Allgemein:

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

Diskussion:

- Man überzeugt sich, dass die kleinste Lösung dieses Ungleichungssystems tatsächlich die Mengen $\mathsf{Follow}_k(B)$ 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:

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

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

Idee:

- Behandle ϵ separat!

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

- Definiere die ϵ-freien First_1-Mengen

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

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

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

... im Beispiel:

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

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

Deshalb erhalten wir:

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

Entsprechend konstruieren wir zur Berechnung von Follow_1 :

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

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:

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

Entsprechend konstruieren wir zur Berechnung von Follow_1 :

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

... im Beispiel:

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

... erhalten wir:

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

Diskussion:

- Diese Ungleichungssysteme bestehen aus Ungleichungen der Form:

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

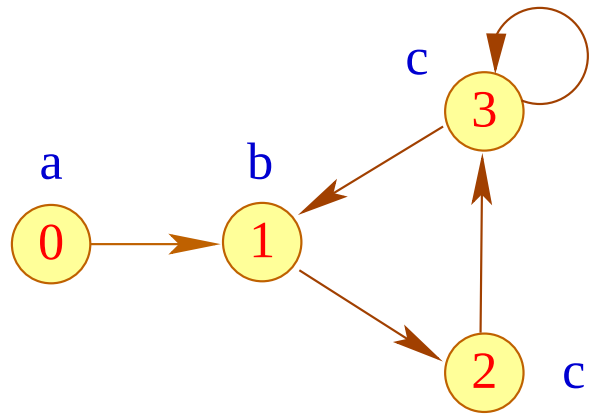
  für Variablen x, y und $d \in \mathbb{D}$.

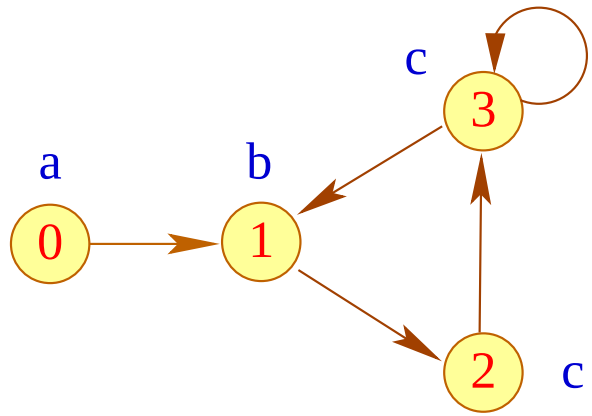
- Solche Ungleichungssysteme heißen reine Vereinigungs-Probleme :-)
- Diese Probleme können mit linearem Aufwand gelöst werden ...

Beispiel:

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

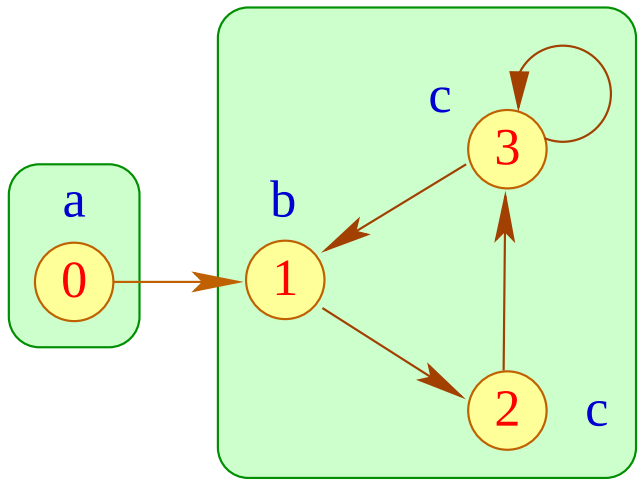
$$\begin{array}{lll}
x_0 \supseteq \{a\} & & \\
x_1 \supseteq \{b\} & x_1 \supseteq x_0 & x_1 \supseteq x_3 \\
x_2 \supseteq \{c\} & x_2 \supseteq x_1 & \\
x_3 \supseteq \{c\} & x_3 \supseteq x_2 & x_3 \supseteq x_3
\end{array}$$





