

Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

# Mathematik und Logik

für WIN

2008W

Franz Binder  
Institut für Algebra  
Johannes Kepler Universität Linz

Vorlesung im 2008W

<http://www.algebra.uni-linz.ac.at/Students/Win/ml>

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

- ▶ Durch Hintereinanderreihung von Objekten entsteht eine **Liste**.
- ▶ Ist  $\Sigma$  eine Menge, dann bezeichne  $\Sigma^*$  die Menge der Listen über dem Grunddatentyp  $\Sigma$ .
- ▶ Die einfachste Liste ist die leere Liste:  $\epsilon \in \Sigma$ .
- ▶ Ist  $\alpha \in \Sigma$  und  $u \in \Sigma^*$ , dann ist auch  $\alpha u \in \Sigma^*$ .

## BEISPIEL

Seien  $a, b, c, d \in \Sigma$ .

$\epsilon, a\epsilon, a(b\epsilon), a(b(c(d\epsilon))), d(c\epsilon), b(c(c(a(b\epsilon))))$  sind Listen.

## NOTATION

Rechts-assoziative Klammerung:  $\alpha\beta u = \alpha(\beta u)$ .

Das Zeichen für die leere Liste ist oft entbehrlich.

## BEISPIEL

$\epsilon, a\epsilon, ab\epsilon, abcde\epsilon, dce\epsilon, bccabe\epsilon$ .

Oder noch einfacher:  $\epsilon, a, ab, abcd, dc, bccab$ .

# Länge einer Liste

## Listen

### Länge

### Induktion

### Rekursion

### Anhängen

### Spiegeln

### Verkettung

### Assoziativität

### Neutrales Element

### Homomorphismus

### Kürzungsregeln

- ▶ Durch die Länge wird jeder Liste eine Zahl zugeordnet.
- ▶ Notation:  $|u|$ .
- ▶ Die leere Liste hat Länge 0.
- ▶ Fügt man einer Liste ein Element hinzu, dann wird sie um 1 länger.
- ▶ Es gelten daher die Gleichungen:

$$|\epsilon| = 0 \quad (\text{len-nil}),$$

$$\forall_{\alpha \in \Sigma} \forall_{u \in \Sigma^*} |\alpha u| = 1 + |u| \quad (\text{len-pre}).$$

- ▶ Frage: Ist die Länge damit ordentlich definiert?
- ▶ Beispiel:

$$\begin{aligned} |abcc| &= |a(b(c(c\epsilon)))| = 1 + |b(c(c\epsilon))| = 1 + (1 + |c(c\epsilon)|) = \\ &= 1 + (1 + (1 + |c\epsilon|)) = 1 + (1 + (1 + (1 + |\epsilon|))) = 1 + (1 + (1 + (1 + 0))) = 4 \end{aligned}$$

- ▶ Die Länge ist für die leere Liste definiert.
- ▶ Ist die Länge für eine Liste  $u$  definiert, dann auch für jede Liste der Form  $au$ .
- ▶ Jede Liste läßt sich aus der leeren Liste durch Hinzufügen von Elementen konstruieren.
- ▶ Daher ist die Länge für jede Liste definiert.
- ▶ Die Schlußfolgerung erfolgte nach folgendem Schema:

$$\frac{P(\epsilon) \quad \forall_{a \in \Sigma} \forall_{u \in \Sigma^*} P(u) \implies P(au)}{\forall_{w \in \Sigma^*} P(w)} \text{List } \mathcal{E}$$

Dabei ist  $P$  eine beliebige Eigenschaft für Listen.

- ▶ Dieses **Induktionsprinzip** bedeutet gerade, daß jede Liste durch die beiden **Konstruktoren**, leere Liste und Voranstellen, konstruiert werden kann.
- ▶ Es schließt insbesondere aus, daß Listen unendlich lange sein könnten.

# Listen-Rekursion

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

- ▶  $|abcc| = 4$ ,  $|abc| = 3$ .
- ▶ Wenn  $abcc = abc$ , dann ist  $4 = 3$ .
- ▶ Dann wäre die Länge nicht wohldefiniert.
- ▶ Listen sollten nur dann gleich sein, wenn sie gleich konstruiert wurden.
- ▶ Dies wird durch das folgende **Rekursionsprinzip** garantiert:  
Für jede Menge  $X$ , jedes  $x \in X$  und jedes  
 $\varphi : \Sigma \rightarrow \Sigma^* \rightarrow X \rightarrow X$  definiert das Gleichungssystem

$$\begin{aligned}
 f(\epsilon) &= x && (f\text{-nil}), \\
 \forall_{\alpha \in \Sigma} \forall_{u \in \Sigma^*} f(\alpha u) &= \alpha \varphi_u f(u) && (f\text{-pre}).
 \end{aligned}$$

genau eine Funktion  $f : \Sigma^* \rightarrow X$ .

