

Konstruktion 2: Item-Kellerautomat

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

⟹ Die Zustände sind jetzt **Items**.

- Ein Item ist eine Regel mit **Punkt**:

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

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

Unser Beispiel:

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

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

Dann konstruieren wir:

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

$[S' \rightarrow \bullet S]$	ϵ	$[S' \rightarrow \bullet S] [S \rightarrow \bullet AB]$
$[S \rightarrow \bullet AB]$	ϵ	$[S \rightarrow \bullet AB] [A \rightarrow \bullet a]$
$[A \rightarrow \bullet a]$	a	$[A \rightarrow a \bullet]$
$[S \rightarrow \bullet AB] [A \rightarrow a \bullet]$	ϵ	$[S \rightarrow A \bullet B]$
$[S \rightarrow A \bullet B]$	ϵ	$[S \rightarrow A \bullet B] [B \rightarrow \bullet b]$
$[B \rightarrow \bullet b]$	b	$[B \rightarrow b \bullet]$
$[S \rightarrow A \bullet B] [B \rightarrow b \bullet]$	ϵ	$[S \rightarrow AB \bullet]$
$[S' \rightarrow \bullet S] [S \rightarrow AB \bullet]$	ϵ	$[S' \rightarrow S \bullet]$

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

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

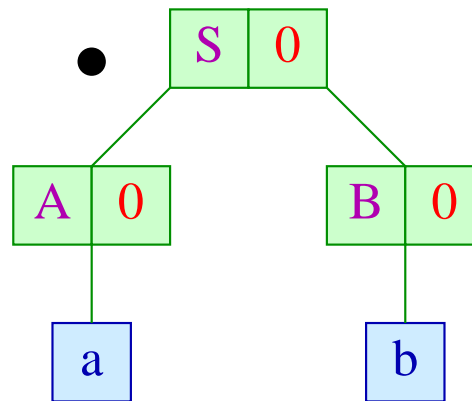
Shifts: $([A \rightarrow \alpha \bullet a \beta], a, [A \rightarrow \alpha a \bullet \beta])$ für $A \rightarrow \alpha a \beta \in P$

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

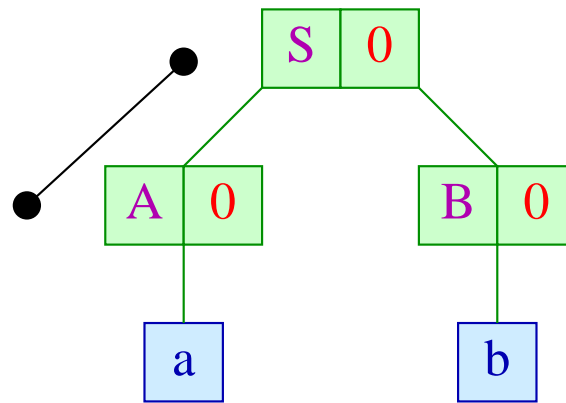
Items der Form: $[A \rightarrow \alpha \bullet]$ heißen auch **vollständig :-)**

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

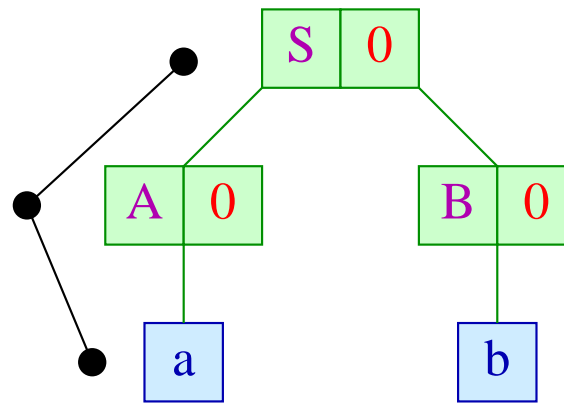
... im Beispiel:



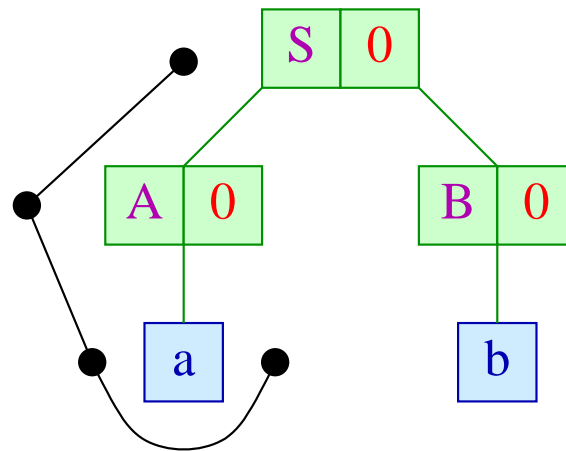
... im Beispiel:



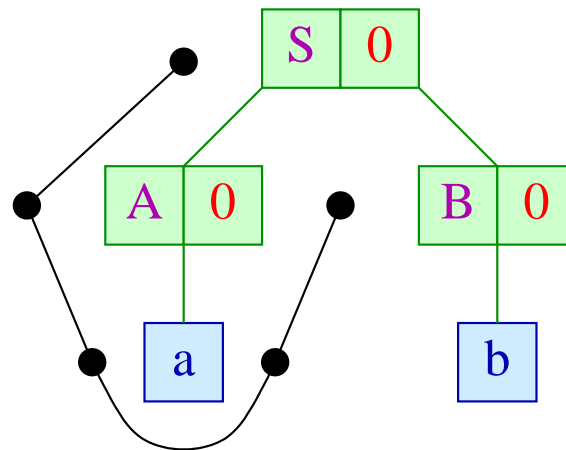
... im Beispiel:



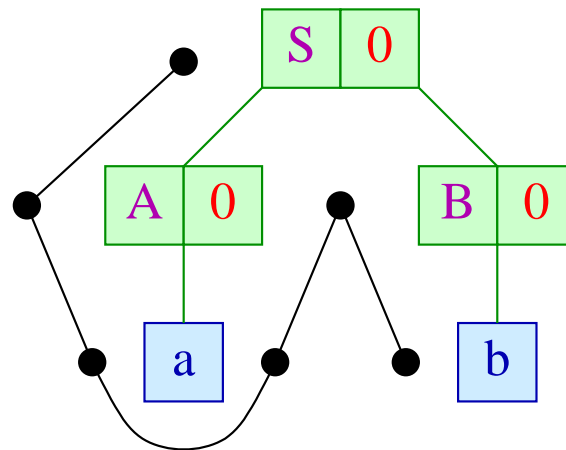
... im Beispiel:



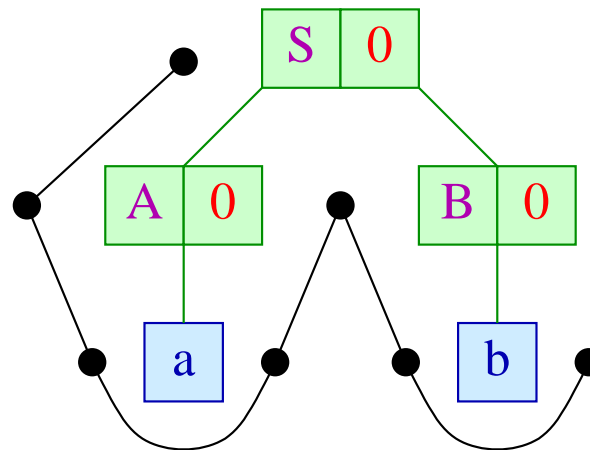
... im Beispiel:



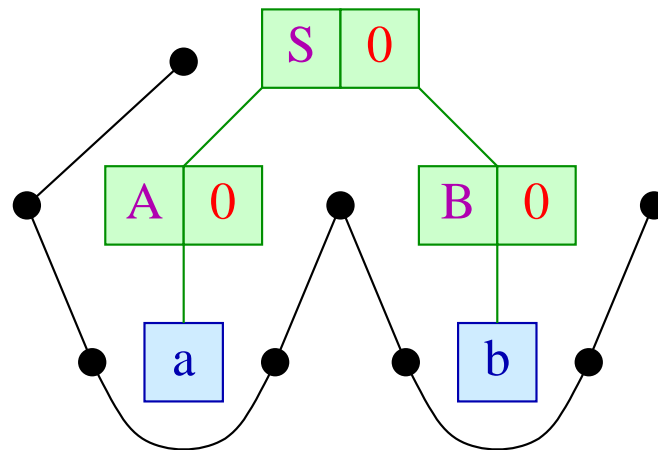
... im Beispiel:



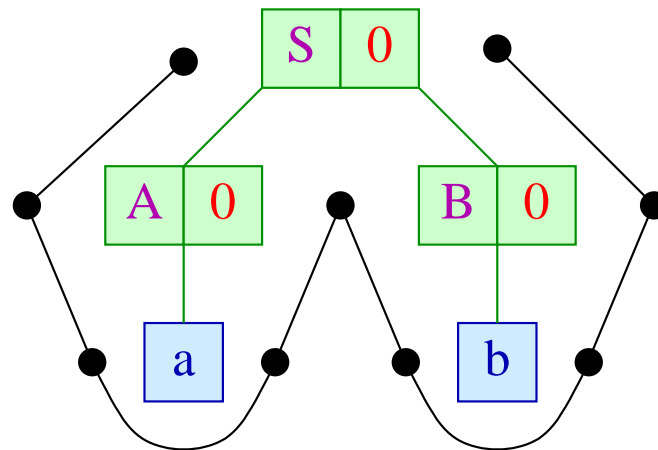
... im Beispiel:



... im Beispiel:



... im Beispiel:

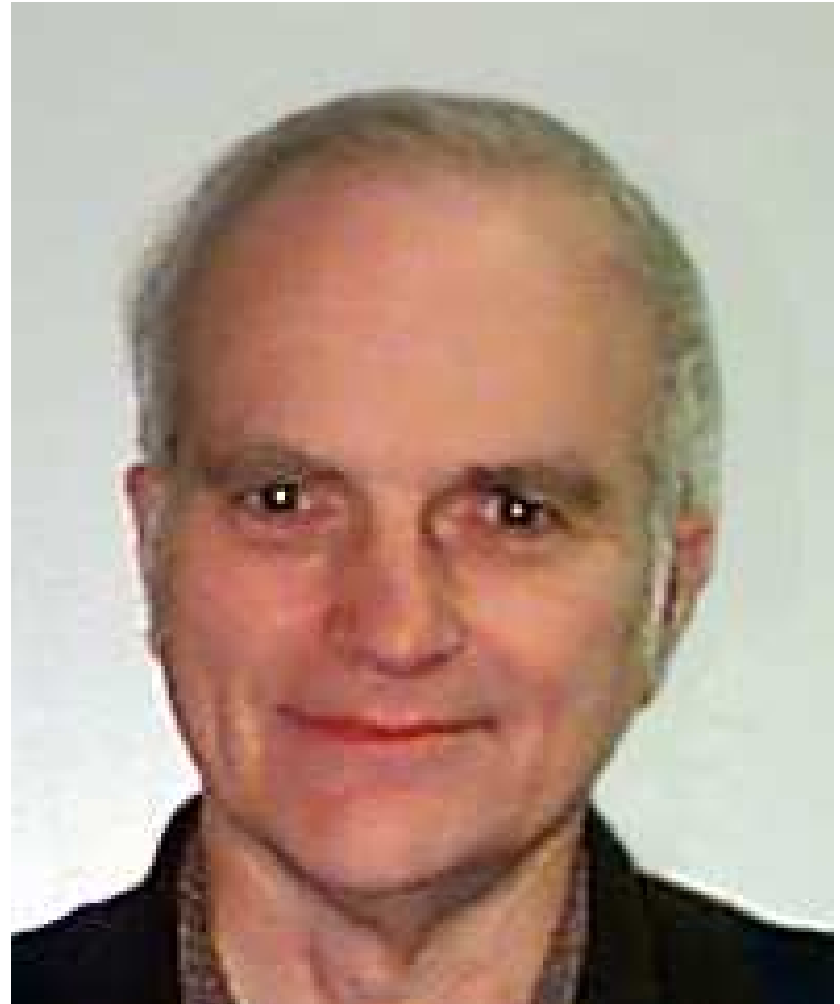


Diskussion:

- 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 $[A \rightarrow \alpha \bullet B \beta]$ gilt:
$$([A \rightarrow \alpha \bullet B \beta], w) \vdash^* ([A \rightarrow \alpha B \bullet \beta], \epsilon) \quad \text{gdw.} \quad B \xrightarrow{*} w$$
- **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: $S \rightarrow \epsilon \mid a S b$

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

2.3 Vorausschau-Mengen

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

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

Beispiel:

ϵ
ab
$aabb$
$aaabbb$

2.3 Vorausschau-Mengen

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

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

Beispiel:

ϵ
ab
aa
aa

die Präfixe der Länge 2 :-)

Rechenregeln:

$\text{First}_k(_)$ ist **verträglich** mit Vereinigung und Konkatination:

$$\begin{aligned}\text{First}_k(\emptyset) &= \emptyset \\ \text{First}_k(L_1 \cup L_2) &= \text{First}_k(L_1) \cup \text{First}_k(L_2) \\ \text{First}_k(L_1 \cdot L_2) &= \text{First}_k(\text{First}_k(L_1) \cdot \text{First}_k(L_2)) \\ &:= \text{First}_k(L_1) \odot \text{First}_k(L_2)\end{aligned}$$

k – Konkatination

Beachte:

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

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

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

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

Für $k \geq 1$ gilt:

$$\text{First}_k(x) = \{x\} \quad \text{für } x \in T \cup \{\epsilon\}$$

$$\text{First}_k(\alpha_1 \alpha_2) = \text{First}_k(\alpha_1) \odot \text{First}_k(\alpha_2)$$

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

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

Für $k \geq 1$ gilt:

$$\text{First}_k(x) = \{x\} \quad \text{für } x \in T \cup \{\epsilon\}$$

$$\text{First}_k(\alpha_1 \alpha_2) = \text{First}_k(\alpha_1) \odot \text{First}_k(\alpha_2)$$

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

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

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

Für $k \geq 1$ gilt:

$$\text{First}_k(x) = \{x\} \quad \text{für } x \in T \cup \{\epsilon\}$$

$$\text{First}_k(\alpha_1 \alpha_2) = \text{First}_k(\alpha_1) \odot \text{First}_k(\alpha_2)$$

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

Idee: Stelle ein **Ungleichungssystem** auf!

Beispiel: $k = 2$

$$\begin{array}{l} E \rightarrow E + T \quad | \quad T \\ T \rightarrow T * F \quad | \quad F \\ F \rightarrow (E) \quad | \quad \text{name} \quad | \quad \text{int} \end{array}$$

Jede Regel gibt Anlass zu einer Inklusionsbeziehung:

$$\begin{array}{ll} \text{First}_2(E) \supseteq \text{First}_2(E + T) & \text{First}_2(E) \supseteq \text{First}_2(T) \\ \text{First}_2(T) \supseteq \text{First}_2(T * F) & \text{First}_2(T) \supseteq \text{First}_2(F) \\ \text{First}_2(F) \supseteq \text{First}_2((E)) & \text{First}_2(F) \supseteq \{\text{name}, \text{int}\} \end{array}$$

Beispiel: $k = 2$

$$\begin{array}{l} E \rightarrow E + T \quad | \quad T \\ T \rightarrow T * F \quad | \quad F \\ F \rightarrow (E) \quad | \quad \text{name} \quad | \quad \text{int} \end{array}$$

Jede Regel gibt Anlass zu einer Inklusionsbeziehung:

$$\begin{array}{ll} \text{First}_2(E) \supseteq \text{First}_2(E + T) & \text{First}_2(E) \supseteq \text{First}_2(T) \\ \text{First}_2(T) \supseteq \text{First}_2(T * F) & \text{First}_2(T) \supseteq \text{First}_2(F) \\ \text{First}_2(F) \supseteq \text{First}_2((E)) & \text{First}_2(F) \supseteq \{\text{name}, \text{int}\} \end{array}$$

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

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

Insgesamt erhalten wir das Ungleichungssystem:

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

$$\text{First}_2(T) \supseteq \text{First}_2(T) \odot \{*\} \odot \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{(\} \odot \text{First}_2(E) \odot \{)\}$$

$$\text{First}_2(E) \supseteq \text{First}_2(T)$$

$$\text{First}_2(T) \supseteq \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{\text{name, int}\}$$

Insgesamt erhalten wir das Ungleichungssystem:

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

$$\text{First}_2(T) \supseteq \text{First}_2(T) \odot \{*\} \odot \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{(\} \odot \text{First}_2(E) \odot \{)\}$$

$$\text{First}_2(E) \supseteq \text{First}_2(T)$$

$$\text{First}_2(T) \supseteq \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{\text{name, int}\}$$

Allgemein:

$$\text{First}_k(A) \supseteq \text{First}_k(X_1) \odot \dots \odot \text{First}_k(X_m)$$

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

Gesucht:

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

... im Beispiel:

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

$$\text{First}_2(T) \supseteq \text{First}_2(T) \odot \{*\} \odot \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{(\} \odot \text{First}_2(E) \odot \{)\}$$

$$\text{First}_2(E) \supseteq \text{First}_2(T)$$

$$\text{First}_2(T) \supseteq \text{First}_2(F)$$

$$\text{First}_2(F) \supseteq \{\text{name, int}\}$$

... 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 $\text{First}_k(A)$ 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 alle $a, b, c \in \mathbb{D}$ gilt:

$$a \sqsubseteq a$$

Reflexivität

$$a \sqsubseteq b \wedge b \sqsubseteq a \implies a = b$$

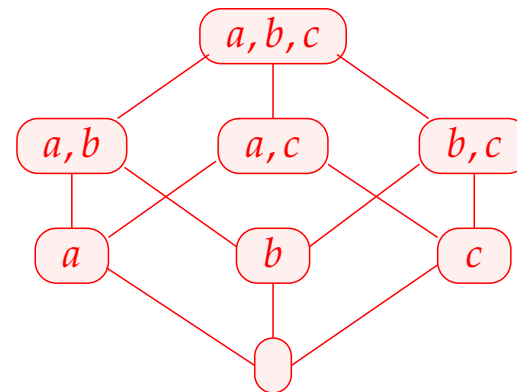
Anti – Symmetrie

$$a \sqsubseteq b \wedge b \sqsubseteq c \implies a \sqsubseteq c$$

Transitivität

Beispiele:

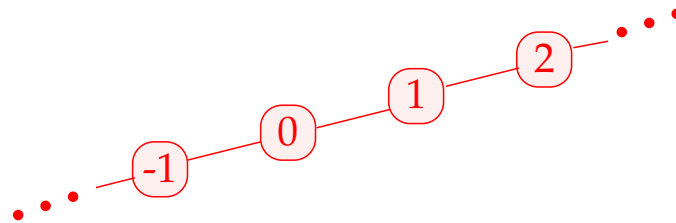
1. $\mathbb{D} = 2^{\{a,b,c\}}$ mit der Relation " \sqsubseteq ":



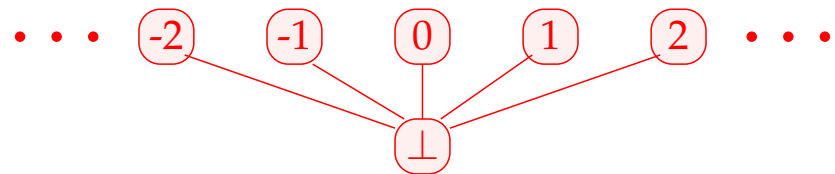
3. \mathbb{Z} mit der Relation “=” :



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

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

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

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

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

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

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:

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

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:

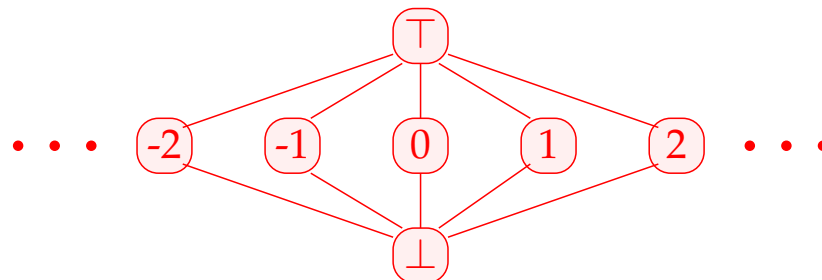
Jeder vollständige Verband besitzt

→ ein **kleinstes** Element $\perp = \bigsqcup \emptyset \in \mathbb{D}$;

→ 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. $\mathbb{D} = \mathbb{Z}_{\perp}$ auch nicht :-)
5. Mit einem zusätzlichen Symbol \top erhalten wir den **flachen** Verband $\mathbb{Z}_{\perp}^{\top} = \mathbb{Z} \cup \{\perp, \top\}$:



Es gilt:

Satz:

In jedem vollständigen Verband \mathbb{D} besitzt jede Teilmenge $X \subseteq \mathbb{D}$ eine größte untere Schranke $\bigwedge X$.

Es gilt:

Satz:

In jedem vollständigen Verband \mathbb{D} besitzt jede Teilmenge $X \subseteq \mathbb{D}$ eine größte untere Schranke $\sqcap X$.

Beweis:

Konstruiere $U = \{u \in \mathbb{D} \mid \forall x \in X : u \sqsubseteq x\}$.

// die Menge der unteren Schranken von X :-)

Es gilt:

Satz:

In jedem vollständigen Verband \mathbb{D} besitzt jede Teilmenge $X \subseteq \mathbb{D}$ eine größte untere Schranke $\sqcap X$.

Beweis:

Konstruiere $U = \{u \in \mathbb{D} \mid \forall x \in X : u \sqsubseteq x\}$.

// die Menge der unteren Schranken von X :-)

Setze: $g := \sqcup U$

Behauptung: $g = \sqcap X$

(1) g ist eine **untere Schranke** von X :

Für $x \in X$ gilt:

$$u \sqsubseteq x \text{ für alle } u \in U$$

$$\implies x \text{ ist obere Schranke von } U$$

$$\implies g \sqsubseteq x \quad :-)$$

(1) g ist eine **untere Schranke** von X :

Für $x \in X$ gilt:

$$u \sqsubseteq x \text{ für alle } u \in U$$

$$\implies x \text{ ist obere Schranke von } U$$

$$\implies g \sqsubseteq x \quad :-)$$

(2) g ist **größte untere Schranke** von X :

Für jede untere Schranke u von X gilt:

$$u \in U$$

$$\implies u \sqsubseteq g \quad :-))$$