

Vorlesungsnotizen

**Mathematik und Logik**

Franz Binder

28. November 2007

# Inhaltsverzeichnis

<b>1</b>	<b>Logik</b>	<b>3</b>
<b>2</b>	<b>Datentypen</b>	<b>4</b>
<b>3</b>	<b>Rekursion</b>	<b>5</b>
3.1	Natürliche Zahlen . . . . .	5
3.2	Arithmetik . . . . .	5
3.3	Listen . . . . .	5
	Introduktion und Elimination . . . . .	5
	Länge einer Liste . . . . .	7
	Konkatenation . . . . .	7
	Iterieren . . . . .	9
	Umkehren . . . . .	9
	Komponentenweise Bearbeitung . . . . .	11
	Akkumulation . . . . .	12
	Listen als Mengen . . . . .	14
	Mehrfache Vorkommen . . . . .	15
	Direkter Zugriff . . . . .	16
	Sortieren . . . . .	16
3.4	Binärbaume . . . . .	16
3.5	Wohlfundierte Relationen . . . . .	16
3.6	Folgen . . . . .	16
<b>4</b>	<b>Algebra</b>	<b>16</b>
<b>5</b>	<b>Graphen</b>	<b>17</b>
<b>6</b>	<b>Formale Sprachen</b>	<b>18</b>
<b>7</b>	<b>Reele Zahlen</b>	<b>19</b>

# 3 Rekursion

## 3.1 Natürliche Zahlen

## 3.2 Arithmetik

## 3.3 Listen

### Introduktion und Elimination

3.3.1 DEFINITION. Zu jeder Menge  $\Sigma$  definieren wir die Menge  $\Sigma^*$  aller *Listen* über  $\Sigma$  induktiv mittels der beiden Konstruktoren:

$$\begin{aligned} \langle \rangle &\in \Sigma^*, \\ \triangleleft \in \Sigma &\rightarrow \Sigma^* \rightarrow \Sigma^*. \end{aligned}$$

Dabei sollten zwei Elemente genau dann gleich sein, wenn sie auf dieselbe Art konstruiert wurden.

3.3.2 BEMERKUNG. Diese Definition besagt sehr viel, vor allem:

- $\langle \rangle$  und  $\triangleleft$  sind Konstruktoren, d.h. daß erstens  $\langle \rangle$  eine Liste ist, und zweitens, daß, wann immer  $\alpha \in \Sigma$  und  $w \in \Sigma^*$ , dann auch  $\alpha \triangleleft w$  eine Liste ist. Formal läßt sich dies durch die folgenden Introduktionsregeln ausdrücken.

$$\frac{}{\langle \rangle \in \Sigma^*} \quad \frac{\alpha \in \Sigma \quad w \in \Sigma^*}{\alpha \triangleleft w \in \Sigma^*}$$

Beispielsweise sind für  $a, b, c, d \in \Sigma$  neben  $\langle \rangle$  auch  $a \triangleleft \langle \rangle$ ,  $a \triangleleft b \triangleleft \langle \rangle$ ,  $a \triangleleft b \triangleleft c \triangleleft d \triangleleft \langle \rangle$ ,  $d \triangleleft c \triangleleft \langle \rangle$ ,  $b \triangleleft c \triangleleft c \triangleleft a \triangleleft b \triangleleft \langle \rangle$  alles Listen.

Ein Ausdruck der Form  $\alpha \triangleleft \beta \triangleleft u$  ist dabei immer wie  $\alpha \triangleleft (\beta \triangleleft u)$  zu lesen.

- Die Definition ist induktiv. D.h., jede Liste läßt sich mit den angegebenen Konstruktoren bilden. Dies läßt sich formal durch eine Eliminationsregeln ( das Induktionsprinzip für Listen) ausdrücken:

$$\frac{P(\langle \rangle) \quad \forall_{a \in \Sigma} \forall_{u \in \Sigma^*} P(u) \implies P(a \triangleleft u)}{\forall_{w \in \Sigma^*} P(w)}$$

Dies schließt insbesondere aus, daß Listen unendlich lange werden könnten.

