Beobachtung:

- Die auszuwählende Regel hängt hier ja gar nicht von den Erweiterungen der Items ab !!!
- Unter dieser Voraussetzung können wir den Item-Kellerautomaten ohne Erweiterung benutzen :-)
- Hängt die auszuwählende Regel nur von der aktuellen Vorausschau w ab, nennen wir G auch stark $LL(k)$ ...

Beobachtung:

- Die auszuwählende Regel hängt hier ja gar nicht von den Erweiterungen der Items ab !!!
- Unter dieser Voraussetzung können wir den Item-Kellerautomaten ohne Erweiterung benutzen :-)
- Hängt die auszuwählende Regel nur von der aktuellen Vorausschau w ab, nennen wir G auch stark $LL(k)$ ...

Wir definieren: $\mathsf{Follow}_k(A) = \bigcup\{\mathsf{First}_k(\beta) \mid S \to_L^* u\, A\, \beta\}$.

Beobachtung:

- Die auszuwählende Regel hängt hier ja gar nicht von den Erweiterungen der Items ab !!!
- Unter dieser Voraussetzung können wir den Item-Kellerautomaten ohne Erweiterung benutzen :-)
- Hängt die auszuwählende Regel nur von der aktuellen Vorausschau w ab, nennen wir G auch stark $LL(k)$ ...

Wir definieren: $\mathsf{Follow}_k(A) = \bigcup\{\mathsf{First}_k(\beta) \mid S \rightarrow_L^* u\, A\, \beta\}$.

Die reduzierte kontextfreie Grammatik G heißt stark $LL(k)$, falls für je zwei verschiedene $A \rightarrow \alpha\,,\ A \rightarrow \alpha' \in P$:

$$\mathsf{First}_k(\alpha) \odot \mathsf{Follow}_k(A) \cap \mathsf{First}_k(\alpha') \odot \mathsf{Follow}_k(A) = \emptyset$$

... im Beispiel: $S \to \epsilon \quad | \quad a\,S\,b$

$$\mathsf{Follow}_1(S) \quad = \quad \{\epsilon, b\}$$

$$\begin{array}{lllll} \mathsf{First}_1(\epsilon) \odot \mathsf{Follow}_1(S) & = & \{\epsilon\} \odot \{\epsilon, b\} & = & \{\epsilon, b\} \\ \mathsf{First}_1(a\,S\,b) \odot \mathsf{Follow}_1(S) & = & \{a\} \odot \{\epsilon, b\} & = & \{a\} \end{array}$$

Wir schließen: Die Grammatik ist in der Tat stark $LL(1)$:-)

Ist G eine starke $LL(k)$-Grammatik, können wir die Vorausschau-Tabelle statt mit (erweiterten) Items mit Nichtterminalen indizieren :-)

Wir setzen $M[B,\ w] = i$ genau dann wenn (B, i) die Regel $B \to \gamma$ ist und: $w \in \mathsf{First}_k(\gamma) \odot \mathsf{Follow}_k(B)$.

... im Beispiel:

$$S \to \epsilon \quad | \quad a\,S\,b$$

	ϵ	a	b
S	0	1	0

Ist G eine starke $LL(k)$-Grammatik, können wir die Vorausschau-Tabelle statt mit (erweiterten) Items mit Nichtterminalen indizieren :-)

Wir setzen $M[B, w] = i$ genau dann wenn (B, i) die Regel $B \to \gamma$ ist und: $w \in \mathsf{First}_k(\gamma) \odot \mathsf{Follow}_k(B)$.

... im Beispiel:

$$S \to \epsilon \quad | \quad a\,S\,b$$

	ϵ	a	b
S	0	1	0

Satz

- Jede starke $LL(k)$-Grammatik ist auch $LL(k)$:-)
- Jede $LL(1)$-Grammatik ist bereits stark $LL(1)$:-))

Beweis:

Sei G stark $LL(k)$.

Betrachte eine Ableitung $S \rightarrow_L^* u\, A\, \beta$ und Regeln $A \rightarrow \alpha\,,\; A \rightarrow \alpha' \;\in\; P$.

