Script generated by TTT

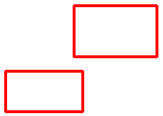
Title: Simon: Compilerbau (09.05.2012)

Date: Wed May 09 14:13:00 CEST 2012

Duration: 91:59 min

Pages: 45

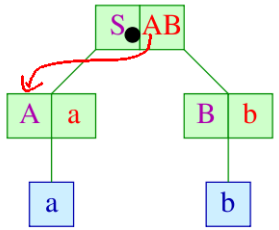
Kapitel 3:
Top-down Parsing



Item-Kellerautomat – Beispiel

Unser Beispiel:

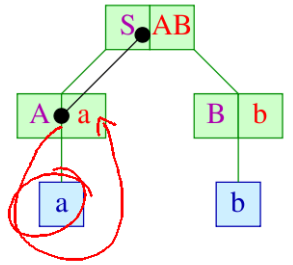
$S \to AB \quad A \to a \quad B \to b$



Item-Kellerautomat – Beispiel

Unser Beispiel:

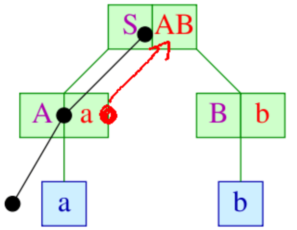
$S \to AB \quad A \to a \quad B \to b$



Item-Kellerautomat – Beispiel

Unser Beispiel:

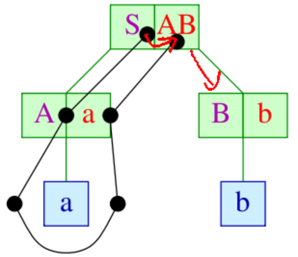
$$S \to A\,B \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

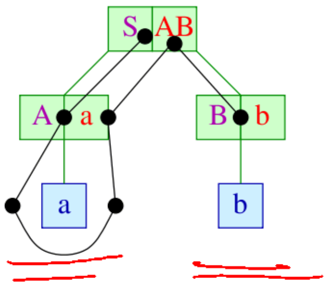
$$S \to A\,B \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

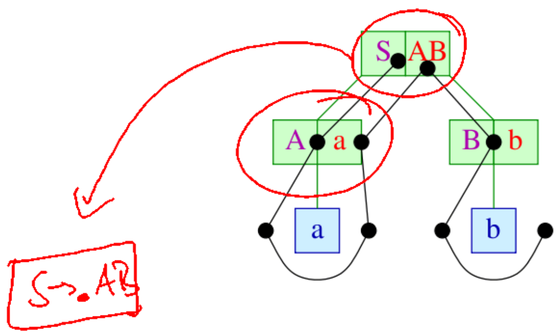
$$S \to A\,B \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

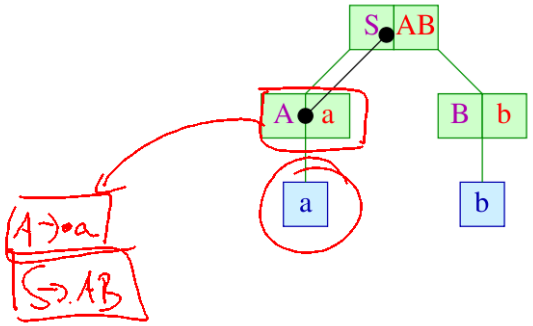
$$S \to A\,B \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

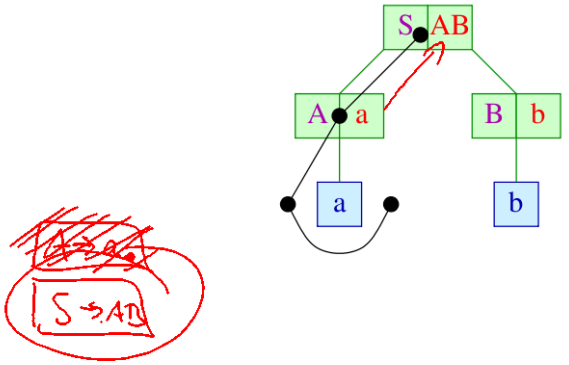
$$S \to AB \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