### 3 Rekursion

- Die beiden Konstruktoren sind wohldefinierte Funktionen, d.h. sie bilden Gleiches auf Gleiches ab:

$$\frac{}{\langle \rangle = \langle \rangle \in \Sigma^*} \quad \frac{\alpha = \beta \in \Sigma \quad u = v \in \Sigma^*}{\alpha \triangleleft u = \beta \triangleleft v \in \Sigma^*}$$

- Wenn zwei Listen gleich sind, dann müssen sie auch auf dieselbe Art konstruiert worden sein. Dies bedeutet insbesondere, daß für  $x \in X$  und  $\varphi \in \Sigma \rightarrow \Sigma^* \rightarrow X \rightarrow X$  durch jede Rekursion der Gestalt

$$\begin{aligned} f(\langle \rangle) &= x \\ f(\alpha \triangleright u) &= \varphi(\alpha, u, f(u)) \end{aligned}$$

eine wohldefinierte Funktion  $f \in \Sigma^* \rightarrow X$  definiert wird.

3.3.3 FOLGERUNG. *Es gilt*

$$\forall_{w \in \Sigma^*} w = \langle \rangle \vee \exists_{\alpha \in \Sigma} \exists_{u \in \Sigma^*} w = \alpha \triangleleft u.$$

3.3.4 FOLGERUNG. *Für alle  $\alpha \in \Sigma$ ,  $u \in \Sigma^*$  gilt*

$$\alpha \triangleleft u \neq \langle \rangle$$

3.3.5 FOLGERUNG. *Für  $\alpha, \beta \in \Sigma$  und  $u, v \in \Sigma^*$  gilt*

$$\alpha \triangleleft u = \beta \triangleleft v \implies \alpha = \beta \wedge u = v.$$

BEWEIS. Wir definieren eine Funktion  $t \in \Sigma^* \rightarrow \Sigma^*$  durch

$$t(\langle \rangle) = \langle \rangle \tag{3.3.1}$$

$$t(\alpha \triangleleft u) = u. \tag{3.3.2}$$

Dann erhalten wir wegen der Wohldefiniertheit von  $t$  aus  $\alpha \triangleleft u = \beta \triangleleft v$  die Gleichung  $t(\alpha \triangleleft u) = t(\beta \triangleleft v)$ . Einsetzen der Definition von  $t$  ergibt dann  $u = v$ .

Für die zweite Gleichung kann man ähnlich vorgehen. □

3.3.6 SATZ. *Wenn  $\Sigma$  diskret ist, dann auch  $\Sigma^*$ .*

3.3.7 NOTATION. Ein Ausdruck wie  $\beta \triangleleft \gamma \triangleleft \gamma \triangleleft \alpha \triangleleft \beta \triangleleft \langle \rangle$  wirkt bei mehrmaliger Verwendung doch etwas plump. Man verwendet daher statt diesem z.B.  $\langle \beta, \gamma, \gamma, \alpha, \beta \rangle$ . Oft werden auch andere Klammern verwendet ( $(\beta, \gamma, \gamma, \alpha, \beta)$ ,  $[\beta, \gamma, \gamma, \alpha, \beta]$ ) oder ganz weggelassen:  $\beta, \gamma, \gamma, \alpha, \beta$ , gerne auch noch ohne Beistriche:  $\beta\gamma\gamma\alpha\beta$ .

### Länge einer Liste

3.3.8 DEFINITION. Die *Länge* einer Liste ist eine Funktion vom Typ  $\Sigma^* \rightarrow \mathbb{N}$ , wird im folgenden mit  $|w|$  bezeichnet und erfüllt die rekursive Definition

$$\begin{aligned} |\langle \rangle| &= 0 \\ |\alpha \triangleleft w| &= 1 + |w|. \end{aligned}$$

3.3.9 BEISPIEL.  $|\langle 3, 7, 7, 3, 3 \rangle| = 5$ .

### Konkatenation

3.3.10 DEFINITION. Die binäre Operation der *Listenverkettung*  $\diamond \in \Sigma^* \rightarrow \Sigma^* \rightarrow \Sigma^*$  ist rekursiv definiert durch

$$\langle \rangle \diamond v = v \tag{3.3.3}$$

$$(\alpha \triangleleft u) \diamond v = \alpha \triangleleft (u \diamond v) \tag{3.3.4}$$

für alle  $\alpha \in \Sigma$  und  $u, v \in \Sigma^*$ .