Dann haben wir:

$$\begin{array}{lll}
\mathsf{First}_k(\alpha\,\beta) \;\cap\; \mathsf{First}_k(\alpha'\,\beta) & = & \mathsf{First}_k(\alpha) \odot \mathsf{First}_k(\beta) \;\cap\; \mathsf{First}_k(\alpha') \odot \mathsf{First}_k(\beta) \\
 & \subseteq & \mathsf{First}_k(\alpha) \odot \mathsf{Follow}_k(A) \;\cap\; \mathsf{First}_k(\alpha') \odot \mathsf{Follow}_k(A) \\
 & = & \emptyset
\end{array}$$

Folglich ist G auch $LL(k)$:-)

Sei G $LL(1)$.

Betrachte zwei verschiedene Regeln $A \rightarrow \alpha\,,\; A \rightarrow \alpha' \;\in\; P$.

Fall 1: $\epsilon \in \mathsf{First}_1(\alpha) \cap \mathsf{First}_1(\alpha')$.

Dann kann G nicht $LL(1)$ sein :-)

Sei G $LL(1)$.

Betrachte zwei verschiedene Regeln $A \to \alpha\,,\; A \to \alpha' \;\in\; P$.

Fall 1: $\epsilon \in \mathsf{First}_1(\alpha) \cap \mathsf{First}_1(\alpha')$.

Dann kann G nicht $LL(1)$ sein :-)

Fall 2: $\epsilon \notin \mathsf{First}_1(\alpha) \cup \mathsf{First}_1(\alpha')$.

Sei $S \to_L^* u\, A\, \beta$. Da G $LL(1)$ ist, gilt:

$$
\begin{aligned}
\mathsf{First}_1(\alpha) \odot \mathsf{Follow}_1(A) \cap \mathsf{First}_1(\alpha') \odot \mathsf{Follow}_1(A) \\
&= \mathsf{First}_1(\alpha) \cap \mathsf{First}_1(\alpha') \\
&= \mathsf{First}_1(\alpha) \odot \mathsf{First}_1(\beta) \cap \mathsf{First}_1(\alpha') \odot \mathsf{First}_1(\beta) \\
&= \emptyset
\end{aligned}
$$

Fall 3: $\epsilon \in \mathsf{First}_1(\alpha)$ und $\epsilon \notin \mathsf{First}_1(\alpha')$.

Dann gilt:

$$\begin{aligned}
&\mathsf{First}_1(\alpha) \odot \mathsf{Follow}_1(A) \cap \mathsf{First}_1(\alpha') \odot \mathsf{Follow}_1(A) \\
&\qquad = \mathsf{First}_1(\alpha) \odot \mathsf{Follow}_1(A) \cap \mathsf{First}_1(\alpha') \\
&\qquad = \mathsf{First}_1(\alpha) \odot (\bigcup\{\mathsf{First}_1(\beta) \mid S \rightarrow^*_L u\, A\, \beta\}) \cap \mathsf{First}_1(\alpha') \\
&\qquad = (\bigcup\{\mathsf{First}_1(\alpha) \odot \mathsf{First}_1(\beta) \mid S \rightarrow^*_L u\, A\, \beta\}) \cap \mathsf{First}_1(\alpha') \\
&\qquad = \bigcup\{\mathsf{First}_1(\alpha) \odot \mathsf{First}_1(\beta) \cap \mathsf{First}_1(\alpha') \mid S \rightarrow^*_L u\, A\, \beta\} \\
&\qquad = \bigcup\{\emptyset \mid S \rightarrow^*_L u\, A\, \beta\} \\
&\qquad = \emptyset
\end{aligned}$$

Fall 3: $\epsilon \in \mathsf{First}_1(\alpha)$ und $\epsilon \notin \mathsf{First}_1(\alpha')$.

Dann gilt:

$$
\begin{aligned}
\mathsf{First}_1(\alpha) \odot \mathsf{Follow}_1(A) \cap \mathsf{First}_1(\alpha') \odot \mathsf{Follow}_1(A) \\
&= \mathsf{First}_1(\alpha) \odot \mathsf{Follow}_1(A) \cap \mathsf{First}_1(\alpha') \\
&= \mathsf{First}_1(\alpha) \odot (\bigcup\{\mathsf{First}_1(\beta) \mid S \to_L^* u\, A\, \beta\}) \cap \mathsf{First}_1(\alpha') \\
&= (\bigcup\{\mathsf{First}_1(\alpha) \odot \mathsf{First}_1(\beta) \mid S \to_L^* u\, A\, \beta\}) \cap \mathsf{First}_1(\alpha') \\
&= \bigcup\{\mathsf{First}_1(\alpha) \odot \mathsf{First}_1(\beta) \cap \mathsf{First}_1(\alpha') \mid S \to_L^* u\, A\, \beta\} \\
&= \bigcup\{\emptyset \mid S \to_L^* u\, A\, \beta\} \\
&= \emptyset
\end{aligned}
$$

