6

Sternfreie Sprachen

Wir studieren nunmehr eine echte Teilklasse der regulären Sprachen: die *sternfreien Sprachen*. Wie die regulären Sprachen erlauben diese mehrere äquivalente Charakterisierungen: durch sternfreie reguläre Ausdrücke, durch erststufige Logik definierbar, in LTL definierbar.

6.1 Erststufige Logik

Die erststufige Logik ist das Fragment der wMSO, in dem keine Quantifikation über zweitstufige Variablen erlaubt ist. Erststufige Quantoren und freie zweitstufige Variablen sind aber erlaubt. Die Quantorentiefe einer erststufigen Formel ist die Schachtelungstiefe der erststufigen Quantoren.

Zum Beispiel ist $\phi \equiv \forall x.P_a(x) \Rightarrow \exists y.x{<}y \wedge P_b(x)$ eine erststufige Formel der Quantorentiefe 2 mit $L(\phi) = (a \cup b \cup c)^*b(b \cup c)^* \cup (b \cup c)^*$ (auf jedes a folgt später noch ein b). Wir zeigen später, dass keine erststufige Formel ψ existiert mit $L(\psi) = (aa)^*$.

Bevor wir ins Detail gehen können, müssen wir uns ein Hilfsmittel aus der endlichen Modelltheorie erarbeiten, das *Ehrenfeucht-Fraïssé-Spiel*.

6.2 Das Ehrenfeucht-Fraïssé-Spiel

Zwei Personen, genannt Spoiler (S) und Duplicator (D) spielen folgendes Spiel: Gegeben sind zwei Wörter u und v über einem Alphabet Σ, die untereinander geschrieben werden. Außerdem wird eine Zahl von Runden vereinbart.

S beginnt das Spiel, indem er auf eine Position in entweder u oder v zeigt. D antwortet, indem er auf eine Position im jeweils anderen Wort zeigt und die beiden Positionen durch eine Linie verbindet. Die Buchstaben an den beiden Positionen müssen dabei aber übereinstimmen. Außerdem darf dabei keine schon vorhandene Verbindungslinie gekreuzt werden. Sodann ist wieder S am Zug usw. Der Spieler S darf eine bereits gespielte Position nochmal spielen; in diesem Fall muss D mit der korrespondierenden Position antworten. In keinem anderen Fall darf D eine schon gespielte Position wiederholen.

Kann D nicht spielen, so verliert er das Spiel. Ist die vereinbarte Rundenzahl vorbei, ohne dass D verloren hätte, so gewinnt D. Wir sagen dann, D gewinnt k Runden.

Dieses Spiel heisst Ehrenfeucht-Fraïssé-Spiel, im folgenden kurz EF-Spiel. Wir bezeichnen das EF-Spiel auf u, v mit k Runden als $G_k(u, v)$. Wir sagen S, bzw. D gewinnt $G_k(u, v)$, wenn S, bzw. D das Spiel immer gewinnen kann, wie geschickt auch immer der Gegner D bzw. S spielen mag. Wir sagen dann, dass der jeweilige Gegner das Spiel verliert (auch wenn er bei ungeschicktem Spiel des anderen vielleicht doch gewinnen könnte.) Wie alle endlichen Spiele ist auch dieses Spiel determiniert: entweder S oder D gewinnt.

Beispiel I:

$\Sigma = \{a, b\}$, $u = aabaacaa$, $v = aacaabaa$. Duplicator gewinnt $G_1(u, v)$, verliert aber $G_2(u, v)$ Um $G_2(u, v)$ zu gewinnen, würde S im ersten Zug das b in u spielen; D muss mit dem B in v antworten. In der zweiten Runde spielt S das c in u und D würde gerne c spielen, kann aber nicht, weil sich die Linien dann überkreuzen würden, und verliert.

Beispiel II:

$\Sigma = \{a\}$, $|u| >= 2^k - 1$, $|v| >= 2^k - 1$. Hier kann Duplicator k Runden überleben.

Beispiel II:

$\Sigma = \{a\}$, $|u| >= 2^k - 1$, $|v| >= 2^k - 1$. Hier kann Duplicator k Runden überleben (D gewinnt $G_k(u, v)$).

Um geeignete Invarianten formulieren zu können, benötigen wir noch die folgende Verallgemeinerung: Seien u, v Wörter, $\mathbf{s} = (s_0, s_1, \ldots, s_{n-1})$ und $\mathbf{t} = (t_0, t_1, \ldots, t_{n-1})$ Zahlenfolgen mit $s_i < |u|, t_i < |v|$. Das Spiel $G_k((u, \mathbf{s}), (v, \mathbf{t}))$ wird so gespielt wie $G_k(u, v)$ mit der Ausnahme, dass die Verbindungen s_i—t_i bereits als vorhanden gelten, somit dürfen die Positionen s_i in u und t_i in v von D nicht mehr gespielt werden, es sei denn, S hätte zuvor die korrespondierende Position gespielt. Außerdem dürfen die Verbindungen s_i—t_i nicht gekreuzt werden. Sollten sich solche vorgegebenen Verbindungen s_i—t_i und s_j—t_j kreuzen (also z.B. $s_i < s_j$ und $t_j < t_i$) oder verschiedene Buchstaben verbinden, so hat D sofort verloren, also schon nach 0 Runden.

Der Satz von Ehrenfeucht und Fraïssé besagt, dass ein Spiel $G_k(u, v)$ von D genau dann gewonnen wird, wenn u und v durch Formeln der erststufigen Logik mit Quantorentiefe k nicht zu unterscheiden sind. Wenn bis auf weiteres von Formeln die Rede ist, so sind immer Formeln der erststufigen Logik über der Signatur $P_a(x)$ für $a \in \Sigma$ und $x < y$ gemeint.

Wir fixieren ein Alphabet Σ und betrachten erststufige Formeln, deren freie Mengenvariablen der Menge $\{P_a \mid a \in \Sigma\}$ entstammen, vgl. Abschnitt 4.2.

Formeln der Quantorentiefe Null sind Boole'sche Kombinationen von atomaren Formeln, also $P_a(x)$ für a in Σ und $x < y$.

