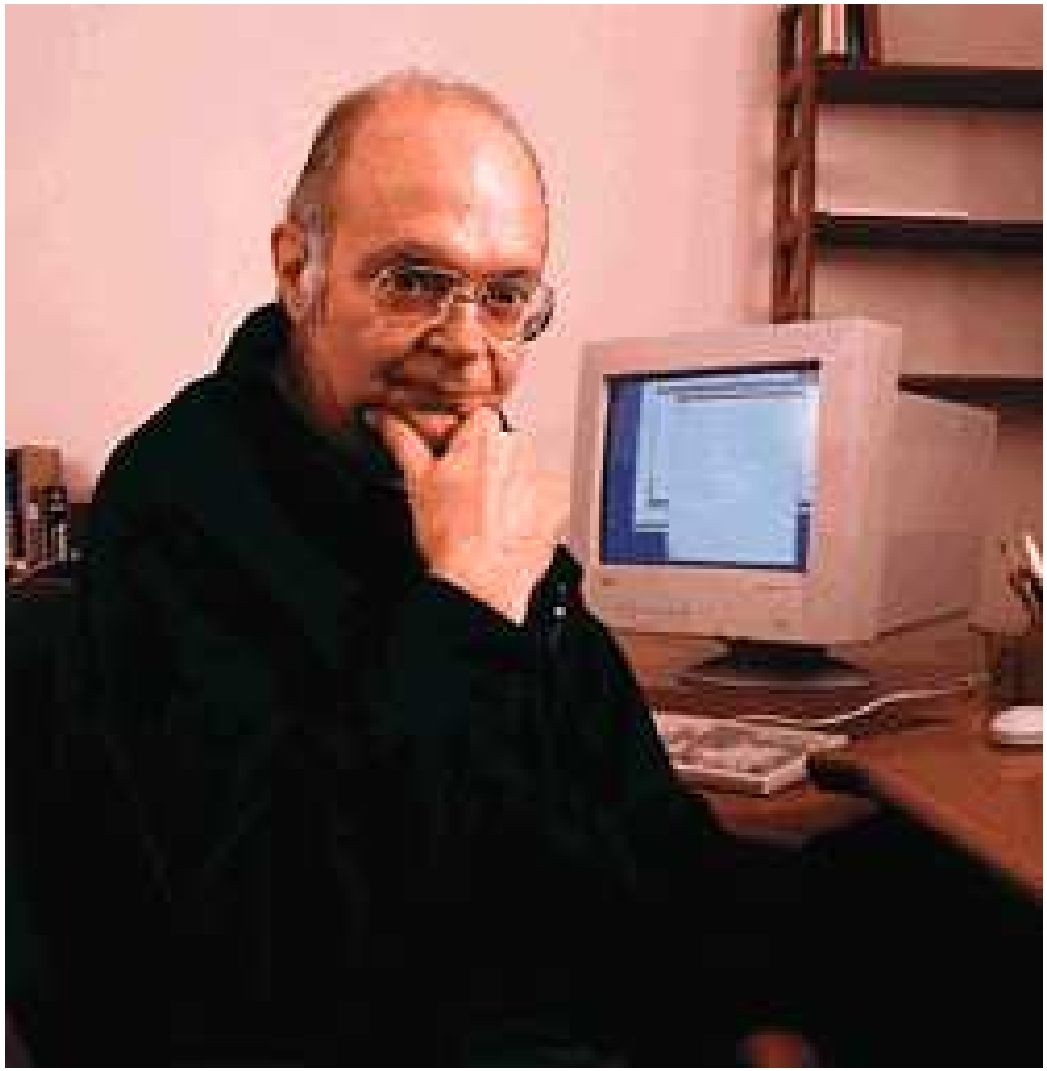
Offenbar gilt:

- Die Folge der Reduktionen entspricht einer reversen Rechtsableitung für die Eingabe :-)
- Zur Korrektheit zeigt man, dass für jedes q gilt:

$$(q, w) \vdash^* (q\,A, \epsilon) \qquad \text{gdw.} \qquad A \rightarrow^* w$$

- Der Kellerautomat $M_G^{(1)}$ ist i.a. nicht-deterministisch :-(
- Um ein deterministisches Parse-Verfahren zu erhalten, muss man die Reduktionsstellen identifizieren $\Longrightarrow$ LR-Parsing



Donald E. Knuth, Stanford

Konstruktion 2: Item-Kellerautomat

- Rekonstruiere eine Linksableitung.
- Expandiere Nichtterminale mithilfe einer Regel.
- Verifiziere sukzessive, dass die gewählte Regel mit der Eingabe übereinstimmt.