Fall 4: $\epsilon \notin \mathsf{First}_1(\alpha)$ und $\epsilon \in \mathsf{First}_1(\alpha')$: analog :-)

Beispiel:

$$\begin{array}{lll} S & \rightarrow & a\,A\,a\,a^{\,0} \;\mid\; b\,A\,b\,a^{\,1} \\ A & \rightarrow & b^{\,0} \;\mid\; \epsilon^{\,1} \end{array}$$

Offenbar ist die Grammatik $LL(2)$:-) Andererseits gilt:

$$\begin{array}{ll} \multicolumn{2}{l}{\mathsf{First}_2(b) \odot \mathsf{Follow}_2(A) \;\cap\; \mathsf{First}_2(\epsilon) \odot \mathsf{Follow}_2(A)} \\ = & \{b\} \odot \{a\,a\,,\, b\,a\} \;\cap\; \{\epsilon\} \odot \{a\,a\,,\, b\,a\} \\ = & \{b\,a\,,\, b\,b\} \;\cap\; \{a\,a\,,\, b\,a\} \\ \neq & \emptyset \end{array}$$

Folglich ist die Grammatik nicht stark $LL(2)$:-(

Beispiel:

$$\begin{array}{lll} S & \to & a\,A\,a\,a^{\,0} \mid b\,A\,b\,a^{\,1} \\ A & \to & b^{\,0} \mid \epsilon^{\,1} \end{array}$$

Offenbar ist die Grammatik $LL(2)$:-) Andererseits gilt:

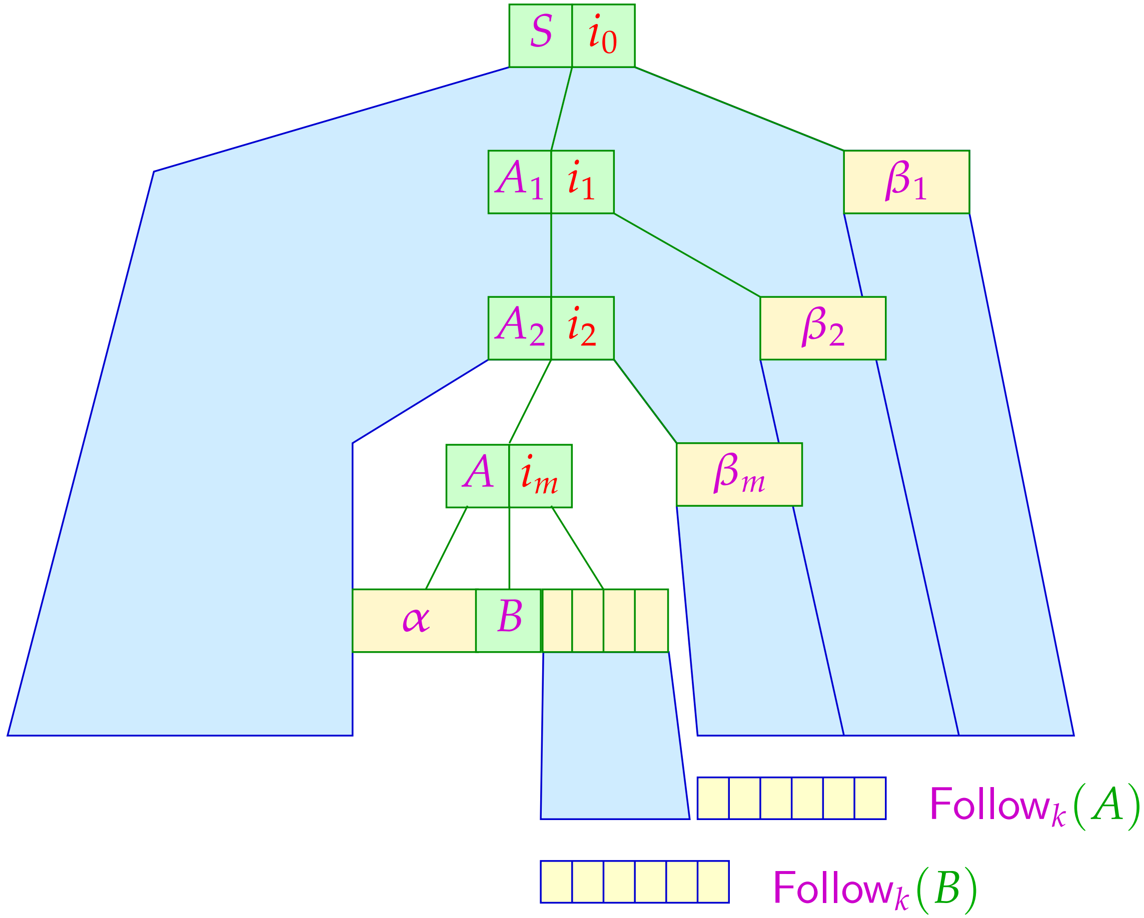
$$\begin{array}{rl} \multicolumn{2}{l}{\mathsf{First}_2(b) \odot \mathsf{Follow}_2(A) \cap \mathsf{First}_2(\epsilon) \odot \mathsf{Follow}_2(A)} \\ = & \{b\} \odot \{a\,a\,,\ b\,a\} \cap \{\epsilon\} \odot \{a\,a\,,\ b\,a\} \\ = & \{b\,a\,,\ b\,b\} \cap \{a\,a\,,\ b\,a\} \\ \neq & \emptyset \end{array}$$