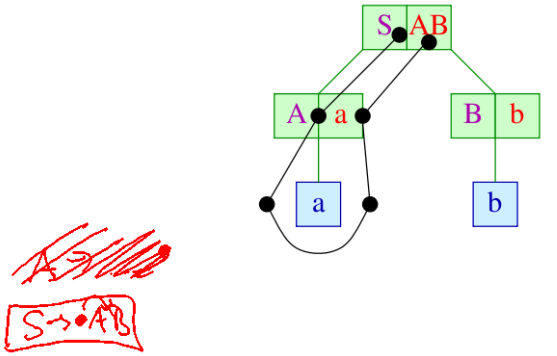
$$S \to AB \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

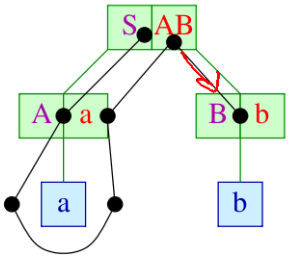
$$S \to AB \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

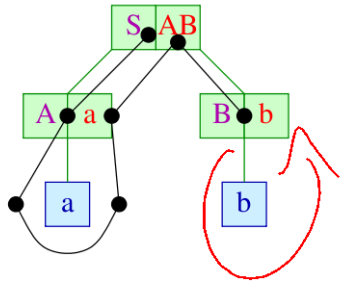
$$S \to AB \qquad A \to a \qquad B \to b$$



Item-Kellerautomat – Beispiel

Unser Beispiel:

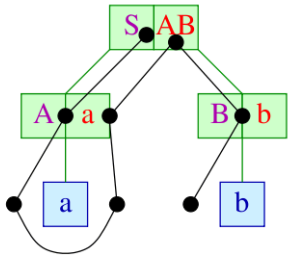
$S \to AB \quad A \to a \quad B \to b$



Item-Kellerautomat – Beispiel

Unser Beispiel:

$S \to AB \quad A \to a \quad B \to b$



86 / 160

Item-Kellerautomat – Beispiel

Wir fügen zur Initialisierung eine Regel: $S' \to S$ hinzu und konstruieren:

Anfangszustand: $[S' \to \bullet\, S]$
Endzustand: $[S' \to S\,\bullet]$
Zustandsübergangsrelation:

$[S' \to \bullet\, S]$	ϵ	$[S' \to \bullet\, S]\,[S \to \bullet\, A\,B]$
$[S \to \bullet\, A\,B]$	ϵ	$[S \to \bullet\, A\,B]\,[A \to \bullet\, a]$
$[A \to \bullet\, a]$	a	$[A \to a\,\bullet]$
$[S \to \bullet\, A\,B]\,[A \to a\,\bullet]$	ϵ	$[S \to A\,\bullet\, B]$
$[S \to A\,\bullet\, B]$	ϵ	$[S \to A\,\bullet\, B]\,[B \to \bullet\, b]$
$[B \to \bullet\, b]$	b	$[B \to b\,\bullet]$
$[S \to A\,\bullet\, B]\,[B \to b\,\bullet]$	ϵ	$[S \to A\,B\,\bullet]$
$[S' \to \bullet\, S]\,[S \to A\,B\,\bullet]$	ϵ	$[S' \to S\,\bullet]$

Item-Kellerautomat

Der Item-Kellerautomat M_G^L hat drei Arten von Übergängen:

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

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

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

Items der Form: $[A \to \alpha\,\bullet]$ heißen auch vollständig
Der Item-Kellerautomat schiebt den Punkt einmal um den Ableitungsbaum herum ...

Item-Kellerautomat

Diskussion:

- Die Expansionen einer Berechnung bilden eine Linksableitung
- Leider muss man bei den Expansionen nichtdeterministisch zwischen verschiedenen Regeln auswählen

- Zur Korrektheit der Konstruktion zeigt man, dass für jedes Item $[A \to \alpha \bullet B\,\beta]$ gilt:

$$([A \to \alpha \bullet B\,\beta],\, w) \vdash^* ([A \to \alpha\, B \bullet \beta],\, \epsilon) \quad \text{gdw.} \quad B \to^* w$$

- LL-Parsing basiert auf dem Item-Kellerautomaten und versucht, die Expansionen deterministisch zu machen ...

Item-Kellerautomat

Beispiel: $S \to \epsilon \mid a\,S\,b$

Die Übergänge des zugehörigen Item-Kellerautomat:

0	$[S' \to \bullet\, S]$	ϵ	$[S' \to \bullet\, S]\,[S \to \bullet]$
1	$[S' \to \bullet\, S]$	ϵ	$[S' \to \bullet\, S]\,[S \to \bullet\, a\,S\,b]$
2	$[S \to \bullet\, a\,S\,b]$	a	$[S \to a \bullet S\,b]$
3	$[S \to a \bullet S\,b]$	ϵ	$[S \to a \bullet S\,b]\,[S \to \bullet]$
4	$[S \to a \bullet S\,b]$	ϵ	$[S \to a \bullet S\,b]\,[S \to \bullet\, a\,S\,b]$
5	$[S \to a \bullet S\,b]\,[S \to \bullet]$	ϵ	$[S \to a\,S \bullet b]$
6	$[S \to a \bullet S\,b]\,[S \to a\,S\,b \bullet]$	ϵ	$[S \to a\,S \bullet b]$
7	$[S \to a\,S \bullet b]$	b	$[S \to a\,S\,b \bullet]$
8	$[S' \to \bullet\, S]\,[S \to \bullet]$	ϵ	$[S' \to S \bullet]$
9	$[S' \to \bullet\, S]\,[S \to a\,S\,b \bullet]$	ϵ	$[S' \to S \bullet]$

Konflikte gibt es zwischen den Übergängen $(0, 1)$ bzw. zwischen $(3, 4)$.

Topdown Parsing

Problem:

Konflikte zwischen Übergängen verhindern die Implementierung des Item-Kellerautomaten als deterministischer Kellerautomat.

Topdown Parsing

Problem:

Konflikte zwischen Übergängen verhindern die Implementierung des Item-Kellerautomaten als deterministischer Kellerautomat.

Idee 1: GLL Parsing

Bei jedem Konflikt wird virtuell der Stack komplett kopiert und parallel weiter geführt.

Idee 2: Recursive Descent & Backtracking

Tiefensuche nach einer passenden Ableitung.

Topdown Parsing

Problem:

Konflikte zwischen Übergängen verhindern die Implementierung des Item-Kellerautomaten als deterministischer Kellerautomat.

Idee 1: GLL Parsing

Bei jedem Konflikt wird virtuell der Stack komplett kopiert und parallel weiter geführt.

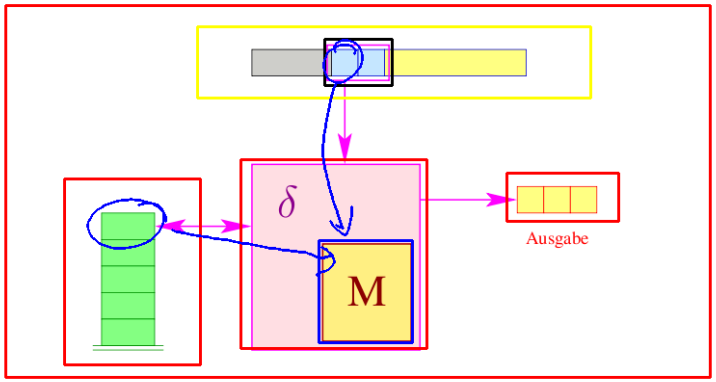
Idee 2: Recursive Descent & Backtracking

Tiefensuche nach einer passenden Ableitung.

Idee 3: Recursive Descent & Lookahead

Konflikte werden durch Betrachten des/der nächsten Zeichens gelöst.

Struktur des $LL(1)$-Parsers:



- Der Parser sieht ein Fenster der Länge 1 der Eingabe;
- er realisiert im Wesentlichen den Item-Kellerautomaten;
- die Tabelle $M[q, w]$ enthält die jeweils zu wählende Regel

Topdown Parsing

Idee:

- Benutze den Item-Kellerautomaten.
- Benutze das nächste Zeichen, um die Regeln für die Expansionen zu bestimmen
- Eine Grammatik heißt $LL(1)$, falls dies immer eindeutig möglich ist.