- ▶  $f(u) = u \overleftarrow{\varphi} x$ .
- ▶  $f(abc) = abc \overleftarrow{\varphi} x = a \underset{bc}{\varphi} b \underset{c}{\varphi} c \underset{\epsilon}{\varphi} x$ .

# Append – Hinten anhängen

## DEFINITION

Analog zum Voranstellen definieren wir eine Funktion die ein Element hinten anhängt, z.B.  $(abcd)b = abcd b$ . Sie sollte erfüllen:

$$\forall_{\beta \in \Sigma} \quad \epsilon\beta = \beta\epsilon$$

(app-nil),

$$\forall_{\beta \in \Sigma} \forall_{\alpha \in \Sigma} \forall_{u \in \Sigma^*} \quad (\alpha u)\beta = \alpha(u\beta)$$

(app-pre).

## BEMERKUNG

In  $\alpha u\beta$  brauchen wir keine Klammern.

## SATZ

Es gilt  $\forall_{\beta \in \Sigma} \forall_{w \in \Sigma^*} |w\beta| = 1 + |w|$ .

## BEWEIS.

- ▶ Es sei  $\beta \in \Sigma$ .
- ▶ Wir müssen zeigen:
 
$$\forall_{w \in \Sigma^*} |w\beta| = 1 + |w|.$$
- ▶ Dies erledigen wir gemäß dem Induktionsprinzip:
  - ▶ Wir zeigen, daß  $\epsilon$  diese Eigenschaft hat.
  - ▶ Und wir zeigen, daß wenn ein  $u \in \Sigma^*$  diese Eigenschaft hat, dann auch jedes  $\alpha u$ .



# Induktionsbeweis: $\forall_{w \in \Sigma^*} |w\beta| = 1 + |w|$

## ► Induktionsanfang:

Wir zeigen  $|\epsilon\beta| = 1 + |\epsilon|$ :

$$\begin{aligned} |\epsilon\beta| &= |\beta\epsilon| && \text{(app-nil)} \\ &= 1 + |\epsilon| && \text{(len-pre)}. \end{aligned}$$

## ► Induktionsschritt:

Wir nehmen nun für ein  $u \in \Sigma^*$  die **Induktionshypothese**  $|u\beta| = 1 + |u|$  an.

Für jedes  $\alpha \in \Sigma$  zeigen wir nun  $|(\alpha u)\beta| = 1 + |\alpha u|$ :

$$\begin{aligned} |(\alpha u)\beta| &= |\alpha(u\beta)| && \text{(app-pre)} \\ &= 1 + |u\beta| && \text{(len-pre)} \\ &= 1 + (1 + |u|) && \text{(IH)} \\ &= 1 + |\alpha u| && \text{(len-pre-R)}. \end{aligned}$$

## reversion - Spiegelung

Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## DEFINITION

Die Spiegelung (*reversion*) $\tilde{\cdot} \in \Sigma^* \rightarrow \Sigma^*$  läßt sich rekursiv definieren durch

$$\tilde{\epsilon} = \epsilon \quad (\text{rev-nil}),$$

$$\widetilde{\alpha u} = \tilde{u} \alpha \quad (\text{rev-pre}).$$

## SATZ

Für  $\alpha \in \Sigma$ ,  $u \in \Sigma^*$  gilt

$$\widetilde{u\beta} = \beta \tilde{u} \quad (\text{rev-app}).$$

## SATZ

 $\tilde{\tilde{u}} = u$  (*rev-inv*).

## SATZ

 $|\tilde{u}| = |u|$  (*len-rev*).

## BEWEIS.

► IA: Zu zeigen  $\widetilde{\epsilon\beta} = \beta\tilde{\epsilon}$ :

$$\widetilde{\epsilon\beta} = \tilde{\beta}\epsilon \quad (\text{app-nil})$$

$$= \tilde{\epsilon}\beta \quad (\text{rev-pre})$$

$$= \epsilon\beta \quad (\text{rev-nil})$$

$$= \beta\epsilon \quad (\text{app-nil})$$

$$= \beta\tilde{\epsilon} \quad (\text{rev-nil-R});$$

► IS mit IH:  $\widetilde{u\beta} = \beta\tilde{u}$ ;Zu zeigen:  $\widetilde{(\alpha u)\beta} = \beta\tilde{(\alpha u)}$ :

$$\widetilde{(\alpha u)\beta} = \widetilde{\alpha(u\beta)} \quad (\text{app-pre})$$

$$= \widetilde{u\beta}\alpha \quad (\text{rev-pre})$$

$$= (\beta\tilde{u})\alpha \quad (\text{IH})$$

$$= \beta(\tilde{u}\alpha) \quad (\text{app-pre})$$

$$= \beta\tilde{\alpha u} \quad (\text{rev-pre-R}).$$

## Spiegelung und Allaussagen

Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## SATZ

Sei  $P$  eine Eigenschaft für Listen. Dann gilt

$$\forall_{u \in \Sigma^*} P(u) \iff \forall_{u \in \Sigma^*} P(\tilde{u}).$$

## BEWEIS.

- ▶ Von links nach rechts:

Annahme:  $\forall_{u \in \Sigma^*} P(u)$ ; zu zeigen:  $\forall_{u \in \Sigma^*} P(\tilde{u})$ .

Weitere Annahme:  $u \in \Sigma^*$ ; nur noch zu zeigen:  $P(\tilde{u})$ .

Es gilt  $\tilde{u} \in \Sigma^*$ ; wir wenden daher die erste Annahme auf  $\tilde{u}$  an und erhalten somit  $P(\tilde{u})$ , was zu zeigen war.

- ▶ Von rechts nach links:

Annahme:  $\forall_{u \in \Sigma^*} P(\tilde{u})$ ; zu zeigen:  $\forall_{u \in \Sigma^*} P(u)$ .

Weitere Annahme:  $u \in \Sigma^*$ ; nur noch zu zeigen:  $P(u)$ .

Es gilt  $\tilde{u} \in \Sigma^*$ ; wir wenden daher die erste Annahme auf  $\tilde{u}$  an und erhalten somit  $P(\tilde{u})$ .

Da  $\tilde{\tilde{u}} = u$ , ist auch  $P(\tilde{u}) \iff P(u)$ , insbesondere somit  $P(\tilde{u}) \implies P(u)$ , woraus unser Beweisziel unmittelbar folgt.



## Induktion von hinten

## SATZ

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

$$\frac{P(\epsilon) \quad \forall_{\beta \in \Sigma} \forall_{w \in \Sigma^*} P(w) \implies P(w\beta)}{\forall_{w \in \Sigma^*} P(w)}$$

## BEWEIS.

Wir setzen die beiden Prämissen voraus. Statt der Konklusion zeigen wir die dazu äquivalente Aussage  $\forall_{w \in \Sigma^*} P(\tilde{w})$ . Mit der Eigenschaft

$$P^\sim(w) : \iff P(\tilde{w})$$

läßt sich dies als  $\forall_{w \in \Sigma^*} P^\sim(w)$  schreiben und mit Induktion beweisen.

IA: Wir zeigen  $P^\sim(\epsilon)$ . Gemäß obiger Definition,  $P^\sim(\epsilon) \iff P(\tilde{\epsilon})$ , und wegen  $\tilde{\epsilon} = \epsilon$ , auch  $P(\tilde{\epsilon}) \iff P(\epsilon)$ . Letzteres ist in der Voraussetzung.

IS mit IH:  $P^\sim(u)$  (äquivalent zu  $P(\tilde{u})$ ). Zu zeigen haben wir  $P^\sim(\alpha u)$ , das ist  $P(\tilde{\alpha u})$ , also  $P(\tilde{u}\alpha)$ . Die Voraussetzung liefert (wir setzen  $\tilde{u}$  für  $w$  ein):  $P(\tilde{u}) \implies P(\tilde{u}\alpha)$ . Mit IH sind wir fertig.  $\square$

Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## 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:

$$\frac{}{\text{PD}(\epsilon)} \quad \frac{\alpha \in \Sigma}{\text{PD}(\alpha)} \quad \frac{u \in \Sigma^* \quad \text{PD}(u) \quad \alpha \in \Sigma}{\text{PD}(\alpha u \alpha)}$$

Die zugehörige Eliminationsregel lautet dann

$$\frac{P(\epsilon) \quad \forall_{\alpha \in \Sigma} P(\alpha) \quad \forall_{\substack{u \in \Sigma^* \\ P(u)}} \forall_{\alpha \in \Sigma} P(\alpha u \alpha)}{\forall_{\substack{u \in \Sigma^* \\ \text{PD}(u)}} P(u)}$$

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

## SATZ

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

- ▶ Wir definieren die **Verkettung** von Listen, eine binäre Verknüpfung von Listen, welche die Elemente einfach aneinanderreihet, z.B.  $(ab)(cbb) = abcbb$ .
- ▶ Sie sollte für alle  $u, v \in \Sigma^*$  und  $\alpha \in \Sigma$  diese beiden Gleichungen erfüllen:

$$\epsilon v = v \quad (\text{cat-nil}),$$

$$(\alpha u)v = \alpha(uv) \quad (\text{cat-pre}).$$

- ▶ Gemäß Rekursionsprinzip gibt es genau eine Funktion mit diesen Eigenschaften.

## BEISPIEL

$$\begin{aligned} (abc)(cbb) &= (a(b(c\epsilon)))(c(b(b\epsilon))) = a((b(c\epsilon))(c(b(b\epsilon)))) = \\ &a(b((c\epsilon)(c(b(b\epsilon))))) = a(b(c(\epsilon(c(b(b\epsilon))))) = a(b(c(c(b(b\epsilon)))) = \\ &abcbb. \end{aligned}$$

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## SATZ

Die Listenverkettung ist assoziativ, d.h.

$$\forall u, v, w \in \Sigma^* \quad (uv)w = u(vw).$$

## BEWEIS.

- ▶ Es seien  $v, w \in \Sigma^*$ .
- ▶ Wir müssen zeigen, daß

$$\forall u \in \Sigma^* \quad (uv)w = u(vw).$$

- ▶ Dies erledigen wir gemäß dem Induktionsprinzip:
  - ▶ Wir zeigen, daß  $\epsilon$  diese Eigenschaft hat.
  - ▶ Und wir zeigen, daß wenn ein  $u \in \Sigma^*$  diese Eigenschaft hat, dann auch jedes  $\alpha u$ .



# Induktionsbeweis: $\forall_{u \in \Sigma^*} (uv)w = u(vw)$

► **Induktionsanfang:**

Wir zeigen  $(\epsilon v)w = \epsilon(vw)$ :

$$\begin{aligned} (\epsilon v)w &= vw && \text{(cat-nil)} \\ &= \epsilon(vw) && \text{(cat-nil-R)}. \end{aligned}$$

► **Induktionsschritt:**

Wir nehmen nun für ein  $u \in \Sigma^*$  die **Induktionshypothese**  $(uv)w = u(vw)$  an.

Für jedes  $\alpha \in \Sigma$  zeigen wir nun  $((\alpha u)v)w = (\alpha u)(vw)$ :

$$\begin{aligned} ((\alpha u)v)w &= (\alpha(uv))w && \text{(cat-pre)} \\ &= \alpha((uv)w) && \text{(cat-pre)} \\ &= \alpha(u(vw)) && \text{(IH)} \\ &= (\alpha u)(vw) && \text{(cat-pre-R)}. \end{aligned}$$

# $\epsilon$ ist ein neutrales Element

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## LEMMA

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

$$\forall u \in \Sigma^* \quad \epsilon u = u = u \epsilon.$$

## BEWEIS.

Der Teil  $\epsilon u = u$  ist Bestandteil der Definition. Wir zeigen noch  $u \epsilon = u$  mittels Listeninduktion:

$$\begin{aligned} \epsilon \epsilon &= \epsilon && \text{(cat-nil);} \\ (\alpha u) \epsilon &= \alpha(u \epsilon) && \text{(cat-pre)} \\ &= \alpha u && \text{(IH).} \end{aligned}$$



## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## LEMMA

*Die Länge ist ein Homomorphismus in die Halbgruppe der natürlichen Zahlen mit Addition, d.h.:*

$$|uv| = |u| + |v|$$

## SATZ

*Die Spiegelung ist ein Anti-Homomorphismus:  $\widetilde{uv} = \widetilde{v}\widetilde{u}$ .*

## Listen

Länge

Induktion

Rekursion

Anhängen

Spiegeln

Verkettung

Assoziativität

Neutrales Element

Homomorphismus

Kürzungsregeln

## SATZ

Für Listen  $u, v, w \in \Sigma^*$  gelten die Kürzungsregeln:

$$uv = uw \implies v = w,$$

$$uw = vw \implies u = v.$$

## BEWEIS.

- ▶ IA: Es sei  $\epsilon v = \epsilon w$ , dann ist  $v = w$  (cat-nil);
- ▶ IS mit IH  $uv = uw \implies v = w$ :

Zu zeigen:  $(\alpha u)v = (\alpha u)w \implies v = w$

Annahme:  $(\alpha u)v = (\alpha u)w$ ;

Nur noch zu zeigen  $v = w$ .

Mit (cat-pre) erhalten wir aus der Annahme  $\alpha(uv) = \alpha(uw)$ . Da Listen nur dann gleich sind, wenn sie gleich konstruiert sind, ist somit  $uv = uw$ . Die (IH) liefert nun  $u = w$ .