Folglich ist die Grammatik nicht stark $LL(2)$:-(

Wir schließen:

- Für $k > 1$ ist nicht jede $LL(k)$-Grammatik automatisch stark $LL(k)$.
- Zu jeder $LL(k)$-Grammatik kann jedoch eine äquivalente starke $LL(k)$-Grammatik konstruiert werden $\Longrightarrow$ Übung!

Berechnung von $\mathsf{Follow}_k(B)$:



Berechnung von $\text{Follow}_k(B)$:

Idee:

- Wir stellen ein Ungleichungssystem auf :-)
- ϵ ist ein möglicher rechter Kontext von S :-)
- Mögliche rechte Kontexte der linken Seite einer Regel propagieren wir ans Ende jeder rechten Seite ...

... im Beispiel:

$$S \to \epsilon \quad | \quad a\,S\,b$$

$$\begin{array}{lll} \text{Follow}_k(S) & \supseteq & \{\epsilon\} \\ \text{Follow}_k(S) & \supseteq & \{b\} \odot \text{Follow}_k(S) \end{array}$$

Allgemein:

$$\begin{array}{lcl} \text{Follow}_k(S) & \supseteq & \{\epsilon\} \\ \text{Follow}_k(B) & \supseteq & \text{First}_k(X_1) \odot \ldots \odot \text{First}_k(X_m) \odot \text{Follow}_k(A) \\ & & \text{für} \quad A \to \alpha\, B\, X_1 \ldots X_m \in P \end{array}$$

Diskussion:

- Man überzeugt sich, dass die kleinste Lösung dieses Ungleichungssystems tatsächlich die Mengen $\text{Follow}_k(B)$ liefert :-)
- Die Größe der auftretenden Mengen steigt mit k rapide :-(
- In praktischen Systemen wird darum meist nur der Fall $k = 1$ implementiert ...

2.5 Schnelle Berechnung von Vorausschau-Mengen

Im Fall $k = 1$ lassen sich First, Follow besonders effizient berechnen ;-)

Beobachtung:

Seien $L_1, L_2 \subseteq T \cup \{\epsilon\}$ mit $L_1 \neq \emptyset \neq L_2$. Dann ist:

$$L_1 \odot L_2 = \begin{cases} L_1 & \text{falls } \epsilon \notin L_1 \\ (L_1 \backslash \{\epsilon\}) \cup L_2 & \text{sonst} \end{cases}$$

Ist G reduziert, sind alle Mengen $\text{First}_1(A)$ nichtleer :-)

Idee:

- Behandle ϵ separat!

  Sei $\text{empty}(X) = \text{true}$ gdw. $X \rightarrow^* \epsilon$.