Formeln der Quantorentiefe $k + 1$ sind von der Form $\exists x.\phi(x)$, wobei ϕ von Quantorentiefe k ist und außerdem Boole'sche Kombinationen solcher Formeln.

Ist ϕ Formel mit n freien erststufigen Variablen $x_0, \dots, x_{n-1}$ (neben den zweitstufigen Variablen P_a) und $\mathbf{s} = (s_0, \dots, s_{n-1})$, so schreiben wir $u, \mathbf{s} \models \phi$ um zu sagen, dass ϕ wahr ist unter der durch u gegebenen Interpretation der Mengenvariablen P_a und der durch $\mathbf{s}$ gegebenen Belegung der freien erststufigen Variablen, formal also $I_u[\mathbf{x} \mapsto \mathbf{s}] \models \phi$. Alternativ schreiben wir auch $u \models \phi(\mathbf{s})$.

Definition 19. *Seien u, v Wörter. Wir schreiben $u \equiv_k v$, wenn gilt $u \models \phi \iff v \models \phi$ für alle Formeln ϕ der Quantorentiefe k, welche außer den P_a keine weiteren freien Variablen haben. Seien u, v Wörter und $\mathbf{s} = (s_0, \dots, s_{n-1})$, $\mathbf{t} = (t_0, \dots, t_{n-1})$. Wir schreiben $(u, \mathbf{s}) \simeq_{k,n} (v, \mathbf{t})$, wenn gilt*

$$u, \mathbf{s} \models \phi \iff v, \mathbf{t} \models \phi$$

für alle Formeln ϕ mit n freien Variablen und Quantorentiefe k.

Proposition 4 (Ehrenfeucht-Fraïssé). *Das Spiel $G_k((u, \mathbf{s}), (v, \mathbf{t}))$ wird von Spieler D genau dann gewonnen, wenn gilt $(u, \mathbf{s}) \simeq_{k,n} (v, \mathbf{t})$.*

Beweis. "$\Rightarrow$": Durch Induktion über k. Spieler D gewinnt 0 Runden, wenn $s_i < s_j \iff t_i < t_j$ und $u(s_i) = v(t_i)$ für alle i. Dann aber gilt für jede atomare Formel ϕ, dass $u \models \phi(\mathbf{s}) \iff v \models \phi(\mathbf{t})$ und durch Induktion über den Formelaufbau setzt sich dies auf beliebige quantorenfreie Formeln fort. Sei nun die Behauptung für k schon gezeigt und gewinne D $k+1$ Runden. Wir zeigen wiederum durch Induktion über den Formelaufbau, dass $u \models \phi(\mathbf{s}) \iff v \models \phi(\mathbf{t})$. Der interessante Fall ist $\phi = \exists x.\psi$. Wenn $u \models \phi(\mathbf{s})$, so existiert eine Position s, derart dass $u \models \psi(s, \mathbf{s})$. Wir betrachten den Fall, in dem S mit der Position s eröffnet. Da ja nach Voraussetzung D eine Gewinnstrategie besitzt, muss er diesen Zug mit einer Position t beantworten können. Das Restspiel $G_k((u, s.\mathbf{s}), (v, t.\mathbf{t}))$ wird nun von D gewonnen, nach Induktionsvoraussetzung haben wir also $u, s.\mathbf{s} \simeq_{k,n+1} v, t.\mathbf{t}$ und es folgt $v \models \psi(t, \mathbf{t})$.

"$\Leftarrow$" Die Idee ist, zu jedem $(u, \mathbf{s})$ und k eine Formel $\chi = \chi^{(n,k)}_{(u,\mathbf{s})}$ anzugeben, derart dass $u, \mathbf{s} \models \chi$ gilt und weiterhin aus $v, \mathbf{t} \models \chi$ folgt, dass D das Spiel $G_k((u, \mathbf{s}), (v, \mathbf{t}))$ gewinnt. Es sollte klar sein, dass die Existenz solch einer Formel die Behauptung liefert. Wir konstruieren die Formel durch Induktion über k. Falls $k = 0$, so wählen wir für χ die Konjunktion der folgenden Bedingungen:

$$\begin{array}{ll} a(x_i) & \text{, falls } u_{s_i} = a \\ x_i < x_j & \text{, falls } s_i < s_j \end{array}$$

Es ist klar, dass $u, \mathbf{s} \models \chi$. Wenn jetzt auch $v, \mathbf{t} \models \chi$, so folgt, dass D das Spiel $G_0((u, \mathbf{s}), (v, \mathbf{t}))$ gewinnt. Falls $k > 0$, so setzen wir

$$\chi(\mathbf{x}) = \bigwedge_{i=0}^{|u|-1} \exists x.\chi_i(x, \mathbf{x}) \wedge \forall x. \bigvee_{i=0}^{|u|-1} .\chi_i(x, \mathbf{x})$$

wobei χ_i die charakterisierende Formel fuer $(k-1, n+1)$ und $(u, i.\mathbf{s})$ ist. Klar, dass $(u, \mathbf{s})$ dieses χ wahrmacht: Gegeben i, wählen wir einfach i für x. Gegeben x, wählen wir einfach x für i. Wenn jetzt $(v, \mathbf{t}) \models \chi$, so beschreiben wir eine Gewinnstrategie für D. Beginnt S in u, so liefert das erste Konjunkt die entsprechende Antwort; beginnt S in v, so nehmen wir das zweite Konjunkt her.

Als einfache Anwendung zeigen wir folgenden Satz:

Proposition 5. *Die Sprache $L = (aa)^*$ lässt sich nicht durch FOL definieren, d.h. es gibt keine Formel ϕ der FOL, sodass $u \models \phi$, gdw $u \in (aa)^*$.*

Beweis. Gäbe es so ein ϕ, dann hätte es eine bestimmte Quantorentiefe k. Das Spiel $G_k(a^{2^k}, a^{2^k-1})$ wird von D gewonnen, somit sind die beiden Wörter durch FOL Formeln der Quantorentiefe k nicht zu unterscheiden, also insbesondere nicht durch ϕ. Das eine Wort ist aber in L, das andere nicht. Widerspruch.

