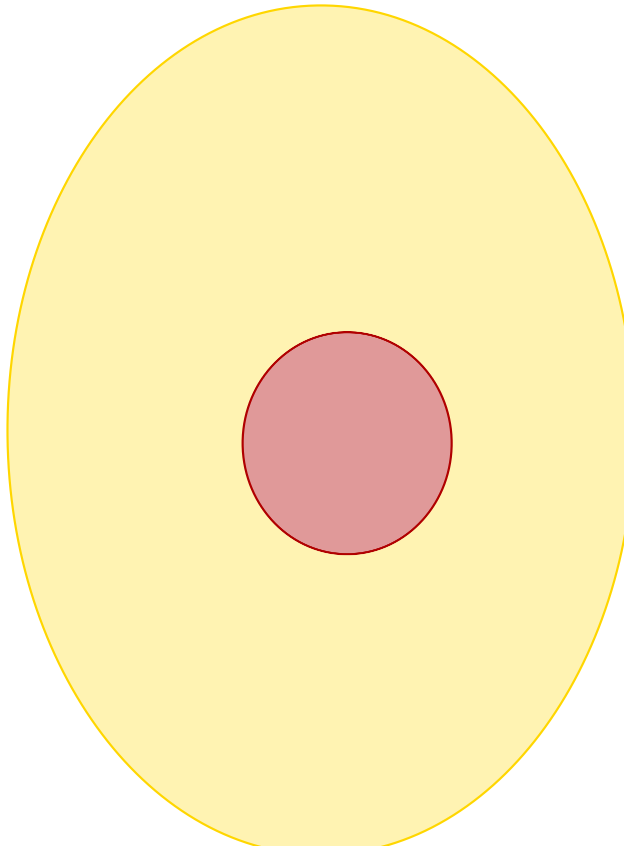
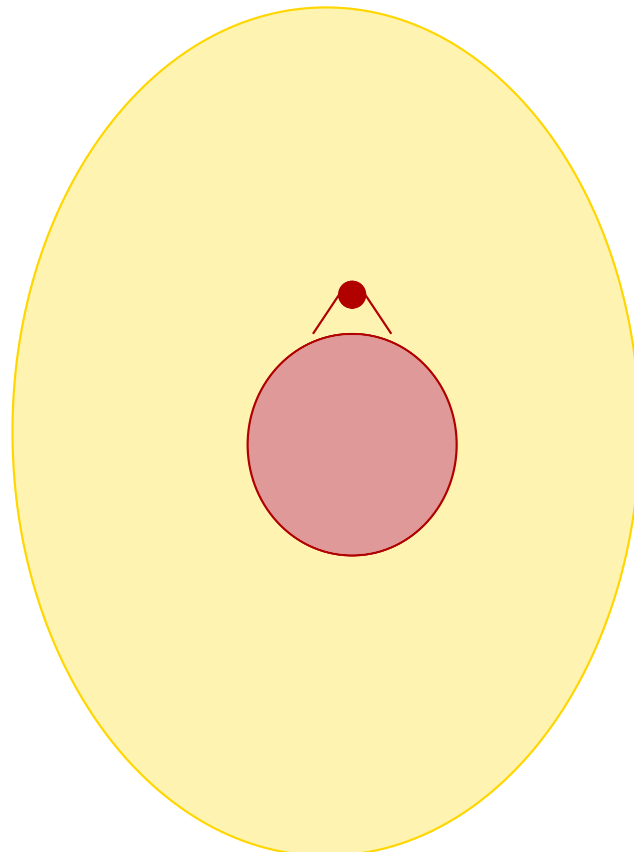
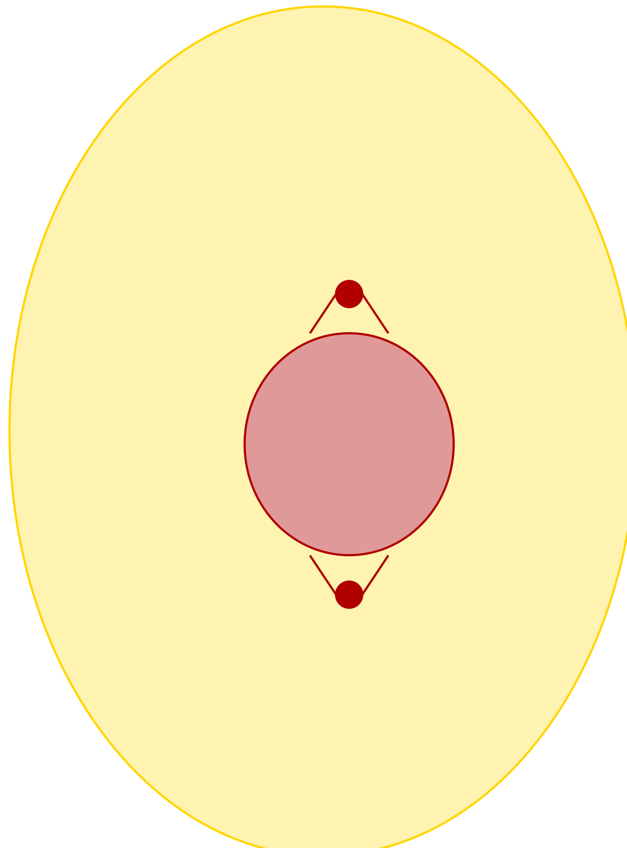