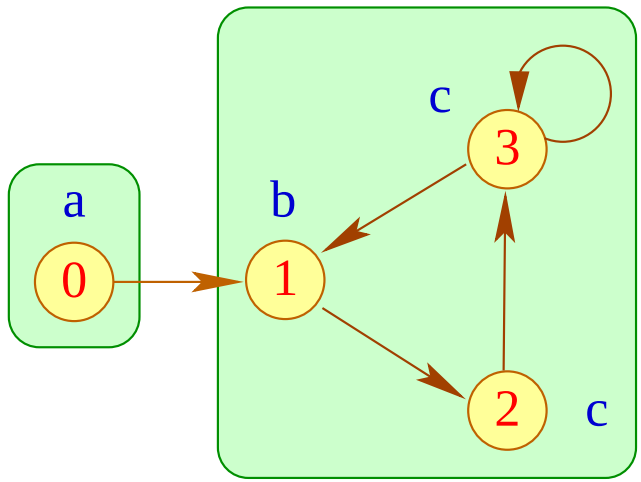
Vorgehen:

- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.



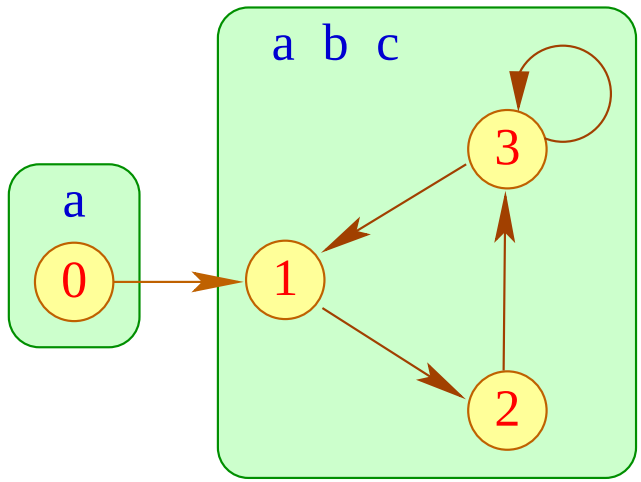
Vorgehen:

- 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 :-)

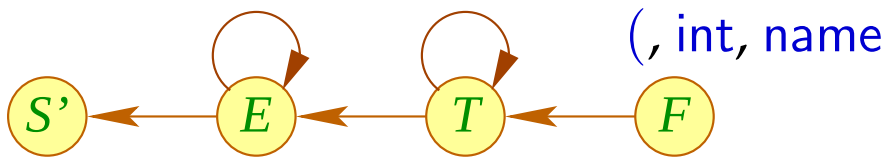


Vorgehen:

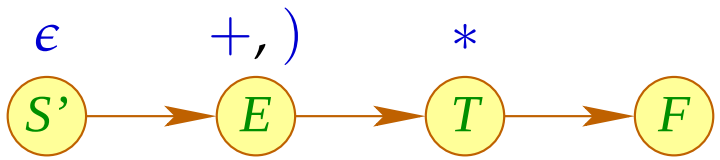
- Konstruiere den Variablen-Abhängigkeitsgraph zum Ungleichungssystem.
- Innerhalb einer starken Zusammenhangskomponente haben alle Variablen den gleichen Wert :-)
- Hat eine SZK keine eingehenden Kanten, erhält man ihren Wert, indem man die kleinste obere Schranke aller Werte in der SZK berechnet :-)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

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

... 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 $n \geq 0$:

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

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

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

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 :-)