3.3.11 BEMERKUNG. Insbesondere ist  $\langle \alpha \rangle \diamond v = (\alpha \triangleleft \langle \rangle) \diamond v = \alpha \triangleleft (\langle \rangle \diamond v) = \alpha \triangleleft v$ . Somit kann  $\diamond$  als Verallgemeinerung von  $\triangleleft$  aufgefaßt werden.

3.3.12 BEISPIEL.  $\langle 3, 4, 2 \rangle \diamond \langle 2, 1, 7, 4 \rangle = \langle 3, 4, 2, 2, 1, 7, 4 \rangle$ , wie leicht durch Einsetzen in die rekursiven Gleichungen erkennbar ist.

3.3.13 LEMMA. *Die Listenverkettung ist assoziativ, d.h.*

$$\forall_{u,v,w \in \Sigma^*} (u \diamond v) \diamond w = u \diamond (v \diamond w)$$

BEWEIS. Wir verwenden Listeninduktion:

$$\begin{aligned} (\langle \rangle \diamond v) \diamond w &= v \diamond w \\ &= \langle \rangle \diamond (v \diamond w). \\ ((\alpha \triangleleft u) \diamond v) \diamond w &= (\alpha \triangleleft (u \diamond v)) \diamond w \\ &= \alpha \triangleleft ((u \diamond v) \diamond w) \\ &= \alpha \triangleleft (u \diamond (v \diamond w)) \\ &= (\alpha \triangleleft u) \diamond (v \diamond w). \end{aligned}$$

□

3.3.14 LEMMA. *Die leere Liste verhält sich bei Listenverkettung neutral, d.h.*

$$\forall_{u \in \Sigma^*} \langle \rangle \diamond u = u = u \diamond \langle \rangle.$$

### 3 Rekursion

BEWEIS. Der Teil  $\langle \rangle \diamond u = u$  ist Bestandteil der Definition. Wir zeigen noch  $u \diamond \langle \rangle = u$  mittels Listeninduktion:

$$\begin{aligned} \langle \rangle \diamond \langle \rangle &= \langle \rangle. \\ (\alpha \triangleleft u) \diamond \langle \rangle &= \alpha \triangleleft (u \diamond \langle \rangle) \\ &= \alpha \triangleleft u. \end{aligned}$$

□

3.3.15 SATZ. *Es gibt genau eine Funktion  $\diamond : \Sigma^* \rightarrow \Sigma^* \rightarrow \Sigma^*$ , welche*

$$\begin{aligned} \langle \alpha \rangle \diamond u &= \alpha \triangleleft u \\ \langle \rangle \diamond u &= u \\ u \diamond (v \diamond w) &= (u \diamond v) \diamond w \end{aligned}$$

*erfüllt.*

3.3.16 LEMMA. *Es gelten:*

$$|u \diamond v| = |u| + |v|$$

3.3.17 LEMMA.

$$\forall_{u,v \in \Sigma^*} \forall_{\alpha \in \Sigma} u \diamond (\alpha \triangleleft v) \neq \langle \rangle$$

3.3.18 LEMMA. *Ist  $u \diamond v = \langle \rangle$ , dann gilt  $u = \langle \rangle \wedge v = \langle \rangle$ .*

3.3.19 SATZ. *Für die Listenverkettung gilt die Kürzungsregel, d.h. für  $u, v, w \in \Sigma^*$  gilt*

$$u \diamond v = u \diamond w \implies v = w, \tag{3.3.5}$$

$$u \diamond w = v \diamond w \implies u = v. \tag{3.3.6}$$

BEWEIS. Wir zeigen die erste Beziehung mittels Induktion nach  $u$ .

Für den Induktionsanfang ist zu zeigen:  $\langle \rangle \diamond v = \langle \rangle \diamond w \implies v = w$ , was trivial ist.

Für den Induktionsschritt nehmen wir  $u \diamond v = u \diamond w \implies v = w$  an und zeigen damit  $(\alpha \triangleleft u)v = (\alpha \triangleleft u)w \implies v = w$ . Es gelte also  $(\alpha \triangleleft u)v = (\alpha \triangleleft u)w$ , und somit  $\alpha \triangleleft (u \diamond v) = \alpha \triangleleft (u \diamond w)$ . Inversion liefert dann  $u \diamond v = u \diamond w$ , woraus mittels der Induktionsannahme sofort  $v = w$  folgt.

Für die zweite Beziehung ist analog vorzugehen. □

### 3 Rekursion

#### Iterieren

3.3.20 DEFINITION. Wir definieren rekursiv eine Funktion  $\mathbb{N} \rightarrow \Sigma^* \rightarrow \Sigma^*$ :

$$\begin{aligned} u^0 &= \langle \rangle \\ u^{1+n} &= u \diamond u^n \end{aligned}$$

3.3.21 LEMMA. *Es gelten die üblichen Rechengesetze für Potenzen:*

$$\begin{aligned} u^n \diamond u^m &= u^{n+m} \\ (u^n)^m &= u^{nm} \\ \langle \rangle^n &= \langle \rangle \end{aligned}$$

3.3.22 LEMMA. *Es gelten:*

$$|u^n| = n \cdot |u|$$

und insbesondere

$$|\langle \alpha \rangle^n| = n$$

3.3.23 LEMMA. *Ist  $u \diamond w = v \diamond w'$  und  $|u| = |v|$ , dann gilt  $u = v$ .*

3.3.24 LEMMA. *Ist  $u \diamond w = v \diamond w'$  und  $|u| \leq |v|$ , dann gibt es ein  $v' \in \Sigma^*$ , sodaß  $v = u \diamond v'$ .*

3.3.25 SATZ.

$$u \diamond v = v \diamond u \iff \exists_{t \in \Sigma^*} \exists_{n, m \in \mathbb{N}} u = t^n \wedge v = t^m$$

BEWEIS. Die Richtung von rechts nach links folgt sofort aus den Rechengesetzen für Potenzen.

Für die andere Richtung seien  $u, v \in \Sigma^*$  mit  $u \diamond v = v \diamond u$ . Wir beweisen nun die Existenz passender  $t, n, m$  durch Induktion über die Länge von  $u$ .

Wenn  $|u| = 0$ , dann ist  $u = \langle \rangle$  und somit  $u = v^0$ . Wir wählen also  $t := u$ ,  $n := 0$ ,  $m := 1$ .

Wenn  $|u| > 0$ , und ist  $|u| \leq |v|$ , dann ist  $u \diamond v' = v$ , mit  $|v'| < |v|$ . Die Induktionsvoraussetzung liefert dann  $u = t^n$ ,  $v' = t^m$ . Dann ist  $v = u \diamond v' = t^n t^m = t^{n+m}$ .

Ist dagegen  $|u| > |v|$ , dann vertauschen wir die Rollen von  $u$  und  $v$ , und die Induktionsvoraussetzung liefert  $v = t^n$ ,  $u = t^m$ .  $\square$

#### Umkehren

3.3.26 DEFINITION. Die binäre Operation  $\triangleright \in \Sigma^* \rightarrow \Sigma \rightarrow \Sigma^*$  wird rekursiv definiert durch

$$\begin{aligned} \langle \rangle \triangleright \beta &= \beta \triangleleft \langle \rangle, & (\triangleright \langle \rangle) \\ (\alpha \triangleleft u) \triangleright \beta &= \alpha \triangleleft (u \triangleright \beta). & (\triangleright \triangleleft) \end{aligned}$$

Eqn (3.3.7)

### 3 Rekursion

3.3.27 SATZ. Für  $\alpha \in \Sigma$ ,  $u \in \Sigma^*$  gilt

$$|u \triangleright \alpha| = 1 + |u|.$$

3.3.28 DEFINITION. Die Spiegelung  $\sim \in \Sigma^* \rightarrow \Sigma^*$  läßt sich rekursiv definieren durch

$$\begin{aligned} \langle \rangle &= \langle \rangle, \\ \widetilde{\alpha \triangleleft u} &= \widetilde{u} \triangleright \alpha. \end{aligned}$$

3.3.29 SATZ. Für  $\alpha \in \Sigma$ ,  $u \in \Sigma^*$  gilt

$$\widetilde{u \triangleright \alpha} = \alpha \triangleleft \widetilde{u}.$$

3.3.30 SATZ.  $|\widetilde{u}| = |u|$ .

3.3.31 SATZ.  $u \widetilde{\diamond} v = \widetilde{v} \diamond u$ .

3.3.32 SATZ.  $\widetilde{\widetilde{u}} = u$ .

3.3.33 SATZ. Sei  $P \in \Sigma^* \rightarrow \mathbb{P}$  eine Eigenschaft. Dann gilt das Induktionsprinzip

$$\frac{P(\langle \rangle) \quad \forall_{\substack{u \in \Sigma^* \\ P(u)}} \quad \forall_{\alpha \in \Sigma} P(u \triangleright \alpha)}{\forall_{w \in \Sigma^*} P(w)}$$

3.3.34 FOLGERUNG. Es gilt

$$\forall_{w \in \Sigma^*} w = \langle \rangle \vee \exists_{\beta \in \Sigma} \exists_{u \in \Sigma^*} w = u \triangleright \beta; .$$

3.3.35 DEFINITION. Eine Liste ist ein *Palindrom*, kurz  $\text{PD}(u)$ , wenn sie von vorne und hinten gelesen gleich erscheint. Dies läßt sich induktiv durch die folgenden Introduktionsregeln ausdrücken:

$$\frac{}{\text{PD}(\langle \rangle)} \quad \frac{\alpha \in \Sigma}{\text{PD}(\langle \alpha \rangle)} \quad \frac{u \in \Sigma^* \quad \text{PD}(u) \quad \alpha \in \Sigma}{\text{PD}(\alpha \triangleleft u \triangleright \alpha)}$$

Die zugehörige Eliminationsregel lautet dann

$$\frac{P(\langle \rangle) \quad \forall_{\alpha \in \Sigma} P(\langle \alpha \rangle) \quad \forall_{\substack{u \in \Sigma^* \\ \text{PD}(u)}} \quad \forall_{\alpha \in \Sigma} P(\alpha \triangleleft u \triangleright \alpha)}{\forall_{\substack{u \in \Sigma^* \\ \text{PD}(u)}} P(u)}$$

für eine beliebige Eigenschaft  $P \in \Sigma^* \rightarrow \mathbb{P}$ .

### 3 Rekursion

3.3.36 SATZ. Eine Liste ist genau dann ein Palindrom, wenn sie mit ihrer Spiegelung übereinstimmt, d.h. für jedes  $u \in \Sigma^*$  gilt

$$\text{PD}(u) \iff u = \tilde{u}.$$

BEWEIS. Wir zeigen zuerst

$$\forall_{\substack{u \in \Sigma^* \\ \text{PD}(u)}} u = \tilde{u}$$

mittels der Eliminationsregel:

- $\langle \rangle = \tilde{\langle \rangle}$  ist per definition erfüllt.
- $\langle \alpha \rangle = \tilde{\langle \alpha \rangle}$  ist ebenfalls für jedes  $\alpha \in \Sigma$  erfüllt (setze  $u = \langle \rangle$  in 3.3.28).
- Für  $\alpha \in \Sigma$ ,  $u \in \Sigma^*$  gelte  $u = \tilde{u}$ . Wir haben zu zeigen, daß dann auch  $\alpha \triangleleft u \triangleright \alpha = \alpha \triangleleft \tilde{u} \triangleright \alpha$ . Tatsächlich gilt  $\alpha \triangleleft (u \triangleright \alpha) = \tilde{u} \triangleright \alpha \triangleright \alpha = (\alpha \triangleleft \tilde{u}) \triangleright \alpha = (\alpha \triangleleft u) \triangleright \alpha$ .

Für die Umkehrung haben wir zu zeigen, daß

$$\forall_{\substack{w \in \Sigma^* \\ w = \tilde{w}}} \text{PD}(w).$$

Sei dazu  $w \in \Sigma^*$  vorgeben. Anwendung von 3.3.3 und 3.3.34 ergibt

$$w = \langle \rangle \vee \exists_{\alpha \in \Sigma} w = \langle \alpha \rangle \vee \exists_{u \in \Sigma^*} \exists_{\alpha, \beta \in \Sigma} w = \alpha \triangleleft u \triangleright \beta.$$

In den ersten beiden Fällen folgt  $P(w)$  sofort. Im letzten Fall folgt aus  $w = \tilde{w}$  schnell  $\alpha = \beta$  sowie  $u = \tilde{u}$ . Da  $|u| < |w|$  können wir mittels Induktion über die Länge von  $w$  fortfahren. Dann ist  $u$  ein Palindrom, und somit auch  $w = \alpha \triangleleft u \triangleright \alpha$ .  $\square$

### Komponentenweise Bearbeitung

3.3.37 DEFINITION. Eine Funktion  $f \in \Sigma_1 \rightarrow \Sigma_2$  läßt sich durch elementweises Anwenden zu einer Funktion  $f^* \in \Sigma_1^* \rightarrow \Sigma_2^*$  hochziehen, die sich rekursiv folgendermaßen ausdrücken läßt:

$$\begin{aligned} f^* \langle \rangle &= \langle \rangle, \\ f^* (\alpha \triangleleft u) &= f \alpha \triangleleft f^* u. \end{aligned}$$

3.3.38 BEISPIEL.  $f^* \langle 1, 2, 3 \rangle = \langle f(1), f(2), f(3) \rangle$ .

3.3.39 SATZ.  $f^*$  ist ein Homomorphismus, d.h.

$$f^* (u \diamond v) = f^* u \diamond f^* v.$$

### 3 Rekursion

BEWEIS. Listeninduktion nach  $u$ :

$$\begin{aligned}
 f^*(\langle \rangle \diamond v) &= f^*v \\
 &= \langle \rangle \diamond f^*v \\
 &= f^*\langle \rangle \diamond f^*v. \\
 f^*((\alpha \triangleleft u) \diamond v) &= f^*(\alpha \triangleleft (u \diamond v)) \\
 &= f\alpha \triangleleft f^*(u \diamond v) \\
 &= f\alpha \triangleleft (f^*u \diamond f^*v) \\
 &= (f\alpha \triangleleft f^*u) \diamond f^*v \\
 &= f^*(\alpha \triangleleft u) \diamond f^*v.
 \end{aligned}$$

□

3.3.40 NOTATION. In Anlehnung an eine weitverbreitete Notation bei Mengen, schreibt man für  $f^*(u)$  gerne auch  $\langle f(\alpha) \mid \alpha \in u \rangle$ , was besonders praktisch ist, wenn  $f$  durch einen Term gegeben ist, z.B.  $\langle x^2 \mid x \in \ell \rangle$  ergibt die Liste der Quadrate der Elemente in der Liste  $\ell$ .

3.3.41 SATZ. Die Abbildung  $*$ :  $(\Sigma \rightarrow \Sigma) \rightarrow (\Sigma^* \rightarrow \Sigma^*)$  erfüllt

$$\begin{aligned}
 (\iota_\Sigma)^* &= \iota_{\Sigma^*}, \\
 (g \circ f)^* &= g^* \circ f^*.
 \end{aligned}$$

Man nennt sie daher auch einen *Funktor*.

3.3.42 SATZ.

$$|f^*u| = |u|.$$

### Akkumulation

3.3.43 DEFINITION. Sei  $A$  eine beliebige Menge und  $\varphi \in \Sigma \rightarrow A \rightarrow A$ . Wir schreiben  $\varphi$  wie eine binäre Operation, also  $\alpha \varphi e$  statt  $\varphi(\alpha, e)$ , und der Ausdruck  $\alpha \varphi \beta \varphi e$  ist wie  $\alpha \varphi (\beta \varphi e)$  zu lesen.

$\varphi$  läßt sich dann zu einer Funktion  $\overleftarrow{\varphi} \in \Sigma^* \rightarrow A \rightarrow A$ , der *Rechtsakkumulation* (engl.: *fold right*) von  $\varphi$ , erweitern:

$$\begin{aligned}
 \langle \rangle \overleftarrow{\varphi} e &= e, \\
 (\alpha \triangleleft u) \overleftarrow{\varphi} e &= \alpha \varphi (u \overleftarrow{\varphi} e).
 \end{aligned}$$

3.3.44 BEMERKUNG. Damit ergibt sich z.B.

$$\langle 4, 3, 2, 1 \rangle \overleftarrow{\varphi} 0 = 4 \varphi 3 \varphi 2 \varphi 1 \varphi 0.$$

Da dies wie  $4 \varphi (3 \varphi (2 \varphi (1 \varphi 0)))$  zu lesen ist beginnt dabei die Abarbeitung der Liste am rechten Rand (daher: Rechtsakkumulation) und setzt sich von dort nach links fort (entsprechend der Pfeilrichtung).

### 3 Rekursion

3.3.45 BEISPIEL. Ist  $A$  eine Menge, in der man vernünftig addieren kann, dann definieren wir eine Funktion  $\sum \in A^* \rightarrow A$ , sodaß für alle  $u \in A^*$ ,

$$\sum u := u \overleftarrow{+} 0.$$

Somit gilt:

$$\begin{aligned} \sum \langle \rangle &= 0, \\ \sum (\alpha \triangleleft u) &= \alpha + \sum u. \end{aligned}$$

3.3.46 NOTATION. Für  $n \in \mathbb{N}$  verwendet man die Notation

$$\sum_{k=1}^n a_k = \sum \langle a_k \mid k \in \langle 1, 2, \dots, n \rangle \rangle.$$

3.3.47 BEISPIEL. Zu  $\triangleleft \in \Sigma \rightarrow \Sigma^* \rightarrow \Sigma^*$  erhalten wir  $\overleftarrow{\triangleleft} \in \Sigma^* \rightarrow \Sigma^* \rightarrow \Sigma^*$ , und es gilt

$$\overleftarrow{\triangleleft} = \diamond$$

Selbstverständlich kann man alles auch verkehrt-herum durchführen:

3.3.48 DEFINITION. Sei  $A$  eine beliebige Menge und  $\varphi \in A \rightarrow \Sigma \rightarrow A$ . Wir schreiben  $\varphi$  wie eine binäre Operation, also  $e \varphi \alpha$  statt  $\varphi(e, \alpha)$ , und der Ausdruck  $e \varphi \alpha \varphi \beta$  ist wie  $(e \varphi \alpha) \varphi \beta$  zu lesen.

$\varphi$  läßt sich dann zu einer Funktion  $\overrightarrow{\varphi} \in A \rightarrow \Sigma^* \rightarrow A$ , der *Linksakkumulation* (engl.: *fold left*) von  $\varphi$ , erweitern:

$$\begin{aligned} e \overrightarrow{\varphi} \langle \rangle &= e, \\ e \overrightarrow{\varphi} (u \triangleright \beta) &= (e \overrightarrow{\varphi} u) \varphi \beta. \end{aligned}$$

3.3.49 BEMERKUNG. Damit ergibt sich z.B.

$$0 \overrightarrow{\varphi} \langle 1, 2, 3, 4 \rangle = 0 \varphi 1 \varphi 2 \varphi 3 \varphi 4.$$

Da dies wie  $((((0 \varphi 1) \varphi 2) \varphi 3) \varphi 4)$  zu lesen ist beginnt dabei die Abarbeitung der Liste am linken Rand (daher: Linksakkumulation) und setzt sich von dort nach rechts fort (entsprechend der Pfeilrichtung).

3.3.50 SATZ.

$$\begin{aligned} e \overrightarrow{\varphi} \langle \rangle &= e, \\ e \overrightarrow{\varphi} (\alpha \triangleleft u) &= (e \varphi \alpha) \overrightarrow{\varphi} u. \end{aligned}$$

### 3 Rekursion

3.3.51 BEISPIEL. Ist  $A$  eine Menge, in der man vernünftig addieren kann, so gilt für alle  $u \in A^*$ ,

$$\sum u := 0 \vec{+} u,$$

sodaß gilt:

$$\begin{aligned} \sum \langle \rangle &= 0, \\ \sum (\alpha \triangleleft u) &= \alpha + \sum u. \end{aligned}$$

3.3.52 BEISPIEL. Zu  $\triangleright \in \Sigma^* \rightarrow \Sigma \rightarrow \Sigma^*$  erhalten wir  $\vec{\triangleright} \in \Sigma^* \rightarrow \Sigma^* \rightarrow \Sigma^*$ , und es gilt

$$\vec{\triangleright} = \diamond$$

.

### Listen als Mengen

3.3.53 DEFINITION. Die zweistellige Relation  $\in: \Sigma \rightarrow \Sigma^* \rightarrow \mathbb{P}$  läßt sich induktiv definieren durch

$$\begin{aligned} \alpha \in \alpha \triangleleft u, \\ \beta \in u \implies \beta \in \alpha \triangleleft u. \end{aligned}$$

3.3.54 LEMMA.  $\alpha \in u \diamond v \iff \alpha \in u \vee \alpha \in v$ .

3.3.55 LEMMA. Für  $\beta \in \Sigma$ ,  $w \in \Sigma^*$  gilt

$$\beta \in w \iff \exists_{u,v \in \Sigma^*} w = u \diamond (\beta \triangleleft v).$$

3.3.56 LEMMA. Es gilt stets

$$\begin{aligned} \alpha \notin \langle \rangle, \\ \alpha \in \beta \triangleleft u \iff \alpha = \beta \vee \alpha \in u. \end{aligned}$$

3.3.57 LEMMA.

$$\begin{aligned} w = \langle \rangle &\iff \forall_{\alpha \in \Sigma} \alpha \notin w, \\ w \neq \langle \rangle &\iff \exists_{\alpha \in \Sigma} \alpha \in w. \end{aligned}$$

3.3.58 LEMMA. Ist  $\Sigma$  diskret, dann auch  $\Sigma^*$ .

### 3 Rekursion

3.3.59 LEMMA. Ist  $\Sigma$  diskret, dann ist  $\in$  entscheidbar:

$$\forall_{u,v \in \Sigma^*} \alpha \in u \vee \alpha \notin u,$$

wobei sich tatsächlich für jeden konkreten Fall bestimmen läßt, welche Alternative zutrifft. Somit ist  $\in$  in diesem Fall eine Funktion vom Typ  $\Sigma \rightarrow \Sigma^* \rightarrow \mathbb{B}$ .

3.3.60 DEFINITION. Für diskretes  $\Sigma$  läßt sich ferner definieren:

$$\begin{aligned} \text{rm}_\beta \langle \rangle &= \langle \rangle, \\ \text{rm}_\beta(\alpha \triangleleft u) &= \text{rm}_\beta u && \text{falls } \alpha = \beta, \\ \text{rm}_\beta(\alpha \triangleleft u) &= \alpha \triangleleft \text{rm}_\beta u && \text{sonst.} \end{aligned}$$

3.3.61 DEFINITION. Sei  $b \in \Sigma^* \rightarrow \mathbb{B}$  eine entscheidbare Eigenschaft. Dann bilden wir die Teilliste  $\langle u \mid b \rangle$  bestehend aus allen Elementen von  $u$ , welche  $b$  erfüllen, durch:

$$\begin{aligned} \langle \langle \rangle \mid b \rangle &= \langle \rangle, \\ \langle \alpha \triangleleft u \mid b \rangle &= \alpha \triangleleft \langle u \mid b \rangle && \text{falls } b(\alpha), \\ \langle \alpha \triangleleft u \mid b \rangle &= \langle u \mid b \rangle && \text{sonst.} \end{aligned}$$

3.3.62 NOTATION. Statt  $\langle u \mid \alpha \mapsto b(\alpha) \rangle$  schreiben wir einfacher  $\langle \alpha \in u \mid b(\alpha) \rangle$ , z.B.  $\langle x \in u \mid (x \mid 60) \rangle$  um die Teiler von 60 auszuwählen; für  $u = \langle 4, 6, 7, 2, 4, 5, 5, 12, 8 \rangle$  ergibt sich dann z.B.  $\langle 4, 6, 2, 4, 5, 5, 12 \rangle$ .

3.3.63 SATZ. Es gilt

$$\text{rm}_\beta u = \langle \alpha \in u \mid \alpha \neq \beta \rangle.$$

### Mehrfache Vorkommen

3.3.64 DEFINITION. Für diskretes  $\Sigma$  kann man definieren.

$$\begin{aligned} |\langle \rangle|_\beta &= 0, \\ |\alpha \triangleleft u|_\beta &= 1 + |u|_\beta && \text{falls } \alpha = \beta, \\ |\alpha \triangleleft u|_\beta &= |u|_\beta && \text{sonst.} \end{aligned}$$

3.3.65 LEMMA. Für diskretes  $\Sigma$  gilt

$$\begin{aligned} \beta \in w &\iff |w|_\beta > 0, \\ w = \langle \rangle &\iff \forall_{\alpha \in \Sigma} |w|_\alpha = 0, \\ |w|_\beta &= |\langle \alpha \in w \mid \alpha = \beta \rangle|. \end{aligned}$$

## Direkter Zugriff

3.3.66 DEFINITION.

$$(\alpha \triangleleft u)[0] = \alpha,$$

$$n < |u| \implies (\alpha \triangleleft u)[1 + n] = u[n].$$

$$0 < n \leq |u| \implies u[-n] = u[|u| - n].$$

Sortieren

## 3.4 Binärbaume

## 3.5 Wohlfundierte Relationen

## 3.6 Folgen