  $\Longrightarrow$ Die Zustände sind jetzt Items.
- Ein Item ist eine Regel mit Punkt:

$$[A \to \alpha \bullet \beta] \,, \quad A \to \alpha\,\beta \;\in\; P$$

Der Punkt gibt an, wieweit die Regel bereits abgearbeitet wurde :-)

Unser Beispiel:

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

Wir fügen eine Regel: $S' \to S$ hinzu ;-)

Dann konstruieren wir:

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

Endzustand: $[S' \to S\ \bullet]$

$[S' \to \bullet\ S]$	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	$[S' \to S\ \bullet]$

Der Item-Kellerautomat $M_G^{(2)}$ hat drei Arten von Übergängen:

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

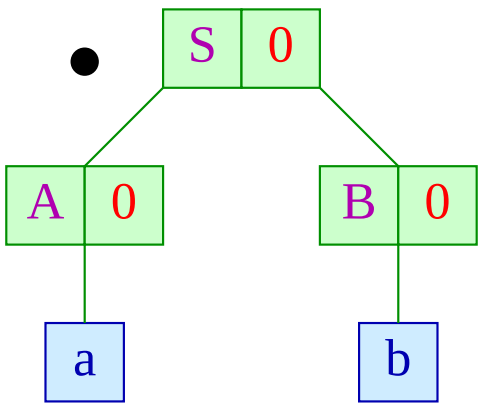
Shifts: $([A \to \alpha \bullet a\, \beta], a, [A \to \alpha\, a \bullet \beta])$ für $A \to \alpha\, a\, \beta \in P$

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

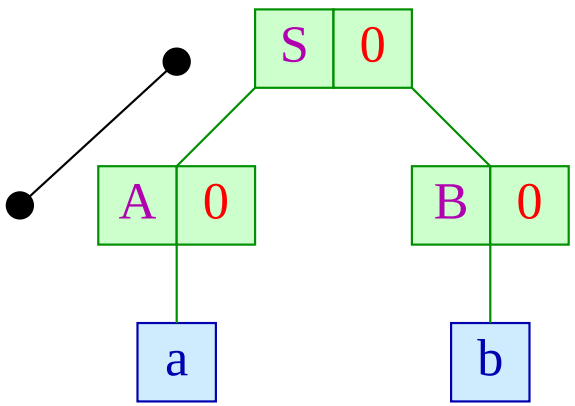
Items der Form: $[A \to \alpha \bullet]$ heißen auch vollständig :-)

Der Item-Kellerautomat schiebt den Punkt einmal um den Ableitungsbaum herum ...

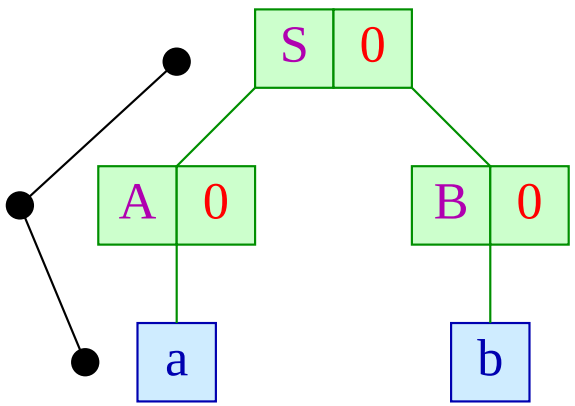
... im Beispiel:



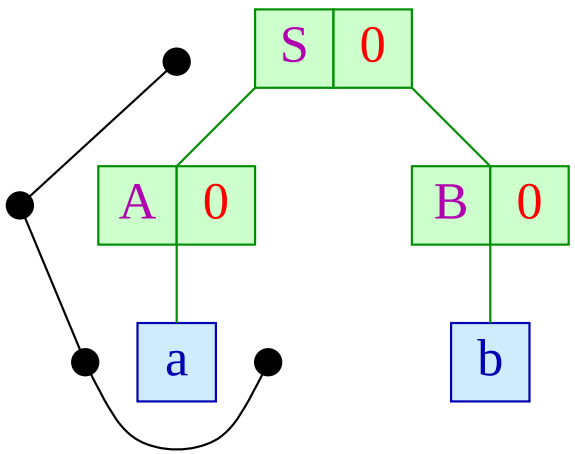
... im Beispiel:



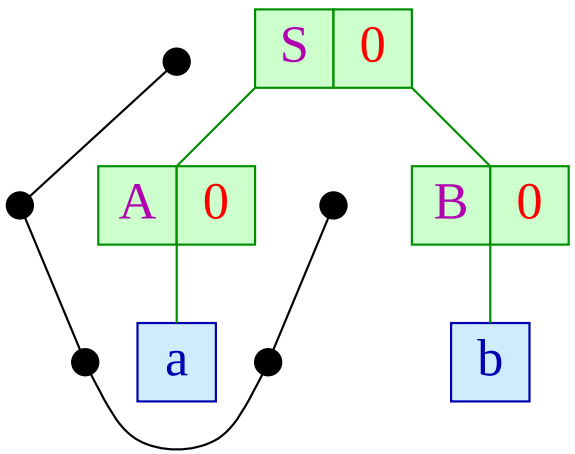
... im Beispiel:



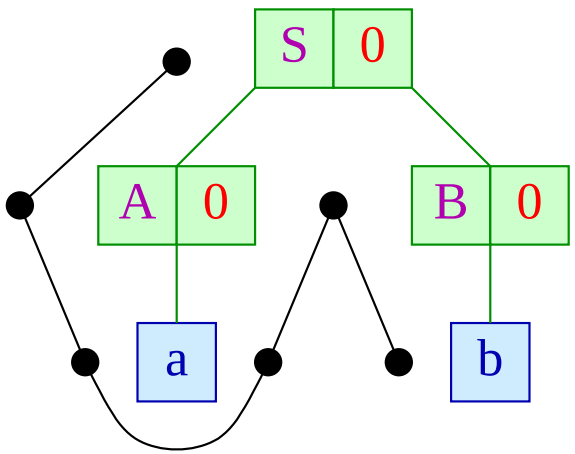
... im Beispiel:



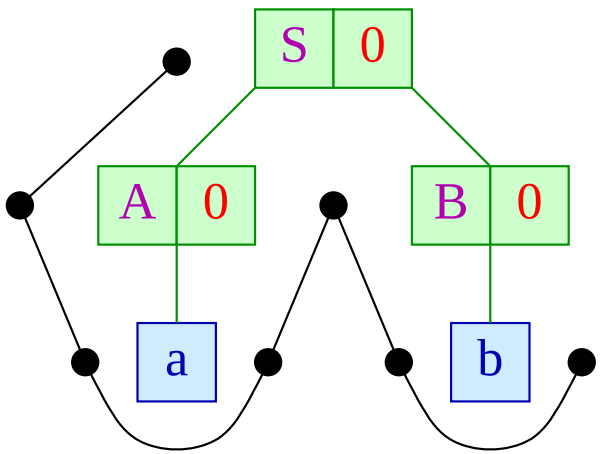
... im Beispiel:



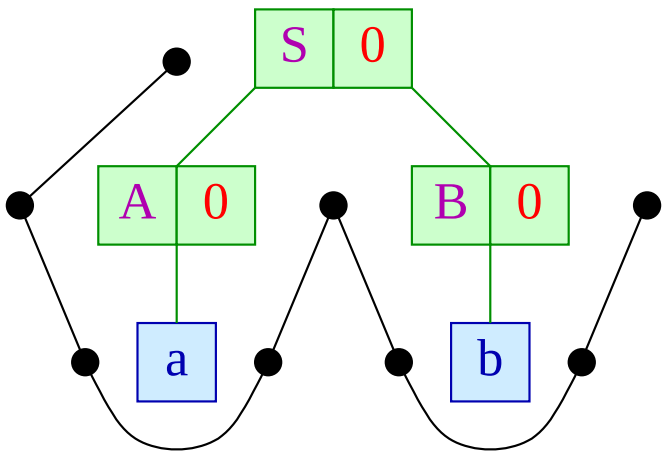
... im Beispiel:



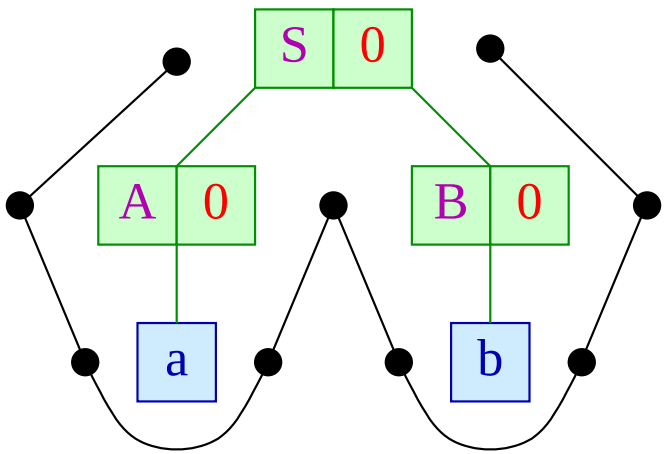
... im Beispiel:



... im Beispiel:



... im Beispiel:



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:

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

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

Philip M. Lewis, SUNY

Richard E. Stearns, SUNY

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 $(3, 4)$ – die sich durch Betrachten des nächsten Zeichens lösen ließen :-)

2.3 Vorausschau-Mengen

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

$$\mathsf{First}_k(L) \;=\; \{u \in L \mid |u| < k\} \cup \{u \in T^k \mid \exists\, v \in T^* :\; uv \in L\}$$

Beispiel:

ϵ
$a\,b$
$a\,a\,b\,b$
$a\,a\,a\,b\,b\,b$

2.3 Vorausschau-Mengen

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

$$\mathsf{First}_k(L) \;=\; \{u \in L \mid |u| < k\} \cup \{u \in T^k \mid \exists\, v \in T^* : \; uv \in L\}$$

Beispiel:

ϵ
$a\,b$
$a\,a$
$a\,a$

die Präfixe der Länge 2 :-)

Rechenregeln:

$\mathsf{First}_k(_)$ ist verträglich mit Vereinigung und Konkatenation:

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

k – Konkatenation

Beachte:

- Die Menge $\mathbb{D}_k = 2^{T^{\leq k}}$ ist endlich :-)
- Die Operation: $\odot : \mathbb{D}_k \times \mathbb{D}_k \to \mathbb{D}_k$ ist distributiv in jedem Argument:

$$\begin{array}{lcl}
L \odot \emptyset & = & \emptyset \\
\emptyset \odot L & = & \emptyset
\end{array}
\qquad
\begin{array}{lcl}
L \odot (L_1 \cup L_2) & = & (L \odot L_1) \cup (L \odot L_2) \\
(L_1 \cup L_2) \odot L & = & (L_1 \odot L) \cup (L_2 \odot L)
\end{array}$$

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

$$\mathsf{First}_k(\alpha) \;=\; \mathsf{First}_k(\{w \in T^* \mid \alpha \to^* w\})$$

Für $k \geq 1$ gilt:

$$\begin{array}{lcll} \mathsf{First}_k(x) & = & \{x\} & \text{für} \quad x \in T \cup \{\epsilon\} \\ \mathsf{First}_k(\alpha_1\, \alpha_2) & = & \mathsf{First}_k(\alpha_1) \odot \mathsf{First}_k(\alpha_2) & \end{array}$$

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

$$\mathsf{First}_k(\alpha) \;=\; \mathsf{First}_k(\{w \in T^* \mid \alpha \rightarrow^* w\})$$

Für $k \geq 1$ gilt:

$$\begin{array}{lcll} \mathsf{First}_k(x) & = & \{x\} & \text{für} \quad x \in T \cup \{\epsilon\} \\ \mathsf{First}_k(\alpha_1\,\alpha_2) & = & \mathsf{First}_k(\alpha_1) \odot \mathsf{First}_k(\alpha_2) & \end{array}$$

Frage: Wie berechnet man $\mathsf{First}_k(A)$??

