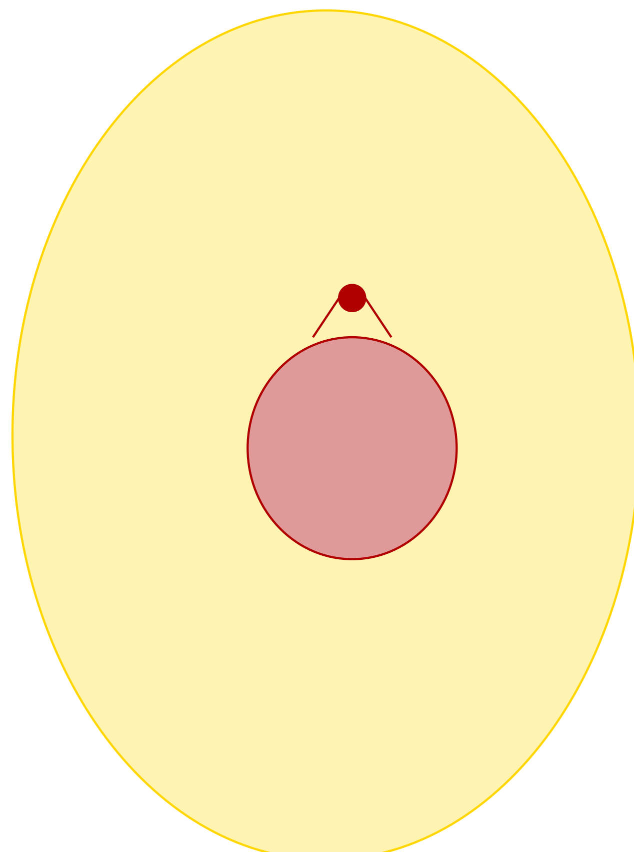
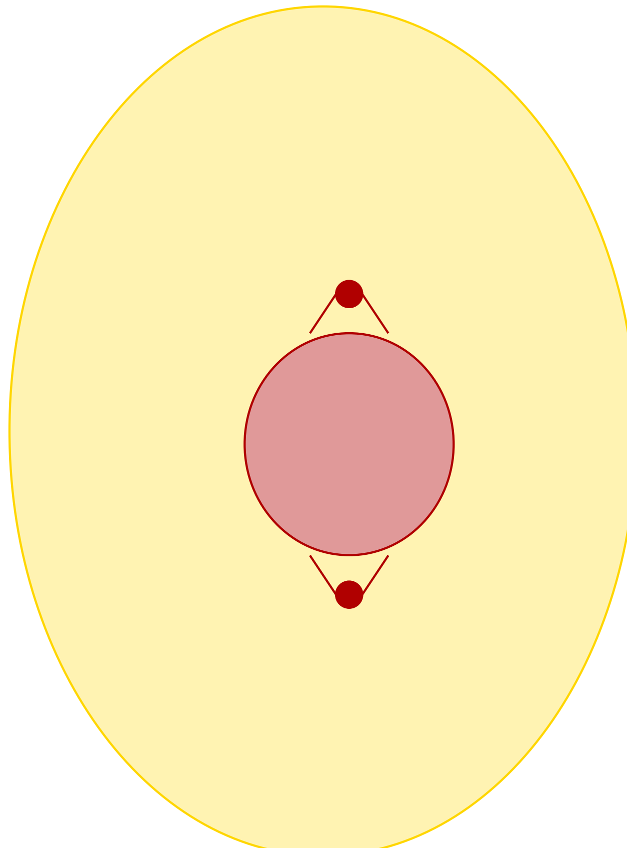






Wir suchen Lösungen für Constraint-Systeme der Form:

$$x_i \quad \sqsupseteq \quad f_i(x_1, \ldots, x_n) \qquad\qquad (*)$$

Wir suchen Lösungen für Constraint-Systeme der Form:

$$x_i \quad \sqsupseteq \quad f_i(x_1, \ldots, x_n) \qquad\qquad (*)$$

wobei:

x_i	Unbekannte	hier:	$\mathcal{A}[u]$
$\mathbb{D}$	Werte	hier:	2^{Expr}
$\sqsubseteq \quad \subseteq \quad \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\supseteq$
$f_i\colon \mathbb{D}^n \longrightarrow \mathbb{D}$	Bedingung	hier:	...

Wir suchen Lösungen für Constraint-Systeme der Form:

$$x_i \quad \sqsupseteq \quad f_i(x_1, \ldots, x_n) \qquad\qquad (*)$$

wobei:

x_i	Unbekannte	hier:	$\mathcal{A}[u]$
$\mathbb{D}$	Werte	hier:	2^{Expr}
$\sqsubseteq \quad \subseteq \quad \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\supseteq$
$f_i \colon \mathbb{D}^n \to \mathbb{D}$	Bedingung	hier:	...

Constraint für $\mathcal{A}[v]$:

$$\mathcal{A}[v] \quad \subseteq \quad \bigcap\{[\![k]\!]^\sharp\,(\mathcal{A}[u]) \mid k = (u, \_, v) \;\; \text{Kante}\}$$

Wir suchen Lösungen für Constraint-Systeme der Form:

$$x_i \quad \sqsupseteq \quad f_i(x_1, \ldots, x_n) \qquad\qquad (*)$$

wobei:

x_i	Unbekannte	hier:	$\mathcal{A}[u]$
$\mathbb{D}$	Werte	hier:	2^{Expr}
$\sqsubseteq \quad \subseteq \quad \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\supseteq$
$f_i \colon \mathbb{D}^n \to \mathbb{D}$	Bedingung	hier:	...

Constraint für $\mathcal{A}[v]$:

$$\mathcal{A}[v] \quad \subseteq \quad \bigcap\{[\![k]\!]^\sharp\,(\mathcal{A}[u]) \mid k = (u, \_, v) \text{ Kante}\}$$

Denn:

$$x \sqsupseteq d_1 \wedge \ldots \wedge x \sqsupseteq d_k \quad \text{gdw.} \quad x \sqsupseteq \bigsqcup\{d_1, \ldots, d_k\} \qquad \text{:-)}$$

Eine Abbildung $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt monoton, falls
$f(a) \sqsubseteq f(b)$ für alle $a \sqsubseteq b$.

Eine Abbildung $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt monoton, falls $f(a) \sqsubseteq f(b)$ für alle $a \sqsubseteq b$.

Beispiele:

(1) $\mathbb{D}_1 = \mathbb{D}_2 = 2^U$ für eine Menge U und $f\,x = (x \cap a) \cup b$.

Offensichtlich ist jedes solche f monoton :-)

Eine Abbildung $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt monoton, falls
$f(a) \sqsubseteq f(b)$ für alle $a \sqsubseteq b$.

Beispiele:

(1) $\mathbb{D}_1 = \mathbb{D}_2 = 2^U$ für eine Menge U und $f\,x = (x \cap a) \cup b$.
Offensichtlich ist jedes solche f monoton :-)

(2) $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{Z}$ (mit der Ordnung "$\leq$"). Dann gilt:

- $\mathsf{inc}\,x = x + 1$ ist monoton.
- $\mathsf{dec}\,x = x - 1$ ist monoton.

Eine Abbildung $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt monoton, falls
$f(a) \sqsubseteq f(b)$ für alle $a \sqsubseteq b$.

