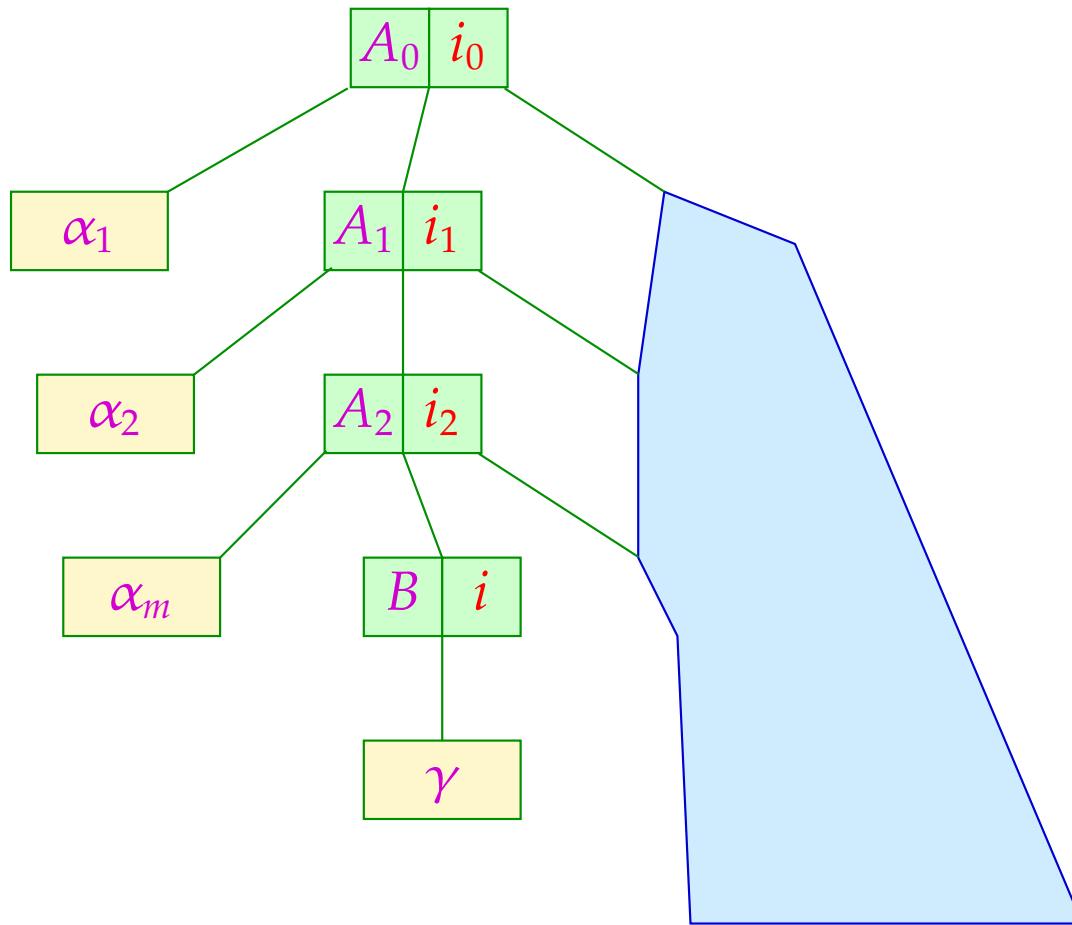


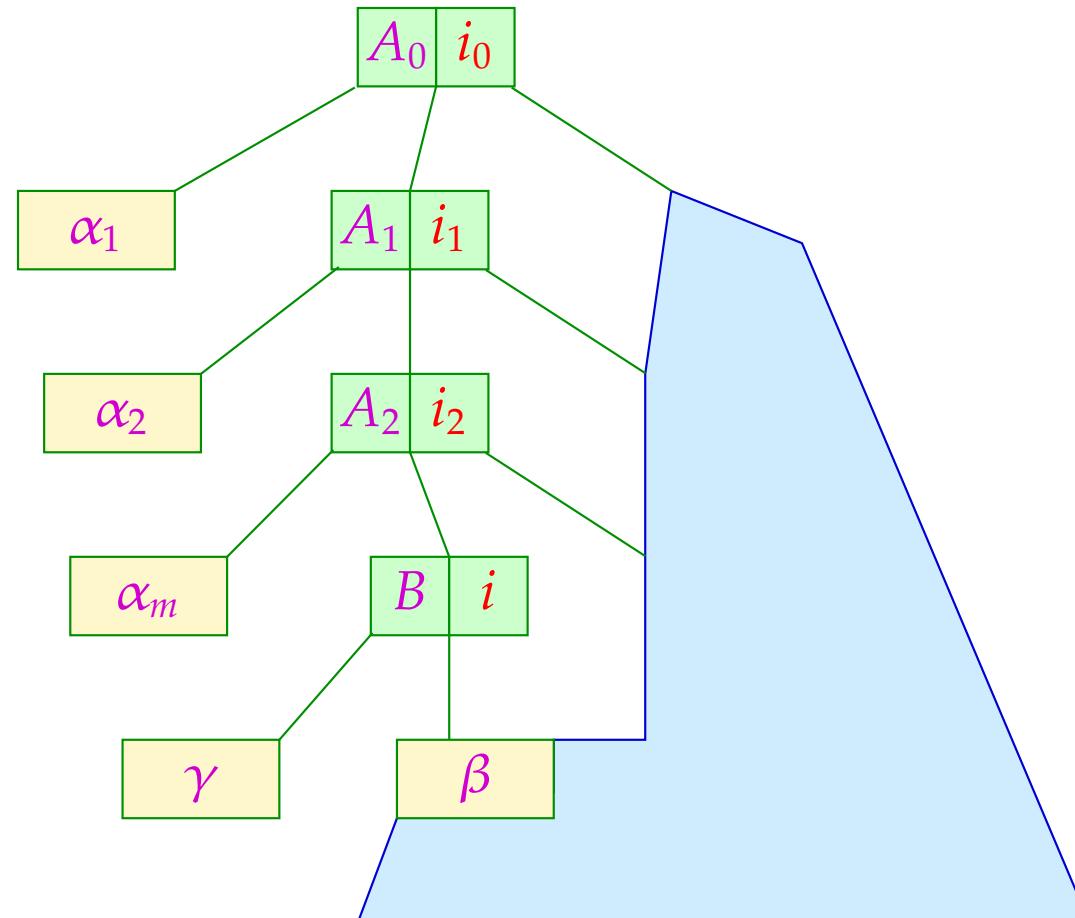
... wobei $\alpha = \alpha_1 \dots \alpha_m \text{ :-)}$



... wobei $\alpha = \alpha_1 \dots \alpha_m \quad :-)$

Umgekehrt können wir zu jedem möglichen Wort α' die Menge aller möglicherweise später passenden Regeln ermitteln ...

Das Item $[B \rightarrow \gamma \bullet \beta]$ heißt **gültig** für α' gdw. $S \xrightarrow{R}^* \alpha B \nu$ mit $\alpha' = \alpha \gamma$:



... wobei $\alpha = \alpha_1 \dots \alpha_m$:-)

Beobachtung:

Die Menge der zuverlässigen Präfixe aus $(N \cup T)^*$ für (vollständige) Items kann mithilfe eines endlichen Automaten berechnet werden :-)

Zustände: Items :-)

Anfangszustand: $[S' \rightarrow \bullet S]$

Endzustände: $\{[B \rightarrow \gamma \bullet] \mid B \rightarrow \gamma \in P\}$

Übergänge:

(1) $([A \rightarrow \alpha \bullet X \beta], X, [A \rightarrow \alpha X \bullet \beta]), \quad X \in (N \cup T), A \rightarrow \alpha X \beta \in P;$

(2) $([A \rightarrow \alpha \bullet B \beta], \epsilon, [B \rightarrow \bullet \gamma]), \quad A \rightarrow \alpha B \beta, B \rightarrow \gamma \in P;$