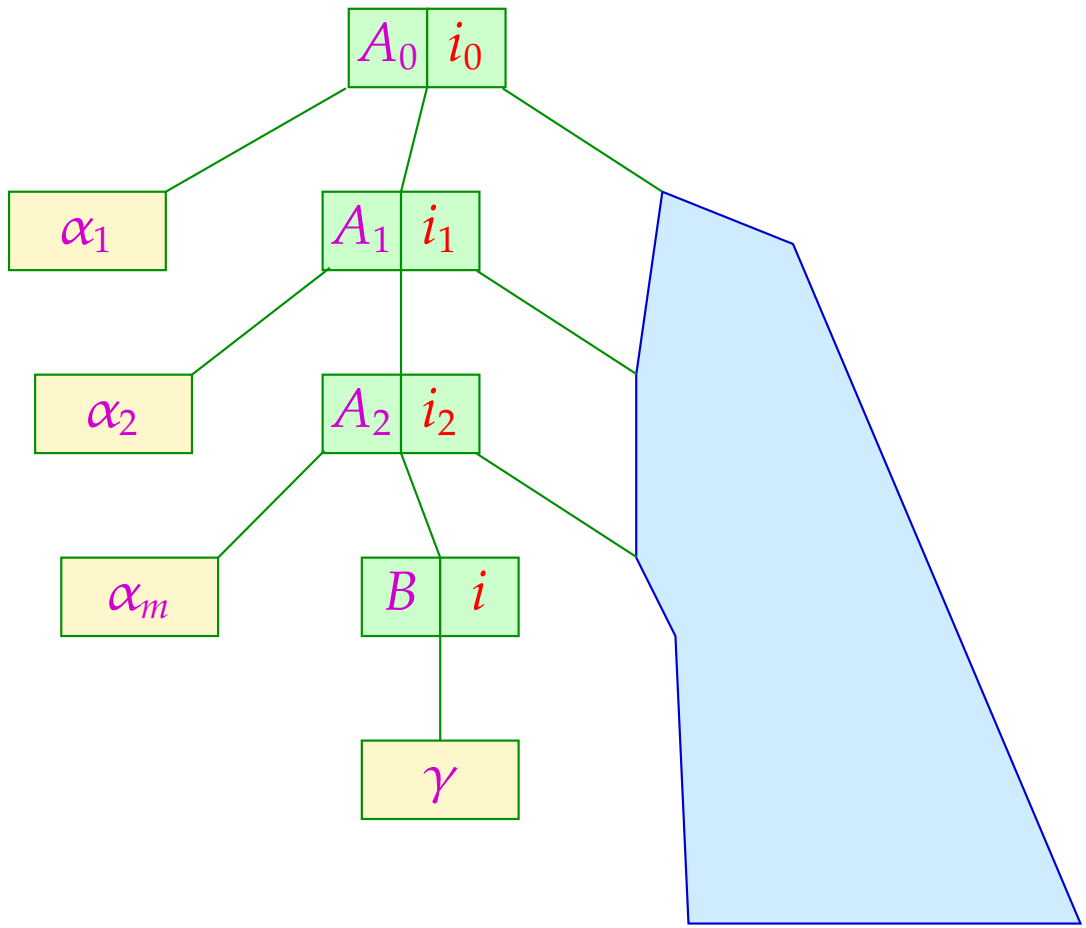
Dazu versuchen wir, für den Shift-Reduce-Parser $M_G^{(1)}$ die Reduktionsstellen zu identifizieren ...

Betrachte eine Berechnung dieses Kellerautomaten:

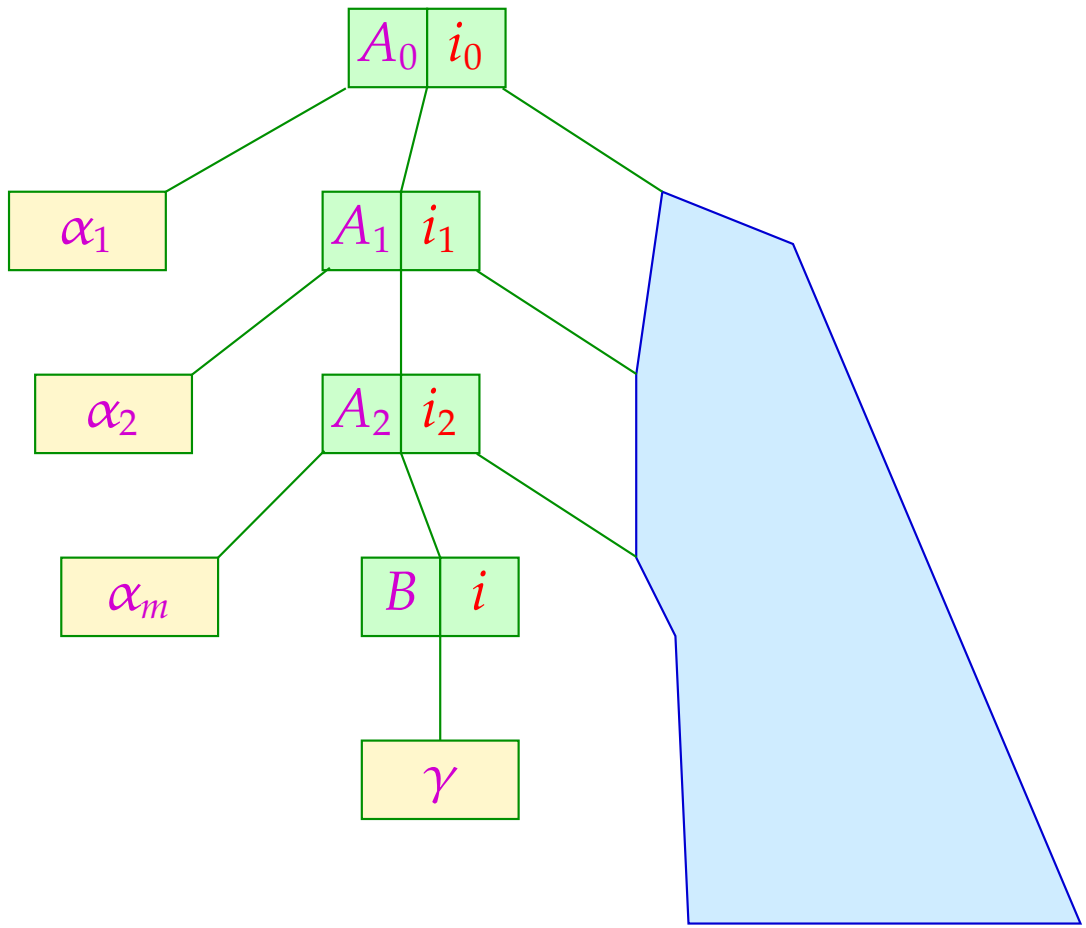
$$(q_0\, \alpha\, \gamma,\ v) \vdash (q_0\, \alpha\, B,\ v) \vdash^* (q_0\, S,\ \epsilon)$$