Beispiele:

(1) $\mathbb{D}_1 = \mathbb{D}_2 = 2^U$ für eine Menge U und $f\,x = (x \cap a) \cup b$.
Offensichtlich ist jedes solche f monoton :-)

(2) $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{Z}$ (mit der Ordnung "$\leq$"). Dann gilt:

- $\mathsf{inc}\,x = x + 1$ ist monoton.
- $\mathsf{dec}\,x = x - 1$ ist monoton.
- $\mathsf{inv}\,x = -x$ ist nicht monoton :-)

Satz:

Sind $f_1 : \mathbb{D}_1 \to \mathbb{D}_2$ und $f_2 : \mathbb{D}_2 \to \mathbb{D}_3$ monoton, dann ist auch $f_2 \circ f_1 : \mathbb{D}_1 \to \mathbb{D}_3$ monoton :-)

Ist $\mathbb{D}_2$ ein vollständiger Verband, dann bildet auch die Menge $[\mathbb{D}_1 \to \mathbb{D}_2]$ der monotonen Funktionen $f : \mathbb{D}_1 \to \mathbb{D}_2$ einen vollständigen Verband, wobei

Satz:

Sind $f_1 : \mathbb{D}_1 \to \mathbb{D}_2$ und $f_2 : \mathbb{D}_2 \to \mathbb{D}_3$ monoton, dann ist auch $f_2 \circ f_1 : \mathbb{D}_1 \to \mathbb{D}_3$ monoton :-)

Satz:

Ist $\mathbb{D}_2$ ein vollständiger Verband, dann bildet auch die Menge $[\mathbb{D}_1 \to \mathbb{D}_2]$ der monotonen Funktionen $f : \mathbb{D}_1 \to \mathbb{D}_2$ einen vollständigen Verband, wobei

$$f \sqsubseteq g \quad \text{gdw.} \quad f\,x \sqsubseteq g\,x \quad \text{für alle } x \in \mathbb{D}_1$$

Satz:

Sind $f_1 : \mathbb{D}_1 \to \mathbb{D}_2$ und $f_2 : \mathbb{D}_2 \to \mathbb{D}_3$ monoton, dann ist auch $f_2 \circ f_1 : \mathbb{D}_1 \to \mathbb{D}_3$ monoton :-)

Satz:

Ist $\mathbb{D}_2$ ein vollständiger Verband, dann bildet auch die Menge $[\mathbb{D}_1 \to \mathbb{D}_2]$ der monotonen Funktionen $f : \mathbb{D}_1 \to \mathbb{D}_2$ einen vollständigen Verband, wobei

$$f \sqsubseteq g \quad \text{gdw.} \quad f\,x \sqsubseteq g\,x \quad \text{für alle } x \in \mathbb{D}_1$$

Insbesondere ist für $F \subseteq [\mathbb{D}_1 \to \mathbb{D}_2]$,

$$\bigsqcup F = f \quad \text{mit} \quad f\,x = \bigsqcup\{g\,x \mid g \in F\}$$

Für Funktionen $f_i\, x = a_i \cap x \cup b_i$ können wir die Operationen "$\circ$", "$\sqcup$" und "$\sqcap$" explizit angeben:

$$
\begin{array}{lll}
(f_2 \circ f_1)\, x & = & \boxed{a_1 \cap a_2} \cap x \cup \boxed{a_2 \cap b_1 \cup b_2} \\
(f_1 \sqcup f_2)\, x & = & \boxed{(a_1 \cup a_2)} \cap x \cup \boxed{b_1 \cup b_2} \\
(f_1 \sqcap f_2)\, x & = & \boxed{(a_1 \cup b_1) \cap (a_2 \cup b_2)} \cap x \cup \boxed{b_1 \cap b_2}
\end{array}
$$

Gesucht: möglichst kleine Lösung für:

$$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

Gesucht: möglichst kleine Lösung für:

$$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

Idee:

- Betrachte $F : \mathbb{D}^n \to \mathbb{D}^n$ mit

  $F(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ wobei $y_i = f_i(x_1, \ldots, x_n)$.

Gesucht: möglichst kleine Lösung für:

$$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

Idee:

- Betrachte $F : \mathbb{D}^n \to \mathbb{D}^n$ mit

  $$F(x_1, \ldots, x_n) = (y_1, \ldots, y_n) \quad \text{wobei} \quad y_i = f_i(x_1, \ldots, x_n).$$

- Sind alle f_i monoton, dann auch F :-)

Gesucht: möglichst kleine Lösung für:

$$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

Idee:

- Betrachte $F : \mathbb{D}^n \to \mathbb{D}^n$ mit

  $F(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ wobei $y_i = f_i(x_1, \ldots, x_n)$.
- Sind alle f_i monoton, dann auch F :-)
- Wir approximieren sukzessive eine Lösung. Wir konstruieren:

$$\underline{\perp}, \quad F\,\underline{\perp}, \quad F^2\,\underline{\perp}, \quad F^3\,\underline{\perp}, \quad \ldots$$

Hoffnung: Wir erreichen irgendwann eine Lösung ... ???

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \, = \, \subseteq$

$$
\begin{array}{lll}
x_1 & \supseteq & \{a\} \cup x_3 \\
x_2 & \supseteq & x_3 \cap \{a, b\} \\
x_3 & \supseteq & x_1 \cup \{c\}
\end{array}
$$

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \; = \; \subseteq$

$$\begin{array}{lll} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a, b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

Die Iteration:

	0	1	2	3	4
x_1	$\emptyset$				
x_2	$\emptyset$				
x_3	$\emptyset$				

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \, = \, \subseteq$

$$\begin{array}{lll} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a, b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

Die Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$			
x_2	$\emptyset$	$\emptyset$			
x_3	$\emptyset$	$\{c\}$			

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \, = \, \subseteq$

$$\begin{array}{lll} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a, b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

Die Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	$\{a, c\}$		
x_2	$\emptyset$	$\emptyset$	$\emptyset$		
x_3	$\emptyset$	$\{c\}$	$\{a, c\}$		

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq \, = \, \subseteq$

$$\begin{array}{lll} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a, b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

Die Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	$\{a, c\}$	$\{a, c\}$	
x_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	
x_3	$\emptyset$	$\{c\}$	$\{a, c\}$	$\{a, c\}$	

Beispiel: $\mathbb{D} = 2^{\{a,b,c\}}, \quad \sqsubseteq = \subseteq$

$$\begin{array}{lll} x_1 & \supseteq & \{a\} \cup x_3 \\ x_2 & \supseteq & x_3 \cap \{a,b\} \\ x_3 & \supseteq & x_1 \cup \{c\} \end{array}$$

Die Iteration:

	0	1	2	3	4
x_1	$\emptyset$	$\{a\}$	$\{a,c\}$	$\{a,c\}$	dito
x_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{a\}$	
x_3	$\emptyset$	$\{c\}$	$\{a,c\}$	$\{a,c\}$	

Offenbar gilt:

- Gilt $F^k \underline{\perp} = F^{k+1} \underline{\perp}$, ist eine Lösung gefunden :-)
- $\underline{\perp}, F \underline{\perp}, F^2 \underline{\perp}, \ldots$ bilden eine aufsteigende Kette :

$$\underline{\perp} \quad \sqsubseteq \quad F \underline{\perp} \quad \sqsubseteq \quad F^2 \underline{\perp} \quad \sqsubseteq \quad \ldots$$

- Sind alle aufsteigenden Ketten endlich, gibt es k immer.

Offenbar gilt:

- Gilt $F^k \underline{\perp} = F^{k+1} \underline{\perp}$, ist eine Lösung gefunden :-)
- $\underline{\perp}, F\underline{\perp}, F^2\underline{\perp}, \ldots$ bilden eine aufsteigende Kette :

$$\underline{\perp} \quad \sqsubseteq \quad F\underline{\perp} \quad \sqsubseteq \quad F^2\underline{\perp} \quad \sqsubseteq \quad \ldots$$

- Sind alle aufsteigenden Ketten endlich, gibt es k immer.

Die zweite Aussage folgt mit vollständiger Induktion:

Offenbar gilt:

- Gilt $F^k \underline{\perp} = F^{k+1} \underline{\perp}$, ist eine Lösung gefunden :-)
- $\underline{\perp}, F\,\underline{\perp}, F^2\,\underline{\perp}, \ldots$ bilden eine aufsteigende Kette :

$$\underline{\perp} \quad \sqsubseteq \quad F\,\underline{\perp} \quad \sqsubseteq \quad F^2\,\underline{\perp} \quad \sqsubseteq \quad \ldots$$

- Sind alle aufsteigenden Ketten endlich, gibt es k immer.

