1 Formale Sprachen

1.1 Wörter

- Typ der *Zeichenketten* oder *Wörter* in *Haskell*

  $$\texttt{type String} = [\texttt{Char}]$$

- Konkatenation (Hintereinanderhängen): ++
- Wort aus Zeichen $a_1, \ldots, a_k$:

  $$"a_1 \cdots a_k"$$

- Leeres Wort: ""

Ziel dieses Kapitels:

- präzise Beschreibung gewisser Mengen von Wörtern
- z.B. alle zulässigen Bezeichner oder Programme einer Programmiersprache
- oder Ausschnitte aus natürlichen Sprachen
- spätere Kapitel: Maschinen (Automaten), die
  - die Zulässigkeit prüfen
  - Struktur erkennen

    (z.B. Programme in Teilausdrücke zerlegen)
  - Übersetzungen vornehmen

Terminologie und Notation beim Argumentieren über Wörter (nicht wenn wir programmieren):

- die Menge $[A]$ aller Wörter über Zeichenvorrat A bezeichnet man auch mit A^*
- Die Menge aller nichtleeren Wörter über A ist

$$A^+ \stackrel{\text{def}}{=} A^* \backslash \{\varepsilon\}$$

- Anführungszeichen um Wörter werden weggelassen:

  $a_1 \cdots a_k$ statt "$a_1 \cdots a_k$"

- damit würde aber das leere Wort unsichtbar
- daher ε statt ""

- das Konkatenationszeichen wird weggelassen:

  uv statt $u{+}{+}v$

- konform mit obiger Schreibweise:

  $a_1 \cdots a_n$ statt "a_1" ++ $\cdots$ ++ "a_k"

- damit werden einzelne Zeichen und Worte mit nur einem Zeichen notationell nicht mehr unterschieden (im Gegensatz zu *Haskell*, wo x $\neq$ [x]!)

Erzeugungsprinzip: Alle Wörter können erzeugt werden

- ausgehend vom leeren Wort ε
- durch fortgesetztes Vornanfügen von Zeichen
  (: in *Haskell*)

Viele Definitionen und Beweise laufen über diesen induktiven Aufbau der Wörter.

Beispiel 1.1 induktive Definition der Konkatenation:

$$\varepsilon u \stackrel{\text{def}}{=} u \qquad (a v) w \stackrel{\text{def}}{=} a(vw)$$

```
[] ++ u   =  u
(a:v) ++ w  =  a:(v ++ w)
```

$(a \in A)$

Eigenschaften (Beweis durch Induktion über den Aufbau):

- die Konkatenation ist assoziativ:

$$u(vw) = (uv)w$$

- das leere Wort ist neutrales Element:

$$\varepsilon u = u = u\varepsilon$$

Diese beiden Eigenschaften sind so häufig, daß man einen eigenen Begriff dafür einführt.

1.2 Monoide

Ein *Monoid* ist ein Tripel $(M, \cdot, 1)$, wobei

- $\cdot : M \times M \to M$ innere Verknüpfung,
- die assoziativ ist
$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$
- und 1 als neutrales Element hat
$$1 \cdot x = x = x \cdot 1$$

Beispiel 1.2

- $(A^*, ++, \varepsilon)$
- $(\mathbb{N}, \cdot, 1)$
- $(\mathbb{N}, +, 0)$
- $(\mathbb{B}, \wedge, True)$
- $(\mathbb{B}, \vee, False)$

In einem Monoid kann man natürlichzahlige Potenzen definieren:

- $x^0 \stackrel{\text{def}}{=} 1$
- $x^{n+1} \stackrel{\text{def}}{=} x \cdot x^n$

Es gelten die üblichen Rechenregeln für Potenzen:

- $x^1 = x$
- $x^{m+n} = x^m \cdot x^n$
- $(x^m)^n = x^{m \cdot n}$

Beispiel 1.3

$$O^3L^5 = OOOLLLLL$$

□

Negative Potenzen sind in allgemeinen Monoiden nicht sinnvoll, sondern nur in Monoiden, die Gruppen sind.
Um beim Rechnen nicht ins "Negative" abzurutschen, verwendet man die *abgeschnittene Subtraktion*:

$$m \dot{-} n \stackrel{\text{def}}{=} \max\{0, m - n\}$$

1.3 Zeichenhäufigkeit und Länge

- Sei $B \subseteq A$ ein Teilzeichenvorrat und $u \in A^*$.
- $|u|_B$: Anzahl der Zeichen von u, die in B liegen

$$|\varepsilon|_B \stackrel{\text{def}}{=} 0$$

$$|av|_B \stackrel{\text{def}}{=} \begin{cases} 1 + |v|_B & \text{wenn } a \in B \\ |v|_B & \text{sonst} \end{cases}$$