Topdown Parsing

Idee:

- Benutze den Item-Kellerautomaten.
- Benutze das nächste Zeichen, um die Regeln für die Expansionen zu bestimmen
- Eine Grammatik heißt $LL(1)$, falls dies immer eindeutig möglich ist.


Philip Lewis


Richard Stearns

Definition:

Eine reduzierte Grammatik heißt dann $LL(1)$, falls für je zwei verschiedene Regeln $A \to \alpha$, $A \to \alpha' \in P$ und jede Ableitung $S \to_L^* uA\beta$ mit $u \in T^*$ gilt:

$$\text{First}_1(\alpha\beta) \cap \text{First}_1(\alpha'\beta) = \emptyset$$

Topdown Parsing

Beispiel 1:

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

ist $LL(1)$, da $\text{First}_1(E) = \{\text{id}\}$

Topdown Parsing

Beispiel 1:

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

ist $LL(1)$, da $\text{First}_1(E) = \{\text{id}\}$

Beispiel 2:

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

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

Vorausschau-Mengen

Definition:

Für eine Menge $L \subseteq T^*$ definieren wir:

$$\text{First}_1(L) \;=\; \{\epsilon \mid \epsilon \in L\} \cup \{u \in T \mid \exists\, v \in T^* \;:\; uv \in L\}$$

Beispiel:

ϵ
ab
$aabb$
$aaabbb$
⋮

Vorausschau-Mengen

Definition:

Für eine Menge $L \subseteq T^*$ definieren wir:

$$\text{First}_1(L) \;=\; \{\epsilon \mid \epsilon \in L\} \cup \{u \in T \mid \exists\, v \in T^* \;:\; uv \in L\}$$

Beispiel:

ϵ
ab
$aabb$
$aaabbb$

die Präfixe der Länge 1

Vorausschau-Mengen

Rechenregeln:

$\text{First}_1(\_)$ ist verträglich mit Vereinigung und Konkatenation:

$$\begin{array}{lcl} \text{First}_1(\emptyset) & = & \emptyset \\ \text{First}_1(L_1 \cup L_2) & = & \text{First}_1(L_1) \cup \text{First}_1(L_2) \\ \text{First}_1(L_1 \cdot L_2) & = & \text{First}_1(\text{First}_1(L_1) \cdot \text{First}_1(L_2)) \\ & := & \text{First}_1(L_1) \odot \text{First}_1(L_2) \end{array}$$

1 – Konkatenation

Vorausschau-Mengen

Rechenregeln:

$\text{First}_1(\_)$ ist verträglich mit Vereinigung und Konkatenation:

$$\begin{array}{lcl} \text{First}_1(\emptyset) & = & \emptyset \\ \text{First}_1(L_1 \cup L_2) & = & \text{First}_1(L_1) \cup \text{First}_1(L_2) \\ \text{First}_1(L_1 \cdot L_2) & = & \text{First}_1(\text{First}_1(L_1) \cdot \text{First}_1(L_2)) \\ & := & \text{First}_1(L_1) \odot \text{First}_1(L_2) \end{array}$$

1 – Konkatenation

Beobachtung:

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

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

Sind alle Regeln von G produktiv, sind alle Mengen $\text{First}_1(A)$ nichtleer.

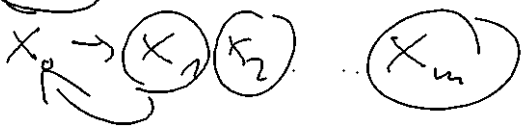
Vorausschau-Mengen

Für $\alpha \in (N \cup T)^*$ sind wir interessiert an der Menge:

$$\text{First}_1(\alpha) = \text{First}_1(\{w \in T^* \mid \alpha \to^* w\})$$

Idee: Behandle ϵ separat: F_ϵ

- Sei $\text{empty}(X) = \text{true}$ gdw. $X \to^* \epsilon$.
- $F_\epsilon(X_1 \ldots X_m) = \bigcup_{i=1}^{j} F_\epsilon(X_i)$ falls $\text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1})$