Wir suchen Lösungen für Ungleichungssysteme der Form:

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

Wir suchen Lösungen für Ungleichungssysteme der Form:

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

wobei:

x_i	Unbekannte	hier:	$\text{First}_k(A)$
$\mathbb{D}$	Werte	hier:	$\mathbb{D}_k = 2^{T^{\leq k}}$
$\sqsubseteq \quad \subseteq \quad \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\subseteq$
$f_i \colon \mathbb{D}^n \longrightarrow \mathbb{D}$	Bedingung	hier:	...

Wir suchen Lösungen für Ungleichungssysteme der Form:

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

wobei:

x_i	Unbekannte	hier:	$\mathsf{First}_k(A)$
$\mathbb{D}$	Werte	hier:	$\mathbb{D}_k = 2^{T^{\leq k}}$
$\sqsubseteq \quad \subseteq \quad \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\subseteq$
$f_i \colon \mathbb{D}^n \to \mathbb{D}$	Bedingung	hier:	...

Ungleichung für $\mathsf{First}_k(A)$:

$$\mathsf{First}_k(A) \quad \supseteq \quad \bigcup \{\mathsf{First}_k(X_1) \odot \ldots \odot \mathsf{First}_k(X_m) \mid A \to X_1 \ldots X_m \;\in\; P\}$$

Wir suchen Lösungen für Ungleichungssysteme der Form:

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

wobei:

x_i	Unbekannte	hier:	$\mathsf{First}_k(A)$
$\mathbb{D}$	Werte	hier:	$\mathbb{D}_k = 2^{T^{\leq k}}$
$\sqsubseteq \;\; \subseteq \;\; \mathbb{D} \times \mathbb{D}$	Ordnungsrelation	hier:	$\subseteq$
$f_i \colon \mathbb{D}^n \to \mathbb{D}$	Bedingung	hier:	...

Ungleichung für $\mathsf{First}_k(A)$:

$$\mathsf{First}_k(A) \quad \supseteq \quad \bigcup\{\mathsf{First}_k(X_1) \odot \ldots \odot \mathsf{First}_k(X_m) \mid A \to X_1 \ldots X_m \;\in\; P\}$$

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 :-)

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

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

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

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

- 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 Kleene

In einer vollständigen Halbordnung $\mathbb{D}$ hat jede stetige Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Dieser ist gegeben durch $d_0 = \bigsqcup_{k \geq 0} f^k \bot$.

Satz **Kleene**

In einer vollständigen Halbordnung $\mathbb{D}$ hat jede stetige Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Dieser ist gegeben durch $d_0 = \bigsqcup_{k \geq 0} f^k \bot$.