- Definiere die ϵ-freien First_1-Mengen

$$\begin{array}{lcll} F_\epsilon(a) & = & \{a\} & \text{für} \quad a \in T \\ F_\epsilon(A) & = & \text{First}_1(A) \backslash \{\epsilon\} & \text{für} \quad A \in N \end{array}$$

- Konstruiere direkt ein Ungleichungssystem für $F_\epsilon(A)$:

$$\begin{array}{lcll} F_\epsilon(A) & \supseteq & F_\epsilon(X_j) & \text{falls} \quad A \rightarrow X_1 \ldots X_m \in P, \\ & & & \text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1}) \end{array}$$

... im Beispiel:

$$\begin{array}{lcl}
E & \to & E + T \quad | \quad T \\
T & \to & T * F \quad | \quad F \\
F & \to & (\, E \,) \quad | \quad \mathsf{name} \quad | \quad \mathsf{int}
\end{array}$$

wobei $\mathsf{empty}(E) = \mathsf{empty}(T) = \mathsf{empty}(F) = \mathsf{false}$.

Deshalb erhalten wir:

$$\begin{array}{lclclcl}
F_\epsilon(S') & \supseteq & F_\epsilon(E) & \quad & F_\epsilon(E) & \supseteq & F_\epsilon(E) \\
F_\epsilon(E) & \supseteq & F_\epsilon(T) & & F_\epsilon(T) & \supseteq & F_\epsilon(T) \\
F_\epsilon(T) & \supseteq & F_\epsilon(F) & & F_\epsilon(F) & \supseteq & \{\, (\, , \mathsf{name}, \mathsf{int} \}
\end{array}$$

Entsprechend konstruieren wir zur Berechnung von Follow_1 :

$$
\begin{array}{lllll}
\mathsf{Follow}_1(S) & \supseteq & \{\epsilon\} & & \\
\mathsf{Follow}_1(B) & \supseteq & F_\epsilon(X_j) & \text{falls} & A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & & \mathsf{empty}(X_1) \wedge \ldots \wedge \mathsf{empty}(X_{j-1}) \\
\mathsf{Follow}_1(B) & \supseteq & \mathsf{Follow}_1(A) & \text{falls} & A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & & \mathsf{empty}(X_1) \wedge \ldots \wedge \mathsf{empty}(X_m)
\end{array}
$$

Entsprechend konstruieren wir zur Berechnung von Follow_1 :

$$
\begin{array}{lcll}
\mathsf{Follow}_1(S) & \supseteq & \{\epsilon\} & \\
\mathsf{Follow}_1(B) & \supseteq & F_\epsilon(X_j) & \text{falls} \quad A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & \qquad \mathsf{empty}(X_1) \wedge \ldots \wedge \mathsf{empty}(X_{j-1}) \\
\mathsf{Follow}_1(B) & \supseteq & \mathsf{Follow}_1(A) & \text{falls} \quad A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & \qquad \mathsf{empty}(X_1) \wedge \ldots \wedge \mathsf{empty}(X_m)
\end{array}
$$

... im Beispiel:

$$
\begin{array}{lclclcl}
E & \to & E + T & \mid & T & & \\
T & \to & T * F & \mid & F & & \\
F & \to & (\ E\ ) & \mid & \mathsf{name} & \mid & \mathsf{int}
\end{array}
$$

Entsprechend konstruieren wir zur Berechnung von $\textsf{Follow}_1$:

$$\begin{array}{lllll}
\textsf{Follow}_1(S) & \supseteq & \{\epsilon\} & & \\
\textsf{Follow}_1(B) & \supseteq & F_\epsilon(X_j) & \text{falls} & A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & & \textsf{empty}(X_1) \wedge \ldots \wedge \textsf{empty}(X_{j-1}) \\
\textsf{Follow}_1(B) & \supseteq & \textsf{Follow}_1(A) & \text{falls} & A \to \alpha\, B\, X_1 \ldots X_m \;\in P, \\
 & & & & \textsf{empty}(X_1) \wedge \ldots \wedge \textsf{empty}(X_m)
\end{array}$$

... im Beispiel:

$$\begin{array}{lllllll}
E & \to & E + T & \mid & T & & \\
T & \to & T * F & \mid & F & & \\
F & \to & (\; E\;) & \mid & \textsf{name} & \mid & \textsf{int}
\end{array}$$