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

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



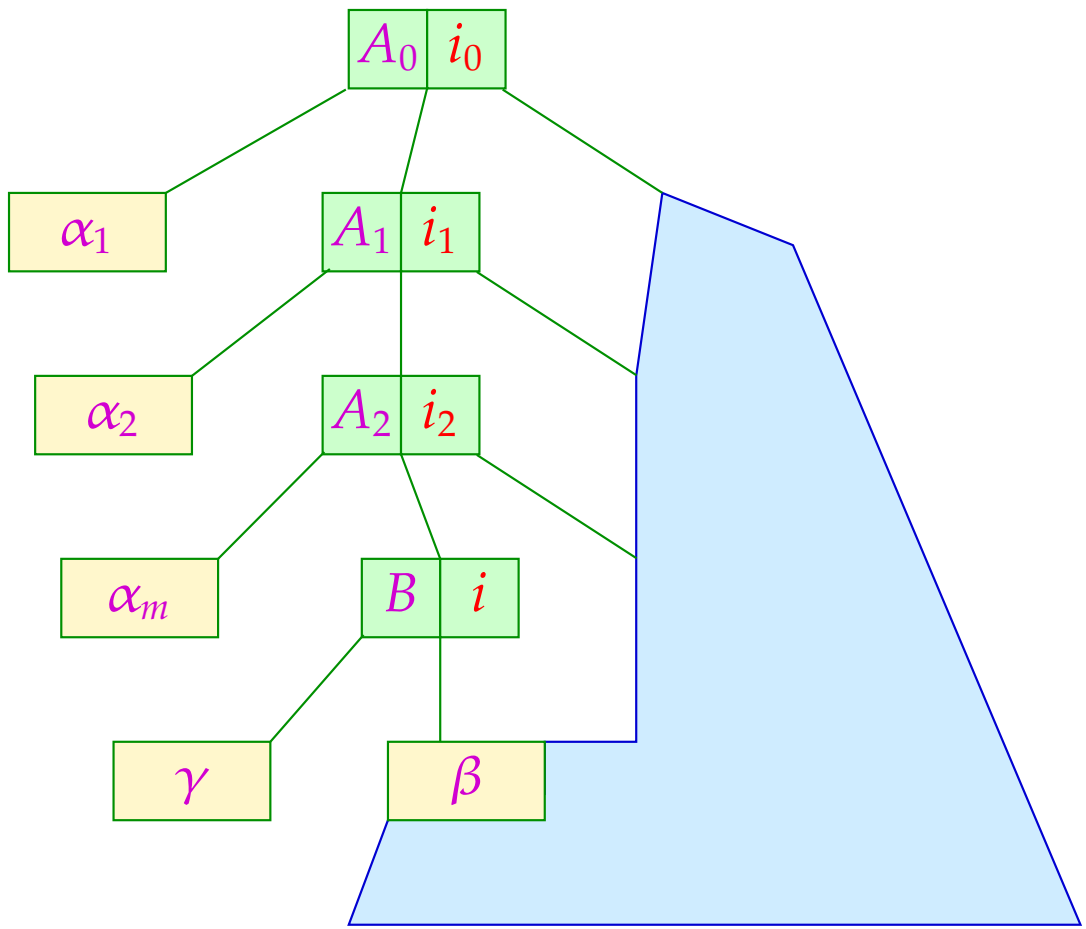
... wobei $\alpha = \alpha_1 \ldots \alpha_m$:-)



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

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

Das Item $[B \to \gamma \bullet \beta]$ heißt gültig für α' gdw. $S \to_R^* \alpha\, B\, v$ mit $\alpha' = \alpha\,\gamma$:



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

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:

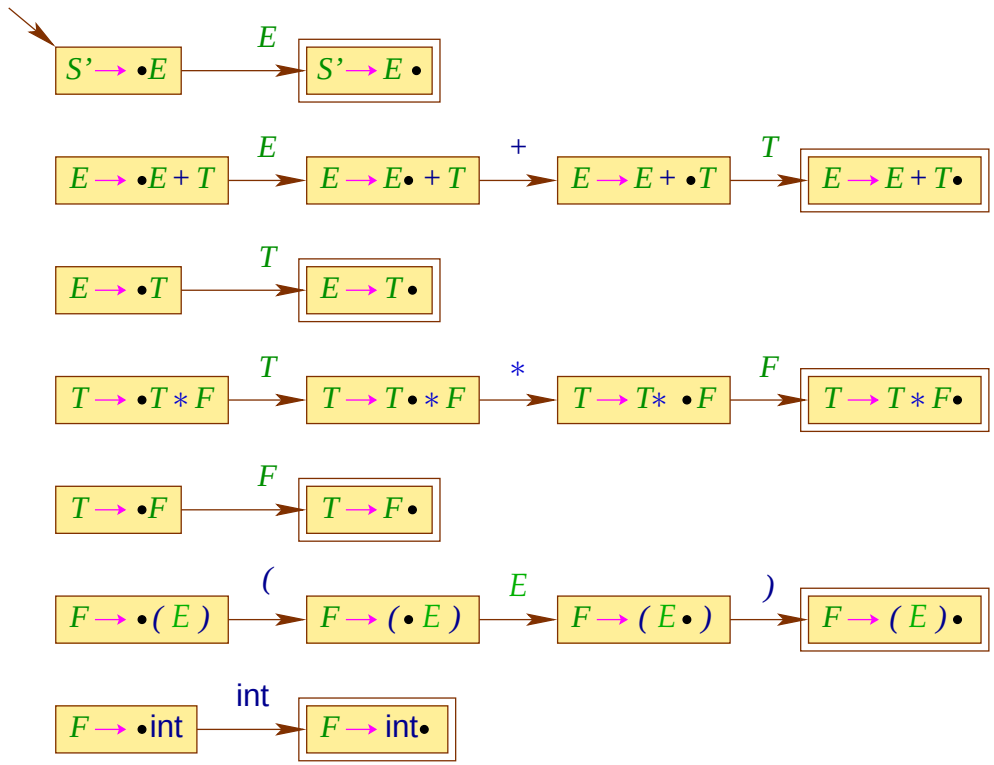
(1) $([A \rightarrow \alpha \bullet X\, \beta], X, [A \rightarrow \alpha\, X \bullet \beta]), \quad X \in (N \cup T), A \rightarrow \alpha\, X\, \beta \in P;$