6.3 Sternfreie Sprachen durch Abschlusseigenschaften

Definition 20. *Sei Σ ein Alphabet. Die sternfreien Sprachen über Σ, geschrieben $SF(\Sigma)$ bilden die kleinste Klasse von Sprachen über Σ, die unter folgendem abgeschlossen ist:*

- *$\emptyset$ ist sternfrei.*
- *$\{\epsilon\}$ ist sternfrei.*
- *$\{a\}$ ist sternfrei für $a \in \Sigma$.*
- *Sind A, B sternfrei, so auch AB, $\overline{A}$, $A \cup B$*

Beispiele: Σ^* ist sternfrei, da $\Sigma^* = \overline{\emptyset}$. Sind A, B sternfrei, so auch $A \cap B = \overline{\overline{A} \cup \overline{B}}$ und $A \setminus B = A \cap \overline{B}$. Falls $D \subseteq \Sigma$, so ist D^* sternfrei, da $D^* = \Sigma^* \setminus (\Sigma^*(\Sigma \setminus D)\Sigma^*)$. Die Sprache $(ab)^*$ ist auch sternfrei, da $(ab)^* = \Sigma^* \setminus b\Sigma^* \setminus \Sigma^* aa\Sigma^* \setminus \Sigma^* bb\Sigma^*$.

Proposition 6. *Sei L eine sternfreie Sprache mit $\epsilon \notin L$. Es gibt eine FOL Formel ϕ, sodass $w \in L \iff w \models \phi$.*

Beweis. Ist L sternfrei, so ist $L\setminus\{\epsilon\}$ FOL-definierbar. Wir beweisen das durch Induktion über die Definition von "sternfrei": $\emptyset$, $\{a\}$ sind alle FOL definierbar. , z.B.: $\emptyset = L(\neg(\forall x.x = x)))), \{a\} = L(\forall x, y.x = y \wedge a(x))$. Sei nun $L = \overline{L_1}$ und $L_2 = L_1 \setminus \epsilon$. Nach Induktionsvoraussetzung ist $L_2 = L(\phi)$ für eine Formel ϕ. Es gilt $L = L(\neg\phi)$.

Sei nun $L = L_1 \cup L_2$ und $L_1 = L(\phi_1), L_2 = L(\phi_2)$. Es ist $L = L(\phi_1 \vee \phi_2)$.

Sei nun $L = L_1L_2$. Wir nehmen o.B.d.A. an, dass ϵ nicht in L_1, L_2 enthalten ist. Sei $L_1 = L(\phi_1)$, $L_2 = L(\phi_2)$. Es ist $L = L(\exists t.\phi_1 \upharpoonright \{0 \dots t\} \wedge \phi_2 \upharpoonright \{t+1 \dots\})$. Hierbei bezeichnet $\phi \upharpoonright \{u \dots v\}$ die Formel, die man aus ϕ durch Relativierung aller Quantoren auf den Bereich $\{u \dots v\}$ erhält. Man ersetzt also ein Vorkommen von $\exists y. \dots$ durch $\exists y.(u \leq y \wedge y \leq v) \wedge \dots$.

Für die Umkehrung dieses Satzes (FOL-definierbare Sprachen sind sternfrei) benötigen wir etwas Vorarbeit.

Lemma 6. *Die Relation $\equiv_k$ (Ununterscheidbarkeit durch Formeln der Quantorentiefe k) ist Äquivalenzrelation mit endlich vielen Äquivalenzklassen.*

Beweis. Direkte Folgerung aus der Tatsache, dass es bis auf logische Äquivalenz nur endliche viele Formeln einer festen Quantorentiefe k gibt. Jede Äquivalenzklasse ist eindeutig bestimmt durch die Menge der Formeln, die in ihr gelten. Gibt es also höchstens p Formeln, dann gibt es höchstens 2^p Äquivalenzklassen.

Lemma 7. *Zu jeder $\equiv_k$-Äquivalenzklasse W gibt es eine Formel ϕ_W, sodass* $(u, \mathbf{s}) \in W \iff (u, \mathbf{s}) \models \phi_W$.

Beweis. Man nimmt die Konjunktion aller (bis auf Äquivalenz) Formeln der Quantorentiefe k, die in W gelten.

Lemma 8. *Jede durch eine Formel der Quantorentiefe k definierte Sprache ist (endliche) Vereinigung von $\equiv_k$-Äquivalenzklassen.*

Beweis. Ist ein Wort u in L, so erfüllt es die definierende Formel. Jedes äquivalente Wort erfüllt dann aber auch diese Formel, ist somit auch in L.

Proposition 7. *Sei ϕ eine Formel der FOL. Die Sprache $L(\phi)$ ist sternfrei.*

Beweis. Induktion über die Quantorentiefe k und den Formelaufbau.

Induktionsanfang: die einzigen (bis auf Äquivalenz) geschlossenen Formeln von Quantorentiefe 0 sind die Formeln `tt` und `ff`. Beide definieren sternfreie Sprachen, nämlich $\emptyset$ und Σ^+.

Induktionsschritt: Habe ϕ die Quantorentiefe $k+1$. Boole'sche Operationen sind auf der Ebene der sternfreien Ausdrücke vorhanden; somit können wir uns auf den Fall $\phi = \exists x.\psi$ beschränken, wobei ψ die Quantorentiefe k hat. Wir dürfen die Behauptung für ψ und alle anderen Formeln der Quantorentiefe k voraussetzen. Insbesondere ist jede $\equiv_k$-Äquivalenzklasse sternfrei.

Wir behaupten nun folgendes:

$$L(\phi) = L(\exists x.\psi) = \bigcup_{(uav,|u|)\models\psi} [u]_{\equiv_k} a[v]_{\equiv_k}$$

Man beachte, dass dies eine endliche Vereinigung ist, da es ja überhaupt nur endlich viele $\equiv_k$-Äquivalenzklassen gibt.

Zum Beweis der Behauptung nehmen wir an, dass $w \models \phi$. Somit ist $w = uav$ und $(uav, |u|) \models \psi(x)$. Dann aber ist w in $[u]_{\equiv_k} a[v]_{\equiv_k}$.