... erhalten wir:

$$\begin{array}{llllll}
\textsf{Follow}_1(S') & \supseteq & \{\epsilon\} & \textsf{Follow}_1(E) & \supseteq & \textsf{Follow}_1(S') \\
\textsf{Follow}_1(E) & \supseteq & \{+,)\} & \textsf{Follow}_1(T) & \supseteq & \{*\} \\
\textsf{Follow}_1(T) & \supseteq & \textsf{Follow}_1(E) & \textsf{Follow}_1(F) & \supseteq & \textsf{Follow}_1(T)
\end{array}$$

Diskussion:

- Diese Ungleichungssysteme bestehen aus Ungleichungen der Form:

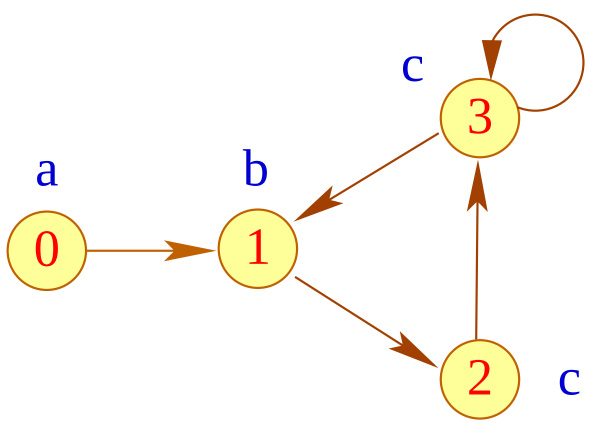
$$x \sqsupseteq y \qquad \text{bzw.} \qquad x \sqsupseteq d$$

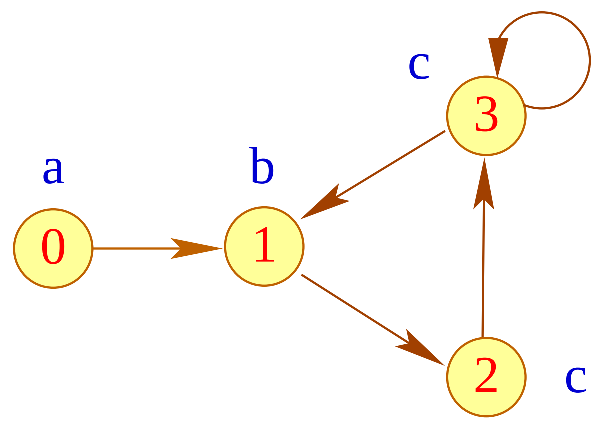
  für Variablen x, y und $d \in \mathbb{D}$.

- Solche Ungleichungssysteme heißen reine Vereinigungs-Probleme :-)
- Diese Probleme können mit linearem Aufwand gelöst werden ...

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

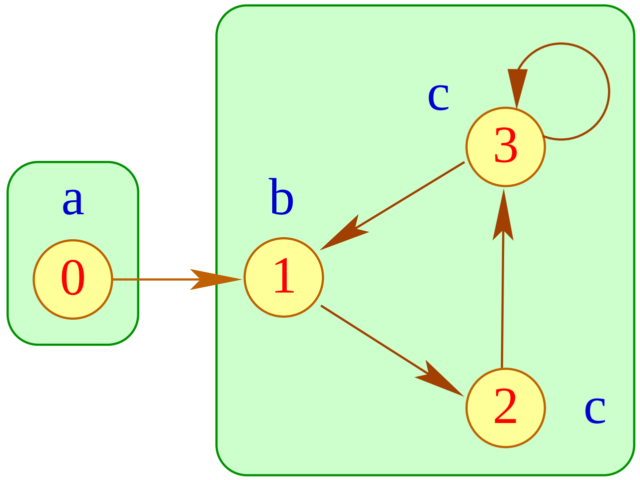
$$\begin{array}{lll} x_0 \supseteq \{a\} & & \\ x_1 \supseteq \{b\} & x_1 \supseteq x_0 & x_1 \supseteq x_3 \\ x_2 \supseteq \{c\} & x_2 \supseteq x_1 & \\ x_3 \supseteq \{c\} & x_3 \supseteq x_2 & x_3 \supseteq x_3 \end{array}$$