Bemerkung:

- Eine Funktion f heißt stetig, falls für jede aufsteigende Kette $d_0 \sqsubseteq \ldots \sqsubseteq d_m \sqsubseteq \ldots$ gilt: $f\,(\bigsqcup_{m \geq 0} d_m) = \bigsqcup_{m \geq 0}(f\,d_m)$.
- Werden alle aufsteigenden Ketten irgendwann stabil, ist jede monotone Funktion automatisch stetig :-)

Satz — Kleene

In einer vollständigen Halbordnung $\mathbb{D}$ hat jede stetige Funktion $f : \mathbb{D} \to \mathbb{D}$ einen kleinsten Fixpunkt d_0.

Dieser ist gegeben durch $d_0 = \bigsqcup_{k \geq 0} f^k \bot$.

Bemerkung:

- Eine Funktion f heißt stetig, falls für jede aufsteigende Kette $d_0 \sqsubseteq \ldots \sqsubseteq d_m \sqsubseteq \ldots$ gilt: $f\,(\bigsqcup_{m \geq 0} d_m) = \bigsqcup_{m \geq 0}(f\,d_m)$.
- Werden alle aufsteigenden Ketten irgendwann stabil, ist jede monotone Funktion automatisch stetig :-)
- Eine Halbordnung heißt vollständig (CPO), falls alle aufsteigenden Ketten kleinste obere Schranken haben :-)
- Jeder vollständige Verband ist auch eine vollständige Halbordnung :-)

Beweis:

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

$$
\begin{array}{lll}
f\,d_0 & = & f\left(\bigsqcup_{m\geq 0}(f^m\bot)\right) \\
& = & \bigsqcup_{m\geq 0}(f^{m+1}\bot) \qquad \text{wegen Stetigkeit} \quad \text{:-)} \\
& = & \bot \sqcup \left(\bigsqcup_{m\geq 0}(f^{m+1}\bot)\right) \\
& = & \bigsqcup_{m\geq 0}(f^m\bot) \\
& = & d_0
\end{array}
$$

(2) $\quad d_0$ ist kleinster Fixpunkt:

Sei $f\,d_1 = d_1$ weiterer Fixpunkt. Wir zeigen: $\forall\, m \geq 0 :\; f^m\bot \sqsubseteq d_1$.

$$
\begin{array}{llllll}
m = 0 : & & \bot & \sqsubseteq & d_1 & \text{nach Definition} \\
m > 0 : & \text{Gelte} & f^{m-1}\bot & \sqsubseteq & d_1 & \text{Dann folgt:} \\
& & f^m\bot & = & f\,(f^{m-1}\bot) & \\
& & & \sqsubseteq & f\,d_1 & \text{wegen Monotonie} \quad \text{:-)} \\
& & & = & d_1 &
\end{array}
$$

Bemerkung:

- Jede stetige Funktion ist auch monoton :-)
- Betrachte die Menge der Postfixpunkte:

$$P = \{x \in \mathbb{D} \mid x \sqsupseteq f\,x\}$$

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:

- Jede stetige Funktion ist auch monoton :-)
- Betrachte die Menge der Postfixpunkte:

$$P = \{x \in \mathbb{D} \mid x \sqsupseteq f\,x\}$$

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:

- Jede stetige Funktion ist auch monoton :-)
- Betrachte die Menge der Postfixpunkte:

$$P = \{x \in \mathbb{D} \mid x \sqsupseteq f\, x\}$$

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 :-)

Der Kleenesche Fixpunkt-Satz liefert uns nicht nur die Existenz einer kleinsten Lösung sondern auch eine Charakterisierung :-)

Satz

Die Mengen $\mathsf{First}_k(\{w \in T^* \mid A \to^* w\})$, $A \in N$, sind die kleinste Lösung des Ungleichungssystems:

$$\mathsf{First}_k(A) \supseteq \mathsf{First}_k(X_1) \odot \ldots \odot \mathsf{First}_k(X_m) \quad , \quad A \to X_1 \ldots X_m \in P$$