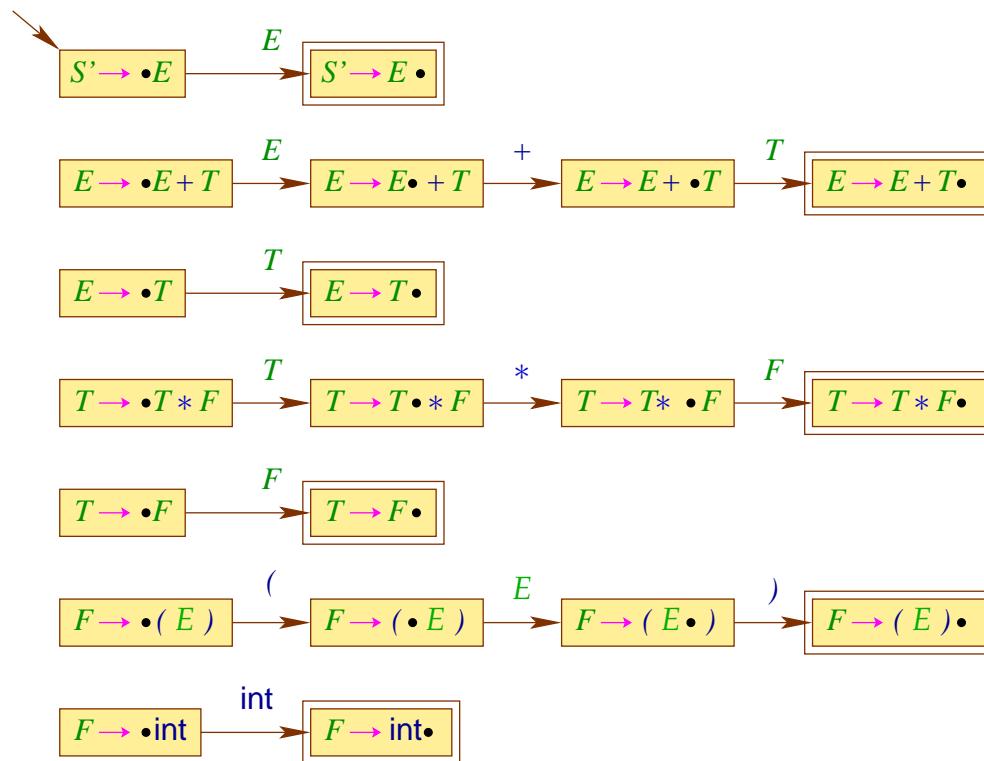
Den Automaten $c(G)$ nennen wir **charakteristischen Automaten** für G .

Beispiel:

$$E \rightarrow E + T \quad | \quad T$$

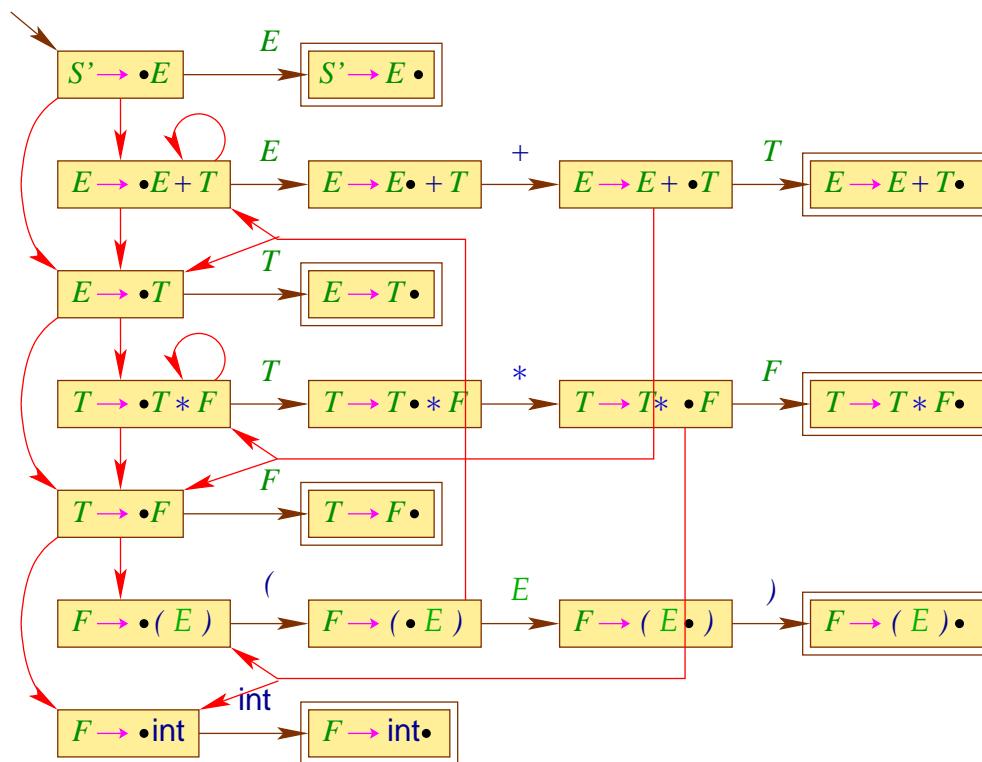
$$T \rightarrow T * F \quad | \quad F$$

$$F \rightarrow (E) \quad | \quad \text{int}$$



Beispiel:

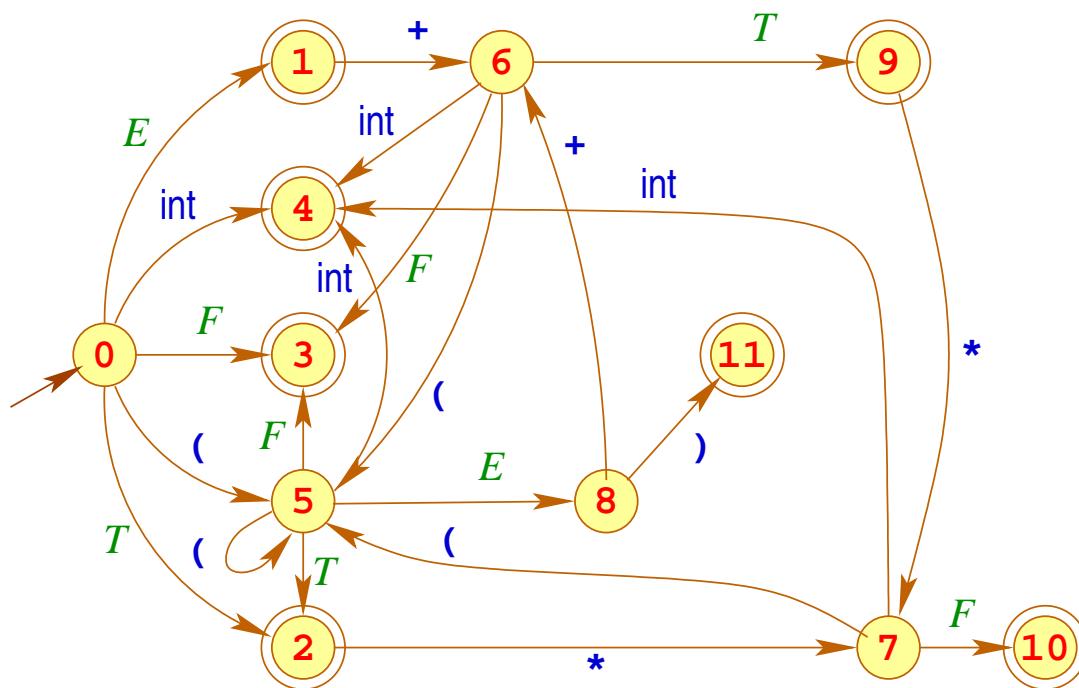
$$\begin{array}{lcl}
 E & \rightarrow & E + T \quad \mid \quad T \\
 T & \rightarrow & T * F \quad \mid \quad F \\
 F & \rightarrow & (E) \quad \mid \quad \text{int}
 \end{array}$$



Den kanonischen $LR(0)$ -Automaten $LR(G)$ erhalten wir aus $c(G)$, indem wir:

- (1) nach jedem lesenden Übergang beliebig viele ϵ -Übergänge einschieben (unsere Konstruktion 1 zur Beseitigung von ϵ -Übergängen :-)
- (2) die Teilmengenkonstruktion anwenden.

... im Beispiel:



Dazu konstruieren wir:

$$\begin{aligned}
 q_0 &= \{ [S' \rightarrow \bullet E], & q_1 &= \delta(q_0, E) = \{ [S' \rightarrow E \bullet], \\
 & [E \rightarrow \bullet E + T], & & [E \rightarrow E \bullet + T] \} \\
 & [E \rightarrow \bullet T], \\
 & [T \rightarrow \bullet T * F] \} & q_2 &= \delta(q_0, T) = \{ [E \rightarrow T \bullet], \\
 & [T \rightarrow \bullet F], & & [T \rightarrow T \bullet * F] \} \\
 & [F \rightarrow \bullet (E)], \\
 & [F \rightarrow \bullet \text{int}] \} & q_3 &= \delta(q_0, F) = \{ [T \rightarrow F \bullet] \} \\
 \\
 & & q_4 &= \delta(q_0, \text{int}) = \{ [F \rightarrow \text{int} \bullet] \}
 \end{aligned}$$

