Tafelmitschriften zur Vorlesung „Automatentheorie und ihre Anwendungen“ im Wintersemester 2019/20

Prof. Dr. Thomas Schneider
AG Theorie der Künstlichen Intelligenz
Fachbereich 3
Universität Bremen

Stand: 17. Januar 2020

Inhaltsverzeichnis

I. Endliche Automaten auf endlichen Wörtern **3**

II. Endliche Automaten auf endlichen Bäumen **8**

III. Endliche Automaten auf unendlichen Wörtern **24**

IV. Alternierung **50**

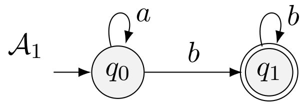
V. Endliche Automaten auf unendlichen Bäumen **60**

Teil I.

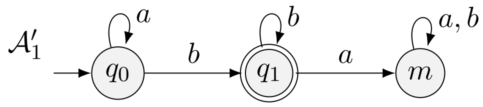
Endliche Automaten auf endlichen Wörtern

T1.1 Beispiel für Papierkorbzustand

Der NEA $\mathcal{A}_1$ von Folie 1.10



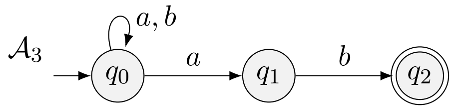
ist nur deshalb kein DEA, weil Zustand q_1 keine ausgehende b-Kante besitzt. Durch Hinzufügen eines neuen Zustands m (für „Mülleimer“ oder „Papierkorb“) wird $\mathcal{A}_1$ zu einem DEA $\mathcal{A}'_1$, ohne dass das Akzeptanzverhalten sich ändert:



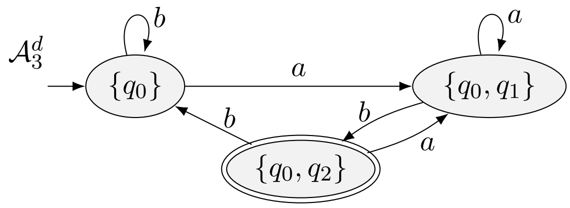
Diese Konstruktion macht aber beispielsweise aus dem NEA $\mathcal{A}_3$ von Folie 1.10 *keinen* DEA, denn dabei behält q_0 seine *zwei* a-Nachfolger.

T1.2 Beispiel für die Potenzmengenkonstruktion

Wir betrachten den NEA $\mathcal{A}_3$ von Folie 1.10:

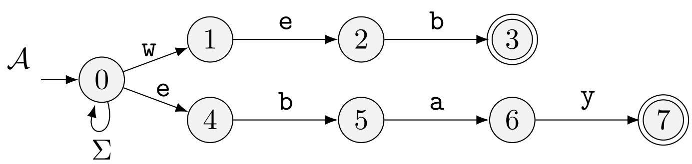


Wenn man die Potenzmengenkonstruktion (Folie 1.13) ausführt, und dabei die unerreichbaren Zustände ignoriert, dann erhält man folgenden DEA $\mathcal{A}_3^d$.



T1.3 Beispiel-NEA für Stichwortsuche

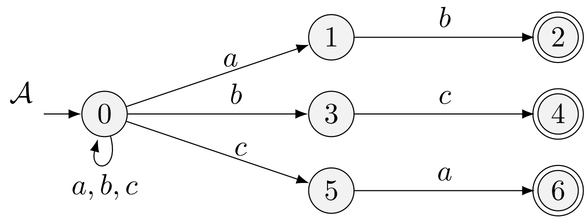
Seien $w_1 = \texttt{web}$ und $w_2 = \texttt{ebay}$, und sei Σ die Menge aller ASCII-Zeichen. Folgender NEA $\mathcal{A}$ akzeptiert genau die Dokumente (Zeichenketten), die w_1 oder w_2 als Teilwort enthalten. Dies gilt aber nur, wenn die leicht geänderte Akzeptanzbedingung von Folie 1.17 zugrunde gelegt wird: $\mathcal{A}$ akzeptiert, sobald ein akzeptierender Zustand erreicht wird, auch wenn das Eingabewort zu diesem Zeitpunkt noch nicht zu Ende gelesen wurde.



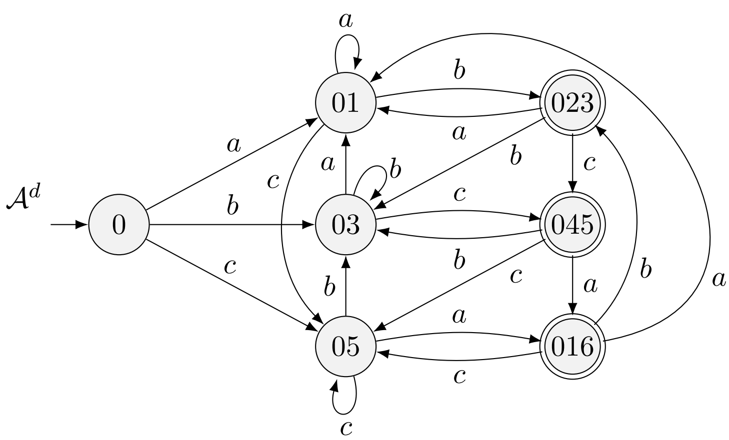
Die Beschriftung Σ der Schleife am Zustand 0 steht für die Folge aller Symbole aus Σ.

T1.4 Beispiel-DEA für Stichwortsuche

Um die Determinisierung in diesem speziellen Fall zu demonstrieren, wählen wir ein einfacheres Beispiel der Stichwortsuche: wir beschränken uns auf das Alphabet $\Sigma = a, b, c$ und die Stichwörter $w_1 = ab$, $w_2 = bc$ und $w_3 = ca$. Analog zu T1.3 erhält man folgenden NEA $\mathcal{A}$.



Wendet man die Potenzmengenkonstruktion an und beschränkt sich dabei wieder auf die erreichbaren Zustände, dann erhält man folgenden DEA $\mathcal{A}^d$.



Dabei sind die Zustandsnamen wie 01, 023 usw. Kurzschreibweisen für die Mengen $\{0, 1\}$, $\{0, 2, 3\}$ usw.

Die Anzahl der Zustände von $\mathcal{A}^d$ ist genauso groß wie die von $\mathcal{A}$, und man kann zeigen, dass dies für jeden NEA $\mathcal{A}$ der Fall ist, den man für die Stichwortsuche gemäß der Beispiele in T1.3 und T1.4 konstruiert. Der DEA hat also immer nur $|w_1| + \cdots + |w_n|$ Zustände, wenn $w_1, \ldots, w_n$ die Stichwörter sind.

T1.5 Beispiel für die Anwendung des Pumping-Lemmas

Wir betrachten die Sprache $L = \{a^n b^n \mid n \geq 0\}$.

Vorbetrachtung für Lesende, die mehr Hintergrundinfo benötigen. Intuitiv kann diese Sprache nicht NEA-erkennbar sein, weil NEAs nicht unbeschränkt zählen können – insbesondere kann ein NEA nicht die Anzahl der gelesenen a's speichern, um diese mit der Anzahl der b's zu vergleichen. Dies bleibt aber ein intuitives Argument, denn es basiert auf der (schwer zu beweisenden) Annahme, dass die skizzierte Vorgehensweise die einzig mögliche ist.

Um unanfechtbar zu beweisen, dass L nicht NEA-erkennbar ist, verwenden wir die Kontraposition des Pumping-Lemmas. Wir müssen also zeigen:

1. Für alle $p \geq 0$
2. gibt es ein Wort $w \in L$ mit $|w| \geq p$, so dass gilt:
3. für alle Zerlegungen $w = xyz$ mit $y \neq \varepsilon$ und $|xy| \leq p$
4. gibt es ein $i \geq 0$ mit $xy^i z \notin L$.

In den Schritten 1 und 3 („für alle“) müssen wir für eine *beliebige* Zahl p bzw. Zerlegung xyz argumentieren; in Schritten 2 und 4 („es gibt“) genügt es, ein Wort w bzw. ein $i \geq 0$ zu wählen (in Abhängigkeit von p bzw. xyz).

Eigentlicher Beweis. Sei $p \geq 0$ beliebig. Wir wählen $w = a^p b^p$, für das offensichtlich $w \in L$ mit $|w| \geq p$ gilt. Sei nun $w = xyz$ eine beliebige Zerlegung mit $y \neq \varepsilon$ und $|xy| \leq p$. Wegen $|xy| \leq p$ kann y nur aus a's bestehen. Wegen $y \neq \varepsilon$ muss y mindestens ein a enthalten. Wenn wir also nun $i = 0$ wählen, dann hat das Wort $xy^0 z = xz$ mindestens ein a weniger als $w = xyz$, aber es hat dieselbe Anzahl b's wie w. Deshalb kann $xy^0 z$ nicht mehr von der Form $a^n b^n$ sein; also ist $xy^0 z \notin L$.