Die zweite Aussage folgt mit vollständiger Induktion:

Anfang: $F^0\,\underline{\perp} = \underline{\perp} \sqsubseteq F^1\,\underline{\perp}$:-)

Offenbar gilt:

- Gilt $F^k \underline{\perp} = F^{k+1} \underline{\perp}$, ist eine Lösung gefunden :-)
- $\underline{\perp}, F\,\underline{\perp}, F^2\,\underline{\perp}, \ldots$ bilden eine aufsteigende Kette :

$$\underline{\perp} \quad \sqsubseteq \quad F\,\underline{\perp} \quad \sqsubseteq \quad F^2\,\underline{\perp} \quad \sqsubseteq \quad \ldots$$

- Sind alle aufsteigenden Ketten endlich, gibt es k immer.

Die zweite Aussage folgt mit vollständiger Induktion:

Anfang: $F^0\,\underline{\perp} = \underline{\perp} \sqsubseteq F^1\,\underline{\perp}$:-)

Schluss: Gelte bereits $F^{i-1}\,\underline{\perp} \sqsubseteq F^i\,\underline{\perp}$. Dann

$$F^i\,\underline{\perp} = F\,(F^{i-1}\,\underline{\perp}) \sqsubseteq F\,(F^i\,\underline{\perp}) = F^{i+1}\,\underline{\perp}$$

da F monoton ist :-)

Fazit:

Wenn $\mathbb{D}$ endlich ist, finden wir mit Sicherheit eine Lösung :-)

Fragen:

Fazit:

Wenn $\mathbb{D}$ endlich ist, finden wir mit Sicherheit eine Lösung :-)

Fragen:

1. Gibt es eine kleinste Lösung ?

Fazit:

Wenn $\mathbb{D}$ endlich ist, finden wir mit Sicherheit eine Lösung :-)

Fragen:

1. Gibt es eine kleinste Lösung ?
2. Wenn ja: findet Iteration die kleinste Lösung ??

Fazit:

Wenn $\mathbb{D}$ endlich ist, finden wir mit Sicherheit eine Lösung :-)

Fragen:

1. Gibt es eine kleinste Lösung ?
2. Wenn ja: findet Iteration die kleinste Lösung ??
3. Was, wenn $\mathbb{D}$ nicht endlich ist ???

Satz Knaster – Tarski

In einem vollständigen Verband $\mathbb{D}$ hat jede monotone Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Sei $P = \{d \in \mathbb{D} \mid d \sqsupseteq f\, d\}$ die Menge der Postfixpunkte.

Dann ist $d_0 = \bigsqcap P$.

Bronisław Knaster (1893-1980), topology

Satz Knaster – Tarski

In einem vollständigen Verband $\mathbb{D}$ hat jede monotone Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Sei $P = \{d \in \mathbb{D} \mid d \sqsupseteq f\, d\}$ die Menge der Postfixpunkte.

Dann ist $d_0 = \bigsqcap P$.

Beweis:

(1) $d_0 \in P$:

Satz — Knaster – Tarski

In einem vollständigen Verband $\mathbb{D}$ hat jede monotone Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Sei $P = \{d \in \mathbb{D} \mid d \sqsupseteq f\, d\}$ die Menge der Postfixpunkte.

Dann ist $d_0 = \bigsqcap P$.

Beweis:

(1) $d_0 \in P$:

$$
\begin{array}{ll}
 & f\, d_0 \sqsubseteq f\, d \sqsubseteq d \quad \text{für alle } d \in P \\
\Longrightarrow & f\, d_0 \quad \text{ist untere Schranke von } P \\
\Longrightarrow & f\, d_0 \sqsubseteq d_0 \quad \text{weil } d_0 = \bigsqcap P \\
\Longrightarrow & d_0 \in P \qquad \text{:-)}
\end{array}
$$

(2) $$d_0 = f\, d_0 :$$

(2) $\quad d_0 = f\, d_0$:

$$f\, d_0 \sqsubseteq d_0 \quad \text{wegen} \quad (1)$$

$$\Longrightarrow \quad f(f\, d_0) \sqsubseteq f\, d_0 \quad \text{wegen Monotonie von } f$$

$$\Longrightarrow \quad f\, d_0 \in P$$

$$\Longrightarrow \quad d_0 \sqsubseteq f\, d_0 \qquad \text{und die Behauptung folgt} \quad \text{:-)}$$

(2) $d_0 = f\, d_0$:

$f\, d_0 \sqsubseteq d_0$ wegen (1)

$\Longrightarrow$ $f(f\, d_0) \sqsubseteq f\, d_0$ wegen Monotonie von f

$\Longrightarrow$ $f\, d_0 \in P$

$\Longrightarrow$ $d_0 \sqsubseteq f\, d_0$ und die Behauptung folgt :-)

(3) d_0 ist kleinster Fixpunkt:

(2) $d_0 = f\, d_0$:

$$f\, d_0 \sqsubseteq d_0 \quad \text{wegen} \quad (1)$$

$$\Longrightarrow \quad f\,(f\, d_0) \sqsubseteq f\, d_0 \quad \text{wegen Monotonie von } f$$

$$\Longrightarrow \quad f\, d_0 \in P$$

$$\Longrightarrow \quad d_0 \sqsubseteq f\, d_0 \qquad \text{und die Behauptung folgt} \quad \text{:-)}$$

(3) d_0 ist kleinster Fixpunkt:

$$d_1 \sqsupseteq d_1 = f\, d_1 \quad \text{weiterer Fixpunkt}$$

$$\Longrightarrow \quad d_1 \in P$$

$$\Longrightarrow \quad d_0 \sqsubseteq d_1 \qquad \text{:-))}$$

Bemerkung:

Der kleinste Fixpunkt d_0 ist in P und untere Schranke :-)

$\Longrightarrow$ d_0 ist der kleinste Wert x mit $x \sqsupseteq f\,x$

Bemerkung:

Der kleinste Fixpunkt d_0 ist in P und untere Schranke :-)

$\Longrightarrow$ d_0 ist der kleinste Wert x mit $x \sqsupseteq f\,x$

Anwendung:

Sei $$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

ein Ungleichungssystem, wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

Bemerkung:

Der kleinste Fixpunkt d_0 ist in P und untere Schranke :-)

$\Longrightarrow$ d_0 ist der kleinste Wert x mit $x \sqsupseteq f\, x$

Anwendung:

Sei $$x_i \sqsupseteq f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \qquad (*)$$

ein Ungleichungssystem, wobei alle $f_i : \mathbb{D}^n \to \mathbb{D}$ monoton sind.

$\Longrightarrow$ kleinste Lösung von $(*)$ $==$ kleinster Fixpunkt von F :-)

Beispiel 1: $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

Beispiel 1: $\mathbb{D} = 2^U, \quad f\, x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	$\top$

Beispiel 1: $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	$\top$
1	b	$a \cup b$

Beispiel 1: $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	$\top$
1	b	$a \cup b$
2	b	$a \cup b$

Beispiel 1: $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k \perp$	$f^k \top$
0	$\emptyset$	$\top$
1	b	$a \cup b$
2	b	$a \cup b$

Beispiel 2: $\mathbb{D} = \mathbb{N} \cup \{\infty\}$

Für die Funktion $\quad f\,x = x + 1 \quad$ ist:

$$f^i \perp = f^i\,0 = i \quad \sqsubset \quad i + 1 = f^{i+1} \perp$$

Beispiel 1: $\mathbb{D} = 2^U, \quad f\,x = x \cap a \cup b$

f	$f^k\,\bot$	$f^k\,\top$
0	$\emptyset$	$\top$
1	b	$a \cup b$
2	b	$a \cup b$

Beispiel 2: $\mathbb{D} = \mathbb{N} \cup \{\infty\}$

Für die Funktion $f\,x = x + 1$ ist:

$$f^i\,\bot = f^i\,0 = i \quad \sqsubset \quad i + 1 = f^{i+1}\,\bot$$

$\Longrightarrow$ Die normale Iteration erreicht nie einen Fixpunkt :-(

$\Longrightarrow$ Man benötigt manchmal transfinite Iteration :-)

Satz:

Sei $f : \mathbb{D} \to \mathbb{D}$ monoton und $X \subseteq \mathbb{D}$ die kleinste Menge mit:

(a) $\perp \in X$;

(b) $f\,d \in X$ falls $d \in X$;

(c) $\bigsqcup X_0 \in X$ für alle $X_0 \subseteq X$.