Sei umgekehrt $w = uav, u \equiv_k u'$, $v \equiv_k v'$ und $u'av', |u'| \models \psi(x)$, dann gilt mithilfe des Satzes von Ehrenfeucht-Fraïssé auch $(uav, |u|) \equiv_k (u'av', |u'|)$, somit $(uav, |u|) \models \psi$, somit $w \models \phi$. Hierzu argumentieren wir wie folgt: Nach dem Satz von Ehrenfeucht-Fraïssé gewinnt D die Spiele $G_k(u, u')$ und $G_k(v, v')$. Duplicator gewinnt dann auch das Spiel $G_k((uav, |u|), (u'av', |u'|))$, denn Züge von S auf u oder u' werden entsprechend der Gewinnstrategie für $G_k(u, u')$ beantwortet, analog für (v, v'). Die Verbindung a—a ist ja schon gespielt. Eine abermalige Anwendung des Satzes von Ehrenfeucht-Fraïssé liefert dann $(u'av', |u'|) \models \psi$, also $u'av' \models \phi$.

Übungsaufgaben

Übung 9. Sei $f \in \mathbb{B}^+(Q)$.

a) Seien $N \subseteq M \subseteq Q$ mit $N \models f$. Zeigen Sie per Induktion über den Formelaufbau, dass auch $M \models f$ gilt.
b) Kann es sein, dass $\emptyset \models f$ gilt?
c) Kann es sein, dass $Q \not\models f$ gilt?

Übung 10. Beweisen Sie Lemma 5.

Übung 11. Sei $M \subseteq Q$. Dann heißt M *minimales Modell* von $f \in \mathbb{B}^+(Q)$, falls $M \models f$ und $N \not\models f$ für alle $N \subsetneq M$.

a) Zeigen oder widerlegen Sie: Jedes f hat genau ein minimales Modell.
b) Ein mAFA sei genauso definiert wie ein AFA mit dem Unterschied, dass in den Laufbäumen die Beschriftungen der umittelbaren Nachfolgerknoten ein *minimales* Modell der entsprechenden Formel aus der Transitionsfunktion bilden müssen. Erkennen mAFAs genau die regulären Sprachen?

Übung 12. Beweise Thm. 13, also die Korrektheit der Dualisierungskonstruktion für AFAs.

Übung 13. Beweisen Sie die Aussagen in Korollar 2 ohne die Potenzmengenkonstruktion für AFAs zu benutzen. Zeigen Sie dazu, wie sich ein möglicherweise nicht gedächtnisloser Lauf eines AFA sukzessive in einen gedächtnislosen umbauen lässt, ohne dass dabei die entsprechenden Bedingungen an akzeptierende Läufe verletzt werden.

Teil III

Unendliche Wörter

7

Automaten auf unendlichen Wörtern

In diesem Teil beschäftigen wir uns mit der Erkennbarkeit von Mengen von *unendlichen* Folgen von Symbolen durch endliche Automaten. Die Tatsache, dass ein Automat ein unendliches Wort verarbeiten soll, mag auf den ersten Moment unnatürlich erscheinen, wenn man sich vorstellt, dass ein Automat ein Wort vom Anfang zum Ende hin abfährt und dann durch Erreichen eines Zustands signalisiert, ob er das Wort akzeptiert oder nicht. Dies ist aber in Wahrheit nur ein Problem mit der Vorstellung. Mathematisch ist die Akzeptanz eines Wortes durch einen Automaten einfach durch die Existenz eines akzeptierenden Laufs gegeben. Genauso kann man sich bei die Frage stellen, ob ein Automat auf einem gegebenen unendlichen Wort einen akzeptierenden Lauf hat. Läufe auf unendlichen Wörtern können ganz leicht als Erweiterung (ins Unendliche) von Läufen auf endlichen Wörtern erklärt werden. Es bleibt lediglich zu definieren, was daran akzeptierend bzw. verwerfend sein soll. Offensichtlich kann dies nicht sinnvoll durch den letzten Zustand auf dem Lauf erklärt werden, da es einfach keinen letzten mehr gibt. Eine sehr natürliche Erweiterung der Akzeptanz durch Erreichen eines Endzustands ins Unendliche ist die sogenannte *Büchi-Bedingung*: Ein Lauf ist akzeptierend, wenn er immer wieder in einen Endzustand gelangt, wenn also unendlich oft Endzustände durchlaufen werden.

Algorithmisch sinnvolle Fragestellungen im Zusammenhang mit solchen Automaten sind dann zum Beispiel, ob ein gegebener Automat überhaupt irgendein unendliches Wort erkennt (Leerheitsproblem), oder ob zwei gegebene Automaten dieselben Wörter erkennen (Äquivalenzproblem). Das Wortproblem, also ob ein Automat ein vorgelegtes Wort erkennt, ist im allgemeinen nicht algorithmisch sinnvoll, da man ja ein unendliches Wort nicht "vorlegen" kann. Anders sieht es wiederum aus, wenn die unendliche Eingabe in endlicher Form durch Verwendung einer geeigneten kompakten Notation gegeben ist.

Wir definieren zunächst unendliche Wörter und Sprachen von solchen, bevor wir dann Automaten, die auf diesen Wörtern operieren, einführen. Diese sind weiterhin endliche Automaten, die sich syntaktisch nicht wesentlich von den herkömmlichen endlichen Automaten unterscheiden. Im einfachsten Fall definieren wir Akzeptanz über die oben genannte Büchi-Bedingung und erhalten dann Büchi-Automaten. Wir werden sehen, dass die algorithmische Behandlung dieser Automaten und der Klasse der von ihnen erkannten Sprachen komplizierter als im Fall der endlichen Wörter ist. Wir betrachten auch andere Arten, Akzeptanz zu definieren und erhalten somit andere Automatenmodelle. Die wichtigsten Fragestellungen sind dann die Determinisierbarkeit von solchen Automaten, die Relationen zwischen den Klassen der von verschiedenen Automatentypen erkannten Sprachen, deren Abschlusseigenschaften. Die wichtigste Anwendung endlicher Automaten auf unendlichen Wörtern liegt in der monadischen Logik zweiter Stufe (MSO). Diese hat dieselbe Syntax wie die wMSO aus Kapitel 4; semantisch unterscheidet sie sich aber darin, dass zweitstufige Variablen über beliebige, nicht notwendigerweise endliche, Teilmengen von $\mathbb{N}$ rangieren. Eine Belegung der Variablen lässt sich dann in Analogie zur Konstruktion im Abschnitt 4.2.2 als *unendliches* Wort auffassen.

7.1 Unendliche Wörter und reguläre ω-Sprachen

Sei Σ wieder ein endliches Alphabet. Ein *unendliches Wort* über Σ — auch ω-Wort genannt — ist eine unendliche Folge $w = a_0a_1a_2\dots$ von Buchstaben $a_i \in \Sigma$. Wir bezeichnen die Menge der unendlichen Wörter über Σ mit Σ^ω.

Eine Teilmenge $L \subseteq \Sigma^\omega$ wird als ω-Sprache bezeichnet. Da es in diesem Teil hauptsächlich um unendliche Wörter geht, sprechen wir über ω-Wörter und ω-Sprachen auch einfach als Wörter und Sprachen. Sollte es in speziellen Fällen dann doch einmal um endliche Wörter und deren Sprachen gehen, so werden wir dies dann explizit klar machen.

Sei $w \in \Sigma^\omega$ und $a \in \Sigma$. Wir schreiben $|w|_a$ für die Anzahl der Vorkommen des Buchstabens a im Wort w. Beachte, dass $|w|_a = \infty$ möglich ist. Da das zugrundeliegende Alphabet endlich ist, muss es sogar für jedes $w \in \Sigma^\omega$ ein $a \in \Sigma$ geben, so dass $|w|_a = \infty$ gilt.

Man kann unendliche Wörter nicht konkatenieren, da man intuitiv niemals durch das gesamte erste Wort laufen kann um dann irgendwann den Übergang in das zweite Wort zu machen. Man kann aber natürlich die Linkskonkatenation eines unendlichen Wortes mit einem endlichen Wort definieren: Ist $v \in \Sigma^*$ und $w \in \Sigma^\omega$, so ist $vw \in \Sigma^\omega$. Dies wird in natürlicher Weise auf Sprachen fortgesetzt: Ist $U \subseteq \Sigma^*$ und $V \subseteq \Sigma^\omega$, so ist $UV := \{uv \mid u \in U, v \in V\} \subseteq \Sigma^\omega$.

Genauso kann man die unendliche Iteration eines endlichen Wortes $w \in \Sigma^*$ als $w^\omega := www\dots$ definieren und dies ebenfalls auf Sprachen erweitern: Sei $L \subseteq \Sigma^*$. Dann ist $L^\omega := \{w_0w_1w_2\dots \mid w_i \in L \text{ für alle } i \in \mathbb{N}\}$.

Auf dieser Basis lassen sich ω-reguläre Ausdrücke definieren. Diese sind wie reguläre Ausdrücke aufgebaut mit einem zusätzlichen Operator $(\cdot)^\omega$ für die unendliche Iteration analog zum endlichen Iterationsoperator $(\cdot)^*$, dem Kleene-Stern. Beachte, dass nur durch unendliche Iteration aus endlichen Wörtern unendliche gewonnen werden können. Außerdem macht es keinen Sinn, die unendliche Iteration links von einer Konkatenation in einem ω-regulären Ausdruck zuzulassen. Aus diesen Gründen verlangen wir immer eine spezielle Form für ω-reguläre Ausdrücke im Gegensatz zu den regulären Ausdrücken.

Zur Wiederholung: Reguläre Ausdrücke *endlicher Wörter* über Σ sind gegeben durch die folgende Grammatik.

$$\alpha \ ::= \ \emptyset \mid \epsilon \mid a \mid \alpha \cup \alpha \mid \alpha\alpha \mid \alpha^*$$

wobei $a \in \Sigma$.

Definition 21. *Ein ω-regulärer Ausdruck über Σ ist von der Form*

$$\gamma \ ::= \ \bigcup_{i=1}^{n} \alpha_i \beta_i^\omega$$

wobei $n \geq 0$ und α_i, β_i jeweils reguläre Ausdruck und zusätzlich gilt $\varepsilon \notin L(\beta_i)$.

Solch ein ω-regulärer Ausdruck definiert eine ω-Sprache wie folgt.

$$L(\bigcup_{i=1}^{n} \alpha_i \beta_i^\omega) \ := \ \bigcup_{i=1}^{n} L(\alpha_i) L(\beta_i)^\omega$$

Der Fall $n = 0$ liefert $\emptyset$ als Sprache.

Eine Sprache $L \subseteq \Sigma^\omega$ heißt ω-regulär *gdw. es einen ω-regulären Ausdruck γ gibt mit $L = L(\gamma)$.*

Die Nebenbedingung $\varepsilon \notin \beta_i$ ist erforderlich, da sonst auch endliche Wörter erzeugt würden: $(\epsilon \cup a)^\omega$ enthält ja a^*, sowie a^ω. Man kann alternativ auch ε-freie reguläre Ausdrücke durch die Grammatik

$$\beta \ ::= \ a \mid \beta \cup \beta \mid \beta\beta \mid \beta\beta^*$$

definieren, wobei $a \in \Sigma$.

Beispiel 7. Die Menge aller Wörter, in denen unendlich viele b's vorkommen, die jeweils durch eine gerade Anzahl von a's getrennt werden, wird durch den Ausdruck $((aa)^* b)^\omega$ beschrieben.

Sei $\Sigma = \{a, b, c\}$. Die Sprache $L = \{w \in \Sigma^\omega \mid |w|_a = \infty \Rightarrow |w|_b = \infty\}$ ist ebenfalls ω-regulär. Einen entsprechenden Ausdruck kann man dadurch konstruieren, dass man die Bedingung in ihrer Definition in eine Disjunktion umschreibt. Ein Wort w ist in L genau dann, wenn es unendlich viele b's enthält oder nur endlich viele a's. Damit wird sie von dem ω-regulären Ausdruck

$$(a \cup b \cup c)^* \big((a \cup c)^* b\big)^\omega \ \cup \ (a \cup b \cup c)^* (b \cup c)^\omega$$

beschrieben.

Aus der Definition der ω-Regularität über diese entsprechenden Ausdrücke folgt sofort die folgende Charakterisierung der ω-regulären Sprachen, die wir im folgenden auch benutzen werden.

Proposition 8. *Sei $L \subseteq \Sigma^\omega$. Es gilt: L is ω-regulär gdw. es ein $n \in \mathbb{N}$ und reguläre Sprachen $U_1, \ldots, U_n$ und nicht-leere reguläre Sprachen $V_1, \ldots, V_n$ gibt mit $V_i \neq \{\epsilon\}$ für alle $i = 1, \ldots, n$, so dass $L = \bigcup_{i=1}^n U_i V_i^\omega$.*

Lemma 9. *Die Klasse der ω-regulären Sprachen ist abgeschlossen unter Vereinigung und Linkskonkatenation mit regulären Sprachen endlicher Wörter.*

Beweis. Abschluss unter Vereinigung ist offensichtlich. Der Abschluss unter dieser Linkskonkatenation folgt aus der Kommutativität von Vereinigung und Konkatenation: Sei α ein regulärer Ausdruck und $\bigcup_{i=1}^n \alpha_i \beta_i^\omega$ ein ω-regulärer Ausdruck. Dann gilt

$$L(\alpha)L(\bigcup_{i=1}^n \alpha_i \beta_i^\omega) \;=\; \bigcup_{i=1}^n L(\alpha)L(\alpha_i \beta_i^\omega) \;=\; L(\bigcup_{i=1}^n (\alpha\alpha_i)\beta_i^\omega)$$

wegen des Konkatenationsabschlusses der regulären Sprachen, womit ein ω-regulärer Ausdruck für diese Linkskonkatenation gefunden ist. □

Weitere Abschlusseigenschaften wie die unter Durchschnitt oder sogar Komplement sind nicht so einfach mithilfe dieser Ausdrücke zu sehen. Dazu braucht man wie bei den regulären Sprachen ein Automatenmodell.

7.2 Büchi-Automaten

Wie oben bereits angedeutet, unterscheiden sich Büchi-Automaten syntaktisch nicht von endlichen Automaten. Lediglich ihre Semantik über unendlichen Wörtern wird durch unendliches Durchlaufen von Endzuständen erklärt.

Definition 22. *Ein* Büchi-Automat *(NBA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$, wobei*

- *Q eine endliche Menge von* Zuständen *ist,*
- *Σ ein endliches* Alphabet *ist,*
- *$q_0 \in Q$ ein ausgezeichneter* Anfangszustand *ist,*
- *$\delta : Q \times \Sigma \to 2^Q$ die* Transitionsfunktion *ist und*
- *$F \subseteq Q$ eine ausgezeichnete Menge von* Endzuständen *ist.*

Ein Lauf *von $\mathcal{A}$ auf einem Wort $w = a_0 a_1 a_2 \ldots \in \Sigma^\omega$ ist eine unendliche Folge von Zuständen $\rho = q_0, q_1, q_2, \ldots$ beginnend mit dem Anfangszustand q_0, so dass $q_i \in \delta(q_{i-1}, a_i)$ für alle $i \geq 1$.*

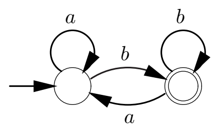
Mit $Inf(\rho)$ bezeichnen wir die Menge derjenigen Zustände, die unendlich oft in ρ vorkommen. Beachte, dass $Inf(\rho)$ niemals leer ist, da Q endlich ist.

Ein solcher Lauf ρ ist akzeptierend, *wenn $Inf(\rho) \cap F \neq \emptyset$, also wenn ein Endzustand unendlich oft durchlaufen wird. Aufgrund der Endlichkeit von Q ist dies gleichbedeutend damit, dass die Endzustandsmenge F in ρ unendlich oft (also immer wieder) besucht wird.*

Die Sprache $L(\mathcal{A})$ des Büchiautomaten ist definiert als die Menge aller ω-Wörter, für die ein akzeptierender Lauf existiert. Es ist wohlgemerkt nicht verlangt, dass jeder Lauf akzeptierend sein muss.

Der NBA $\mathcal{A}$ heißt deterministisch *(DBA), falls $|\delta(q,a)| = 1$ für alle $q \in Q$ und alle $a \in \Sigma$.*

Beispiel 8. Sei $\Sigma = \{a, b\}$ und $L_1 = L((a^*b)^\omega)$ die Sprache aller Wörter, die unendlich viele b's enthalten. Diese kann von einem NBA erkannt werden, welcher grafisch wie folgt repräsentiert ist.



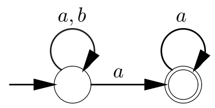
Wie bei endlichen Automaten lassen sich Zustandsmenge und Transitionsrelation als gerichteter, kantenbeschrifteter Graph darstellen. Die Endzustandsmenge wird durch doppelte Kreise angezeigt, und der Startzustand hat einen eingehenden Pfeil, der von außen kommt.

Es sollte klar sein, dass dieser NBA $\mathcal{A}$ wirklich L erkennt. Hat ein vorgelegtes Wort unendlich viele b's, so muss $\mathcal{A}$ zwangsläufig den rechten Zustand unendlich oft besuchen. Da dieser ein Endzustand ist, wird in solch einem Fall das Wort akzeptiert. Hat umgekehrt ein Wort nur endlich viele Vorkommen von b, so wird $\mathcal{A}$ irgendwann nur noch im linken Zustand verweilen und dadurch den einzigen Endzustand nur endlich oft besuchen. Dadurch wird das Wort nicht akzeptiert.

Beachte, dass der NBA aus diesem Beispiel sogar ein DBA ist. Dies wird in der Argumentation über seine Korrektheit benutzt, indem implizit davon gesprochen wird, dass er nur einen einzigen Lauf auf jedem Wort hat. Dies ist wie bei Automaten auf endlichen Wörtern ein charakteristisches Merkmal für Determinismus.

Beispiel 9. Sei weiterhin $\Sigma = \{a, b\}$ aber $L_2 = L((a \cup b)^* a^\omega)$ die Sprache der Wörter, die nur endlich viele b's enthalten. Beachte, dass L_2 das Komplement von L_1 ist: $L_2 = \Sigma^\omega \setminus L_1$.

L_2 wird von dem folgenden NBA erkannt.



Beachte, dass der NBA in diesem Beispiel kein DBA ist. Dies ist kein Zufall, denn man kann zeigen, dass L_2 von keinem DBA erkannt werden kann. Dies ist ein wichtiges Resultat, welches in Kontrast zu den Resultaten über endliche Automaten auf endlichen Wörtern steht. Es gilt ja, dass jede von einem NFA erkannte Sprache auch von einem DFA erkannt wird. Entsprechendes gilt also nicht für NBAs und DBAs. Damit sind nichtdeterministische Büchi-Automaten also in ihrer Ausdrucksstärke echt stärker als deterministische Büchi-Automaten.

Theorem 17. *Es gibt Sprachen, die von einem NBA, aber nicht von einem DBA erkannt werden.*

Beweis. Als Zeugen für diese Aussagen nehmen wir gerade die Sprache $L = \{w \in \{a,b\}^\omega \mid |w|_b < \infty\}$ aus Bsp. 9. Dieses Beispiel hat gezeigt, dass L von einem NBA erkannt werden kann. Sei nun angenommen, dass es einen DBA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ gibt, so dass $L = L(\mathcal{A})$.

Betrachte zunächst das Wort $w_0 = ba^\omega$. Da $w_0 \in L$, muss es einen akzeptierenden Lauf $\rho_0 = q_0, q_1, \ldots$ von $\mathcal{A}$ auf w_0 geben. Dieser muss unendliche viele Endzustände durchlaufen, also gibt es auch ein $i_0 \in \mathbb{N}$, so dass ρ_0 nach Lesen des einzigen b und i_0 weiteren Schritten einen Endzustand besitzt.

Betrachte als nächstes das Wort $w_1 = ba^{i_0}ba^\omega$. Da $w_1 \in L$ muss es auch einen akzeptierenden Lauf ρ_1 von $\mathcal{A}$ auf w_1 geben. Da zusätzlich w_0 und w_1 in den ersten $i_0 + 1$ Buchstaben übereinstimmen und $\mathcal{A}$ deterministisch ist, müssen auch ρ_0 und ρ_1 in den ersten $i_0 + 2$ Zuständen übereinstimmen. Da ρ_2 akzeptierend ist, gibt es wiederum ein $i_1 \in \mathbb{N}$, so dass ρ_2 nach Lesen des letzten b und i_1 weiteren Schritten einen Endzustand besucht.

Dann betrachtet man das Wort $w_2 = ba^{i_0}ba^{i_1}ba^\omega$ und führt diese Argumentaton fort. So kann man ein unendliches Wort $w = ba^{i_0}ba^{i_1}ba^{i_2}ba^{i_3}b\ldots$ konstruieren, welches von $\mathcal{A}$ akzeptiert wird, da dieser Automat darauf einen Lauf hat, welcher sich als Limes der $\rho_0, \rho_1, \ldots$ ergibt und unendliche viele Endzustände aufweist. Dies ist aber ein Widerspruch zu der Annahme $L(\mathcal{A}) = L$, da $w \notin L$. □

Beispiel 10. Sei nun $\Sigma = \{a,b,c\}$ und L die Sprache aller Wörter, in denen auf jedes b irgendwann ein c folgt. Es ist

$$L \;=\; L\big((a \cup c)^\omega \cup (a \cup b \cup c)^* c(a \cup c)^\omega \cup (a \cup b \cup c)^*(b(a \cup b \cup c)^* c)^\omega\big) \;.$$

Ein passender Büchi-Automat $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ für L ist gegeben durch die Zustandsmenge $Q = \{q_0, q_1\}$, die Transitionsfunktion

$$\begin{array}{rcrcl} & & \delta(q_0,b) & := & \{q_1\} \\ \delta(q_0,a) & = & \delta(q_0,c) & := & \{q_0\} \\ & & \delta(q_1,c) & := & \{q_0\} \\ \delta(q_1,a) & = & \delta(q_1,b) & := & \{q_1\} \end{array}$$

und die Endzustandsmenge $F = \{q_0\}$.

In den obigen Beispielen wurde deutlich, dass sich zumindest einige Sprachen sowohl über NBAs wie auch über ω-reguläre Ausdrücke definieren lassen. Dies ist kein Zufall, denn diese beiden Spezifikationsformalismen für ω-Sprachen sind in der Tat äquivalent. Man beachte, dass dies im Gegensatz zur Determinisierbarkeit eine Eigenschaft ist, die sich aus der Welt der endlichen Wörter auf die Welt der unendlichen Wörter überträgt. Bevor wir sie beweisen, überlegen wir uns noch drei Konstruktionen auf NFAs und NBAs.

Lemma 10. *Seien $\mathcal{A}, \mathcal{B}$ zwei NBAs. Dann existiert ein NBA $\mathcal{C}$ mit $L(\mathcal{C}) = L(\mathcal{A}) \cup L(\mathcal{B})$.*

Beweis. Übung.

Lemma 11. *Seien $\mathcal{A}$ ein NFA und $\mathcal{B}$ ein NBA. Dann existiert ein NBA $\mathcal{C}$ mit $L(\mathcal{C}) = L(\mathcal{A})L(\mathcal{B})$.*

Beweis. Übung.

Lemma 12. *Sei $\mathcal{A}$ ein NFA. Dann existiert ein NBA $\mathcal{B}$ mit $L(\mathcal{B}) = L(\mathcal{A})^\omega$.*

Beweis. Übung.

Theorem 18 (Büchi). *Eine Sprache L ist ω-regulär genau dann, wenn ein Büchi-Automat $\mathcal{A}$ existiert mit $L(\mathcal{A}) = L$.*

Beweis. "$\Rightarrow$" Sei ein ω-regulärer Ausdruck $\gamma = \bigcup_{i=1}^{n} \alpha_i \beta_i^\omega$ gegeben. Aus dem entsprechenden Satz über die Äquivalenz von regulären Ausdrücken und endlichen Automaten über endlichen Wörtern erhalten wir NFAs $\mathcal{A}_i, \mathcal{B}_i$ mit $L(\mathcal{A}_i) = L(\alpha_i)$ und $L(\mathcal{B}_i) = L(\beta_i)$ für $i = 1, \ldots, n$. Durch sukzessive Anwendung von Lemmas 10–12 erhält man auch einen NBA für $L(\gamma)$.

"$\Leftarrow$" Ist umgekehrt ein Büchiautomat $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ gegeben, so bezeichne $L_{q,q'}$ die Sprache aller Wörter, für die ein endlicher Lauf von q nach q' existiert. Diese ist offensichtlich regulär, denn sie wird beispielsweise von dem NFA erkannt, den man aus $\mathcal{A}$ erhält, wenn man q zum Anfangszustand und q' zum einzigen Endzustand macht. Dann gilt

$$L(\mathcal{A}) = \bigcup_{q \in F} L_{q_0,q} L_{q,q}^\omega \tag{7.1}$$

Die Richtung "$\supseteq$" gilt, da jedes Wort in der rechten Seite offensichtlich einen Lauf in $\mathcal{A}$ hat, welcher unendlich oft einen Endzustand durchläuft. Die Richtung "$\subseteq$" gilt, das jeder akzeptierende Lauf von $\mathcal{A}$ einen bestimmten Endzustand unendlich oft durchlaufen muss. Dieser unterteilt das gelesene Wort in unendlich endliche viele Stücke, die jeweils in den Disjunkten der rechten Seite liegen.

Da es zu jeder regulären Sprachen endlicher Wörter aber auch einen regulären Ausdruck gibt, der sie beschreibt, stellt die rechte Seite der Gleichung (7.1) einen ω-regulären Ausdruck dar. □

Mithilfe dieser Äquivalenz kann man Abschlusseigenschaften der Klasse der ω-regulären Sprachen durch Angabe entsprechender Automatenkonstruktionen zeigen. Wir tun das jetzt sofort für den Abschluss unter Schnitt. Im nächsten Kapitel beweisen wir dann ebenso den Abschluss unter Komplement, welcher sich allerdings im Gegensatz zur Situation bei den endlichen Automaten recht kompliziert gestaltet. Der Grund dafür ist zum einen die bereits bewiesene Tatsache, dass DBA echt schwächer als NBA sind. Erschwerend hinzu kommt noch die Tatsache, dass selbst DBA nicht einfach durch Vertauschen der End- mit den Nicht-Endzuständen komplementiert werden können. Ein solcher DBA würde ja akzeptieren, wenn ein Nicht-Endzustand des ursprünglichen Automaten unendlich oft erreicht wird. Das bedeutet aber nicht, dass kein Endzustand unendlich oft erreicht wird.

Theorem 19. *Seien $\mathcal{A}$ und $\mathcal{B}$ jeweils NBAs mit n bzw. m Zuständen. Dann existiert en NBA $\mathcal{A}$ mit höchstens $3 \cdot n \cdot m$ Zuständen, so dass $L(\mathcal{C}) = L(\mathcal{A}) \cap L(\mathcal{B})$.*

Beweis. Die Grundidee liefert der von NFAs bekannte Produktautomat. Diese bedarf jedoch noch einer kleinen Modifikation um der Tatsache zu begegnen, dass ein Wort w akzeptierende Läufe in $\mathcal{A}$ und $\mathcal{B}$ jeweils zu verschiedenen Momenten Endzustände besuchen können. Wir fügen also noch eine weitere Komponente hinzu, welche protokolliert, ob zuletzt kein Endzustand, nur einer in $\mathcal{A}$ oder einer in beiden gesehen wurde. Ist letzteres der Fall, so setzt man diesen Zähler wieder zurück.

Formal ist die Konstruktion folgendermaßen gegeben. Seien $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ und $\mathcal{B} = (Q', \Sigma, q'_0, \delta', F')$. Definiere

$$\mathcal{C} \;:=\; (Q \times Q' \times \{0,1,2\}, \Sigma, (q_0, q'_0, 0), \Delta, Q \times Q' \times \{2\})$$

mit

$$\Delta((q, q', i), a) \;:=\; \{(p, p', j) \mid\; p \in \delta(q, a), p' \in \delta'(q', a)\}$$

wobei

$$j = \begin{cases} 1, & \text{falls } i = 0 \text{ und } p \in F, \text{ oder } i = 1 \text{ und } p' \notin F' \\ 2, & \text{falls } i = 1 \text{ und } p' \in F' \\ 0, & \text{sonst} \end{cases}$$

Die angegebene Größenbeschränkung an $\mathcal{C}$ ist offensichtlich. Es bleibt noch die Korrektheit zu zeigen.

"$\supseteq$" Sei $w \in L(\mathcal{A}) \cap L(\mathcal{B})$. Dann existieren akzeptierende Läufe $\rho = q_0, q_1, \ldots$ von $\mathcal{A}$ auf w und $\rho' = q'_0, q'_1, \ldots$ von $\mathcal{B}$ auf w. Beachte, dass die Transitionsfunktion Δ von $\mathcal{C}$ deterministisch in ihrer dritten Komponente ist. D.h. bei einem Übergang von (q, q', i) zu (p, p', j) ist j eindeutig durch q, q' und i gegeben. Somit gibt es einen Lauf $\rho'' = (q_0, q'_0, i_0), (q_1, q'_1, i_1), \ldots$ der in seinen ersten beiden Komponenten die Läufe ρ und ρ' simuliert. Setzt man nun $i_0 = 0$ fest, so beginnt dieser im Anfangszustand und alle i_j, $j \in \mathbb{N}$, sind damit ebenfalls festgelegt. Man beachte nun, dass es unendlich viele $i_j = 2$ geben muss, da ρ unendlich oft Zustände in F und ρ' unendlich oft Zustände in F' durchläuft. Dadurch hat die Sequenz $i_0, i_1, i_2, \ldots$ die Form $(0^+1^+2^+)^\omega$, und damit ist ρ'' akzeptierender Lauf von $\mathcal{C}$ auf w.

"$\subseteq$" Analog. $\square$