Kapitel 7

Reguläre Ausdrücke

7.1 Motivation

Reguläre Ausdrücke für Wortsprachen sind ein handliches Beschreibungsmittel. Bei der Übertragung auf Baumsprachen müssen wir uns überlegen, welche Operation auf Bäumen der Verkettung (dem Nebeneinanderschreiben) von Wörtern entsprechen soll.

Zum Verketten benutzen wir spezielle Blätter mit Markierungen aus einer Menge V, die nicht zum ursrünglichen Alphabet Σ gehört. Diese Blätter können wir dann durch Bäume ersetzen.

Wir können reguläre Ausdrücke erweitern (um Komplemente) und dann wieder einschränken (um Sterne). Das wirft Fragen auf, die für Wortsprachen gelöst sind, aber für Baumsprachen nicht (weil, wie erwähnt, die Begriffe vielleicht noch nicht die richtigen sind).

7.2 Reguläre Baum-Ausdrücke

7.2.1 Syntax

Def. Sei Σ eine Signatur, V eine Menge von nullstelligen Symbolen, und $\Sigma' = \Sigma \cup V$. Die Menge der *erweiterten regulären Ausdrücke* über einer Signatur Σ ist das kleinste M, für das gilt:

1. (leere Menge) $\emptyset \in M$
2. (Konstruktor-Anwendung) wenn $f \in \Sigma'_k, X_i \in M$, dann $f(X_1, \dots, X_k) \in M$
3. (Verkettung) wenn $X \in M, v \in V, Y \in M$, dann $X \cdot_v Y \in M$.
4. (Stern) wenn $X \in M, v \in V$, dann $X^v \in M$.
5. (Komplement) wenn $X \in M$, dann $\overline{X} \in M$

7.2.2 Semantik

Def. Ein erw. regulärer Ausdruck beschreibt eine Baumsprache durch

1. (leere Menge) $\mathsf{Wert}(\emptyset) = \emptyset$

   Kommentar: Die leere Menge ist als Elementar-Ausdruck zugelassen, damit wir $\mathsf{Tree}(\Sigma')$ als sternfreien Ausdruck bekommen, nämlich $\overline{\emptyset}$, siehe unten (Komplemente).
2. (Konstruktoren) $\mathsf{Wert}(f(X_1, \dots, X_k)) = \{f(x_1, \dots) \mid x_i \in \mathsf{Wert}(X_i)\}$
3. (Verkettung) $\mathsf{Wert}(X \cdot_v Y) = M$ mit $t \in M \iff \exists x \in \mathsf{Wert}(X)$, und t entsteht aus x, indem *alle* Vorkommen von v in x durch ein $y \in \mathsf{Wert}(Y)$ ersetzt werden

   Beachte: Verschiedene Vorkommen von v dürfen durch verschiedene y ersetzt werden. y darf v enthalten, das interessiert uns nicht, diese bleiben stehen.
4. (Stern) $\mathsf{Wert}(X^v)$ ist die kleinste Menge M mit $v \in M$ und $x \in M, y \in \mathsf{Wert}(X) \to x \cdot_v y \in M$.

5. (Komplement) $\mathsf{Wert}(\overline{X}) = \mathsf{Tree}(\Sigma') \setminus \mathsf{Wert}(X)$

   Beachte: Das Komplement wird gegen die erweiterte Signatur genommen

In Wirklichkeit sind das alles Ausdrücke, die Sprachen über Σ' beschreiben. Wir vereinbaren, daß wir ganz zum Schluß mit $\mathsf{Tree}(\Sigma)$ schneiden, d. h. alle Bäume ignorieren, die noch $v \in V$ enthalten.

7.2.3 Teilklassen

Def. Ein erweiterter regulärer Ausdruck heißt *regulär*, wenn er kein $\emptyset$ und kein Komplement enthält.

Def. Ein erw. reg. Ausdr. heißt *sternfrei*, wenn er keinen Stern enthält.

7.3 Sternfreie Baumsprachen

Die typische sternfreie Wortsprache sieht so aus:

$$(ab)^* = \overline{b\overline{\emptyset} \cup \overline{\emptyset}a \cup \overline{\emptyset}(aa \cup bb)\overline{\emptyset}}$$

(darf nicht mit b anfangen, nicht mit a aufhören, und kein aa oder bb enthalten), und die typische nicht sternfreie Sprache ist $(aa)^*$. Das heißt, sternfreie Ausdrücke können nicht zählen, jedenfalls nicht ohne Hilfe des Alphabets.