T1.6 Beispiel für die Anwendung des Satzes von Myhill-Nerode

Wir betrachten wieder die Sprache $L = \{a^n b^n \mid n \geq 0\}$. Es ist leicht zu sehen, dass für zwei beliebige Zahlen k_1, k_2 mit $k_1 \neq k_2$ gilt: $a^{k_1} \not\sim_L a^{k_2}$. Dies ist so, weil beispielsweise für $w = b^{k_1}$ zwar das Wort $a^{k_1} w$ in L liegt, aber das Wort $a^{k_2} w$ nicht. Deshalb bilden alle a^k, $k \geq 0$, paarweise verschiedene Äquivalenzklassen, und somit ist der Index von $\sim_L$ unendlich. Mit dem Satz von Myhill-Nerode folgt, dass L nicht NEA-erkennbar ist.

T1.7 Beispiel einer Polynomialzeitreduktion

Wir betrachten die folgenden Mengen.

$$\begin{aligned} M &= \{\mathcal{A} \mid \mathcal{A} \text{ ist NEA}\} \\ X &= \{\mathcal{A} \mid \mathcal{A} \text{ ist NEA, } L(\mathcal{A}) \neq \emptyset\} \end{aligned}$$

$$\begin{aligned} M' &= \{(G, s, t) \mid G = (V, E) \text{ ist gerichteter Graph und } s, t \in V\} \\ X' &= \{(G, s, t) \mid (G, s, t) \in M' \text{ und es gibt Pfad von } s \text{ nach } t \text{ in } G\} \end{aligned}$$

Also ist X das Komplement des Leerheitsproblems für NEAs und X' das Erreichbarkeitsproblem für gerichtete Graphen.

Es gilt $X \leq_{\mathrm{p}} X'$, was durch folgende Polynomialzeitreduktion π bezeugt wird. Gegeben $\mathcal{A} = (Q, \Sigma, \Delta, I, F) \in M$, definiere $\pi(\mathcal{A}) = (G_{\mathcal{A}}, s, t)$ so, dass die Knoten von $G_{\mathcal{A}}$ die Zustände von $\mathcal{A}$ plus zwei neue Knoten s, t sind und die Kanten genau den Übergängen von $\mathcal{A}$ entsprechen (wenn man die Zeichen ignoriert); zusätzlich hat G Kanten von s zu allen Anfangszuständen sowie von allen akzeptierenden Zuständen zu t. Genauer:

$$\begin{aligned} G_{\mathcal{A}} &= (V, E) \qquad \text{mit} \\ V &= Q \uplus \{s, t\} \\ E &= \{(q, q') \mid (q, a, q') \in \Delta, a \in \Sigma\} \cup \{(s, q) \mid q \in I\} \cup \{(q, t) \mid q \in F\} \end{aligned}$$

Nun ist leicht zu sehen, dass gilt:

- Wenn $\mathcal{A} \in X$, dann $\pi(\mathcal{A}) \in X'$:

  Da $L(\mathcal{A}) \neq \emptyset$, gilt $q_0 \vdash_{\mathcal{A}}^{w} q_f$ für ein Wort $w \in \Sigma^*$ und Zustände $q_0 \in I$, $q_f \in F$. Wegen der Konstruktion von $G_{\mathcal{A}}$ gibt es somit einen Pfad von q_0 nach q_f in $G_{\mathcal{A}}$ und damit auch von s nach t.
- Wenn $\pi(\mathcal{A}) \in X'$, dann $\mathcal{A} \in X$:

  Ein Pfad von s nach t in $G_{\mathcal{A}}$ muss nach Konstruktion auch einen Pfad von einem $q_0 \in I$ zu einen $q_f \in F$ enthalten; die Beschriftungen der entsprechenden Kanten in $\mathcal{A}$ liefern ein Wort $w \in L(\mathcal{A})$, welches $L(\mathcal{A}) \neq \emptyset$ bezeugt.
- π lässt sich in Polynomialzeit berechnen:

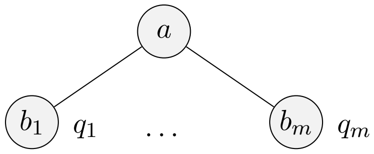
  $G_{\mathcal{A}}$ lässt sich in Polynomialzeit aus $\mathcal{A}$ konstruieren.

Teil II.

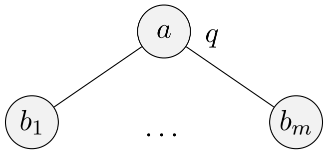
Endliche Automaten auf endlichen Bäumen

T2.1 Skizze zur Intuition der Übergänge im Baumautomaten

Der Übergang $a(q_1, \dots, q_m) \to q$ bedeutet:
Wenn $\mathcal{A}$ in Position p Zeichen a liest und in p's Kindern Zustände $q_1, \dots, q_m$ eingenommen hat,



dann darf $\mathcal{A}$ in p Zustand q einnehmen.

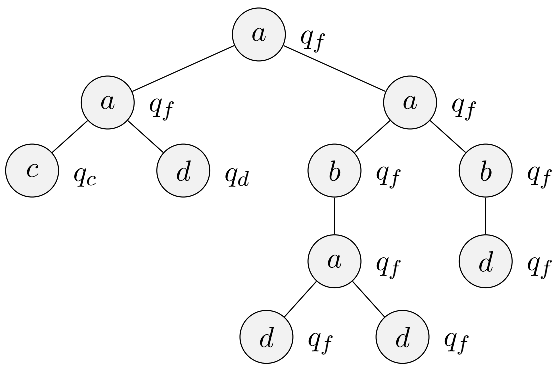


T2.2 Beispiel-Run

Sei $\Sigma = \{a/2,\ b/1,\ c/0,\ d/0\}$ und $\mathcal{A} = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\})$ mit

$$\begin{aligned}\Delta = \{ \qquad\qquad c \to q_c, \quad d \to q_d, \quad d \to q_f,\\ a(q_c, q_d) \to q_f,\\ a(q_f, q_f) \to q_f,\\ b(q_f) \to q_f \quad \}.\end{aligned}$$

Dann gibt es folgenden Run von $\mathcal{A}$ auf dem Baum $T = a(a(cd)a(b(a(dd))b(d)))$.



T2.3 Korrektheit der Potenzmengenkonstruktion

Es ist noch zu zeigen: $L(\mathcal{A}^d) = L(\mathcal{A})$. Dazu zeigen wir zunächst zwei Hilfsaussagen (HA).

HA1. Für jeden Baum $T = (P, t)$ gibt es *genau einen* Run des DEBAs $\mathcal{A}^d$ auf T; wir nennen diesen Run r_T^d.

Beweis von HA1. Dazu definieren wir $r_T^d(p)$ induktiv über die Höhe der Position p in T (zur Erinnerung: „Höhe" ist so definiert, dass Blätter die Höhe 0 haben). Simultan zeigen wir, dass zu jedem Zeitpunkt für den bis dahin definierten Teilrun r_T^d gilt:

(a) r_T^d erfüllt die Eigenschaften eines Runs (Def. 2.3).

(b) Für alle Runs r von $\mathcal{A}^d$ auf T und alle Nachfolger p' der Position p (einschließlich p selbst) gilt $r(p') = r_T^d(p')$.

Induktionsanfang. Sei p eine Position der Höhe 0 (also ein Blatt) mit $t(p) = a \in \Sigma_0$. Wir betrachten die *eindeutig bestimmte* Menge $S \subseteq Q$ mit

$$a \to S \in \Delta^d \tag{$*$}$$

und setzen $r_T^d(p) = S$. Dann gilt (a) wegen ($*$), und (b) gilt, weil S nach Definition von Δ^d eindeutig bestimmt ist.

Induktionsschritt. Sei p eine Position der Höhe ≥ 0 mit Kindern $p1, \dots, pm$, $m > 0$, und sei $t(p) = a \in \Sigma_m$. Weil die Höhe der pi geringer ist als die von p, ist $S_i := r_T^d(pi)$ für alle $i \leq m$ definiert, und nach Induktionsvoraussetzung (IV) erfüllt r_T^d bis dahin (a) und (b). Wir betrachten die *eindeutig bestimmte* Menge $S \subseteq Q$ mit

$$a(S_1, \dots, S_m) \to S \in \Delta^d \tag{$**$}$$

und setzen $r_T^d(p) = S$. Dann gilt (a) wegen IV und ($*$), und (b) gilt wegen IV und weil S nach Definition von Δ^d eindeutig bestimmt ist.

Dies beendet den Beweis von HA1.

Wir benutzen ab jetzt r_T^d.