(Definition durch Induktion über den Aufbau von u)

- Als *Länge* erhält man dann $|u| \stackrel{\text{def}}{=} |u|_A$

Beispiel 1.4

- Wörter mit gleich vielen O und L: $|u|_O = |u|_L$
- Wörter gerader Länge: $|u| \bmod 2 = 0$
- Beschreibung von Kommunikation über einen Pufferspeicher:

e	Eingabe in den Puffer
a	Ausgabe auf Kanal A
b	Ausgabe auf Kanal B

$u \in \{a, b, e\}^*$ beschreibt korrekte Ein-/Ausgabefolgen gdw. für alle Anfangsstücke v von u gilt $|v|_a + |v|_b \leq |v|_e$ $\square$

Lemma 1.5 *Es gilt* $|uv|_B = |u|_B + |v|_B$

Beweis: Durch Induktion über den Aufbau von u.

$u = \varepsilon$:

$$
\begin{aligned}
& |uv|_B \\
= & \quad \{\!\!\{ \text{ Annahme } \}\!\!\} \\
& |\varepsilon v|_B \\
= & \quad \{\!\!\{ \text{ Neutralität von } \varepsilon \}\!\!\} \\
& |v|_B \\
= & \quad \{\!\!\{ \text{ Neutralität von } 0 \}\!\!\} \\
& 0 + |v|_B \\
= & \quad \{\!\!\{ \text{ Definition von } |\,| \}\!\!\} \\
& |\varepsilon|_B + |v|_B \\
= & \quad \{\!\!\{ \text{ Annahme } \}\!\!\} \\
& |u|_B + |v|_B
\end{aligned}
$$

$u = aw$ mit $a \in A$ und $w \in A^*$:

$$
\begin{aligned}
& |uv|_B \\
= \quad & \{\!\!\{ \text{ Annahme } \}\!\!\} \\
& |(aw)v|_B \\
= \quad & \{\!\!\{ \text{ Definition der Konkatenation } \}\!\!\} \\
& |a(wv)|_B \\
= \quad & \{\!\!\{ \text{ Definition von } || \ \}\!\!\} \\
& \begin{cases} 1 + |wv|_B & \text{wenn } a \in B \\ |wv|_B & \text{sonst} \end{cases} \\
= \quad & \{\!\!\{ \text{ Induktionsvoraussetzung } \}\!\!\} \\
& \begin{cases} 1 + |w|_B + |v|_B & \text{wenn } a \in B \\ |w|_B + |v|_B & \text{sonst} \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
&= \quad \{\![\text{ Ausklammern }]\!\} \\
&\quad \left\{ \begin{array}{ll} 1 + |w|_b & \text{wenn } a \in B \\ |w|_B & \text{sonst} \end{array} \right\} + |v|_B \\
&= \quad \{\![\text{ Definition von } \| \,]\!\} \\
&\quad |aw|_B + |v|_B \\
&= \quad \{\![\text{ Annahme }]\!\} \\
&\quad |u|_B + |v|_B
\end{aligned}
$$

□

1.4 Formale Sprachen

Eine *(formale) Sprache* über A ist eine Teilmenge von A^*.

Beispiel 1.6

Sprachen über $A = \{O, L\}$

- leere Sprache $\emptyset$
- die Sprache, die nur aus dem leeren Wort besteht: $\{\varepsilon\}$
- die Sprache aller Wörter mit genau 3 Zeichen :

  $\{OOO, OOL, OLO, OLL, LOO, LOL, LLO, LLL\}$

Notation:

- Eine Sprache, die nur ein einziges Wort u enthält, müßte strenggenommen als $\{u\}$ geschrieben werden.
- Da solche Sprachen aber sehr häufig vorkommen, wird das lästig.
- Man läßt daher in diesem Fall die Mengenklammern weg.
- Z.B. schreibt man lediglich ε für die Sprache, die genau das leere Wort enthält.
- Auch einzelne Zeichen können so als Sprachen aufgefaßt werden: Für $a \in A$ ist dann a die Sprache, die genau das Wort a der Länge 1 enthält.
- Dies scheint zuerst verwirrend. Aus dem Kontext wird aber stets klar werden, was gemeint ist.

1.5 Konkatenation von Sprachen

Zu einer Menge M sei $\wp(M)$ die Potenzmenge von M , d.h. die Menge aller Teilmengen von M. Damit ist $\wp(A^*)$ die Menge aller Sprachen über A.