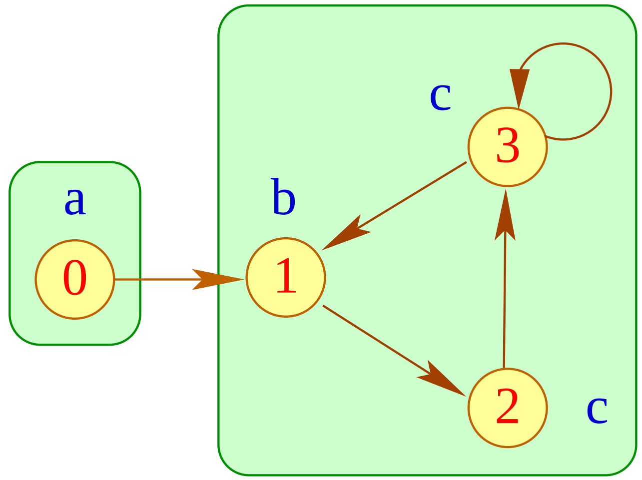
Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.



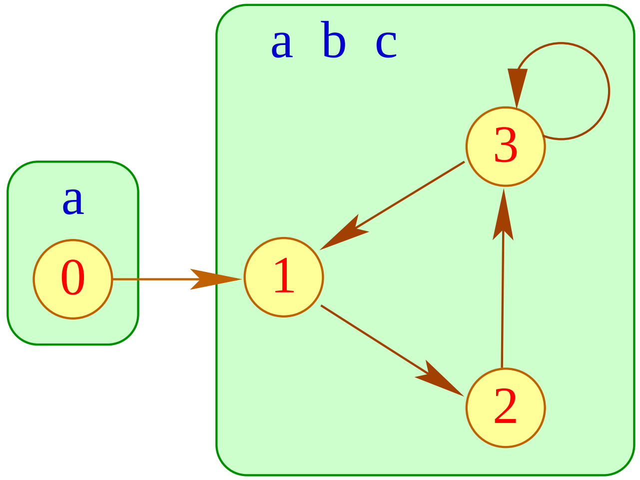
Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente haben alle Variablen den gleichen Wert :-)



Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente haben alle Variablen den gleichen Wert :-)
- Hat eine SZK keine eingehenden Kanten, erhält man ihren Wert, indem man die kleinste obere Schranke aller Werte in der SZK berechnet :-)

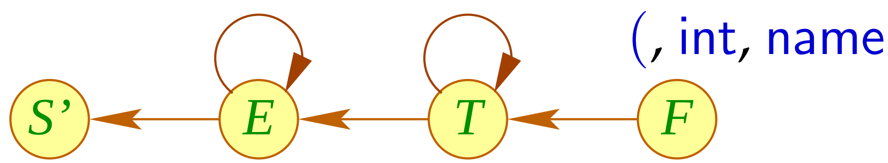


Vorgehen:

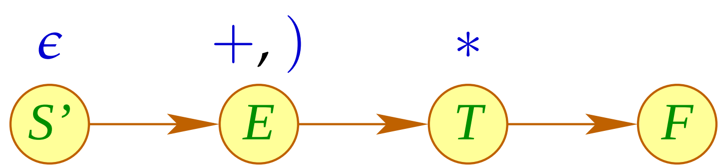
- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente haben alle Variablen den gleichen Wert :-)
- Hat eine SZK keine eingehenden Kanten, erhält man ihren Wert, indem man die kleinste obere Schranke aller Werte in der SZK berechnet :-)
- Gibt es eingehende Kanten, muss man zusätzlich die Werte an deren Startknoten hinzufügen :-)

... für unsere Beispiel-Grammatik:

First$_1$:



Follow$_1$:



2.6 Bottom-up Analyse

Achtung:

- Viele Grammatiken sind nicht $LL(k)$:-(
- Eine Grund ist Links-Rekursivität ...
- Die Grammatik G heißt links-rekursiv, falls

$$A \rightarrow^{+} A\,\beta \qquad \text{für ein} \quad A \in N\,,\ \beta \in (T \cup N)^*$$

2.6 Bottom-up Analyse

Achtung:

- Viele Grammatiken sind nicht $LL(k)$:-(
- Eine Grund ist Links-Rekursivität ...
- Die Grammatik G heißt links-rekursiv, falls

$$A \rightarrow^+ A\,\beta \qquad \text{für ein} \quad A \in N\,,\ \beta \in (T \cup N)^*$$

Beispiel:

$$\begin{array}{lcllll} E & \rightarrow & E + T & \mid & T & \\ T & \rightarrow & T * F & \mid & F & \\ F & \rightarrow & (\ E\ ) & \mid & \mathsf{name} & \mid\ \mathsf{int} \end{array}$$

... ist links-rekursiv :-)

Satz

Ist die Grammatik G reduziert und links-rekursiv, dann ist G nicht $LL(k)$ für jedes k.

