Kapitel 4

Reguläre Baum-Sprachen

4.1 Kongruenzen

(Motto: Wie für Wortsprachen.)

Def: Sei L eine Sprache. Die Relation $\sim_L$ auf $\mathsf{Tree}(\Sigma)$ ist erklärt durch

$$X \sim_L Y \iff \forall C \in \mathsf{Con}(\Sigma) : C[X] \in L \iff C[Y] \in L$$

Def: Der Index eine Äquivalenzrelation $\sim$ auf einer Menge M ist die Anzahl ihrer Äquivalenzklassen.

Satz: Folgende Aussagen sind äquivalent:

1. L ist regulär
2. $\sim_L$ hat endlichen Index

Die Äquivalenzklassen von $\sim_L$ entsprechen den Zuständen des minimalen deterministischen (Bottom-Up-)Automaten für L.

Beweis ($1 \Rightarrow 2$): Wenn L regulär, dann gibt es einen det. bottom-up-Automaten $A = (\Sigma, Q, Q_F, \Delta)$ mit $\mathsf{Lang}(A) = L$. (Beweis: Potenzmengenkonstruktion).

Wir zeigen: die Klassen von $\sim_L$ sind Vereinigungen von $\mathsf{Lang}(A, q)$ für $q \in Q$. Das sind jedenfalls nur endlich viele, höchstens $|Q|$ Stück.

Genauer: Wir definieren

$$X \sim_A Y \iff \exists q \in Q : X \in \mathsf{Lang}(A, q) \wedge Y \in \mathsf{Lang}(A, q)$$

d. h. zwei Bäume sind äquivalent bzgl. A, wenn sie im gleichen Zustand akzeptiert werden.

Und zeigen: die Relation $\sim_A$ ist eine Verfeinerung der Relation $\sim_L$. (D. h. $X \sim_A Y \Rightarrow X \sim_L Y$). Dabei müssen wir die Voraussetzung benutzen, daß A deterministisch ist:

Jetzt sind wir fertig, denn $\sim_A$ hat genau $|Q|$ Äquivalenzklassen, weil es aber eine Verfeinerung von $\sim_L$ ist, hat $\sim_L$ auch höchstens Index $|Q|$, also endlich.

Beweis ($2 \Rightarrow 1$): Aus den endlich vielen Äquivalenzklassen $K_1, \ldots, K_n$ von $\sim_L$ bauen wir uns einen Automaten B für L. Wir fixieren für jedes i ein $X_i \in K_i$ (Das geht: die K_i sind alle nicht leer, weil es eben Äquivalenzklassen sind.) Die Menge $\{X_1, \ldots, X_n\}$ nennen wir M.

Wir schreiben nun für jeden Term

$$\{f(t_1, \ldots, t_k) \mid f \in \Sigma_k, \forall i : t_i \in M\}$$

seine Äquivalenzklasse auf und erhalten so die Übergangsrelation des Automaten. Als akzeptierende Zustände nehmen wir alle Klassen K_i, für die $X_i \in L$.

Wir müssen zeigen, daß das wohldefiniert ist, d. h. nicht von der anfänglichen Wahl der $X_i \in K_i$ abhängt. Dazu brauchen wir:

$$\forall f \in \Sigma_k : \forall i : t_i \sim_L s_i \Rightarrow f(t_1, \ldots) \sim_L f(s_1, \ldots)$$

Wir zeigen das durch Induktion:

$$f(t_1, \ldots, t_k) \sim_L f(s_1, t_2, \ldots, t_k) \sim_L \ldots \sim_L f(s_1, \ldots, s_{k-1}, t_k) \sim_L f(s_1, \ldots, s_k)$$

Das geht für die erste Behauptung so: Wir müssen für jeden Kontext C zeigen, daß

$$C[f(t_1, \ldots, t_k)] \in L \iff C[f(s_1, t_2, \ldots, t_k)] \in L$$

Wir benutzen den Kontext $D = C[f(\bullet, t_2, \ldots, t_k)]$. Dafür gilt $D[t_1] = C[f(t_1, \ldots, t_k)]$ und $D[s_i] = C[f(s_1, t_2, \ldots, t_k)]$. Wegen $t_1 \sim_L s_1$ folgt $D[t_1] \in L \iff D[s_1] \in L$ und daraus die Behauptung.

4.2 Morphismen zwischen Automaten

Der verangegangene Beweis enthielt zwei Beobachtungen, die wir hier separat aufschreiben wollen. Die erste:

Satz: Zu jeder Sprache $L \in \mathsf{REG}$ gibt es einen eindeutigen minimalen deterministischen Automaten B mit $L = \mathsf{Lang}(B)$.

Zur zweiten brauchen wir eine Definition

Def: Ein Morphismus vom Automaten $A = (\Sigma, Q, Q_F, \Delta)$ zum Automaten $A' = (\Sigma, Q', Q'_F, \Delta')$ ist eine Abbildung $h : Q \to Q'$ mit $h(Q_F) \subseteq Q'_F$ (der Morphismus respektiert akzeptierende Zustände)

$$\forall f(q_1, \ldots, q_k) \to q \in \Delta : f(h(q_1), \ldots, h(q_k)) \to h(q) \in \Delta'$$

(der Morphismus respektiert die Übergangsrelation).

Anschaulich: ein Morphismus identifiziert Zustände.

Satz: Sei $h : A \to A'$ ein Morphismus. Dann gilt $\mathsf{Lang}(A) = \mathsf{Lang}(A')$.

Die Umkehrung gilt natürlich nicht.

4.3 Minimale Automaten

Def: Der Automat $A' = (\Sigma, Q', Q'_F, \Delta')$ ist eine Verfeinerung des Automaten $A = (\Sigma, Q, Q_F, \Delta)$, wenn es einen Morphismus $h : A' \to A$ gibt.

Satz: Jeder deterministische Automat A mit $\mathsf{Lang}(A) = L$ ist eine Verfeinerung des minimalen deterministischen Automaten für L.

Die Definitionen (Morphismus, Verfeinern) sind für beliebige (nicht notwendig deterministische) Automaten sinnvoll. Für die Existenz und Konstruierbarkeit des minimalen Automaten brauchen wir aber Determinismus. Kleine nichtdeterministische Automaten sind 1) schwer zu finden und 2) nicht notwendig eindeutig bestimmt.