HA2. Für alle Bäume $T = (P, t)$ und alle Positionen $p \in P$ gilt:

$$r_T^d(p) = \{q \in Q \mid \text{es gibt Run } r \text{ von } \mathcal{A} \text{ auf } T_p \text{ mit } r(\varepsilon) = q\}$$

HA2 gibt genau die Intuition der Potenzmengenkonstruktion wieder: der eindeutig bestimmte Run von $\mathcal{A}^d$ auf einem Baum T „versammelt" in allen Positionen p genau diejenigen Zustände, die sämtliche Runs von $\mathcal{A}$ dort annehmen können. Dazu gehören auch Teilruns, die sich oberhalb der Position p nicht mehr bis zur Wurzel fortsetzen lassen; deshalb muss man sich in der Formulierung auf den Teilbaum T_p beschränken.

Beweis von HA2. Wir gehen per Induktion über die Höhe von p in T vor.

Induktionsanfang. Sei p Blattposition mit $t(p) = a \in \Sigma_0$. Wegen der Definition eines Runs und des Beweises von HA1 ist $r_T^d(p)$ diejenige Menge $S \subseteq Q$, für die $a \to S \in \Delta^d$ gilt. Mit der Definition von Δ^d erhalten wir daraus

$$r_T^d(p) = \{q \in Q \mid a \to q \in \Delta\}.$$

Da die Bedingung $a \to q \in \Delta$ wegen der Definition eines Runs (für Blattpositionen) genau dann gilt, wenn es einen Run r von $\mathcal{A}$ auf T_p gibt mit $r(\varepsilon) = q$, folgt die Behauptung.

Induktionsschritt. Sei p eine Position mit Kindern $p1, \dots, pm$, $m > 0$, und sei $t(p) = a \in \Sigma_m$. Weil die Höhe der pi geringer ist als die von p, ist die Induktionsvoraussetzung (IV) anwendbar und liefert:

$$S_i := r_T^d(p) = \{q \in Q \mid \text{es gibt Run } r_i \text{ von } \mathcal{A} \text{ auf } T_{pi} \text{ mit } r_i(\varepsilon) = q_i\} \qquad (***)$$

Wegen der Definition eines Runs und des Beweises von HA1 ist $r_T^d(p)$ diejenige Menge $S \subseteq Q$, für die $a(S_1, \dots, S_m) \to S \in \Delta^d$ gilt. Mit der Definition von Δ^d erhalten wir daraus

$$r_T^d(p) = \{q \in Q \mid \underbrace{\exists q_1 \in S_1 \cdots \exists q_m \in S_m : \underbrace{a(q_1, \dots, q_m) \to q \in \Delta}_{(\times\times)}}_{(\times)}\}.$$

Es bleibt zu zeigen, dass für alle $q \in Q$ die Bedingung $(\times)$ gilt gdw. es einen Run r von $\mathcal{A}$ auf T_p gibt mit $r(\varepsilon) = q$.

„$\Rightarrow$" Wir nehmen an, dass $(\times)$ gilt. Wegen $(***)$ gibt es für jedes q_i aus $(\times)$ einen Run r_i von $\mathcal{A}$ auf T_{pi} mit $r(\varepsilon) = q_i$. Daraus konstruieren wir einen Run r auf T_p wie folgt:

- $r(\varepsilon) = q$
- $r(iw) = r_i(w)$ für alle $i \leq m$ und $w \in \mathbb{N}_+^*$

Da alle r_i Runs sind und die Eigenschaft $(\times\times)$ gilt, ist auch r ein Run von $\mathcal{A}$ auf T_p; außerdem gilt nach Konstruktion wie gewünscht $r(\varepsilon) = q$.

„$\Leftarrow$" Sei r ein Run von $\mathcal{A}$ auf T_p mit $r(\varepsilon) = q$. Um $(\times)$ zu zeigen, definieren wir $q_i := r(i)$ für alle $i \leq m$. Da r ein Run ist, gilt $(\times\times)$. Es bleibt zu zeigen, dass $q_i \in S_i$ für alle $i \leq m$, also wegen $(***)$: dass es Runs r_i von $\mathcal{A}$ auf T_{pi} gibt mit $r_i(\varepsilon) = q_i$. Diese Runs lassen sich nun leicht aus r konstruieren, indem man $r_i(w) = r(iw)$ für alle $i \leq m$ und $w \in \mathbb{N}_+^*$ setzt. Offenbar sind das Runs, weil sie Einschränkungen von r sind; außerdem erfüllen sie $r_i(\varepsilon) = q_i$, weil die q_i so gewählt wurden, dass $r(i) = q_i$ gilt.

Dies beendet den Beweis von HA2, und wir können nun die Hauptaussage beweisen.

$\boldsymbol{L(\mathcal{A}^d) = L(\mathcal{A})}$**.** Dazu beobachten wir:

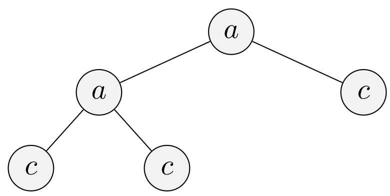
$$
\begin{array}{llll}
T \in L(\mathcal{A}^d) & \text{gdw.} & r_T^d(\varepsilon) \in F^d & \text{(Def. Akzeptanz, HA1)} \\
& \text{gdw.} & r_T^d(\varepsilon) \cap F \neq \emptyset & \text{(Def. } F^d\text{)} \\
& \text{gdw.} & \text{es gibt Run } r \text{ von } \mathcal{A} \text{ auf } T_\varepsilon \text{ mit } r(\varepsilon) \in F & \text{(HA2)} \\
& \text{gdw.} & T \in L(\mathcal{A}) & (T_\varepsilon = T\text{, Def. Akzeptanz)}
\end{array}
$$

T2.4 Vorbetrachtungen für Nicht-Erkennbarkeit

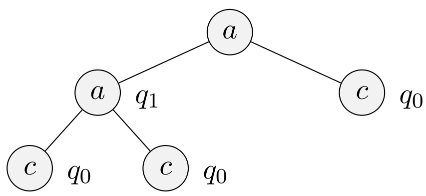
Für den NEBA $\mathcal{A}$ auf Folie 2.29 gilt

$$L(\mathcal{A}) \neq \{T \mid T \text{ hat gerade Höhe}\},$$

denn der Baum



hat zwar gerade Höhe, wird aber nicht von $\mathcal{A}$ akzeptiert, denn wenn es einen Run gäbe, dann müsste dieser wie folgt beginnen:



Dieser Teilrun kann aber nicht mehr zu einem vollständigen Run fortgesetzt werden.

T2.5 Intuitionen für Nicht-Erkennbarkeit

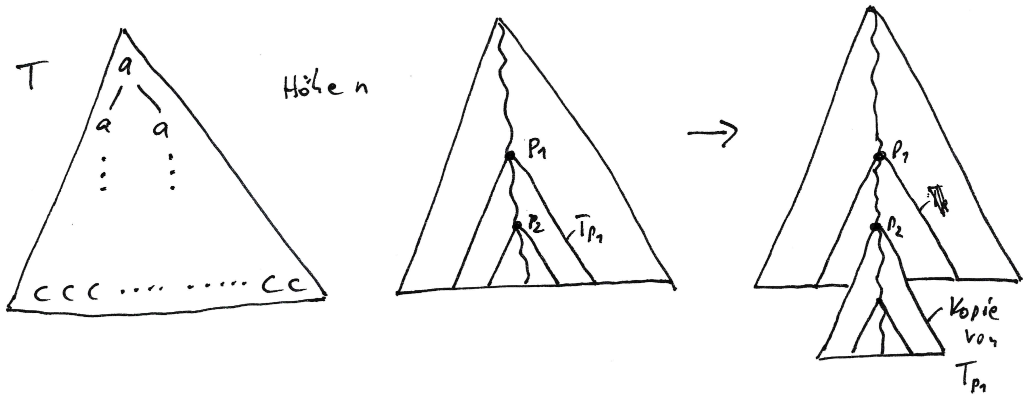
Wir betrachten $L_2 = \{T \mid T \text{ ist vollständiger Binärbaum}\}$, wobei „vollständiger Binärbaum“ bedeutet, dass für jede Nicht-Blattposition p in T gilt:

- p hat genau 2 Kinder $p1, p2$ und
- deren Teilbäume T_{p1}, T_{p2} haben gleiche Höhe.