Satz

Ist die Grammatik G reduziert und links-rekursiv, dann ist G nicht $LL(k)$ für jedes k .

Beweis:

Vereinfachung: $A \to A\,\beta \in P$

A erreichbar $\Longrightarrow$ $S \to_L^* u\,A\,\gamma \to_L^* u\,A\,\beta^n\,\gamma$ für jedes $n \geq 0$.

A produktiv $\Longrightarrow$ $\exists\, A \to \alpha \,:\, \alpha \neq A\,\beta$.

Satz

Ist die Grammatik G reduziert und links-rekursiv, dann ist G nicht $LL(k)$ für jedes k .

Beweis: Vereinfachung: $A \to A\,\beta \in P$

A erreichbar $\Longrightarrow$ $S \to_L^* u\,A\,\gamma \to_L^* u\,A\,\beta^n\,\gamma$ für jedes $n \geq 0$.

A produktiv $\Longrightarrow$ $\exists\, A \to \alpha \;:\; \alpha \neq A\,\beta$.

Annahme: G ist $LL(k)$;-) Dann gilt für alle $n \geq 0$:

	$\text{First}_k(\alpha\,\beta^n\,\gamma)$	$\cap$	$\text{First}_k(A\,\beta\,\beta^n\,\gamma)$	$=$	$\emptyset$
Weil	$\text{First}_k(\alpha\,\beta^{n+1}\,\gamma)$	$\subseteq$	$\text{First}_k(A\,\beta^{n+1}\,\gamma)$		
folgt:	$\text{First}_k(\alpha\,\beta^n\,\gamma)$	$\cap$	$\text{First}_k(\alpha\,\beta^{n+1}\,\gamma)$	$=$	$\emptyset$

Satz

Ist die Grammatik G reduziert und links-rekursiv, dann ist G nicht $LL(k)$ für jedes k .

Beweis:

Vereinfachung: $A \to A\,\beta \in P$

A erreichbar $\Longrightarrow$ $S \to_L^* u\,A\,\gamma \to_L^* u\,A\,\beta^n\,\gamma$ für jedes $n \geq 0$.

A produktiv $\Longrightarrow$ $\exists\, A \to \alpha \;:\; \alpha \neq A\,\beta$.

Annahme: G ist $LL(k)$;-) Dann gilt für alle $n \geq 0$:

$$\mathsf{First}_k(\alpha\,\beta^n\,\gamma) \cap \mathsf{First}_k(A\,\beta\,\beta^n\,\gamma) = \emptyset$$

Weil $\mathsf{First}_k(\alpha\,\beta^{n+1}\,\gamma) \subseteq \mathsf{First}_k(A\,\beta^{n+1}\,\gamma)$

folgt: $\mathsf{First}_k(\alpha\,\beta^n\,\gamma) \cap \mathsf{First}_k(\alpha\,\beta^{n+1}\,\gamma) = \emptyset$

Fall 1: $\beta \to^* \epsilon$ — Widerspruch !!!

Fall 2: $\beta \to^* w \neq \epsilon$ $\Longrightarrow$ $\mathsf{First}_k(\alpha\,\beta^k\,\gamma) \cap \mathsf{First}_k(\alpha\,\beta^{k+1}\,\gamma) \neq \emptyset$:-(

Bottom-up Parsing:

Wir rekonstruieren reverse Rechtsableitungen :-)

Dazu versuchen wir, für den Shift-Reduce-Parser $M_G^{(1)}$ die Reduktionsstellen zu identifizieren ...

Betrachte eine Berechnung dieses Kellerautomaten:

$$(q_0\,\alpha\,\gamma,\; v) \vdash (q_0\,\alpha\,B,\; v) \vdash^* (q_0\,S,\; \epsilon)$$

$\alpha\,\gamma$ nennen wir zuverlässiges Präfix für das vollständige Item $[B \to \gamma\bullet]$.

Dann ist $\alpha\,\gamma$ zuverlässig für $[B \to \gamma\bullet]$ gdw. $S \to_R^* \alpha\,B\,v$:-)