Vorausschau-Mengen

Für $\alpha \in (N \cup T)^*$ sind wir interessiert an der Menge:

$$\text{First}_1(\alpha) = \text{First}_1(\{w \in T^* \mid \alpha \to^* w\})$$

Idee: Behandle ϵ separat: F_ϵ

- Sei $\text{empty}(X) = \text{true}$ gdw. $X \to^* \epsilon$.
- $F_\epsilon(X_1 \ldots X_m) = \bigcup_{i=1}^{j} F_\epsilon(X_i)$ falls $\text{empty}(X_1) \wedge \ldots \wedge \text{empty}(X_{j-1})$

Definiere die ϵ-freien First_1-Mengen als Ungleichungssystem

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

Vorausschau-Mengen

im Beispiel...

$$\begin{array}{lcll} E & \to & E+T & \mid T \\ T & \to & T*F & \mid F \\ F & \to & (\,E\,) & \mid \text{name} \mid \text{int} \end{array}$$

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

Vorausschau-Mengen

im Beispiel...

$$\begin{array}{lcll} E & \to & E+T & \mid T \\ T & \to & T*F & \mid F \\ F & \to & (\,E\,) & \mid \text{name} \mid \text{int} \end{array}$$

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

... erhalten wir:

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

Schnelle Berechnung von Vorausschau-Mengen

Beobachtung:

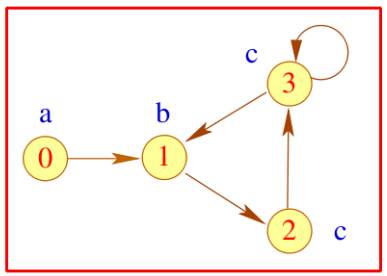
- Die Form der Ungleichungen dieser Ungleichungssysteme ist:

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

  für Variablen x, y und $d \in \mathrm{D}$.
- Solche Systeme heißen reine Vereinigungs-Probleme
- Diese Probleme können mit linearem Aufwand gelöst werden

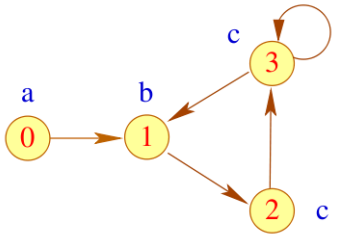
im Beispiel: $\mathrm{D} = 2^{\{a,b,c\}}$

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



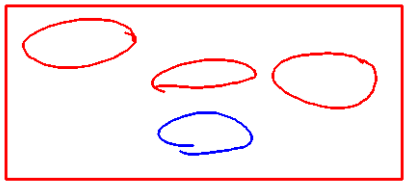
Schnelle Berechnung von Vorausschau-Mengen

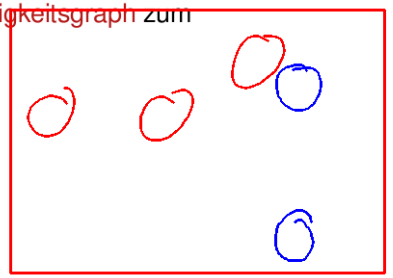
Frank DeRemer
& Tom Pennello



Vorgehen:

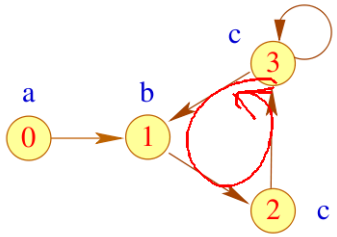
- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.





Schnelle Berechnung von Vorausschau-Mengen

Frank DeRemer
& Tom Pennello

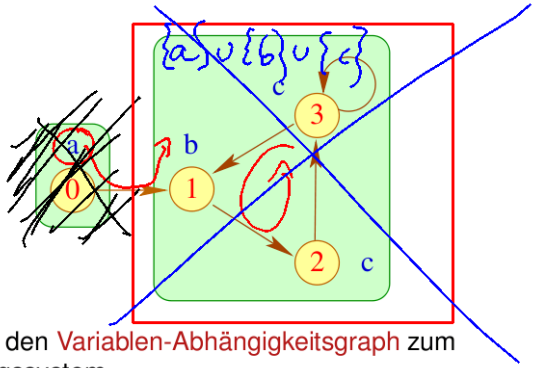


Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.