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

$$\mathsf{First}_k(\alpha) \;=\; \mathsf{First}_k(\{w \in T^* \mid \alpha \to^* w\})$$

Für $k \geq 1$ gilt:

$$\begin{array}{lcll} \mathsf{First}_k(x) & = & \{x\} & \text{für} \quad x \in T \cup \{\epsilon\} \\ \mathsf{First}_k(\alpha_1\,\alpha_2) & = & \mathsf{First}_k(\alpha_1) \odot \mathsf{First}_k(\alpha_2) & \end{array}$$

Frage: Wie berechnet man $\mathsf{First}_k(A)$??

Idee: Stelle ein Ungleichungssystem auf!

Beispiel: $k = 2$

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

Jede Regel gibt Anlass zu einer Inklusionsbeziehung:

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

Beispiel: $k = 2$

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

Jede Regel gibt Anlass zu einer Inklusionsbeziehung:

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

Eine Inklusion $\mathsf{First}_2(E) \supseteq \mathsf{First}_2(E + T)$ kann weiter vereinfacht werden zu:

$$\mathsf{First}_2(E) \supseteq \mathsf{First}_2(E) \odot \{+\} \odot \mathsf{First}_2(T)$$

Insgesamt erhalten wir das Ungleichungssystem:

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

Insgesamt erhalten wir das Ungleichungssystem:

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

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

Allgemein:

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

für jede Regel $A \to X_1 \ldots X_m \in P$ mit $X_i \in T \cup N$.

Gesucht:

- möglichst kleine Lösung (??)
- Algorithmus, der diese berechnet :-)

... im Beispiel:

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

... hat die Lösung:

E	name, int, (name, (int, ((, name $*$, int $*$, name $+$, int $+$
T	name, int, (name, (int, ((, name $*$, int $*$
F	name, int, (name, (int, ((

Beobachtung:

- Die Menge $\mathbb{D}_k$ der möglichen Werte für $\mathsf{First}_k(A)$ bilden einen vollständigen Verband :-)

- Die Operatoren auf den rechten Seiten der Ungleichungen sind monoton, d.h. verträglich mit "$\subseteq$" :-)

Exkurs: Vollständige Verbände

Eine Menge $\mathbb{D}$ mit einer Relation $\sqsubseteq \subseteq \mathbb{D} \times \mathbb{D}$ ist eine Halbordnung falls für allea $a, b, c \in \mathbb{D}$ gilt:

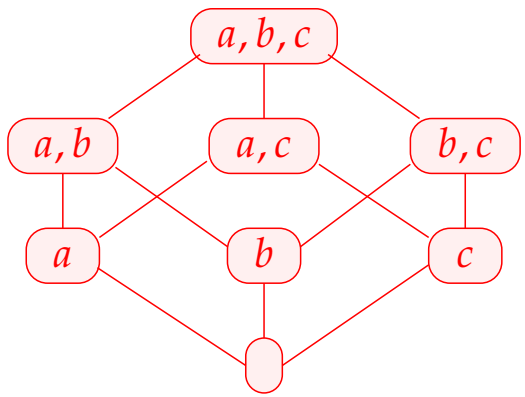
$$a \sqsubseteq a \qquad \textit{Reflexivität}$$

$$a \sqsubseteq b \wedge b \sqsubseteq a \implies a = b \qquad \textit{Anti} - \textit{Symmetrie}$$

$$a \sqsubseteq b \wedge b \sqsubseteq c \implies a \sqsubseteq c \qquad \textit{Transitivität}$$

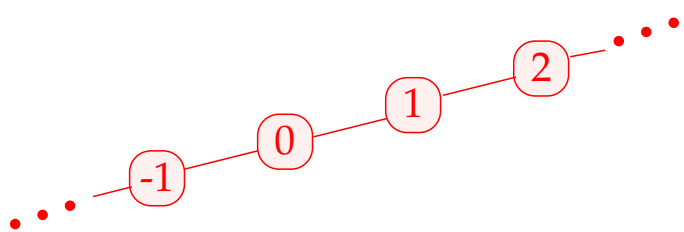
Beispiele:

1. $\mathbb{D} = 2^{\{a,b,c\}}$ mit der Relation "$\subseteq$" :

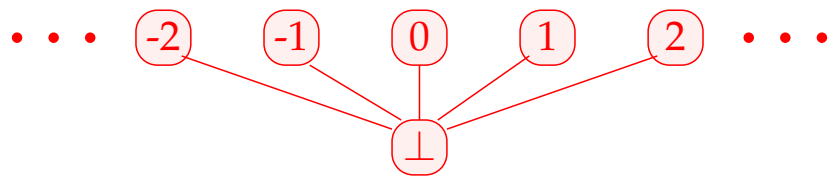


3. $\mathbb{Z}$ mit der Relation "$=$" :

3. $\mathbb{Z}$ mit der Relation "$\leq$" :



4. $\mathbb{Z}_\perp = \mathbb{Z} \cup \{\perp\}$ mit der Ordnung:



$d \in \mathbb{D}$ heißt obere Schranke für $X \subseteq \mathbb{D}$ falls

$$x \sqsubseteq d \qquad \text{für alle} \quad x \in X$$

$d \in \mathbb{D}$ heißt obere Schranke für $X \subseteq \mathbb{D}$ falls

$$x \sqsubseteq d \quad \text{für alle} \quad x \in X$$

d heißt kleinste obere Schranke (lub) falls

1. d eine obere Schranke ist und
2. $d \sqsubseteq y$ für jede obere Schranke y für X.

$d \in \mathbb{D}$ heißt obere Schranke für $X \subseteq \mathbb{D}$ falls

$$x \sqsubseteq d \quad \text{für alle} \quad x \in X$$

d heißt kleinste obere Schranke (lub) falls

1. d eine obere Schranke ist und
2. $d \sqsubseteq y$ für jede obere Schranke y für X.

Achtung:

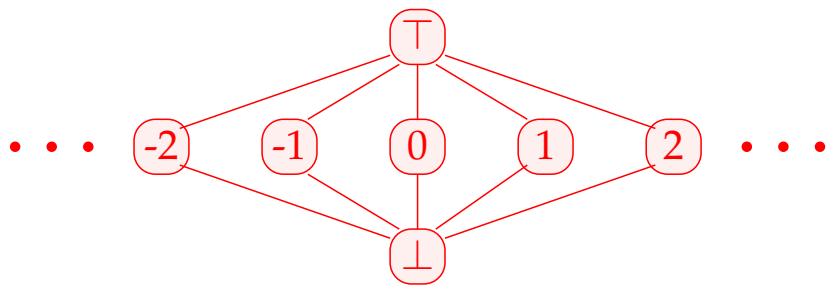
- $\{0, 2, 4, \ldots\} \subseteq \mathbb{Z}$ besitzt keine obere Schranke!
- $\{0, 2, 4\} \subseteq \mathbb{Z}$ besitzt die oberen Schranken $4, 5, 6, \ldots$

Ein vollständiger Verband (cl) $\mathbb{D}$ ist eine Halbordnung, in der jede Teilmenge $X \subseteq \mathbb{D}$ eine kleinste obere Schranke $\bigsqcup X \in \mathbb{D}$ besitzt.

Beachte:

Jeder vollständige Verband besitzt

- $\rightarrow$ ein kleinstes Element $\bot = \bigsqcup \emptyset \in \mathbb{D}$;
- $\rightarrow$ ein größtes Element $\top = \bigsqcup \mathbb{D} \in \mathbb{D}$.

Beispiele:

1. $\mathbb{D} = 2^{\{a,b,c\}}$ ist ein cl :-)
2. $\mathbb{D} = \mathbb{Z}$ mit "$=$" ist keiner.
3. $\mathbb{D} = \mathbb{Z}$ mit "$\leq$" ebenfalls nicht.
4. $\mathbb{D} = \mathbb{Z}_\bot$ auch nicht :-(
5. Mit einem zusätzlichen Symbol $\top$ erhalten wir den flachen Verband $\mathbb{Z}_\bot^\top = \mathbb{Z} \cup \{\bot, \top\}$: