Listen
Länge
Induktion
Rekursion
Anhängen
Spiegeln
Verkettung
Assoziativität
Neutrales Element
Homomorphismus
Kürzungsregeln

Formale Grundlagen
für WIN

2010S

Franz Binder
Institut für Algebra
Johannes Kepler Universität Linz

Vorlesung im 2010S

http://www.algebra.uni-linz.ac.at/Students/Win/fg

Inhalt

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

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

Beispiel

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

Notation

Rechts-assoziative Klammerung: $\alpha \lhd \beta \lhd u = \alpha \lhd (\beta \lhd u)$.
Die Zeichen $\rhd$ und ϵ sind oft entbehrlich.

Beispiel

$\epsilon,\ a \lhd \epsilon,\ a \lhd b \lhd \epsilon,\ a \lhd b \lhd c \lhd d \lhd \epsilon,\ d \lhd c \lhd \epsilon,\ b \lhd c \lhd c \lhd a \lhd b \lhd \epsilon.$
Oder einfacher: $\epsilon,\ a,\ ab,\ abcd,\ dc,\ bccab.$

Länge einer Liste

- 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:

$$\begin{aligned} |\epsilon| &= 0 && \text{(len-nil)}, \\ \underset{\alpha \in \Sigma}{\forall} \underset{u \in \Sigma^*}{\forall} \quad |\alpha \triangleleft u| &= 1 + |u| && \text{(len-pre)}. \end{aligned}$$

- Frage: Ist die Länge damit ordentlich definiert?
- Beispiel:
$|a \triangleleft b \triangleleft c \triangleleft c \triangleleft \epsilon| = |a \triangleleft (b \triangleleft (c \triangleleft (c \triangleleft \epsilon)))| = 1 + |b \triangleleft (c \triangleleft (c \triangleleft \epsilon))| =$
$1 + (1 + |c \triangleleft (c \triangleleft \epsilon)|) = 1 + (1 + (1 + |c \triangleleft \epsilon|)) =$
$1 + (1 + (1 + (1 + |\epsilon|))) = 1 + (1 + (1 + (1 + 0))) = 4$

Listen
Länge
Induktion
Rekursion
Anhängen
Spiegeln
Verkettung
Assoziativität
Neutrales Element
Homomorphismus
Kürzungsregeln

Listen-Induktion

- Die Länge ist für die leere Liste definiert.
- Ist die Länge für eine Liste u definiert, dann auch für jede Liste der Form $\alpha \lhd u$.
- 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:

$$\frac{P(\epsilon) \qquad \underset{a \epsilon \Sigma}{\forall} \; \underset{u \epsilon \Sigma^*}{\forall} \; P(u) \implies P(a \lhd u)}{\underset{w \epsilon \Sigma^*}{\forall} \; P(w)} \text{ List } \mathcal{E}$$

  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

- $|a \triangleleft b \triangleleft c \triangleleft c \triangleleft \epsilon| = 4$, $|a \triangleleft b \triangleleft c \triangleleft \epsilon| = 3$.
- Wenn $a \triangleleft b \triangleleft c \triangleleft c \triangleleft \epsilon = a \triangleleft b \triangleleft c \triangleleft \epsilon$, 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 $\varphi : \Sigma \to \Sigma^* \to X \to X$ definiert das Gleichungssystem

$$\begin{aligned} & f(\epsilon) = x && (f\text{-nil}), \\ \underset{\alpha\in\Sigma}{\forall}\ \underset{u\in\Sigma^*}{\forall}\quad & f(\alpha \triangleleft u) = \alpha \underset{u}{\varphi} f(u) && (f\text{-pre}). \end{aligned}$$

  genau eine Funktion $f : \Sigma^* \to X$.
- $f(u) = u \overleftarrow{\varphi} x$.
- $f(a \triangleleft b \triangleleft c \triangleleft \epsilon) = (a \triangleleft b \triangleleft c \triangleleft \epsilon) \overleftarrow{\varphi} x = a \underset{b\triangleleft c\triangleleft\epsilon}{\varphi} b \underset{c\triangleleft\epsilon}{\varphi} c \underset{\epsilon}{\varphi} x$.

Listen
- Länge
- Induktion
- Rekursion
- Anhängen
- Spiegeln
- Verkettung
- Assoziativität
- Neutrales Element
- Homomorphismus
- Kürzungsregeln

Append – Hinten anhängen

Definition

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

$$\forall_{\beta \epsilon \Sigma} \quad \epsilon \triangleright \beta = \beta \triangleleft \epsilon \qquad \text{(app-nil)},$$

$$\forall_{\beta \epsilon \Sigma} \forall_{\alpha \epsilon \Sigma} \forall_{u \epsilon \Sigma^*} \quad (\alpha \triangleleft u) \triangleright \beta = \alpha \triangleleft (u \triangleright \beta) \qquad \text{(app-pre)}.$$

Bemerkung

In $\alpha \triangleleft u \triangleright \beta$ brauchen wir keine Klammern. Man kann es auch einfach als $\alpha u \beta$ schreiben.

Satz

Es gilt

$$\forall_{\beta \epsilon \Sigma} \forall_{w \epsilon \Sigma^*} \quad |w \triangleright \beta| = 1 + |w| \,.$$

Beweis.

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

$\square$

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

- Induktionsanfang:
  Wir zeigen $|\epsilon \triangleright \beta| = 1 + |\epsilon|$:

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

- Induktionsschritt:
  Wir nehmen nun für ein $u \in \Sigma^*$ die Induktionshypothese $|u \triangleright \beta| = 1 + |u|$ an.
  Für jedes $\alpha \in \Sigma$ zeigen wir nun $|(\alpha \triangleleft u) \triangleright \beta| = 1 + |\alpha \triangleleft u|$:

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

reversion - Spiegelung

Definition

Die Spiegelung (*reversion*) $\widetilde{\ }\ \epsilon\ \Sigma^* \to \Sigma^*$ läßt sich rekursiv definieren durch

$$\widetilde{\epsilon} = \epsilon \qquad \text{(rev-nil)},$$
$$\widetilde{\alpha \triangleleft u} = \widetilde{u} \triangleright \alpha \qquad \text{(rev-pre)}.$$

Satz

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

$$\widetilde{u \triangleright \beta} = \beta \triangleleft \widetilde{u} \qquad \textit{(rev-app)}.$$

Satz

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

Satz

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

Beweis.

- IA: Zu zeigen $\widetilde{\epsilon \triangleright \beta} = \beta \triangleleft \widetilde{\epsilon}$:

$$\begin{aligned}
\widetilde{\epsilon \triangleright \beta} &= \widetilde{\beta \triangleleft \epsilon} && \text{(app-nil)}\\
&= \widetilde{\epsilon} \triangleright \beta && \text{(rev-pre)}\\
&= \epsilon \triangleright \beta && \text{(rev-nil)}\\
&= \beta \triangleleft \epsilon && \text{(app-nil)}\\
&= \beta \triangleleft \widetilde{\epsilon} && \text{(rev-nil-R)};
\end{aligned}$$

- IS mit IH: $\widetilde{u \triangleright \beta} = \beta \triangleleft \widetilde{u}$;

  Zu zeigen: $\widetilde{(\alpha \triangleleft u) \triangleright \beta} = \beta \triangleleft \widetilde{\alpha \triangleleft u}$:

$$\begin{aligned}
\widetilde{(\alpha \triangleleft u) \triangleright \beta} &= \widetilde{\alpha \triangleleft (u \triangleright \beta)} && \text{(app-pre)}\\
&= \widetilde{u \triangleright \beta} \triangleright \alpha && \text{(rev-pre)}\\
&= (\beta \triangleleft \widetilde{u}) \triangleright \alpha && \text{(IH)}\\
&= \beta \triangleleft (\widetilde{u} \triangleright \alpha) && \text{(app-pre)}\\
&= \beta \triangleleft \widetilde{\alpha \triangleleft u} && \text{(rev-pre-R)}
\end{aligned}$$

Spiegelung und Allaussagen

Satz

Sei P eine Eigenschaft für Listen. Dann gilt

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

Beweis.

- Von links nach rechts:
  Annahme: $\forall_{u \in \Sigma^*} P(u)$; zu zeigen: $\forall_{u \in \Sigma^*} P(\widetilde{u})$.
  Weitere Anname: $u \in \Sigma^*$; nur noch zu zeigen: $P(\widetilde{u})$.
  Es gilt $\widetilde{u} \in \Sigma^*$; wir wenden daher die erste Annahme auf $\widetilde{u}$ an und erhalten somit $P(\widetilde{u})$, was zu zeigen war.
- Von rechts nach links:
  Annahme: $\forall_{u \in \Sigma^*} P(\widetilde{u})$; zu zeigen: $\forall_{u \in \Sigma^*} P(u)$.
  Weitere Annahme: $u \in \Sigma^*$; nur noch zu zeigen: $P(u)$.
  Es gilt $\widetilde{u} \in \Sigma^*$; wir wenden daher die erste Annahme auf $\widetilde{u}$ an und erhalten somit $P(\widetilde{\widetilde{u}})$.
  Da $\widetilde{\widetilde{u}} = u$, ist auch $P(\widetilde{\widetilde{u}}) \iff P(u)$, insbesondere somit $P(\widetilde{\widetilde{u}}) \implies P(u)$, woraus unser Beweisziel unmittelbar folgt.

$\square$

Listen
Länge
Induktion
Rekursion
Anhängen
Spiegeln
Verkettung
Assoziativität
Neutrales Element
Homomorphismus
Kürzungsregeln

Induktion von hinten

Satz

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

$$\frac{P(\epsilon) \qquad \underset{\beta\in\Sigma}{\forall}\ \underset{w\in\Sigma^*}{\forall}\ P(w) \implies P(w \triangleright \beta)}{\underset{w\in\Sigma^*}{\forall}\ P(w)}$$

Beweis.

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

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

läßt sich dies als $\underset{w\in\Sigma^*}{\forall} P^{\sim}(w)$ schreiben und mit Induktion beweisen.
IA: Wir zeigen $P^{\sim}(\epsilon)$. Gemäß obiger Definition, $P^{\sim}(\epsilon) \iff P(\widetilde{\epsilon})$, und wegen $\widetilde{\epsilon} = \epsilon$, auch $P(\widetilde{\epsilon}) \iff P(\epsilon)$. Letzteres ist in der Voraussetzung.
IS mit IH: $P^{\sim}(u)$ (äquivalent zu $P(\widetilde{u})$). Zu zeigen haben wir $P^{\sim}(\alpha \triangleleft u)$, das ist $P(\widetilde{\alpha \triangleleft u})$, also $P(\widetilde{u} \triangleright \alpha)$. Die Voraussetzung liefert (wir setzen $\widetilde{u}$ für w ein): $P(\widetilde{u}) \implies P(\widetilde{u} \triangleright \alpha)$. Mit IH sind wir fertig. $\square$

Listen
- Länge
- Induktion
- Rekursion
- Anhängen
- Spiegeln
- Verkettung
- Assoziativität
- Neutrales Element
- Homomorphismus
- Kürzungsregeln

Palindrom

Definition

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

$$\frac{}{\mathrm{PD}(\epsilon)} \qquad \frac{\alpha \in \Sigma}{\mathrm{PD}(\alpha)} \qquad \frac{u \in \Sigma^* \quad \mathrm{PD}(u) \quad \alpha \in \Sigma}{\mathrm{PD}(\alpha \triangleleft u \triangleright \alpha)}$$

Die zugehörige Eliminationsregel lautet dann

$$\frac{P(\epsilon) \qquad \forall_{\alpha \in \Sigma} P(\alpha) \qquad \forall_{\substack{u \in \Sigma^* \\ P(u)}} \forall_{\alpha \in \Sigma} P(\alpha \triangleleft u \triangleright \alpha)}{\forall_{\substack{u \in \Sigma^* \\ \mathrm{PD}(u)}} P(u)}$$

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

Satz

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

Listen
Länge
Induktion
Rekursion
Anhängen
Spiegeln
Verkettung
Assoziativität
Neutrales Element
Homomorphismus
Kürzungsregeln

Verkettung

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

$$
\begin{aligned}
\epsilon \diamond v &= v && \text{(cat-nil)},\\
(\alpha \triangleleft u) \diamond v &= \alpha \triangleleft (u \diamond v) && \text{(cat-pre)}.
\end{aligned}
$$

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

Beispiel

$abc \diamond cbb = \big(a \triangleleft (b \triangleleft (c \triangleleft \epsilon))\big) \diamond \big(c \triangleleft (b \triangleleft (b \triangleleft \epsilon))\big) =$
$a \triangleleft (\big(b \triangleleft (c \triangleleft \epsilon)\big) \diamond \big(c \triangleleft (b \triangleleft (b \triangleleft \epsilon))\big)) = a \triangleleft (b \triangleleft (\big(c \triangleleft \epsilon\big) \diamond \big(c \triangleleft (b \triangleleft (b \triangleleft \epsilon))\big))) =$
$a \triangleleft (b \triangleleft (c \triangleleft (\epsilon \diamond \big(c \triangleleft (b \triangleleft (b \triangleleft \epsilon))\big)))) = a \triangleleft (b \triangleleft (c \triangleleft (c \triangleleft (b \triangleleft (b \triangleleft \epsilon))))) = abccbb.$

Listen
- Länge
- Induktion
- Rekursion
- Anhängen
- Spiegeln
- Verkettung
- Assoziativität
- Neutrales Element
- Homomorphismus
- Kürzungsregeln

Assoziativität

Satz

Die Listenverkettung ist assoziativ, d.h.

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

Beweis.

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

$$\forall_{u \epsilon \Sigma^*} (u \diamond v) \diamond w = u \diamond (v \diamond w).$$

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

$\square$

Listen
- Länge
- Induktion
- Rekursion
- Anhängen
- Spiegeln
- Verkettung
- Assoziativität
- Neutrales Element
- Homomorphismus
- Kürzungsregeln

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

- Induktionsanfang:
  Wir zeigen $(\epsilon \diamond v) \diamond w = \epsilon \diamond (v \diamond w)$:

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

- Induktionsschritt:
  Wir nehmen nun für ein $u \in \Sigma^*$ die Induktionshypothese $(u \diamond v) \diamond w = u \diamond (v \diamond w)$ an.
  Für jedes $\alpha \in \Sigma$ zeigen wir nun $((\alpha \lhd u) \diamond v) \diamond w = (\alpha \lhd u) \diamond (v \diamond w)$:

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

ϵ ist ein neutrales Element

Lemma

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

$$\underset{u \in \Sigma^*}{\forall} \epsilon \diamond u = u = u \diamond \epsilon.$$

Beweis.

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

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

$\square$

Länge als Homomorphismus

Lemma

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

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

Satz

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

Listen
Länge
Induktion
Rekursion
Anhängen
Spiegeln
Verkettung
Assoziativität
Neutrales Element
Homomorphismus
Kürzungsregeln

Kürzungsregeln

Satz

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

$$u \diamond v = u \diamond w \implies v = w,$$
$$u \diamond w = v \diamond w \implies u = v.$$

Beweis.

- IA: Es sei $\epsilon v = \epsilon w$, dann ist $v = w$ (cat-nil);
- IS mit IH $u \diamond v = u \diamond w \implies v = w$:
  Zu zeigen: $(\alpha \lhd u) \diamond v = (\alpha \lhd u) \diamond w \implies v = w$
  Annahme: $(\alpha \lhd u) \diamond v = (\alpha \lhd u) \diamond w$;
  Nur noch zu zeigen $v = w$.
  Mit (cat-pre) erhalten wir aus der Annahme
  $\alpha \lhd (u \diamond v) = \alpha \lhd (u \diamond w)$. Da Listen nur dann gleich sind, wenn sie gleich konstruiert sind, ist somit $u \diamond v = u \diamond w$. Die (IH) liefert nun $u = w$.

$\square$