// diese Menge existiert offenbar :-)

Satz:

Sei $f : \mathbb{D} \to \mathbb{D}$ monoton und $X \subseteq \mathbb{D}$ die kleinste Menge mit:

(a) $\bot \in X$;

(b) $f\, d \in X$ falls $d \in X$;

(c) $\bigsqcup X_0 \in X$ für alle $X_0 \subseteq X$.

// diese Menge existiert offenbar :-)

Dann ist $d_0 = \bigsqcup X$ der kleinste Fixpunkt von f.

Beweis:

(1) $\quad f\, d_0 \sqsubseteq d_0 \quad$ d.h. $\quad d_0 \quad$ ist Präfixpunkt:

Beweis:

(1) $f\, d_0 \sqsubseteq d_0$ d.h. d_0 ist Präfixpunkt:

$$
\begin{array}{lll}
 & d_0 \in X & \text{wegen (c)} \\
\Longrightarrow & f\, d_0 \in X & \text{wegen (b)} \\
\Longrightarrow & f\, d_0 \sqsubseteq d_0 & \text{:-)}
\end{array}
$$

Beweis:

(1) $f\, d_0 \sqsubseteq d_0$ d.h. d_0 ist Präfixpunkt:

$$\begin{array}{lll} & d_0 \in X & \text{wegen (c)} \\ \Longrightarrow & f\, d_0 \in X & \text{wegen (b)} \\ \Longrightarrow & f\, d_0 \sqsubseteq d_0 & \text{:-)} \end{array}$$

(2) d_0 ist kleinster Präfixpunkt:

Beweis:

(1) $f\, d_0 \sqsubseteq d_0$ d.h. d_0 ist Präfixpunkt:

$$\begin{array}{lll} & d_0 \in X & \text{wegen (c)} \\ \Longrightarrow & f\, d_0 \in X & \text{wegen (b)} \\ \Longrightarrow & f\, d_0 \sqsubseteq d_0 & \text{:-)} \end{array}$$

(2) d_0 ist kleinster Präfixpunkt:

Sei d_1 weiterer Präfixpunkt, d.h. $f\, d_1 \sqsubseteq d_1$.
Dann erfüllt die Menge: $X_1 = \{x \in \mathbb{D} \mid x \sqsubseteq d_1\}$
die Eigenschaften (a), (b) und (c) :-)

Beweis:

(1) $f\,d_0 \sqsubseteq d_0$ d.h. d_0 ist Präfixpunkt:

$$\begin{array}{lll} & d_0 \in X & \text{wegen (c)} \\ \Longrightarrow & f\,d_0 \in X & \text{wegen (b)} \\ \Longrightarrow & f\,d_0 \sqsubseteq d_0 & \text{:-)} \end{array}$$

(2) d_0 ist kleinster Präfixpunkt:

Sei d_1 weiterer Präfixpunkt, d.h. $f\,d_1 \sqsubseteq d_1$.
Dann erfüllt die Menge: $X_1 = \{x \in \mathbb{D} \mid x \sqsubseteq d_1\}$
die Eigenschaften (a), (b) und (c) :-)

$$\begin{array}{ll} \Longrightarrow & X \subseteq X_1 \\ \Longrightarrow & d_1 \ \text{ist obere Schranke von} \ X \\ \Longrightarrow & d_0 = \bigsqcup X \sqsubseteq d_1 \quad \text{:-))} \end{array}$$

Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen,
d.h. durch wiederholtes Einsetzen :-)

Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen,
d.h. durch wiederholtes Einsetzen :-)

Achtung: Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

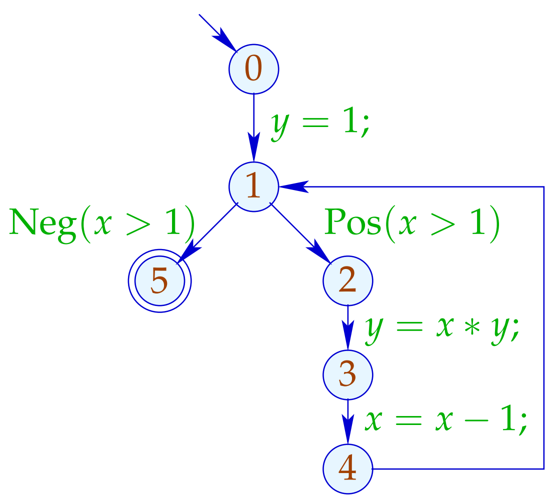
Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen, d.h. durch wiederholtes Einsetzen :-)

Achtung:

Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



0
1
2
3
4
5

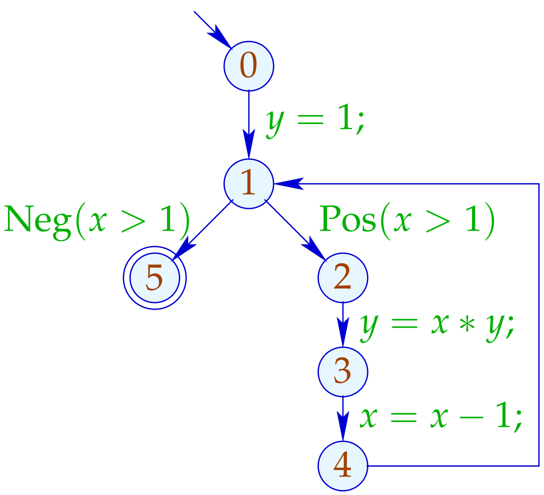
Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen,
d.h. durch wiederholtes Einsetzen :-)

Achtung:

Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



	1
0	$\emptyset$
1	$\{1, x > 1, x - 1\}$
2	*Expr*
3	$\{1, x > 1, x - 1\}$
4	$\{1\}$
5	*Expr*

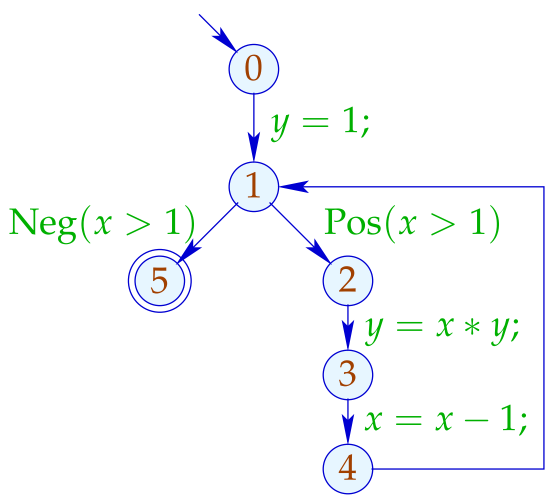
Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen, d.h. durch wiederholtes Einsetzen :-)

Achtung:

Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



	1	2
0	$\emptyset$	$\emptyset$
1	$\{1, x > 1, x - 1\}$	$\{1\}$
2	*Expr*	$\{1, x > 1, x - 1\}$
3	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$
4	$\{1\}$	$\{1\}$
5	*Expr*	$\{1, x > 1, x - 1\}$