Schnelle Berechnung von Vorausschau-Mengen

Frank DeRemer
& Tom Pennello

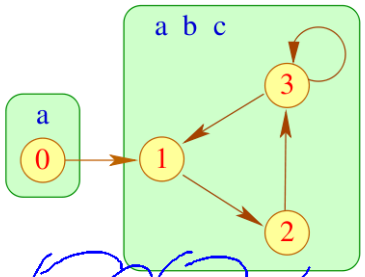


Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente ($\rightarrow$ Tarjan) 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

Schnelle Berechnung von Vorausschau-Mengen

Frank DeRemer
& Tom Pennello



Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente ($\rightarrow$ Tarjan) 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

Item-Kellerautomat als LL(1)-Parser

zurück zum Beispiel: $S \rightarrow \epsilon \mid a\,S\,b$

Die Übergänge des zugehörigen Item-Kellerautomat:

0	$[S' \rightarrow \bullet\, S]$	ϵ	$[S' \rightarrow \bullet\, S]\ [S \rightarrow \bullet]$
1	$[S' \rightarrow \bullet\, S]$	ϵ	$[S' \rightarrow \bullet\, S]\ [S \rightarrow \bullet\, a\,S\,b]$
2	$[S \rightarrow \bullet\, a\,S\,b]$	a	$[S \rightarrow a \bullet S\,b]$
3	$[S \rightarrow a \bullet S\,b]$	ϵ	$[S \rightarrow a \bullet S\,b]\ [S \rightarrow \bullet]$
4	$[S \rightarrow a \bullet S\,b]$	ϵ	$[S \rightarrow a \bullet S\,b]\ [S \rightarrow \bullet\, a\,S\,b]$
5	$[S \rightarrow a \bullet S\,b]\ [S \rightarrow \bullet]$	ϵ	$[S \rightarrow a\,S \bullet b]$
6	$[S \rightarrow a \bullet S\,b]\ [S \rightarrow a\,S\,b \bullet]$	ϵ	$[S \rightarrow a\,S \bullet b]$
7	$[S \rightarrow a\,S \bullet b]$	b	$[S \rightarrow a\,S\,b \bullet]$
8	$[S' \rightarrow \bullet\, S]\ [S \rightarrow \bullet]$	ϵ	$[S' \rightarrow S \bullet]$
9	$[S' \rightarrow \bullet\, S]\ [S \rightarrow a\,S\,b \bullet]$	ϵ	$[S' \rightarrow S \bullet]$

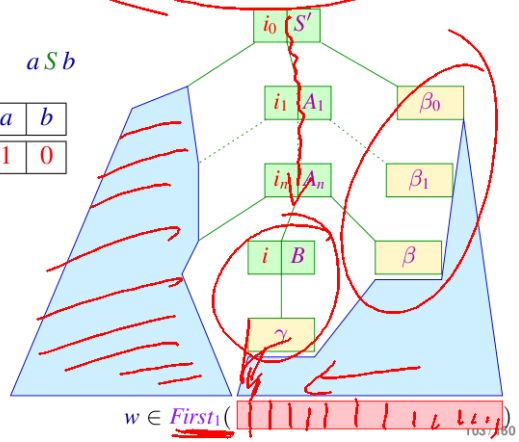
Konflikte gibt es zwischen den Übergängen $(0, 1)$ bzw. zwischen $(3, 4)$.