Seien $S, T \in \wp(A^*)$ Sprachen. Ihre *Konkatenation* ST ("S gefolgt von T") wird punktweise erklärt:

$$ST = \{st : s \in S \wedge t \in T\}$$

Beispiel 1.7 Sprache D_5 der durch 5 teilbaren Dezimalzahlen (mit führenden Nullen)

$$
\begin{aligned}
A &= \{0, 1, ..., 9\} \\
A^* &= \text{Menge aller Dezimalzahlen} \\
D_5 &= A^* \{0, 5\}
\end{aligned}
$$

Gesetze:

- $S(TU) = (ST)U$ (Assoziativität)
- $\varepsilon S = S = S\varepsilon$ (Neutralität)

Insbesondere ist $(\wp(A^*), ++, \varepsilon)$ wieder ein Monoid.

Beispiel 1.8 Die Sprache aller Wörter mit genau 3 Zeichen aus $A = \{O, L\}$ schreibt sich jetzt kurz als A^3.

Allgemeiner: wird eine Operation punktweise auf Mengen hochgehoben, so vererben sich alle Gesetze, in denen die Elementbezeichner auf linker und rechter Seite je genau einmal vorkommen.

Beispiele von Gesetzen dieser Form: Assoziativität, Neutralität, Kommutativität

Achtung: Konkatenation ist *nicht* kommutativ!

weitere Gesetze:

- $(S \cup T)U = SU \cup TU$ (Linksdistributivität)
- $S(T \cup U) = ST \cup SU$ (Rechtsdistributivität)

Hier wurde folgende *Verabredung* getroffen:
Konkatenation bindet stärker als die Mengenoperationen
(sonst müßte man oben etwa $(SU) \cup (TU)$ schreiben)

Die Konkatenation distributiert sogar über beliebige Vereinigungen:

$$(\bigcup_{S \in \mathcal{S}} S)U = \bigcup_{S \in \mathcal{S}} SU$$

Speziell für $\mathcal{S} = \emptyset$ folgt

$$\emptyset U = \emptyset = S\emptyset \qquad \text{(Striktheit)}$$

Allgemeiner: wird eine Operation punktweise auf Mengen hochgehoben, so ist sie distributiv (und daher auch strikt) in allen Argumenten.

1.6 Induktive Definition von Sprachen

Endliche Sprachen kann man durch Aufzählung definieren (wenn auch nicht unbedingt bequem).

Für unendliche Sprachen braucht man weitere Hilfsmittel.

Wir geben zwei Möglichkeiten zur Definition von Bezeichnern über $A = Bu \cup Zi$ mit

$$Bu = \{a, ..., z\} \qquad Zi = \{0, ..., 9\}$$

- Angabe einer *charakteristischen Eigenschaft*: Ein Bezeichner ist eine nichtleere Folge von Buchstaben und Ziffern, die mit einem Buchstaben beginnt:

$$Idc = \{s \in A^* : |s| > 0 \wedge \text{head}\, s \in Bu\}$$

- induktiver Aufbau über ein *Bildungsgesetz*:

  1. Ist $x \in Bu$, so ist $x \in Idi$.
  2. Ist $s \in Idi$ und $x \in A$, so ist $sx \in Idi$.
  3. Idi ist die (bezüglich Inklusion) kleinste Menge, die 1 und 2 erfüllt ("nichts sonst ist ein Bezeichner").

Wir wollen nun zeigen, daß beide Definitionen äquivalent sind, d.h. daß $Idc = Idi$.

(Idc $\subseteq$ Idi). Sei $s \in$ Idc. Induktion über $|s|$.

- Induktionsanfang: $|s| = 1$, etwa $s = x$ für ein $x \in A$. Es folgt

$$\mathtt{head}\, s = x \in \mathrm{Bu}\ .$$

  Also $s \in$ Idi wegen Bildungsregel 1.

- Induktionsschluß: Sei $|s| = n + 1 \geq 2$. Zerlegung

$$s = tx$$

  mit $t = \mathtt{init}\, s$ und $x = \mathtt{last}\, s$.
  Es gilt $|t| = n \geq 1$ und $\mathtt{head}\, t = \mathtt{head}\, s \in \mathrm{Bu}$.
  Also ist $t \in$ Idc und nach Induktionsvoraussetzung folgt
  $t \in$ Idi.
  Also ist auch

$$s = tx \in \mathrm{Idi}$$

  wegen Bildungsregel 2. $\square$

(Idi $\subseteq$ Idc). Sei $s \in \mathrm{Idi}$. Induktion über den Aufbau von s.

- Induktionsanfang: $s = x$ für ein $x \in \mathrm{Bu}$. Es folgt

  $\mathrm{head}\, s = x \in \mathrm{Bu}$.

  Also $s \in \mathrm{Idc}$.
- Induktionsschluß: Sei $s = tx$ mit $t \in \mathrm{Idi}$.

  Nach Induktionsvoraussetzung gilt $t \in \mathrm{Idc}$ und $\mathrm{head}\, t \in \mathrm{Bu}$.