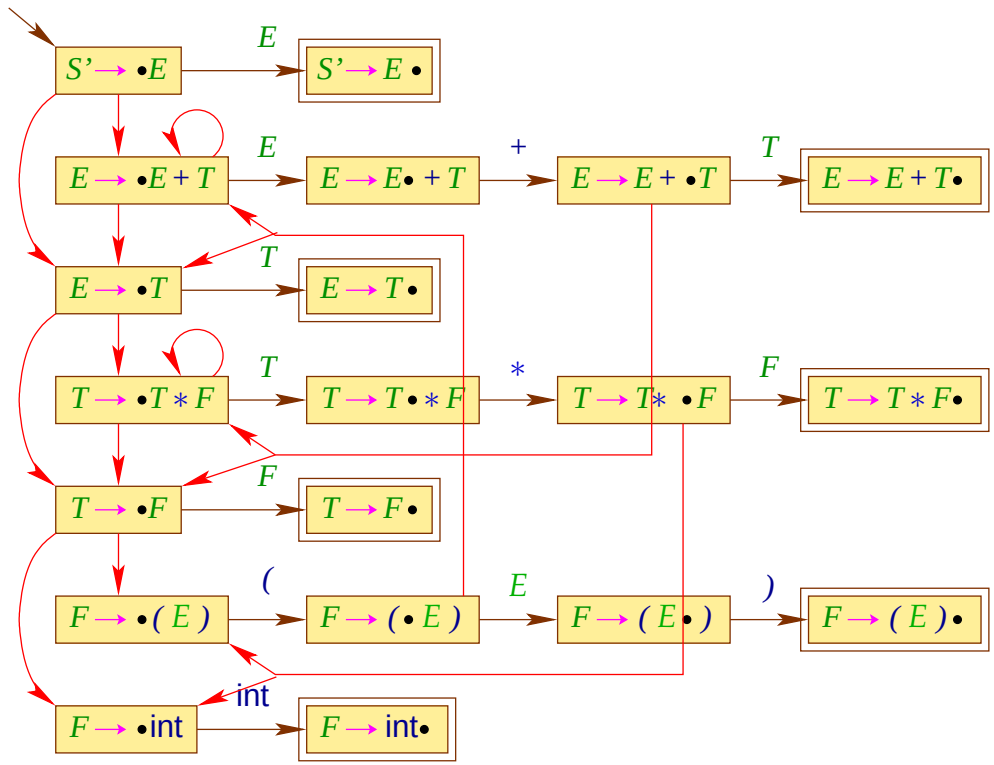
(2) $([A \rightarrow \alpha \bullet B\, \beta], \epsilon, [B \rightarrow \bullet\, \gamma]), \quad A \rightarrow \alpha\, B\, \beta\,,\ B \rightarrow \gamma \in P;$

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

Beispiel:

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



Beispiel:

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