Von A. Potthoff stammt das folgende Beispiel, das zeigt, daß wir mit erweiterten sternfreien Ausdrücken über Bäumen *doch* zählen können:

Die Signatur Σ enthält a (nullstellig) und f (zweistellig). Wir wollen die Sprache L der Bäume, in denen alle Pfade (von der Wurzel zu Blättern) gerade Länge haben.

Ein regulärer Ausdruck für L läßt sich leicht angeben. Wir wählen das Verkettungsalphabet $V = \{c\}$.

$$L = \left(f(f(c,c), f(c,c)) \right)^{c} \cdot_c a$$

Die Intuition sagt: wir müssen (modulo 2) zählen, und das geht sternfrei nicht. Falsch!

Wir benutzen Verkettungszeichen $\{g, u, x\}$ (Bedeutung: gerade, ungerade, auxiliary) und betrachten die Mengen

$$G = f(f(g, X), u), U = f(X, g) \text{ mit } X = \left(f(u, f(g, x)) \right)^{x} \cdot_x a$$

(Übungsaufgabe!)

7.4 Unäre reguläre Ausdrücke

Wir haben gesehen, daß wir sternfrei zählen können, wenn wir mehrere Verkettungsvariablen zur Verfügung haben. Das widerspricht der Vorstellung, die von Wortsprachen kommt. Wir versuchen, reguläre Ausdrücke so weit einzuschränken, bis die Intuition wieder stimmt.

In regulären Ausdrücken für Wortsprachen gibt es genau eine Verkettungsvariable, und sie kommt in jedem erzeugten Wort genau einmal vor, obwohl wir sie nie explizit sehen: das linke Wort-Ende.

7.4.1 Anzahl der Vorkommen der Verkettungsvariablen

(Literatur: U. Heuter, ungefähr 1990)

Def. Ein Ausdruck heißt *speziell*, wenn in jedem Baum der erzeugten Sprache jedes Verkettungssymbol höchstens einmal vorkommt.

Fragen: welche regulären Baumsprachen haben spezielle reguläre Ausdrücke? Ist das entscheidbar?

Bem. reguläre Ausdrücke für Wortsprachen sind immer speziell.

Satz: speziell und sternfrei $\iff$ first-order-definierbar.

Bemerkung: Es ist bis heute kein Algorithmus bekannt, der FO-Definierbarkeit entscheidet.

7.4.2 Anzahl der Verkettungsvariablen

(von A. Potthoff 1994 vorgeschlagen, aber nicht weiter untersucht)

Def. Ein Ausdruck heißt unär, wenn das Zusatz-Alphabet aus nur einem Zeichen besteht.

(Das darf dann aber in erzeugten Bäumen mehrfach vorkommen.)

Fragen: Welche Sprachen haben unäre Ausdrücke?

Bem. reguläre Ausdrücke für Wortsprachen sind immer unär.

Fragen: Welche Sprachen haben unäre und sternfreie Ausdrücke?

7.5 Definite Sprachen

Wir wollen lokale Eigenschaften von Sprachen vergleichen.

7.5.1 Definite Wort-Sprachen

Def. Für eine Wortmenge M bezeichnet $\mathsf{pre}_k(M)$ die Menge der Präfixe der Länge k:

$$\mathsf{pre}_k(M) = \{u \mid u \cdot w \in M, |u| = k\},$$

$\mathsf{suff}_k(M)$ die Menge der Suffixe der Länge k:

$$\mathsf{suff}_k(M) = \{w \mid u \cdot w \in M, |w| = k\},$$

$\mathsf{fact}_k(M)$ die Menge aller Faktoren der Länge k:

$$\mathsf{fact}_k(M) = \{v \mid u \cdot v \cdot w \in M, |v| = k\}.$$

Def. Eine Sprache L heißt *k-präfix-definit*, wenn

$$\forall u, v \in \Sigma^* : \quad (\mathsf{pre}_k(u) = \mathsf{pre}_k(v)) \to (u \in L \iff v \in L)$$

L heißt *präfix-definit*, wenn ein k existiert, für das L k-präfix-definit ist.

Def. Eine Sprache L heißt *k-suffix-definit*, wenn

$$\forall u, v \in \Sigma^* : \quad (\mathsf{suff}_k(u) = \mathsf{suff}_k(v)) \to (u \in L \iff v \in L)$$

Def. Eine Sprache heißt *k-definit*, wenn

$$\forall u, v \in \Sigma^* : \quad (\mathsf{pre}_k(u) = \mathsf{pre}_k(v) \wedge \mathsf{suff}_k(u) = \mathsf{suff}_k(v)) \to (u \in L \iff v \in L)$$

Obacht: Bei anderern Autoren heißt präfix-definit einfach definit, und definit heißt generalized definite.