  Wegen $|t| > 0$ gilt aber $\mathrm{head}\, s = \mathrm{head}\, t \in \mathrm{Bu}$. Außerdem ist

  $$|s| = |t| + 1 > 0 \ .$$

  Insgesamt folgt $s \in \mathrm{Idc}$. $\square$

- Ein Werkzeug zur induktiven Erzeugung, formale Grammatiken, werden wir im nächsten Kapitel studieren.
- Daraus werden sich auch Mechanismen zur Erkennung von Sprachen ergeben.
- Für eine eingeschränkte Klasse von Sprachen, unter die auch die Bezeichner fallen, können wir aber auch algebraische Beschreibungsmittel verwenden.

1.7 Iterationsoperatoren

- Wir geben nochmals eine Charakterisierung von Bezeichnern:
- Ein Bezeichner besteht aus einem Buchstaben gefolgt von beliebig vielen (auch 0) Buchstaben oder Ziffern.
- "Gefolgt von" können wir durch Konkatenation ausdrücken.
- Es fehlt aber ein Operator für "beliebig viele".
- Hier helfen uns die Potenzen: Sei wieder $A = \mathrm{Bu} \cup \mathrm{Zi}$.
- Was ist A^0? Das neutrale Element im Monoid der Sprachen unter Konkatenation, also ε.
- Damit ist $\mathrm{Bu}\ A^0 = \mathrm{Bu}$. Also beschreibt $\mathrm{Bu}\ A^0$ die Teilsprache von Bezeichnern, die aus einem Buchstaben gefolgt von 0 Ziffern bestehen.

- Allgemein beschreibt Bu A^i die Teilsprache von Bezeichnern, die aus einem Buchstaben gefolgt von i Zeichen bestehen.
- Dies übertragen wir auf beliebige Sprachen $U \subseteq A^*$.
- $U^i = \underbrace{U \cdots U}_{i}$ besteht aus allen Wörtern, die durch Konkatenation von i Wörtern aus U entstehen.
- Vereinigen wir über alle Wahlmöglichkeiten für i, so erhalten wir die beliebige (endlichmalige) Iteration

$$U^* \stackrel{\text{def}}{=} \bigcup_{i \geq 0} U^i$$

- Dies erklärt im nachhinein die Notation A^* für die Menge aller endlichen Wörter über A.

Beispiel 1.9

- Unsere Bezeichnersprache läßt sich nun schreiben als

$$\mathrm{Bu}\ A^*$$

- Wir wollen den Ausdruck noch etwas umformen: Wegen der Distributivität gilt

$$\mathrm{Bu}\ A^* = \mathrm{Bu}\ (\bigcup_{i \geq 0} A^i) = \bigcup_{i \geq 0} \mathrm{Bu}\ A^i$$

- Dies zeigt etwas direkter die Wahlmöglichkeiten auf.

Die nichtleere (endlichmalige) Iteration von $U \subseteq A^*$ ist

$$U^+ \stackrel{\text{def}}{=} \bigcup_{i>0} U^i$$

- Hier ist das leere Wort nicht explizit eingeschlossen.
- Es gilt $\varepsilon \in U^+ \Leftrightarrow \varepsilon \in U$.
- Hintergrund: Das neutrale Element ε ist *unzerlegbar*, d.h.

$$uv = \varepsilon \Rightarrow u = \varepsilon = v$$

Einige Rechenregeln:

- $U \subseteq U^+ \subseteq U^*$
- $U^+ = U\,U^* = U^*\,U$
- $(U\,V)^*\,U = U\,(V\,U)^*$
- $(U \cup V)^* = U^*\,(V\,U^*)^*$
- $\varepsilon^* = \varepsilon$
- $(U^*)^* = U^*\,U^* = U^*$
- $U \subseteq V \Rightarrow U^* \subseteq V^* \wedge U^+ \subseteq V^+$
- $U^* = \varepsilon \cup U\,U^* = \varepsilon \cup U^*\,U$
- $U^+ = U \cup U\,U^+ = U \cup U^+\,U$

Beispiel 1.10 Wir zeigen $(U \cup \varepsilon)^* = U^*$

$$
\begin{aligned}
& (U \cup \varepsilon)^* \\
= & \quad \{\!\!\{ \; (U \cup V)^* = U^* \; (V \; U^*)^* \; \}\!\!\} \\
& U^* \; (\varepsilon \; U^*)^* \\
= & \quad \{\!\!\{ \text{ Neutralität } \}\!\!\} \\
& U^* \; (U^*)^* \\
= & \quad \{\!\!\{ \text{ Doppelstern I } \}\!\!\} \\
& U^* \; U^* \\
= & \quad \{\!\!\{ \text{ Doppelstern II } \}\!\!\} \\
& U^*
\end{aligned}
$$

□