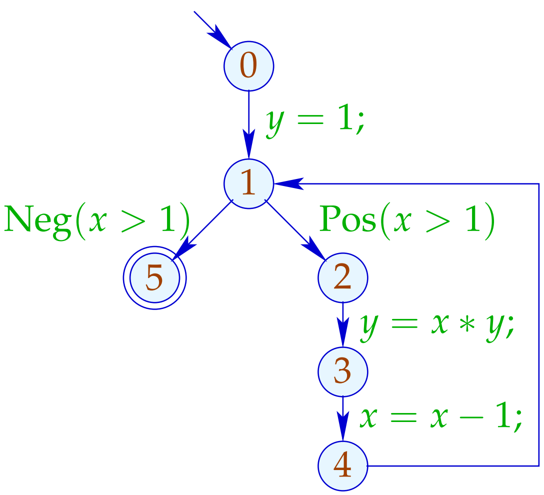
Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen, d.h. durch wiederholtes Einsetzen :-)

Achtung:

Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



	1	2	3
0	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{1, x > 1, x - 1\}$	$\{1\}$	$\{1\}$
2	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$
3	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$
5	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$

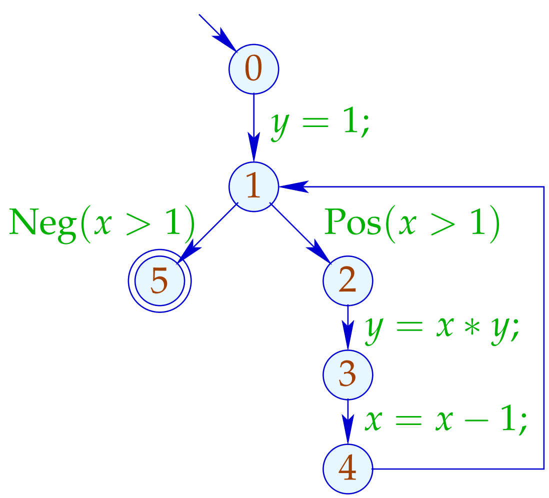
Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen, d.h. durch wiederholtes Einsetzen :-)

Achtung:

Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



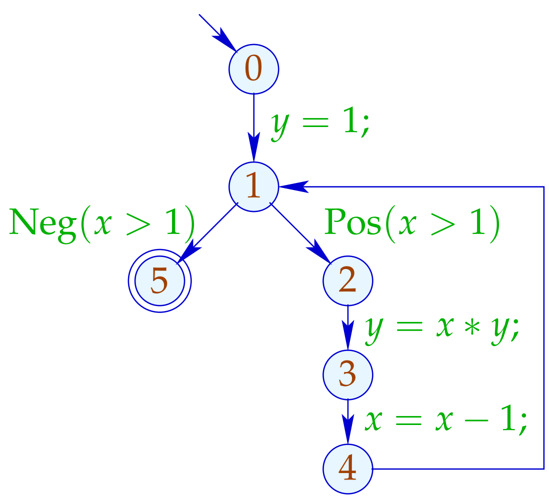
	1	2	3	4
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
1	$\{1, x > 1, x - 1\}$	$\{1\}$	$\{1\}$	$\{1\}$
2	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$
3	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$	$\{1\}$
5	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$

Fazit:

Wir können Constraint-Systeme durch Fixpunkt-Iteration lösen, d.h. durch wiederholtes Einsetzen :-)

Achtung: Naive Fixpunkt-Iteration ist ziemlich ineffizient :-(

Beispiel:



	1	2	3	4	5
0	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	
1	$\{1, x > 1, x - 1\}$	$\{1\}$	$\{1\}$	$\{1\}$	
2	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$	
3	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$	dito
4	$\{1\}$	$\{1\}$	$\{1\}$	$\{1\}$	
5	*Expr*	$\{1, x > 1, x - 1\}$	$\{1, x > 1\}$	$\{1, x > 1\}$	

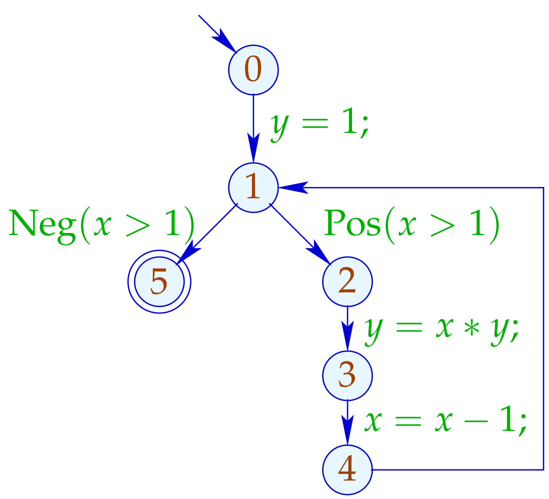
Idee: Round Robin Iteration

Benutze bei der Iteration nicht die Werte der letzten Iteration, sondern die jeweils aktuellen :-)

Idee: Round Robin Iteration

Benutze bei der Iteration nicht die Werte der letzten Iteration, sondern die jeweils aktuellen :-)

Beispiel:

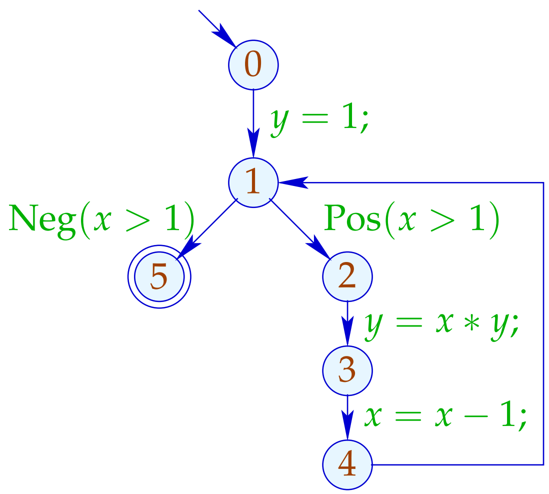


0
1
2
3
4
5

Idee: Round Robin Iteration

Benutze bei der Iteration nicht die Werte der letzten Iteration, sondern die jeweils aktuellen :-)

Beispiel:

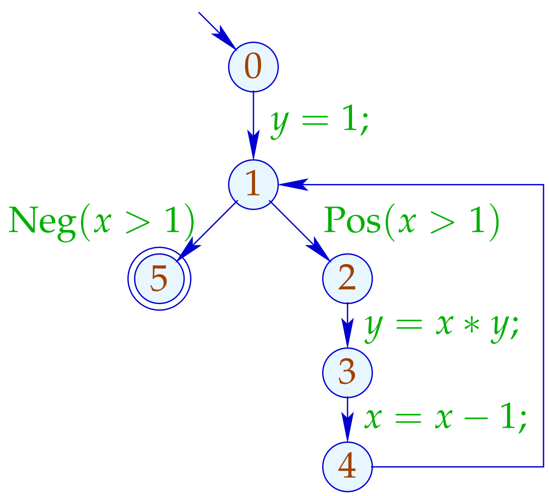


	1
0	$\emptyset$
1	$\{1\}$
2	$\{1, x > 1\}$
3	$\{1, x > 1\}$
4	$\{1\}$
5	$\{1, x > 1\}$

Idee: Round Robin Iteration

Benutze bei der Iteration nicht die Werte der letzten Iteration, sondern die jeweils aktuellen :-)

Beispiel:



	1	2
0	$\emptyset$	
1	$\{1\}$	
2	$\{1, x > 1\}$	
3	$\{1, x > 1\}$	dito
4	$\{1\}$	
5	$\{1, x > 1\}$	

Der Code für Round Robin Iteration sieht in Java so aus:

$$
\begin{array}{l}
\textsf{for}\ (i = 1; i \leq n; i{+}{+})\ x_i = \bot; \\
\textsf{do}\ \{ \\
\qquad \mathit{finished} = \textsf{true}; \\
\qquad \textsf{for}\ (i = 1; i \leq n; i{+}{+})\ \{ \\
\qquad\qquad \mathit{new} = f_i(x_1, \ldots, x_n); \\
\qquad\qquad \textsf{if}\ (!(x_i \sqsupseteq \mathit{new}))\ \{ \\
\qquad\qquad\qquad \mathit{finished} = \textsf{false}; \\
\qquad\qquad\qquad x_i = x_i \sqcup \mathit{new}; \\
\qquad\qquad \} \\
\qquad \} \\
\}\ \textsf{while}\ (!\mathit{finished});
\end{array}
$$

Zur Korrektheit:

Sei $y_i^{(d)}$ die i-te Komponente von $F^d \underline{\perp}$.

Sei $x_i^{(d)}$ der Wert von x_i nach der i-ten RR-Iteration.

Zur Korrektheit:

Sei $y_i^{(d)}$ die i-te Komponente von $F^d \underline{\perp}$.

Sei $x_i^{(d)}$ der Wert von x_i nach der i-ten RR-Iteration.

Man zeigt:

(1) $y_i^{(d)} \sqsubseteq x_i^{(d)}$:-)

Zur Korrektheit:

Sei $y_i^{(d)}$ die i-te Komponente von $F^d \underline{\perp}$.

Sei $x_i^{(d)}$ der Wert von x_i nach der i-ten RR-Iteration.

Man zeigt:

(1) $y_i^{(d)} \sqsubseteq x_i^{(d)}$:-)

(2) $x_i^{(d)} \sqsubseteq z_i$ für jede Lösung $(z_1, \ldots, z_n)$:-)

Zur Korrektheit:

Sei $y_i^{(d)}$ die i-te Komponente von $F^d \underline{\perp}$.

Sei $x_i^{(d)}$ der Wert von x_i nach der i-ten RR-Iteration.

Man zeigt:

(1) $y_i^{(d)} \sqsubseteq x_i^{(d)}$:-)

(2) $x_i^{(d)} \sqsubseteq z_i$ für jede Lösung $(z_1, \ldots, z_n)$:-)

(3) Terminiert RR-Iteration nach d Runden, ist $(x_1^{(d)}, \ldots, x_n^{(d)})$ eine Lösung :-))

Achtung:

Die Effizienz von RR-Iteration hängt von der Anordnung der Variablen ab !!!

Achtung:

Die Effizienz von RR-Iteration hängt von der Anordnung der Variablen ab !!!

Günstig:

$\longrightarrow$ u vor v, falls $u \longrightarrow^{*} v$;

$\longrightarrow$ Eintrittsbedingung vor Schleifen-Rumpf :-)

Achtung:

Die Effizienz von RR-Iteration hängt von der Anordnung der Variablen ab !!!

Günstig:

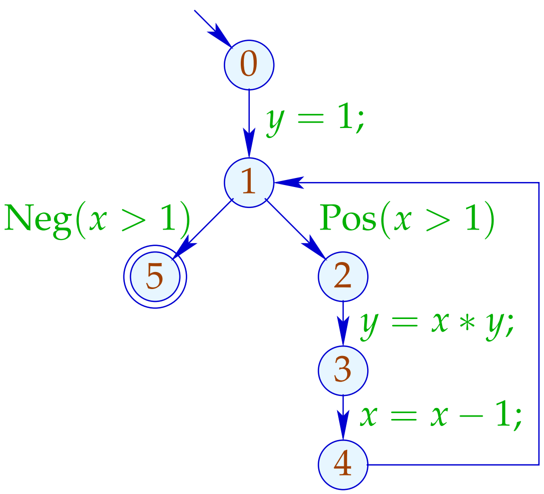
$\longrightarrow$ u vor v, falls $u \longrightarrow^* v$;

$\longrightarrow$ Eintrittsbedingung vor Schleifen-Rumpf :-)

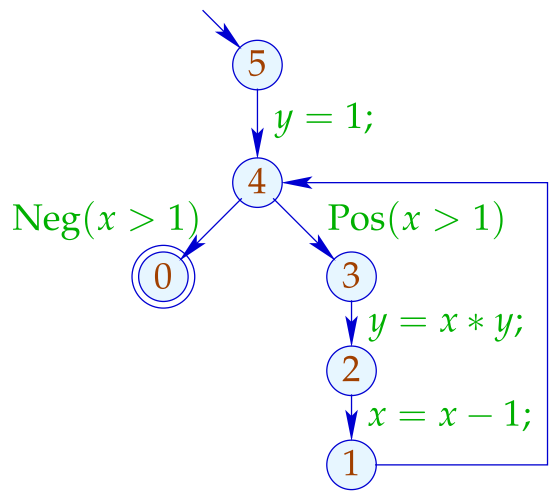
Ungünstig:

z.B. post-order DFS auf dem CFG, startend von start :-)

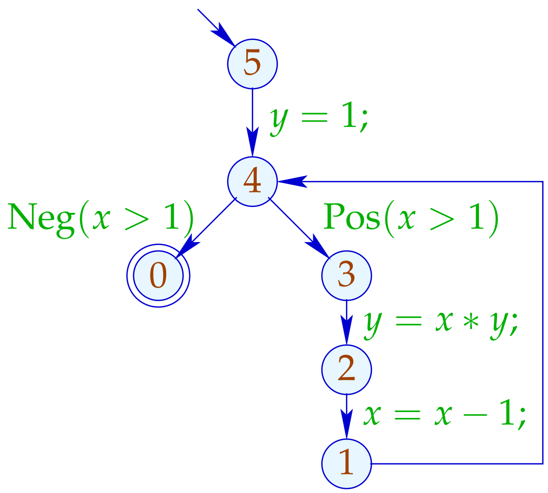
Günstig:



Ungünstig:

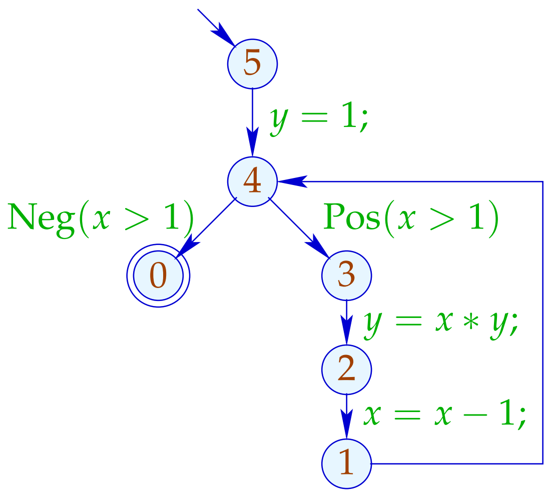


Ungünstige Round Robin Iteration:



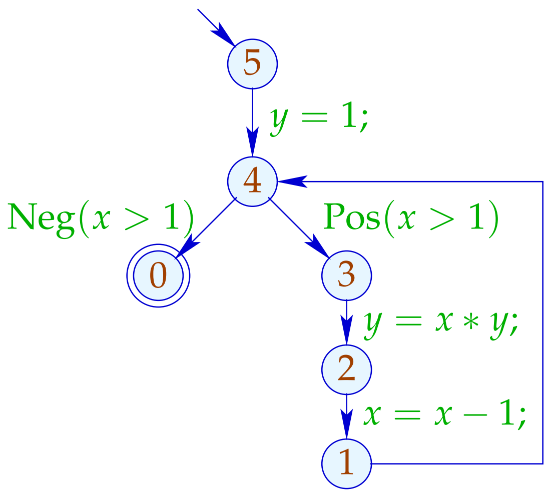
0
1
2
3
4
5

Ungünstige Round Robin Iteration:



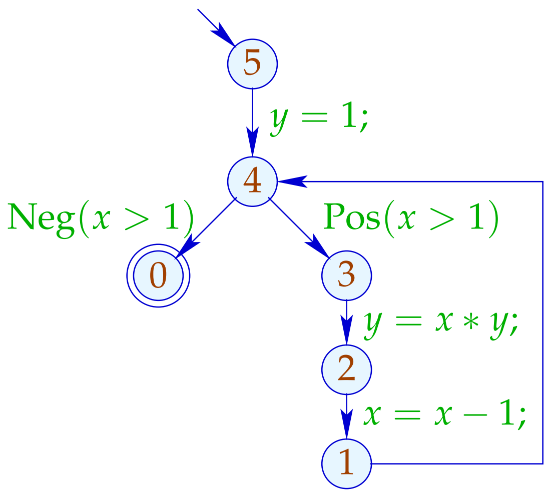
	1
0	*Expr*
1	$\{1\}$
2	$\{1, x - 1, x > 1\}$
3	*Expr*
4	$\{1\}$
5	$\emptyset$