$$q_5 = \delta(q_0, ()) = \{ [F \rightarrow (\bullet E)], [E \rightarrow \bullet E + T], [E \rightarrow \bullet T], [T \rightarrow \bullet T * F], [T \rightarrow \bullet F], [F \rightarrow \bullet (E)], [F \rightarrow \bullet \text{int}] \}$$

$$q_6 = \delta(q_1, +) = \{ [E \rightarrow E + \bullet T], [T \rightarrow \bullet T * F], [T \rightarrow \bullet F], [F \rightarrow \bullet (E)], [F \rightarrow \bullet \text{int}] \}$$

$$q_7 = \delta(q_2, *) = \{ [T \rightarrow T * \bullet F], [F \rightarrow \bullet (E)], [F \rightarrow \bullet \text{int}] \}$$

$$q_8 = \delta(q_5, E) = \{ [F \rightarrow (E \bullet)] \}$$

$$q_9 = \delta(q_6, T) = \{ [E \rightarrow E + T \bullet], [T \rightarrow T \bullet * F] \}$$

$$q_{10} = \delta(q_7, F) = \{ [T \rightarrow T * F \bullet] \}$$

$$q_{11} = \delta(q_8, ()) = \{ [F \rightarrow (E \bullet)] \}$$

Beachte:

Der kanonische $LR(0)$ -Automat kann auch direkt aus der Grammatik konstruiert werden :-)

Man benötigt die Hilfsfunktion:

$$\delta_{\epsilon}^*(q) = q \cup \{ [B \rightarrow \bullet \gamma] \mid \exists [A \rightarrow \alpha \bullet B' \beta'] \in q, \beta \in (N \cup T)^* : B' \rightarrow^* B \beta \}$$

Zustände: Mengen von Items;

Anfangszustand: $\delta_{\epsilon}^* \{ [S' \rightarrow \bullet S] \}$

Endzustände: $\{ q \mid \exists A \rightarrow \alpha \in P : [A \rightarrow \alpha \bullet] \in q \}$

Übergänge: $\delta(q, X) = \delta_{\epsilon}^* \{ [A \rightarrow \alpha X \bullet \beta] \mid [A \rightarrow \alpha \bullet X \beta] \in q \}$

Idee zu einem Parser:

- Der Parser verwaltet ein zuverlässiges Präfix $\alpha = X_1 \dots X_m$ auf dem Keller und benutzt $LR(G)$, um Reduktionsstellen zu entdecken.
- Er kann mit einer Regel $A \rightarrow \gamma$ reduzieren, falls $[A \rightarrow \gamma \bullet]$ für α gültig ist :-)
- Damit der Automat nicht immer wieder neu über den Kellerinhalt laufen muss, kellern wir anstelle der X_i jeweils die **Zustände !!!**

Achtung:

Dieser Parser ist nur dann **deterministisch**, wenn jeder Endzustand des kanonischen $LR(0)$ -Automaten keine **Konflikte** enthält ...

... im Beispiel:

$$q_1 = \{ [S' \rightarrow E \bullet], \\ [E \rightarrow E \bullet + T] \}$$

$$q_2 = \{ [E \rightarrow T \bullet], \quad q_9 = \{ [E \rightarrow E + T \bullet], \\ [T \rightarrow T \bullet * F] \} \quad [T \rightarrow T \bullet * F] \}$$

$$q_3 = \{ [T \rightarrow F \bullet] \} \quad q_{10} = \{ [T \rightarrow T * F \bullet] \}$$

$$q_4 = \{ [F \rightarrow \text{int} \bullet] \} \quad q_{11} = \{ [F \rightarrow (E) \bullet] \}$$

Die Endzustände q_1, q_2, q_9 enthalten mehr als ein Item :-

Aber wir haben ja auch noch nicht Vorausschau eingesetzt :-)

Die Konstruktion des $LR(0)$ -Parsers:

Zustände: $Q \cup \{f\}$ (f neu :-)

Anfangszustand: q_0

Endzustand: f

Übergänge:

Shift: (p, a, pq) falls $q = \delta(p, a) \neq \emptyset$

Reduce: $(pq_1 \dots q_m, \epsilon, pq)$ falls $[A \rightarrow X_1 \dots X_m \bullet] \in q_m,$
 $q = \delta(p, A)$

Finish: $(q_0 p, \epsilon, f)$ falls $[S' \rightarrow S \bullet] \in p$

wobei $LR(G) = (Q, T, \delta, q_0, F)$.