Der Kleenesche Fixpunkt-Satz liefert uns nicht nur die Existenz einer kleinsten Lösung sondern auch eine Charakterisierung :-)

Satz

Die Mengen $\mathsf{First}_k(\{w \in T^* \mid A \rightarrow^* w\})$, $A \in N$, sind die kleinste Lösung des Ungleichungssystems:

$$\mathsf{First}_k(A) \supseteq \mathsf{First}_k(X_1) \odot \ldots \odot \mathsf{First}_k(X_m) \;, \quad A \rightarrow X_1 \ldots X_m \in P$$

Beweis-Idee:

Sei $\mathsf{F}^{(m)}(A)$ die m-te Approximation an den Fixpunkt.

(1) Falls $A \rightarrow^m u$, dann $\mathsf{First}_k(u) \subseteq \mathsf{F}^{(m)}(A)$.

(2) Falls $w \in \mathsf{F}^{(m)}(A)$, dann $A \rightarrow^* u$ für $u \in T^*$ mit $\mathsf{First}_k(u) = \{w\}$:-)

Fazit:

Wir können First_k durch Fixpunkt-Iteration berechnen, d.h. durch wiederholtes Einsetzen :-)

Fazit:

Wir können First_k durch Fixpunkt-Iteration berechnen, d.h. durch wiederholtes Einsetzen :-)

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

Fazit:

Wir können First_k durch Fixpunkt-Iteration berechnen, d.h. durch wiederholtes Einsetzen :-)

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

Idee: Round Robin Iteration

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

Unser Mini-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 Round-Robin-Iteration:

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

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 :-))

Unsere Anwendung:

$$\begin{array}{lcl}
\mathsf{First}_2(E) & \supseteq & \mathsf{First}_2(E) \odot \{+\} \odot \mathsf{First}_2(T) \quad \cup \quad \mathsf{First}_2(T) \\
\mathsf{First}_2(T) & \supseteq & \mathsf{First}_2(T) \odot \{*\} \odot \mathsf{First}_2(F) \quad \cup \quad \mathsf{First}_2(F) \\
\mathsf{First}_2(F) & \supseteq & \{(\} \odot \mathsf{First}_2(E) \odot \{)\} \quad \cup \quad \{\mathsf{name}, \mathsf{int}\}
\end{array}$$

Die RR-Iteration:

First_2	1	2	3
F	name, int	(name, (int	((
T	name, int	(name, (int, name $*$, int $*$	((
E	name, int	(name, (int, name $*$, int $*$, name $+$, int $+$	((

Der Einfachheit halber haben wir in jeder Iteration nur die neuen Elemente vermerkt :-)

Diskussion:

- Die Länge h der längsten echt aufsteigenden Kette nennen wir auch Höhe von $\mathbb{D}$ ...
- Im Falle von $\textsf{First}_k$ ist die Höhe des Verbands exponentiell in k :-(
- Die Anzahl der Runden von RR-Iteration ist beschränkt durch $\mathcal{O}(n \cdot h)$ (n die Anzahl der Variablen)
- Die praktische Effizienz von RR-Iteration hängt allerdings auch von der Anordnung der Variablen ab :-)
- Anstelle von RR-Iteration gibt es auch schnellere Fixpunkt-Verfahren, die aber im schlimmsten Fall immer noch exponentiell sind :-((

$\Longrightarrow$ Man beschränkt sich i.a. auf kleine k !!!

2.4 Topdown Parsing

Idee:

- Benutze den Item-Kellerautomaten.
- Benutze die nächsten k Zeichen, um die Regeln für die Expansionen zu bestimmen ;-)
- Eine Grammatik heißt $LL(k)$, falls dies immer eindeutig möglich ist.

2.4 Topdown Parsing

Idee:

- Benutze den Item-Kellerautomaten.
- Benutze die nächsten k Zeichen, um die Regeln für die Expansionen zu bestimmen ;-)
- Eine Grammatik heißt $LL(k)$, falls dies immer eindeutig möglich ist.