Ein intuitives, aber möglichst schlüssiges Argument dafür dass L_2 nicht NEBA-erkennbar ist, ist folgendes:

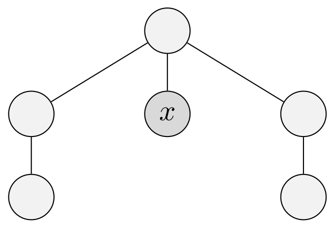
Angenommen L_2 werde von einem NEBA $\mathcal{A}$ erkannt; dieser habe n Zustände. Dann hat $\mathcal{A}$ einen erfolgreichen Run r auf dem vollständigen Binärbaum T der Höhe n. Auf jedem

Pfad hat T aber $n+1$ Knoten, und damit muss auf jedem Pfad in r irgendeinen Zustand doppelt vorkommen. Wenn wir einen Pfad festhalten und die Positionen des doppelten Vorkommens desselben Zustandes p_1 und p_2 sind (mit p_1 Vorgänger von p_2), dann können wir den Teilbaum T_{p_2} durch den höheren Teilbaum T_{p_1} ersetzen und erhalten einen nicht mehr vollständigen Binärbaum, auf dem $\mathcal{A}$ trotzdem einen erfolgreichen Run hat (letzteren erhält man auf die offensichtliche Weise aus r) – ein Widerspruch zur Annahme $L(\mathcal{A}) = L_2$.

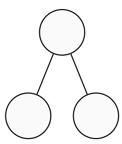
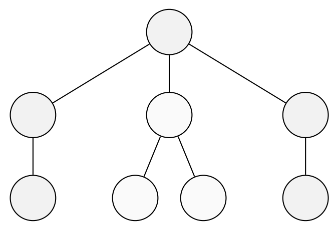


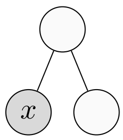
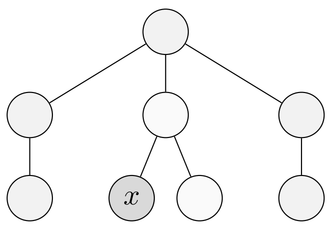
T2.6 Beispiele für Kontexte

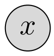
- Unärer Kontext, schematisch:

C 

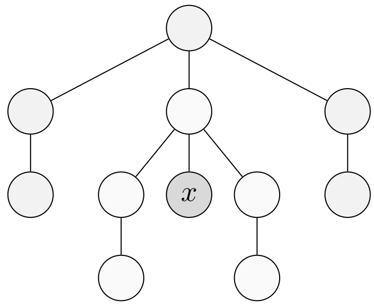
- Einsetzen:

Baum T  $\rightsquigarrow$ Baum $C[T]$ 

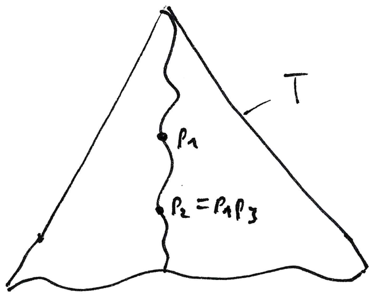
Kontext C'  $\rightsquigarrow$ Kontext $C[C']$ 

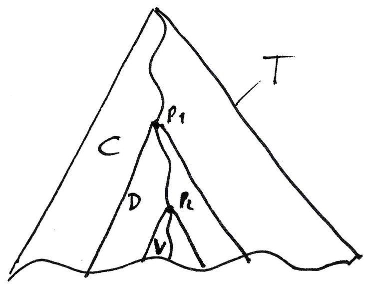
- Trivialer Kontext: C_0 

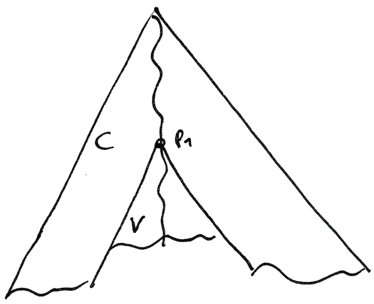
- Iterierte Kontexte:

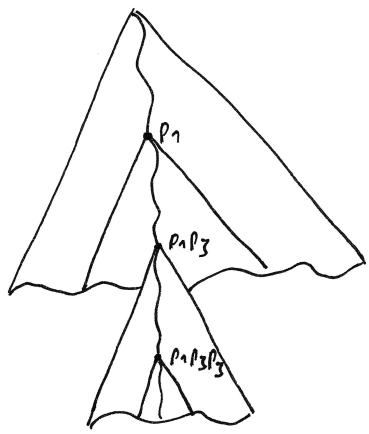
C^2 

T2.7 Illustrationen zum Beweis des Pumping-Lemmas

Folie 2.32: 

Folie 2.33: 

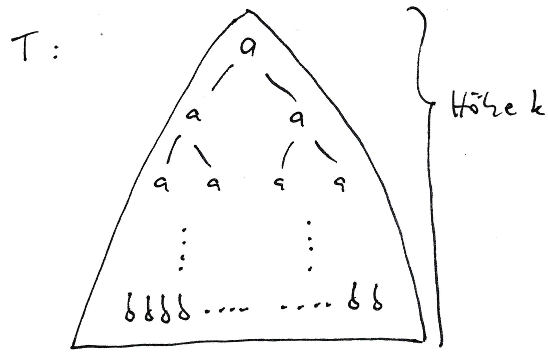
Folie 2.34: 

Folie 2.35: 

T2.8 Anwendung des Pumping-Lemmas

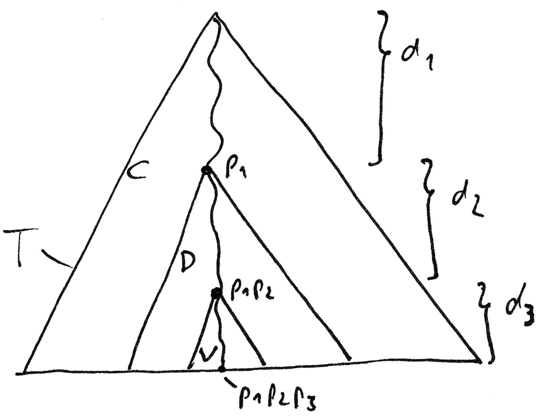
Wir betrachten wieder $L_2 = \{T \mid T \text{ ist vollständiger Binärbaum}\}$, aber diesmal über dem r-Alphabet $\Sigma = \{a/2,\, b/0\}$, denn einstellige Symbole können in einem vollständigen Binärbaum sowieso nicht vorkommen.

Sei $k \geq 0$ beliebig. Wir wählen als T den vollständigen Binärbaum der Höhe k:



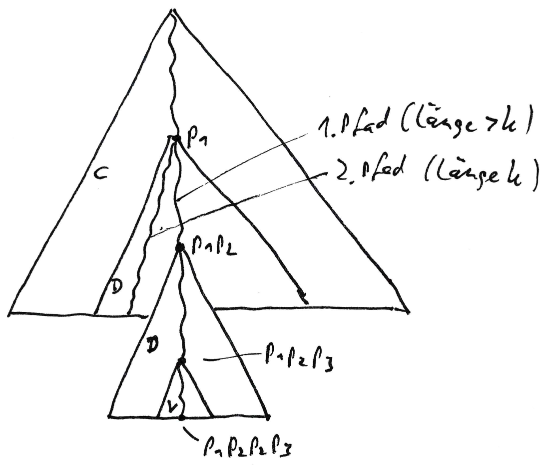
Sei nun $T = C[D[V]]$ eine Zerlegung von T in Kontexte C, D und Baum V mit D nichttrivial. Seien weiterhin

- p_1 die Position von x im Kontext C;
- d_1 die Tiefe von p_1 in C, d. h. $d_1 = |p_1|$;
- p_2 die Position von x im Kontext D;
- d_2 die Tiefe von p_2 in D, d. h. $d_2 = |p_2| > 0$, da D nichttrivial;
- p_3 eine Blattposition in V;
- d_3 die Höhe von V (und damit die Tiefe aller Blätter, denn T ist vollständig).



Weil T vollständig ist, haben alle Pfade in T die Länge $k = d_1 + d_2 + d_3$.

Wir wählen nun $i = 2$ und betrachten den „gepumpten" Baum $C[D^2[V]]$:



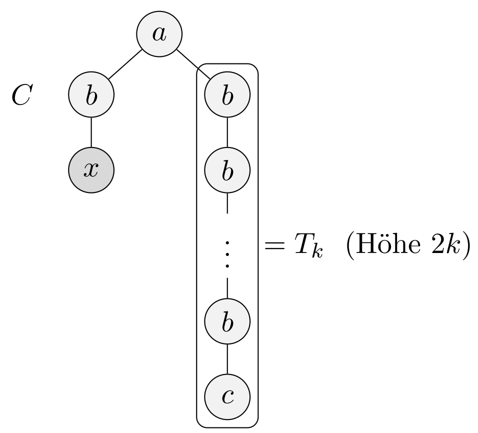
In diesem Baum gibt es einen Pfad der Länge $d_1 + 2d_2 + d_3 > k$, nämlich zur Position $p_1p_2p_2p_3$. Da T vollständig ist, hat p_1 ein Kind, das nicht auf diesem Pfad liegt und dessen Teilbaum also nicht vom Pumpen betroffen ist. Deshalb gibt es in $C[D^2[V]]$ auch einen Pfad der Länge k, woraus folgt, dass $C[D^2[V]]$ nicht vollständig ist und damit auch nicht zu L_2 gehören kann.

Anmerkung. Abpumpen, d. h. die Wahl von $i = 0$, funktioniert hier nicht, denn es ist nicht ausgeschlossen, dass C der triviale Kontext ist. In diesem Fall ist $C[D^0[V]] = C[V] = V$ nach wie vor ein vollständiger Binärbaum, gehört also zu L_2.

T2.9 Anwendung Nerode-Rechtskongruenz für Baumsprachen

Wir betrachten $L_1 = \{T \mid T \text{ hat gerade Höhe}\}$ über dem r-Alphabet $\Sigma = \{a/2,\, b/1\, c/0\}$. Um zu zeigen, dass L_1 nicht NEBA-erkennbar ist, müssen wir eine Folge $(T_n)_{n\geq 1}$ von Bäumen über Σ finden, so dass $T_n \not\sim_{L_1} T_k$ für alle $1 \leq n < k$ gilt.

Wir wählen diese Bäume so, dass T_n der Pfad $b(b(b\cdots(b(c))\cdots))$ der Höhe $2n$ ist. Seien nun n, k mit $1 \leq n < k$ beliebig. Um $T_n \not\sim_{L_1} T_k$ zu zeigen, betrachten wir den Kontext $C = a(b(x)T_k)$:



Da $1 \leq n < k$, hat C die Höhe $2k+1$. Nun gilt:

- $C[T_n]$ hat ebenfalls Höhe $2k+1$, da die Höhe von T_n um mindestens 2 kleiner ist als die Höhe von T_k und damit in $C[T_n]$ der linke Pfad nicht länger werden kann als der rechte.
- $C[T_k]$ hat jedoch Höhe $2k+2$, da der linker Pfad um 1 länger ist als der rechte (denn das linke Vorkommen von T_k beginnt eine Ebene tiefer als das rechte).

Damit ist $C[T_n] \notin L_1$ und $C[T_k] \in L_1$, woraus wie gewünscht $T_n \not\sim_{L_1} T_k$ folgt.

T2.10 Mächtigkeit von DETDBAs

Lemma 2.14. Die erkennbare Baumsprache $L = \{a(bc), a(cb)\}$ wird von keinem DETDBA erkannt.

Beweis. Angenommen, es gebe einen DETDBA $\mathcal{A} = (Q, \Sigma, \Delta, I)$ mit $L(\mathcal{A}) = L$. Wegen des Determinismus gibt es nur einen einzigen Anfangszustand q_I und genau einen Übergang $(a, q_I) \to (q_1, q_2)$ mit (a, q_I) auf der linken Seite. Da $a(bc), a(cb) \in L$, muss Δ auch die folgenden Übergänge enthalten:

$$\begin{array}{ll} (b, q_1) \to () & \qquad (c, q_1) \to () \\ (c, q_2) \to () & \qquad (b, q_2) \to () \end{array}$$

Also hat $\mathcal{A}$ auch einen erfolgreichen Run auf $a(bb)$; also $a(bb) \in L$, Widerspruch zur Definition von L. ❏

T2.11 Korrektheit des Algorithmus fürs Leerheitsproblem

Behauptung: $L(\mathcal{A}) = \emptyset$ gdw. $R \cap F = \emptyset$.

Beweis. Wir zeigen beide Richtungen per Kontraposition.

„$\Leftarrow$“ Sei $L(\mathcal{A}) \neq \emptyset$. Dann gibt es einen Baum $T = (P, t) \in L(\mathcal{A})$. Sei r ein erfolgreicher Run von $\mathcal{A}$ auf T. Man zeigt leicht per Induktion über die Höhe, dass für alle Positionen $p \in P$ gilt: $r(p) \in R$. Also gilt insbesondere $r(\varepsilon) \in R$. Da auch $r(\varepsilon) \in F$ gilt (r ist erfolgreich), ist $R \cap F \neq \emptyset$.

„$\Rightarrow$“ Gelte $R \cap F \neq \emptyset$ und sei $q \in R \cap F$. Um zu zeigen, dass $L(\mathcal{A}) \neq \emptyset$ gilt, konstruieren wir einen Baum T und einen Run r von $\mathcal{A}$ auf T mit $r(\varepsilon) = q$ (woraus mit $q \in R \cap F$ folgt, dass r erfolgreich ist). Wir definieren P, t, r iterativ wie folgt.

(1) Sei $a(q_1, \ldots, q_m) \to q$ der eindeutig bestimmte Übergang, der zu $q \in R$ geführt hat. Setze:
- $P = \{\varepsilon, 1, \ldots, m\}$

- $t(\varepsilon) = a$
- $r(\varepsilon) = q$, $r(1) = q_1$, …, $r(m) = q_m$

(2) Solange es ein $p \neq \varepsilon$ gibt mit $r(p) = q'$ definiert, aber $t(p)$ noch nicht definiert, tue folgendes:

Sei $a'(q'_1, \ldots, q'_n) \to q'$ der eindeutig bestimmte Übergang, der zu $q' \in R$ geführt hat. Setze:

- $P = P \cup \{pj \mid j = 1, \ldots, n\}$
- $t(p) = a'$
- $r(p_1) = q'_1$, …, $r(p_n) = q'_n$

Dieser Prozess endet nach endlich vielen Anwendungen von (2), weil auf jedem Pfad von r kein Zustand doppelt auftreten kann, denn nach Konstruktion gilt:

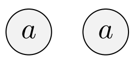
> Wenn $r(p) = \widehat{q}$ und $r(pi) = \widehat{q}'$, dann ist $\widehat{q}$ später als $\widehat{q}'$ zu R hinzugefügt worden.

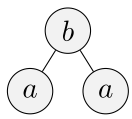
Also ist $T = (P, t)$ endlich. Außerdem ist r ein Run, denn die Übergangsrelation Δ wird nach Konstruktion respektiert. (Erfolgreich ist r, weil $r(\varepsilon) = q \in R \cap F$.) Folglich ist $L(\mathcal{A}) \neq \emptyset$. ❑

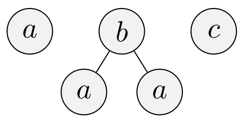
T2.12 Beispiele für Hecken und Bäume

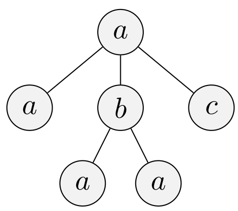
Sei $\Sigma = \{a, b, c\}$.

- ε ist eine Hecke.
- Dann ist auch $a = a()$ ein Baum: 

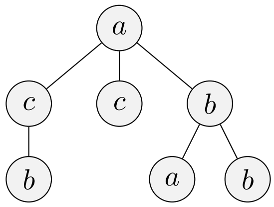
- Dann ist auch aa eine Hecke: 

- Dann ist auch $b(aa)$ ein Baum: 

- Dann ist auch $ab(aa)c$ eine Hecke: 

- Dann ist auch $a(ab(aa)c)$ ein Baum: 

Auch $a(c(b)cb(ab))$ ist ein Baum:



T2.13 Problem mit NEBAs auf Bäumen ohne Stelligkeit

Sei $\Sigma = \{a\}$, wobei a ohne Stelligkeit ist. Wir betrachten die Sprache L aller Bäume mit Höhe 1 über Σ, also $L = \{a(a), a(aa), a(aaa), \dots\}$. Wenn wir nun einen (Bottom-up-)Baumautomaten $\mathcal{A} = (Q, \Sigma, \Delta, F)$ mit $L(\mathcal{A}) = L$ konstruieren wollen, dann sollte dieser zwei Zustände q_0, q_1 haben, um zwischen den zwei Ebenen zu unterscheiden; außerdem müsste er für jedes $k \geq 1$ einen Übergang

$$a(\underbrace{q_0, \dots, q_0}_{k\text{-mal}}) \to q_1$$

haben – damit wäre aber Δ nicht mehr endlich.

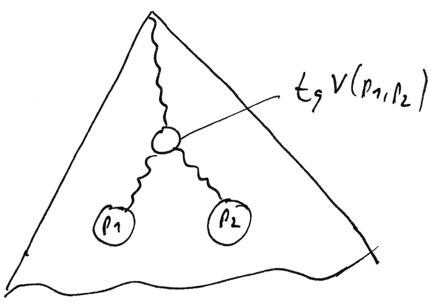
Abhilfe kann man schaffen, indem man eine reguläre Sprache $R \subseteq Q^*$ benutzt, die genau die unendlich vielen Folgen $q_0, \dots, q_0$ beschreibt. Wenn man R mittels eines regulären Ausdrucks beschreibt, erhält man hier einen einzigen Übergang, nämlich

$$a(q_0^+) \to q_1,$$

wobei q_0^+ die übliche Abkürzung für $q_0 q_0^*$ ist (q_0^* genügt nicht, denn oben ist $k = 0$ nicht erlaubt).

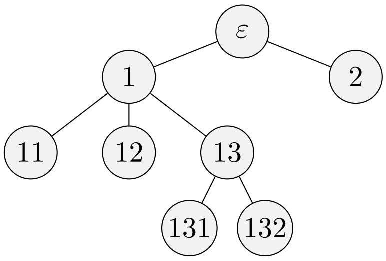
T2.14 Beispiel-NEHA

Wir beginnen mit einem Beispiel für den tiefsten gemeinsamen Vorgänger (tgV) zweier Positionen p_1, p_2. Intuitiv gesprochen ist $\mathsf{tgV}(p_1, p_2)$ die Stelle, an der sich die zwei Pfade zu p_1 bzw. p_2 trennen:



Eine solche Position existiert immer, denn je zwei Pfade haben mindestens die Position ε gemeinsam.

Wenn wir untenstehenden Baum betrachten (nur die Positionen), dann gilt Nebenstehendes.

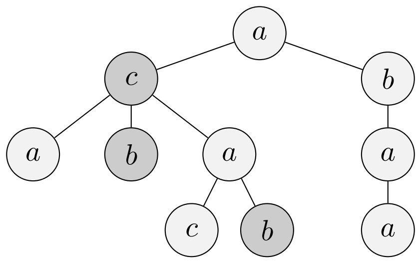


$$\begin{aligned} \mathsf{tgV}(131, 132) &= 13 \\ \mathsf{tgV}(132, 12) &= 1 \\ \mathsf{tgV}(132, 2) &= \varepsilon \end{aligned}$$

Nun betrachten wir die Sprache

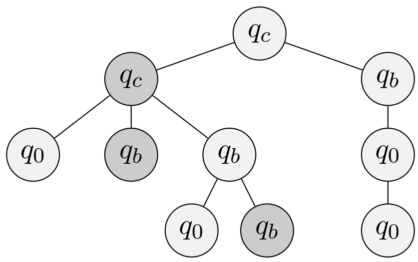
$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit } t(p_1) = t(p_2) = b \text{ und } t(\text{tgV}(p_1, p_2)) = c\}$$

und den folgenden Baum L über dem Alphabet $\Sigma = \{a, b, c\}$ (ohne Stelligkeit).



An den hervorgehobenen Positionen erkennt man, dass $T \in L$.

Der NEHA $\mathcal{A}$ von Folie 2.79 hat folgenden erfolgreichen Run auf T:



T2.15 Beispielableitung in DTD

Wir betrachten die durch folgende erweiterte kontextfreie Grammatik gegebene DTD.

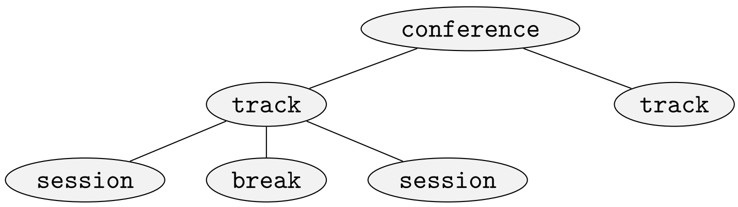
$$\begin{array}{lll}
\texttt{conference} & \to & \texttt{track}^{+} + (\texttt{session}\ (\texttt{break} + \varepsilon))^{+} \\
\texttt{track} & \to & (\texttt{session}\ (\texttt{break} + \varepsilon))^{+} \\
\texttt{session} & \to & \texttt{chair}\ \texttt{talk}^{+} \\
\texttt{talk} & \to & (\texttt{title}\ \texttt{authors}) + (\texttt{title}\ \texttt{speaker}) \\
\texttt{chair} & \to & \texttt{DATA} \\
\ldots & & \\
\texttt{title} & \to & \texttt{DATA}
\end{array}$$

Eine mögliche Ableitung beginnt wie folgt.

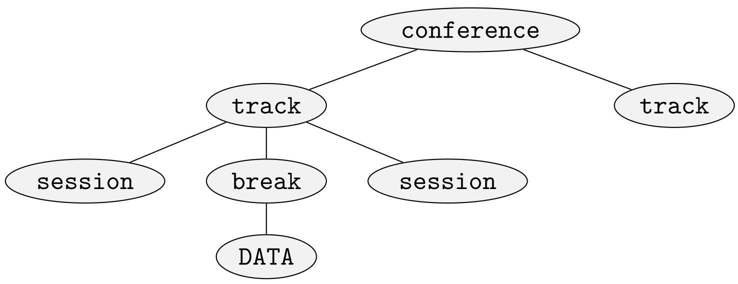
- Beginne mit dem Startsymbol:

- Wende Regel 1 auf diesen Knoten an:

- Wende Regel 2 auf das linke `track`-Blatt an:

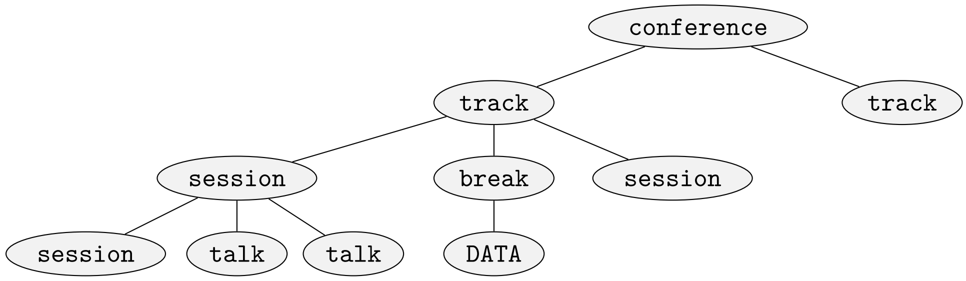


- Wende Regel 6 auf das `break`-Blatt an:



  Auf das neu entstandene `DATA`-Blatt ist nun keine Regel mehr anwendbar.

- Wende Regel 3 auf das linke `session`-Blatt an:



- (usw.)

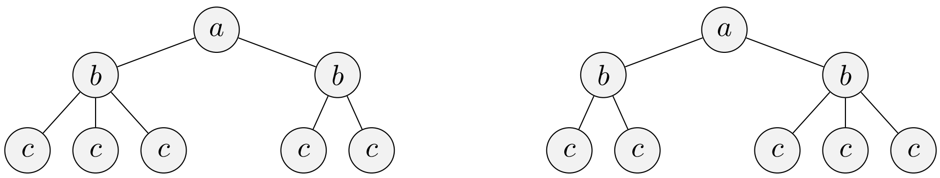
Es wird solange jeweils eine Regel auf ein Blatt angewendet, bis keine Regel mehr anwendbar ist.

T2.16 Beispiele nicht lokaler Sprachen

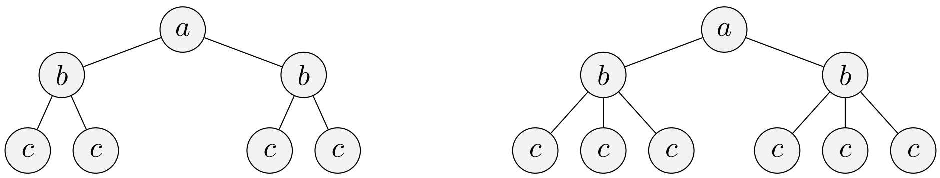
Beispiel 1.

$$L_1 = \{a(b(ccc)b(cc)),\, a(b(cc)b(c))\},$$

d. h. L_1 besteht genau aus den Bäumen



L_1 ist nicht lokal, denn wenn L_1 von einer DTD erzeugt würde, dann müsste diese DTD auch die folgenden Bäume erzeugen:

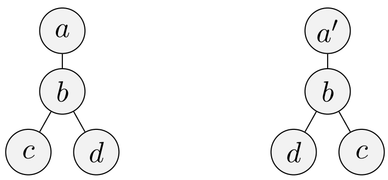


Man kann also mit DTDs nicht in Tiefe ≥ 2 zählen. Für unser Konferenz-Beispiel heißt das, dass man z. B. keine DTD schreiben kann, die erfordert, dass alle Konferenzen genau 5 Vorträge haben.

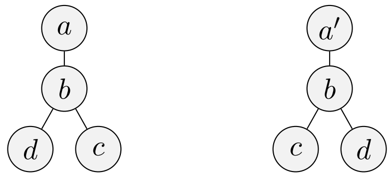
Beispiel 2.

$$L_2 = \{a(b(cd)),\, a'(b(dc))\},$$

d. h. L_2 besteht genau aus den Bäumen



L_2 ist nicht lokal, denn wenn L_2 von einer DTD erzeugt würde, dann müsste diese DTD auch die folgenden Bäume erzeugen:



Man kann also mit DTDs keine Abhängigkeiten modellieren, die sich über mehrere Ebenen erstrecken, wie hier z. B.: „bei den Enkeln von a kommt c vor d, aber bei den Enkeln von a' ist es anders herum.“ Bezogen auf das Konferenzbeispiel kann man z. B. mit einer DTD nicht fordern, dass in Informatik-Konferenzen (a) die Publikation (c) vor dem Vortrag (d) erscheint, während die Reihenfolge bei Mathematik-Konferenzen (a') umgekehrt ist.

T2.17 Beispiel für nicht deterministischen regulären Ausdruck

Der reguläre Ausdruck (RA) $r = ab + ac$ ist nicht deterministisch: Mittels Nummerieren der Buchstaben erhalten wir $r' = a_1b_1 + a_2c_1$. Nun sind aber $a_1b_1, a_2c_1 \in L(r')$ (wenn man $u = \varepsilon$, $v = b_1$ und $w = c_1$ wählt), was Definition 2.28 verletzt.

Es ist aber leicht zu sehen, dass der RA $a(b+c)$ äquivalent zu r ist. Dieser ist trivialerweise deterministisch, denn jedes Zeichen kommt darin nur einmal vor.

Daraus folgt, dass die Zeile

$$\texttt{talk} \;\rightarrow\; (\texttt{title authors}) + (\texttt{title speaker})$$

aus der Beispiel-DTD (Folie 2.86) zwar kein zulässiges Inhaltsmodell enthält, man sie aber äquivalent mit deterministischem RA so umschreiben kann, dass sie zulässig wird:

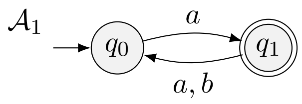
$$\texttt{talk} \;\rightarrow\; \texttt{title}\ (\texttt{authors} + \texttt{speaker})$$

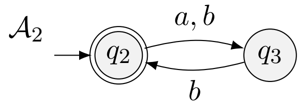
Teil III.

Endliche Automaten auf unendlichen Wörtern

T3.1 Produktkonstruktion für NEAs ist für NBAs nicht korrekt

Sei $\Sigma = \{a, b\}$. Wir betrachten folgende **NEAs** $\mathcal{A}_1, \mathcal{A}_2$.



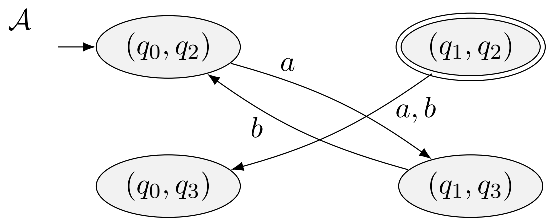


Dann gilt:

$$L(\mathcal{A}_1) = \{a_0 \cdots a_{n-1} \mid n \text{ ist ungerade und } a_0 = a_2 = \cdots = a\}$$
$$L(\mathcal{A}_2) = \{a_0 \cdots a_{n-1} \mid n \text{ ist gerade und } a_1 = a_3 = \cdots = b\}$$

$$L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = \emptyset$$

Der Produktautomat $\mathcal{A}$ ist folgender:



Diese Konstruktion ist korrekt für NEAs; in diesem Beispiel ist $L(\mathcal{A}) = \emptyset$, da der einzige akzeptierende Zustand (q_1, q_2) unerreichbar ist.

Betrachten wir nun dieselben Automaten $\mathcal{A}_1, \mathcal{A}_2$ als **NBAs.** Dann gilt:

$$L_\omega(\mathcal{A}_1) = \{\alpha \mid n \text{ ist ungerade und } \alpha_0 = \alpha_2 = \cdots = a\}$$
$$L_\omega(\mathcal{A}_2) = \{\alpha \mid n \text{ ist gerade und } \alpha_1 = \alpha_3 = \cdots = b\},$$

also ist jetzt

$$L_\omega(\mathcal{A}_1) \cap L_\omega(\mathcal{A}_2) = \{(ab)^\omega\},$$

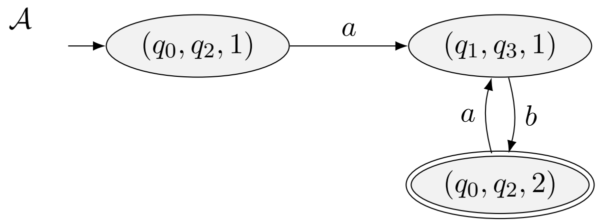
aber nach wie vor ist

$$L_\omega(\mathcal{A}) = \emptyset,$$

also ist die Konstruktion für NBAs nicht korrekt! Der Grund dafür ist, dass die erfolgreichen Runs von $\mathcal{A}_1$ bzw. $\mathcal{A}_2$ die akzeptierenden Zustände asynchron erreichen, nämlich nach $1, 3, 5, \ldots$ Schritten ($\mathcal{A}_1$) bzw. $0, 2, 4, \ldots$ Schritten ($\mathcal{A}_2$). Dadurch erreicht der entsprechende Run von $\mathcal{A}$ niemals einen (kombinierten) akzeptierenden Zustand.

T3.2 Produktkonstruktion für NBAs: Beispiel und Korrektheit

Beispiel. Wendet man die Produktkonstruktion (Folie 30) auf die obigen NBAs $\mathcal{A}_1, \mathcal{A}_2$ an, so erhält man folgenden NBA $\mathcal{A}$ (im Bild sind nur die erreichbaren Zustände wiedergegeben):



Tatsächlich ist nun $L_\omega(\mathcal{A}) = \{(ab)^\omega\}$.

Korrektheitsbeweis. Zu zeigen ist:

$$L_\omega(\mathcal{A}) = L_\omega(\mathcal{A}_1) \cap L_\omega(\mathcal{A}_2)$$

„⊆“ Sei $\alpha \in L_\omega(\mathcal{A})$. Dann gibt es einen erfolgreichen Run $r = q_0 q_1 q_2 \cdots$ von $\mathcal{A}$ auf α mit $q_0 \in I$ und $\mathsf{Inf}(r) \cap F \neq \emptyset$. Nach Konstruktion von $\mathcal{A}$ muss jedes q_i die Form

$$q_i = (s_i, t_0, n_i)$$

haben mit $s_i \in Q_1$, $t_i \in Q_2$ und $n_i \in \{1, 2\}$ für alle $i \geq 0$.

Wir betrachten die Folge $s = s_0 s_1 s_2 \cdots$. Diese ist ein Run von $\mathcal{A}_1$ auf α: da r ein Run ist, folgt mit der Definition von I bzw. Δ, dass $s_0 \in I_1$ und $(s_i, \alpha_i, s_{i+1}) \in \Delta_1$ für alle $i \geq 0$. Außerdem ist s erfolgreich, denn wegen $\mathsf{Inf}(r) \cap F \neq \emptyset$ (und der Definition von F) enthält r unendlich viele Zustände der Form

$$q_i = (s_i, t_i, 2) \qquad \text{mit } t_i \in F_2; \tag{$*$}$$

also enthält r auch unendlich viele Zustände der Form

$$q_j = (s_j, t_j, 1) \qquad \text{mit } s_j \in F_1, \tag{$**$}$$

weil nach jedem q_i der Form ($*$) in $(s_{i+1}, t_{i+1}, 1)$ gewechselt wird und erst dann wieder ins nächste q_i der Form ($*$) gegangen werden kann, wenn ein q_j der Form ($**$) gefunden wurde. Also ist $\mathsf{Inf}(s) \cap F_1 \neq \emptyset$ und damit s erfolgreich.

Analog argumentiert man, dass $t = t_0 t_1 t_2 \cdots$ ein erfolgreicher Run von $\mathcal{A}_2$ auf α ist. Folglich ist $\alpha \in L_\omega(\mathcal{A}_1) \cap L_\omega(\mathcal{A}_2)$.

„⊇“ Sei $\alpha \in L_\omega(\mathcal{A}_1) \cap L_\omega(\mathcal{A}_2)$. Dann gibt es erfolgreiche Runs

$$\begin{aligned} s &= s_0 s_1 s_2 \ldots \quad \text{von } \mathcal{A}_1 \text{ auf } \alpha \quad \text{und} \\ t &= t_0 t_1 t_2 \ldots \quad \text{von } \mathcal{A}_2 \text{ auf } \alpha. \end{aligned}$$

Wir betrachten die Folge

$$r = (s_0, t_0, n_0)\ (s_1, t_1, n_1)\ (s_2, t_2, n_2)\ \cdots ,$$

wobei die n_i induktiv wie folgt definiert sind:

$$n_0 = 1$$
$$n_i = \begin{cases} 1 & \text{falls } n_{i-1} = 1 \text{ und } s_{i-1} \notin F_1 \\ & \text{oder } n_{i-1} = 2 \text{ und } t_{i-1} \in F_2 \\ 2 & \text{sonst} \end{cases}$$

Man zeigt leicht unter Zuhilfenahme der Konstruktion von I, F, Δ, dass r ein erfolgreicher Run von $\mathcal{A}$ auf α ist. Folglich ist $\alpha \in L_\omega(\mathcal{A})$. ❑

T3.3 Büchi-Erkennbarkeit von W^ω für reguläre Sprachen W

Noch zu zeigen: $L_\omega(\mathcal{A}_2) = L(\mathcal{A}_1)^\omega$

„$\subseteq$“ Sei $\alpha \in L_\omega(\mathcal{A}_2)$. Dann gibt es einen erfolgreichen Run $r = q_0q_1q_2\cdots$ von $\mathcal{A}_2$ auf α, d. h. $q_0 = q_I$ (der einzige Anfangszustand von $\mathcal{A}_2$), und q_I kommt unendlich oft in r vor (weil es auch der einzige akzeptierende Zustand von $\mathcal{A}_2$ ist).