Zur Korrektheit:

Man zeigt:

Die akzeptierenden Berechnungen des $LR(0)$ -Parsers stehen in eins-zu-eins Beziehung zu denen des Shift-Reduce-Parsers $M_G^{(1)}$.

Wir folgern:

- ===== Die akzeptierte Sprache ist genau $\mathcal{L}(G) \text{ :-)}$
- ===== Die Folge der Reduktionen einer akzeptierenden Berechnung für ein Wort $w \in T$ liefert eine **reverse Rechts-Ableitung** von G für w :-)

Leider ist der $LR(0)$ -Parser i.a. nicht-deterministisch :-)

Wir identifizieren zwei Gründe:

Reduce-Reduce-Konflikt:

$$[A \rightarrow \gamma \bullet], [A' \rightarrow \gamma' \bullet] \in q \quad \text{mit} \quad A \neq A' \vee \gamma \neq \gamma'$$

Shift-Reduce-Konflikt:

$$[A \rightarrow \gamma \bullet], [A' \rightarrow \alpha \bullet a \beta] \in q \quad \text{mit} \quad a \in T$$

für einen Zustand $q \in Q$.

Solche Zustände nennen wir **ungeeignet**.

Idee:

Benutze k -Vorausschau, um Konflikte zu lösen.

Wir definieren:

Die reduzierte kontextfreie Grammatik G heißt $LR(k)$ -Grammatik, falls für $\text{First}_k(w) = \text{First}_k(x)$ aus:

$$\left. \begin{array}{llll} S & \xrightarrow{R}^* & \alpha A w & \rightarrow \quad \alpha \beta w \\ S & \xrightarrow{R}^* & \alpha' A' w' & \rightarrow \quad \alpha \beta x \end{array} \right\} \quad \text{folgt:} \quad \alpha = \alpha' \wedge A = A' \wedge w' = x$$

Beispiele:

$$(1) \quad S \rightarrow A \mid B \quad A \rightarrow a \underline{A} b \mid 0 \quad B \rightarrow a \underline{B} b b \mid 1$$

... ist nicht $LL(k)$ für jedes k — aber $LR(0)$:

Sei $S \xrightarrow{R}^* \alpha \underline{X} w \rightarrow \alpha \beta \underline{w}$. Dann ist $\alpha \beta$ von einer der Formen:

$$\underline{A}, \underline{B}, a^n \underline{a} \underline{A} b, a^n \underline{a} \underline{B} b b, a^n \underline{0}, a^n \underline{1} \quad (n \geq 0)$$

Beispiele:

$$(1) \quad S \rightarrow A \mid B \quad A \rightarrow a \underline{A} b \mid 0 \quad B \rightarrow a \underline{B} b b \mid 1$$

... ist nicht $LL(k)$ für jedes k — aber $LR(0)$:

Sei $S \xrightarrow{R}^* \alpha \underline{X} w \rightarrow \alpha \beta w$. Dann ist $\alpha \underline{\beta}$ von einer der Formen:

$$\underline{A}, \underline{B}, a^n \underline{a} \underline{A} b, a^n \underline{a} \underline{B} b b, a^n \underline{0}, a^n \underline{1} \quad (n \geq 0)$$

$$(2) \quad S \rightarrow a \underline{A} c \quad A \rightarrow \underline{A} b b \mid b$$

... ist ebenfalls $LR(0)$:

Sei $S \xrightarrow{R}^* \alpha \underline{X} w \rightarrow \alpha \beta w$. Dann ist $\alpha \underline{\beta}$ von einer der Formen:

$$\underline{a} \underline{b}, \underline{a} \underline{A} b b, \underline{a} \underline{A} c$$

(3) $S \rightarrow a A c \quad A \rightarrow b b A \mid b$... ist nicht $LR(0)$, aber $LR(1)$:

Für $S \rightarrow_R^* \alpha X w \rightarrow \alpha \beta w$ mit $\{y\} = \text{First}_k(w)$ ist $\alpha \underline{\beta} y$ von einer der Formen:

$a b^{2n} \underline{b} c$, $a b^{2n} \underline{b} \underline{b} A c$, $\underline{a} A c$

(3) $S \rightarrow a A c \quad A \rightarrow b b A \mid b$... ist nicht $LR(0)$, aber $LR(1)$:

Für $S \rightarrow_R^* \alpha X w \rightarrow \alpha \beta w$ mit $\{y\} = \text{First}_k(w)$ ist $\alpha \underline{\beta} y$ von einer der Formen:

$$a b^{2n} \underline{b} c, a b^{2n} \underline{b} \underline{b} A c, \underline{a} A c$$

(4) $S \rightarrow a A c \quad A \rightarrow b A b \mid b$... ist nicht $LR(k)$ für jedes $k \geq 0$:

Betrachte einfach die Rechtsableitungen:

$$S \rightarrow_R^* a b^n A b^n c \rightarrow a b^n \underline{b} b^n c$$

In der Tat gilt:

Satz:

Die reduzierte Grammatik G ist genau dann $LR(0)$ wenn der kanonische $LR(0)$ -Automat $LR(G)$ keine ungeeigneten Zustände enthält.

In der Tat gilt:

Satz:

Die reduzierte Grammatik G ist genau dann $LR(0)$ wenn der kanonische $LR(0)$ -Automat $LR(G)$ keine ungeeigneten Zustände enthält.

Beweis:

Enthalte G einen ungeeigneten Zustand q .

Fall 1: $[A \rightarrow \gamma \bullet], [A' \rightarrow \gamma' \bullet] \in q$ mit $A \rightarrow \gamma \neq A' \rightarrow \gamma'$

Dann gibt es ein zuverlässiges Präfix $\alpha \gamma = \alpha' \gamma'$ mit

$$\begin{aligned} S \xrightarrow{R}^* \alpha A w \rightarrow \alpha \gamma w \quad \wedge \quad S \xrightarrow{R}^* \alpha' A' x \rightarrow \alpha' \gamma' x \\ \implies G \text{ ist nicht } LR(0) \end{aligned}$$

Fall 2: $[A \rightarrow \gamma \bullet], [A' \rightarrow \beta \bullet a \beta'] \in q$

Dann gibt es ein zuverlässiges Präfix $\alpha \gamma = \alpha' \beta$ mit

$$S \xrightarrow{R}^* \alpha A w \rightarrow \alpha \gamma w \quad \wedge \quad S \xrightarrow{R}^* \alpha' A' x \rightarrow \alpha' \beta a \beta' x$$

Ist $\beta' \in T^*$, dann ist G nicht $LR(0)$:-)

Andernfalls $\beta' \xrightarrow{R}^* v_1 X v_2 \rightarrow v_1 u v_2$. Damit erhalten wir:

$$S \xrightarrow{R}^* \alpha' \beta a v_1 X v_2 x \rightarrow \alpha' \beta a v_1 u v_2 x$$

$\implies G$ ist nicht $LR(0)$:-)

Fall 2: $[A \rightarrow \gamma \bullet], [A' \rightarrow \beta \bullet a \beta'] \in q$

Dann gibt es ein zuverlässiges Präfix $\alpha \gamma = \alpha' \beta$ mit

$$S \xrightarrow{R}^* \alpha A w \rightarrow \alpha \gamma w \quad \wedge \quad S \xrightarrow{R}^* \alpha' A' x \rightarrow \alpha' \beta a \beta' x$$

Ist $\beta' \in T^*$, dann ist G nicht $LR(0)$:-)

Andernfalls $\beta' \xrightarrow{R}^* v_1 X v_2 \rightarrow v_1 u v_2$. Damit erhalten wir:

$$S \xrightarrow{R}^* \alpha' \beta a v_1 X v_2 x \rightarrow \alpha' \beta a v_1 u v_2 x$$

$\implies G$ ist nicht $LR(0)$:-)

Enthalte $LR(G)$ keine ungeeigneten Zustände. Betrachte:

$$S \xrightarrow{R}^* \alpha A w \rightarrow \alpha \gamma w \quad S \xrightarrow{R}^* \alpha' A' w' \rightarrow \alpha' \gamma' x$$

Sei $\delta(q_0, \alpha \gamma) = q$. Insbesondere ist $[A \rightarrow \gamma \bullet] \in q$.

Annahme: $(\alpha, A, w') \neq (\alpha', A', x)$.

Fall 1: $w' = x$. Dann muss q $[A' \rightarrow \gamma' \bullet]$ enthalten :-)

Fall 2: $w' \neq x$. Weitere Fallunterscheidung :-))