Wir definieren:

Eine reduzierte Grammatik heißt dann $LL(k)$, falls für je zwei verschiedene Regeln $A \to \alpha$, $A \to \alpha' \in P$ und jede Ableitung $S \to_L^* u\, A\, \beta$ mit $u \in T^*$ gilt:

$$\mathsf{First}_k(\alpha\, \beta) \;\cap\; \mathsf{First}_k(\alpha'\, \beta) = \emptyset$$

Beispiel 1:

$$\begin{array}{lll} S & \rightarrow & \text{if (} E \text{) } S \text{ else } S \quad \mid \\ & & \text{while (} E \text{) } S \quad \mid \\ & & E \text{ ;} \\ E & \rightarrow & \text{id} \end{array}$$

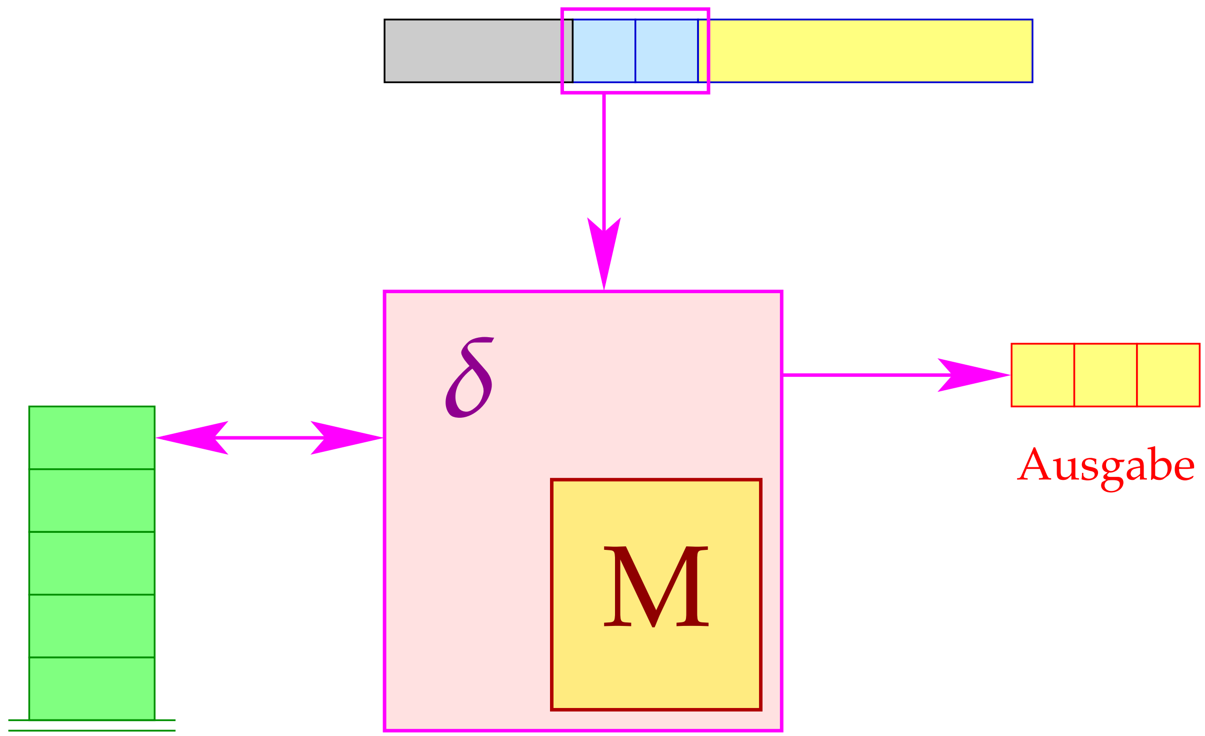
ist $LL(1)$, da $\text{First}_k(E) = \{\text{id}\}$:-)

Beispiel 2:

$$\begin{array}{lll} S & \to & \text{if} \ (\ E \ ) \ S \ \text{else} \ S \quad | \\ & & \text{if} \ (\ E \ ) \ S \quad | \\ & & \text{while} \ (\ E \ ) \ S \quad | \\ & & E \ ; \\ E & \to & \text{id} \end{array}$$

... ist nicht $LL(k)$ für jedes $k > 0$.