Def. Eine Sprache L heißt *k-lokal*, wenn

$$\forall u, v \in \Sigma^* : \quad (\mathsf{pre}_k(u) = \mathsf{pre}_k(v) \wedge \mathsf{fact}_k(u) = \mathsf{fact}_k(v) \wedge \mathsf{suff}_k(u) = \mathsf{suff}_k(v)) \\ \to (u \in L \iff v \in L)$$

Satz. Präfix-definite Sprachen sind darstellbar als endliche Vereinigung

$$\bigcup u_i \Sigma^* \quad \text{mit } u_i \in \Sigma^*$$

Satz. Präfix-definite, suffix-definite, definite, lokale Sprachen sind abgeschlossen unter Booleschen Operationen.

Satz. Es ist entscheidbar, ob eine Sprache (gegeben durch einen Automaten) präfix-definit, suffix-definit, definit, lokal ist.

Beweis: kann man aus dem syntaktischen Monoid ablesen.

7.5.2 Definite Baum-Sprachen

Um diese Begriffe auf Bäume zu übertragen, müssen wir Präfix und Suffix passend festlegen.

Ein Suffix eines Baumes t ist einfach ein Teilbaum von t.

Def. Die Menge $\mathsf{suff}_k(t)$ der Suffixe der Größe (Höhe) k eines Baumes t ist

$$\mathsf{suff}_k(t) = \{t' \mid t' \trianglelefteq t, \mathsf{depth}(t') = k\}$$

Einen Präfix von t erhalten wir durch Abschneiden von Teilbäumen. Dabei ändern die Wurzeln der abgeschnittenen Teilbäume ihre Arität, wir erweitern deswegen die Signatur:

Def. Für $k > 0$: wenn $f \in \Sigma_k$ ein Symbol der Arität k ist, dann ist f_c ein nullstelliges Symbol. Wenn Σ eine Signatur ist, bezeichnet Σ_c die Vereinigung von Σ mit der Menge aller solcher f_c für $f \in \Sigma$.

Def. Der Präfix der Größe (Höhe) k eines Baumes t ist

$$\begin{aligned} f \in \Sigma_0 : \quad \mathsf{pre}_k(f) &= f \\ \mathsf{pre}_0(f(t_1, \dots, f_n)) &= f_c \\ \mathsf{pre}_{k+1}(f(t_1, \dots, f_n)) &= f(\mathsf{pre}_k(t_1), \dots, \mathsf{pre}_k(t_n)) \end{aligned}$$

Def. Eine Baumsprache L heißt k-*Wurzel-definit*, wenn

$$\forall u, v \in \mathsf{Tree}(\Sigma) : \quad (\mathsf{pre}_k(u) = \mathsf{pre}_k(v)) \to (u \in L \iff v \in L)$$

L heißt *Wurzel-definit*, wenn ein k existiert, für das L k-Wurzel-definit ist.

Def. Eine Sprache L heißt k-*Blatt-definit*, wenn

$$\forall u, v \in \mathsf{Tree}(\Sigma) : \quad (\mathsf{suff}_k(u) = \mathsf{suff}_k(v)) \to (u \in L \iff v \in L)$$

Def. Eine Sprache heißt k-*definit*, wenn

$$\forall u, v \in \mathsf{Tree}(\Sigma) : \quad (\mathsf{pre}_k(u) = \mathsf{pre}_k(v) \wedge \mathsf{suff}_k(u) = \mathsf{suff}_k(v)) \to (u \in L \iff v \in L)$$

Obacht: Bei anderern Autoren heißt Wurzel-definit einfach definit, Blatt-definit heißt reverse definit oder frontier testable, und definit heißt generalized definit.

Satz. Wurzel-definite Sprachen sind darstellbar durch erweiterte reguläre Ausdrücke ohne Verkettungen und Sterne.

Satz. Präfix-definite, suffix-definite, definite sind abgeschlossen unter Booleschen Operationen.

Satz. (U. Heuter, 1988) Es ist entscheidbar, ob eine Sprache (gegeben durch einen Automaten) präfix-definit, suffix-definit, definite ist.

Beweis: (Nivat, Podelski, 1989) Kann man auch aus dem syntaktischen Monoid ablesen.

Bemerkung: Wurzel-definite Sprachen werden erkannt durch (eine Teilklasse von) deterministischen top-down-Automaten.