Item-Kellerautomat als LL(1)-Parser

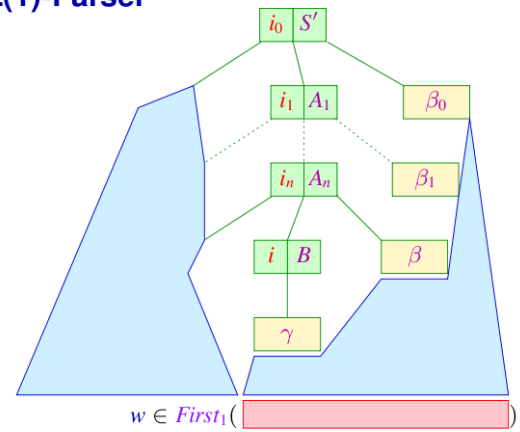
Ist G eine $LL(1)$-Grammatik, können wir eine Vorausschau-Tabelle mit Items und Nichtterminalen indizieren:
Wir setzen $M[B, w] = i$ genau dann wenn (B, i) die Regel $B \to \gamma$ ist und: $w \in \mathsf{First}_1(\gamma) \odot \bigcup\{\mathsf{First}_1(\beta) \mid S' \to_L^* u\,B\,\beta\}$.

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

	ϵ	a	b
S	0	1	0



Item-Kellerautomat als LL(1)-Parser



Ungleichungssystem für $\mathsf{Follow}_1(B) = \bigcup\{\mathsf{First}_1(\beta) \mid S' \to_L^* u\,B\,\beta\}$

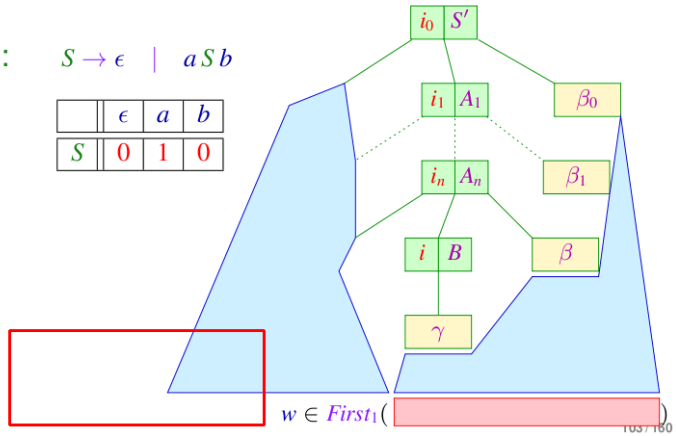
$$\begin{array}{lcll} \mathsf{Follow}_1(S) & \supseteq & \{\epsilon\} & \\ \mathsf{Follow}_1(B) & \supseteq & F_\epsilon(X_j) & \text{falls } A \to \alpha\,B\,X_1 \ldots X_m \in P, \\ & & & \mathsf{empty}(X_1) \wedge \ldots \wedge \mathsf{empty}(X_{j-1}) \\ \mathsf{Follow}_1(B) & \supseteq & \mathsf{Follow}_1(A)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

Item-Kellerautomat als LL(1)-Parser

Ist G eine $LL(1)$-Grammatik, können wir eine Vorausschau-Tabelle mit Items und Nichtterminalen indizieren:
Wir setzen $M[B, w] = i$ genau dann wenn (B, i) die Regel $B \to \gamma$ ist und: $w \in \mathsf{First}_1(\gamma) \odot \bigcup\{\mathsf{First}_1(\beta) \mid S' \to_L^* u\,B\,\beta\}$.

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

	ϵ	a	b
S	0	1	0



Topdown-Parsing

Diskussion

- Einer praktischen Implementation von $LL(1)$-Parsern über recursive Descent steht nichts im Wege
- Jedoch kann nur eine Teilmenge der deterministisch kontextfreien Sprachen auf diesem Weg gelesen werden.

Diskussion

- Einer praktischen Implementation von $LL(1)$-Parsern über recursive Descent steht nichts im Wege
- Jedoch kann nur eine Teilmenge der deterministisch kontextfreien Sprachen auf diesem Weg gelesen werden.
- Ausweg: Schritt von $LL(1)$ auf $LL(k)$
- Die Größe der auftretenden Mengen steigt mit k rapide
- Leider reichen auch $LL(k)$ Parser nicht aus, um alle deterministisch kontextfreien Sprachen zu erkennen.
- In praktischen Systemen wird darum meist nur der Fall $k = 1$ implementiert ...

Kapitel 4:
Bottom-up Analyse