Ungünstige Round Robin Iteration:



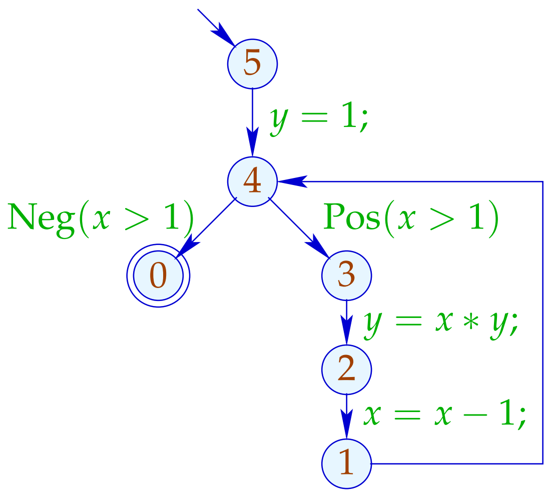
	1	2
0	*Expr*	$\{1, x > 1\}$
1	$\{1\}$	$\{1\}$
2	$\{1, x - 1, x > 1\}$	$\{1, x - 1, x > 1\}$
3	*Expr*	$\{1, x > 1\}$
4	$\{1\}$	$\{1\}$
5	$\emptyset$	$\emptyset$

Ungünstige Round Robin Iteration:



	1	2	3
0	*Expr*	$\{1, x > 1\}$	$\{1, x > 1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1, x - 1, x > 1\}$	$\{1, x - 1, x > 1\}$	$\{1, x > 1\}$
3	*Expr*	$\{1, x > 1\}$	$\{1, x > 1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$
5	$\emptyset$	$\emptyset$	$\emptyset$

Ungünstige Round Robin Iteration:



	1	2	3	4
0	*Expr*	$\{1, x > 1\}$	$\{1, x > 1\}$	
1	$\{1\}$	$\{1\}$	$\{1\}$	
2	$\{1, x-1, x > 1\}$	$\{1, x-1, x > 1\}$	$\{1, x > 1\}$	dito
3	*Expr*	$\{1, x > 1\}$	$\{1, x > 1\}$	
4	$\{1\}$	$\{1\}$	$\{1\}$	
5	$\emptyset$	$\emptyset$	$\emptyset$	

$\Longrightarrow$ deutlich weniger effizient :-)

... Ende des Exkurses: Vollständige Verbände

... Ende des Exkurses: Vollständige Verbände

Letzte Frage:

Wieso hilft uns eine (oder die kleinste) Lösung des Constraint-Systems weiter ???

... Ende des Exkurses: Vollständige Verbände

Letzte Frage:

Wieso hilft uns eine (oder die kleinste) Lösung des Constraint-Systems weiter ???

Betrachte für einen vollständigen Verband $\mathbb{D}$ Systeme:

$$\begin{array}{lll} \mathcal{I}[start] & \sqsupseteq & d_0 \\ \mathcal{I}[v] & \sqsupseteq & [\![k]\!]^\sharp\,(\mathcal{I}[u]) \qquad k = (u, \_, v) \quad \text{Kante} \end{array}$$

wobei $d_0 \in \mathbb{D}$ und alle $[\![k]\!]^\sharp : \mathbb{D} \to \mathbb{D}$ monoton sind ...

... Ende des Exkurses: Vollständige Verbände

Letzte Frage:

Wieso hilft uns eine (oder die kleinste) Lösung des Constraint-Systems weiter ???

Betrachte für einen vollständigen Verband $\mathbb{D}$ Systeme:

$$\begin{array}{lll} \mathcal{I}[start] & \sqsupseteq & d_0 \\ \mathcal{I}[v] & \sqsupseteq & [\![k]\!]^\sharp\,(\mathcal{I}[u]) \qquad k = (u, \_, v) \quad \text{Kante} \end{array}$$

wobei $d_0 \in \mathbb{D}$ und alle $[\![k]\!]^\sharp : \mathbb{D} \to \mathbb{D}$ monoton sind ...

$\Longrightarrow$ monotoner Analyse-Rahmen

Gesucht: MOP (Merge Over all Paths)

$$\mathcal{I}^*[v] = \bigsqcup\{[\![\pi]\!]^\sharp\, d_0 \mid \pi : start \rightarrow^* v\}$$

Gesucht: MOP (Merge Over all Paths)

$$\mathcal{I}^*[v] = \bigsqcup \{ [\![\pi]\!]^\sharp\, d_0 \mid \pi : \mathit{start} \rightarrow^* v \}$$

Theorem Kam, Ullman 1975

Sei $\mathcal{I}$ die kleinste Lösung des Constraint-Systems. Dann gilt:

$$\mathcal{I}[v] \sqsupseteq \mathcal{I}^*[v] \qquad \text{für jedes} \quad v$$

Jeffrey D. Ullman, Stanford

Gesucht: MOP (Merge Over all Paths)

$$\mathcal{I}^*[v] = \bigsqcup\{[\![\pi]\!]^\sharp\, d_0 \mid \pi : \textit{start} \to^* v\}$$

Theorem Kam, Ullman 1975

Sei $\mathcal{I}$ die kleinste Lösung des Constraint-Systems. Dann gilt:

$$\mathcal{I}[v] \sqsupseteq \mathcal{I}^*[v] \qquad \text{für jedes} \quad v$$

Insbesondere: $\mathcal{I}[v] \sqsupseteq [\![\pi]\!]^\sharp\, d_0$ für jedes $\pi : \textit{start} \to^* v$

Beweis: Induktion nach der Länge von π.

Beweis: Induktion nach der Länge von π.

Anfang: $\pi = \epsilon$ (leerer Pfad)

Beweis: Induktion nach der Länge von π.

Anfang: $\pi = \epsilon$ (leerer Pfad)

Dann gilt:

$$[\![\pi]\!]^\sharp\, d_0 = [\![\epsilon]\!]^\sharp\, d_0 = d_0 \sqsubseteq \mathcal{I}[start]$$

Beweis: Induktion nach der Länge von π.

Anfang: $\pi = \epsilon$ (leerer Pfad)

Dann gilt:

$$[\![\pi]\!]^\sharp\, d_0 = [\![\epsilon]\!]^\sharp\, d_0 = d_0 \sqsubseteq \mathcal{I}[start]$$

Schluss: $\pi = \pi' k$ für $k = (u, \_, v)$ Kante.

Beweis: Induktion nach der Länge von π.

Anfang: $\pi = \epsilon$ (leerer Pfad)

Dann gilt:

$$[\![\pi]\!]^\sharp\, d_0 = [\![\epsilon]\!]^\sharp\, d_0 = d_0 \sqsubseteq \mathcal{I}[\mathit{start}]$$

Schluss: $\pi = \pi' k$ für $k = (u, \_, v)$ Kante.

Dann gilt:

$$\begin{array}{rrcll}
 & [\![\pi']\!]^\sharp\, d_0 & \sqsubseteq & \mathcal{I}[u] & \text{wegen I.H. für } \pi \\
\Longrightarrow & [\![\pi]\!]^\sharp\, d_0 & = & [\![k]\!]^\sharp\, ([\![\pi']\!]^\sharp\, d_0) & \\
 & & \sqsubseteq & [\![k]\!]^\sharp\, (\mathcal{I}[u]) & \text{da } [\![k]\!]^\sharp \text{ monoton} \\
 & & \sqsubseteq & \mathcal{I}[v] & \text{da } \mathcal{I} \text{ Lösung} \quad \text{:-))}
\end{array}$$

Enttäuschung:

Liefern Lösungen des Constraint-Systems nur obere Schranken ???

Enttäuschung:

Liefern Lösungen des Constraint-Systems nur obere Schranken ???

Antwort:

Im allgemeinen: ja   :-(

Enttäuschung:

Liefern Lösungen des Constraint-Systems nur obere Schranken ???

Antwort:

Im allgemeinen: ja :-(

Es sei denn, alle Funktionen $[\![k]\!]^\sharp$ sind distributiv ... :-)

Die Funktion $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt

- distributiv, falls $f\,(\bigsqcup X) = \bigsqcup\{f\,x \mid x \in X\}$ für alle $\emptyset \neq X \subseteq \mathbb{D}$;
- strikt, falls $f\,\bot = \bot$.
- total distributiv, falls f distributiv und strikt ist.

Die Funktion $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt

- **distributiv**, falls $f\,(\bigsqcup X) = \bigsqcup\{f\,x \mid x \in X\}$ für alle $\emptyset \neq X \subseteq \mathbb{D}$;
- **strikt**, falls $f\,\bot = \bot$.
- **total distributiv**, falls f distributiv und strikt ist.

Beispiele:

- $f\,x = x \cap a \cup b$ für $a, b \subseteq U$.

Die Funktion $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt

- distributiv, falls $f\,(\bigsqcup X) = \bigsqcup\{f\,x \mid x \in X\}$ für alle $\emptyset \neq X \subseteq \mathbb{D}$;
- strikt, falls $f\,\bot = \bot$.
- total distributiv, falls f distributiv und strikt ist.

Beispiele:

- $f\,x = x \cap a \cup b$ für $a, b \subseteq U$.

  Striktheit: $f\,\emptyset = a \cap \emptyset \cup b = b \quad = \quad \emptyset$ sofern $b = \emptyset$:-(

Die Funktion $f : \mathbb{D}_1 \to \mathbb{D}_2$ heißt

- distributiv, falls $f\,(\bigsqcup X) = \bigsqcup\{f\,x \mid x \in X\}$ für alle $\emptyset \neq X \subseteq \mathbb{D}$;
- strikt, falls $f\,\bot = \bot$.
- total distributiv, falls f distributiv und strikt ist.

Beispiele:

- $f\,x = x \cap a \cup b$ für $a, b \subseteq U$.

  Striktheit: $f\,\emptyset = a \cap \emptyset \cup b = b \quad = \quad \emptyset$ sofern $b = \emptyset$:-(

  Distributivität:

$$
\begin{aligned}
f\,(x_1 \cup x_2) &= a \cap (x_1 \cup x_2) \cup b \\
&= a \cap x_1 \cup a \cap x_2 \cup b \\
&= f\,x_1 \cup f\,x_2 \qquad \text{:-)}
\end{aligned}
$$

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}$, $\quad \textsf{inc}\, x = x + 1$

für

,

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}$, $\quad \mathsf{inc}\, x = x + 1$

  Striktheit: $\quad f \perp = \mathsf{inc}\, 0 = 1 \quad \neq \quad \perp \quad$:-(

für

,

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \mathsf{inc}\, x = x + 1$

  Striktheit: $f\,\bot = \mathsf{inc}\, 0 = 1 \quad \neq \quad \bot$:-(

  Distributivität: $f\,(\bigsqcup X) \quad = \quad \bigsqcup\{x+1 \mid x \in X\}$ für $\emptyset \neq X$:-)

,

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \mathsf{inc}\, x = x + 1$

  Striktheit: $\quad f\,\bot = \mathsf{inc}\, 0 = 1 \quad \neq \quad \bot \quad$:-(

  Distributivität: $\quad f\,(\bigsqcup X) \quad = \quad \bigsqcup\{x + 1 \mid x \in X\}$ für $\emptyset \neq X$:-)

- $\mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2, \quad \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad f\,(x_1, x_2) = x_1 + x_2$

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \mathsf{inc}\, x = x + 1$

  Striktheit: $f\,\bot = \mathsf{inc}\, 0 = 1 \quad \neq \quad \bot \quad$:-(

  Distributivität: $f\,(\bigsqcup X) \quad = \quad \bigsqcup\{x + 1 \mid x \in X\}$ für $\emptyset \neq X \quad$:-)

- $\mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2, \quad \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2$:

  Striktheit: $f\,\bot = 0 + 0 \quad = \quad 0 \qquad$:-)

- $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad \mathsf{inc}\, x = x + 1$

  Striktheit: $f\,\bot = \mathsf{inc}\, 0 = 1 \quad \neq \quad \bot \quad$:-(

  Distributivität: $f\,(\bigsqcup X) \quad = \quad \bigsqcup\{x + 1 \mid x \in X\}$ für $\emptyset \neq X \quad$:-)

- $\mathbb{D}_1 = (\mathbb{N} \cup \{\infty\})^2, \quad \mathbb{D}_2 = \mathbb{N} \cup \{\infty\}, \quad f(x_1, x_2) = x_1 + x_2:$

  Striktheit: $f\,\bot = 0 + 0 \quad = \quad 0 \qquad$:-)

  Distributivität:

$$
\begin{array}{rcl}
f\,((1,4) \sqcup (4,1)) & = & f\,(4,4) \quad = \quad 8 \\
 & \neq & 5 \quad = \quad f\,(1,4) \sqcup f\,(4,1) \qquad \text{:-)}
\end{array}
$$

Bemerkung:

Ist $f : \mathbb{D}_1 \to \mathbb{D}_2$ distributiv, dann auch monoton :-)

Bemerkung:

Ist $f : \mathbb{D}_1 \to \mathbb{D}_2$ distributiv, dann auch monoton :-)

Offenbar gilt: $a \sqsubseteq b$ gdw. $a \sqcup b = b$.

Bemerkung:

Ist $f : \mathbb{D}_1 \to \mathbb{D}_2$ distributiv, dann auch monoton :-)

Offenbar gilt: $a \sqsubseteq b$ gdw. $a \sqcup b = b$.

Daraus folgt:

$$
\begin{array}{lrcl}
 & f\,b & = & f\,(a \sqcup b) \\
 & & = & f\,a \sqcup f\,b \\
\Longrightarrow & f\,a & \sqsubseteq & f\,b \qquad\qquad \text{:-)}
\end{array}
$$

Annahme: alle v sind von $start$ erreichbar.

Annahme: alle v sind von *start* erreichbar.
Dann gilt:

Theorem

Kildall 1972

Sind alle Kanten-Effekte $[\![k]\!]^\sharp$ distributiv, dann ist:
$\mathcal{I}^*[v] = \mathcal{I}[v]$ für alle v .

Gary A. Kildall (1942-1994).

Hat später am Betriebssystem CP/M und an GUIs für PCs gearbeitet.

Annahme: alle v sind von *start* erreichbar.
Dann gilt:

Theorem Kildall 1972

Sind alle Kanten-Effekte $[\![k]\!]^\sharp$ distributiv, dann ist:
$\mathcal{I}^*[v] = \mathcal{I}[v]$ für alle v .

Annahme: alle v sind von *start* erreichbar.
Dann gilt:

Theorem

Kildall 1972

Sind alle Kanten-Effekte $[\![k]\!]^\sharp$ distributiv, dann ist:
$\mathcal{I}^*[v] = \mathcal{I}[v]$ für alle v .

Beweis:

Offenbar genügt es zu zeigen, dass $\mathcal{I}^*$ eine Lösung ist :-)

Wir zeigen, dass $\mathcal{I}^*$ alle Ungleichungen erfüllt :-))

Wir zeigen, dass $\mathcal{I}^*$ alle Ungleichungen erfüllt :-))

(1) Für *start* zeigen wir:

$$\begin{array}{lll} \mathcal{I}^*[\textit{start}] & = & \bigsqcup\{[\![\pi]\!]^\sharp\, d_0 \mid \pi : \textit{start} \rightarrow^* \textit{start}\} \\ & \sqsupseteq & [\![\epsilon]\!]^\sharp\, d_0 \\ & \sqsupseteq & d_0 \qquad \text{:-)} \end{array}$$