Struktur des $LL(k)$-Parsers:



- Der Parser sieht ein Fenster der Länge k der Eingabe;
- er realisiert im Wesentlichen den Item-Kellerautomaten;
- die Tabelle $M[q, w]$ enthält die jeweils zuwählende Regel :-)

... im Beispiel:

$$
\begin{array}{lll}
S & \to & \text{if } (\; E \;) \; S \text{ else } S^{\,0} \quad | \\
 & & \text{while } (\; E \;) \; S^{\,1} \quad | \\
 & & E \; ;^{\,2} \\
E & \to & \text{id}^{\,0}
\end{array}
$$

Zustände: Items

Tabelle:

	if	while	id
$[\ldots \to \ldots \bullet S \ldots]$	0	1	2
$[\ldots \to \ldots \bullet E \ldots]$	—	—	0

Im Allgemeinen ...

- ist die Menge der möglichen nächsten k Zeichen gegeben durch:

$$\mathsf{First}_k(\alpha\,\beta) \;=\; \mathsf{First}_k(\alpha) \odot \mathsf{First}_k(\beta)$$

wobei:

(1) α die rechte Seite der passenden Regel;

(2) β ein möglicher rechter Kontext von A ist :-)

- $\mathsf{First}_k(\beta)$ müssen wir dynamisch akkumulieren.

$\Longrightarrow$ Wir erweitern Items um Vorausschau-Mengen ...

Ein erweitertes Item ist ein Paar: $[A \to \alpha \bullet \gamma\, ,\ L] \quad (A \to \alpha\,\gamma \in P,\ L \subseteq T^{\leq k})$

Die Menge L benutzen wir, um $\mathsf{First}_k(\beta)$ für den rechten Kontext β von A zu repräsentieren :-)

Ein erweitertes Item ist ein Paar: $[A \to \alpha \bullet \gamma\,,\ L]$ $(A \to \alpha\,\gamma\ \in\ P,\ L \subseteq T^{\leq k})$

Die Menge L benutzen wir, um $\mathsf{First}_k(\beta)$ für den rechten Kontext β von A zu repräsentieren :-)

Konstruktion:

Zustände: erweiterte Items

Anfangszustand: $[S' \to \bullet\ S,\ \{\epsilon\}]$

Endzustand: $[S' \to S\,\bullet,\ \{\epsilon\}]$

Übergänge:

Ein erweitertes Item ist ein Paar: $[A \to \alpha \bullet \gamma\,,\ L]$ $(A \to \alpha\,\gamma \in P,\ L \subseteq T^{\leq k})$

Die Menge L benutzen wir, um $\text{First}_k(\beta)$ für den rechten Kontext β von A zu repräsentieren :-)

Konstruktion:

Zustände: erweiterte Items

Anfangszustand: $[S' \to \bullet\ S,\ \{\epsilon\}]$

Endzustand: $[S' \to S\,\bullet,\ \{\epsilon\}]$

Übergänge:

Expansionen: $([A \to \alpha \bullet B\,\beta\,,\ L], \epsilon, [A \to \alpha \bullet B\,\beta\,,\ L]\ [B \to \bullet\,\gamma\,,\ \boxed{\text{First}_k(\beta) \odot L}])$ für $A \to \alpha\,B\,\beta,\ B \to \gamma \in P$

Shifts: $([A \to \alpha \bullet a\,\beta\,,\ L], a, [A \to \alpha\,a \bullet \beta\,,\ L])$ für $A \to \alpha\,a\,\beta \in P$

Reduce: $([A \to \alpha \bullet B\,\beta\,,\ L]\ [B \to \gamma\bullet\,,\ L'], \epsilon, [A \to \alpha\,B \bullet \beta\,,\ L])$ für $A \to \alpha\,B\,\beta,\ B \to \gamma \in P$