Seien $q_{i_0}, q_{i_1}, q_{i_2}, \ldots$ alle Vorkommen von q_I in r. Für jedes $j \geq 0$ betrachten wir die Folge

$$r_j := \underset{\substack{\| \\ q_I}}{q_{i_j}} q_{i_j+1} \cdots q_{i_{j+1}-1} \underset{\substack{\| \\ q_I}}{q_{i_{j+1}}} .$$

Nach Konstruktion von Δ_2 und wegen der Annahmen über $\mathcal{A}_1$ gibt es ein $q_f \in F$, so dass

$$\underset{\substack{\| \\ q_I}}{q_{i_j}} q_{i_j+1} \cdots q_{i_{j+1}-1} \underset{\substack{\cap \\ F}}{q_f}$$

ein erfolgreicher Run von $\mathcal{A}_1$ auf $w_j := \alpha[i_j, i_{j+1} - 1]$ ist. Folglich gehört für alle j das Wort w_j zur Sprache $L(\mathcal{A}_1)$, und damit gilt $\alpha \in L(\mathcal{A}_1)^\omega$.

„$\supseteq$“ Sei $\alpha \in L(\mathcal{A}_1)^\omega$. Dann ist $\alpha = w_0w_1w_2\cdots$ mit $w_i \in L(\mathcal{A}_1)$ für alle $i \geq 0$. Wir nehmen o. B. d. A. an, dass $\varepsilon \notin L(\mathcal{A}_1)$ ist, also $|w_j| > 0$ für alle i. Also gibt es für jedes $j \geq 0$ einen erfolgreichen Run

$$r_j := q_{j,0}q_{j,1}\cdots q_{j,|w_j|}$$

von $\mathcal{A}_1$ auf w_j. Nach Konstruktion von Δ_2 ist dann

$$r \ := \ q_{0,0}q_{0,1}\cdots q_{0,|w_0|-1} \ \ q_{1,0}q_{1,1}\cdots q_{1,|w_1|-1} \ \ \cdots$$

ein erfolgreicher Run von $\mathcal{A}_2$ auf α. Folglich ist $\alpha \in L_\omega(\mathcal{A}_2)$. ❑

T3.4 Beweis Charakterisierung NBA-erkennbarer Sprachen

Satz 3.10. Eine Sprache $L \subseteq \Sigma^\omega$ ist Büchi-erkennbar genau dann, wenn es reguläre Sprachen $V_1, W_1, \dots, V_n, W_n$ gibt mit $n \geqslant 1$ und

$$L = V_1 W_1^\omega \cup \cdots \cup V_n W_n^\omega.$$

Beweis.

„⇐“ Folgt aus Lemmas 3.6, 3.8 und 3.9.

„⇒“ Sei $L \subseteq \Sigma^\omega$ Büchi-erkennbar, also $L = L_\omega(\mathcal{A})$ für einen NBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$. Wir nutzen folgende Beobachtung: für jedes ω-Wort $\alpha \in L_\omega(\mathcal{A})$, auf dem $\mathcal{A}$ einen erfolgreichen Run $r = q_0 q_1 q_2 \cdots$ mit $q_f \in \mathsf{Inf}(r) \cap F$ hat, gibt es

- ein endliches Präfix von α, das q_I nach q_f überführt und
- unendlich viele nicht-leere Infixe, die q_f nach q_f überführen.

Diese beiden Sorten von Infixen können wir durch reguläre Sprachen beschreiben. Dazu verwenden wir folgende Notation: Für zwei beliebige Zustände $q_1, q_2 \in Q$ sei $\mathcal{A}_{q_1,q_2} = (Q, \Sigma, \Delta, \{q_1\}, \{q_2\})$ und $W_{q_1,q_2} = L(\mathcal{A}_{q_1,q_2})$. Nach Definition sind die W_{q_1,q_2} regulär. Wegen der Akzeptanzbedingung von Büchi-Automaten und unserer Beobachtung gilt nun:

$$L_\omega = \bigcup_{\substack{q_i \in I \\ q_f \in F}} W_{q_i,q_f} W_{q_f,q_f}^\omega$$

❑

T3.5 Beispiele für $\overrightarrow{W}$

$$W_1 = \{a^n b^m \mid n, m \geq 0\}$$
$$\overrightarrow{W_1} = \{a^n b^\omega \mid n \geq 0\} \cup \{a^\omega\}$$

$$W_2 = \{a^n b^n \mid n \geq 0\}$$
$$\overrightarrow{W_2} = \emptyset$$

$$W_3 = \{a, b\}^*$$
$$\overrightarrow{W_3} = \{a, b\}^\omega$$

$$W_4 = \{w \in \{a, b\}^* \mid \#_a(w) \text{ ist gerade}\}$$
$$\overrightarrow{W_4} = \{\alpha \in \{a, b\}^\omega \mid \#_a(\alpha) = \infty \text{ oder } \alpha = wb^\omega \text{ mit } \#_a(w) \text{ gerade}\}$$

$$W_5 = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$
$$\overrightarrow{W_5} = \Big\{\alpha \in \{a, b\}^\omega \;\Big|\; \#\{i \mid \#_a(\alpha[0, i]) = \#_b(\alpha[0, i])\} = \infty\Big\}$$

T3.6 Beweis Charakterisierung DBA-erkennbarer Sprachen

Satz 3.11. Eine ω-Sprache $L \subseteq \Sigma^\omega$ ist DBA-erkennbar genau dann, wenn es eine reguläre Sprache $W \subseteq \Sigma^*$ gibt mit $L = \overrightarrow{W}$.

Beweis. Es genügt zu zeigen, dass für jeden **D**EA/**D**BA $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, F)$ gilt:

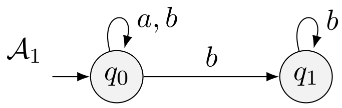
$$L_\omega(\mathcal{A}) = \overrightarrow{L(\mathcal{A})}$$

„$\subseteq$“ (Diese Richtung funktioniert sogar, wenn $\mathcal{A}$ ein **N**EA/**N**BA ist.)
Sei $\alpha \in L_\omega(\mathcal{A})$. Dann gibt es einen erfolgreichen Run $r = q_0 q_1 q_2 \cdots$ von $\mathcal{A}$ auf α. Seien $i_0, i_1, i_2, \ldots \in \mathbb{N}$ die Positionen mit $q_{i_j} \in F$. Dann ist für jedes $j \geq 0$ das Präfix $q_0 \cdots q_{i_j}$ von r ein erfolgreicher Run des **NEAs** $\mathcal{A}$ auf $\alpha_0 \cdots \alpha_{i_j - 1}$. Damit gibt es unendlich viele Präfixe von α, die in $L(\mathcal{A})$ sind, und damit ist $\alpha \in \overrightarrow{L(\mathcal{A})}$.

„$\supseteq$“ Sei $\alpha \in \overrightarrow{L(\mathcal{A})}$. Dann hat α unendlich viele Präfixe in $L(\mathcal{A})$. Also muss der **eindeutig bestimmte** Run r von $\mathcal{A}$ (der Automat ist deterministisch!) einen akzeptierenden Zustand unendlich oft erreichen. Damit ist $\alpha \in L_\omega(\mathcal{A})$. ❑

T3.7 Beispiele für Muller-Automaten

- Betrachte den folgenden Muller-Automaten $\mathcal{A}_1 = (Q_1, \Sigma, \Delta_1, I_1, \mathcal{F}_1)$ über dem Alphabet $\Sigma = \{a, b\}$.



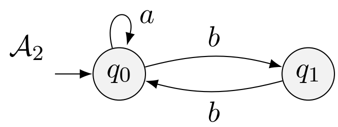
(M1) Wenn $\mathcal{F}_1 = \{\{q_0\}\}$, dann $L_\omega(\mathcal{A}_1) = \Sigma^\omega$.

(M2) Wenn $\mathcal{F}_1 = \{\{q_1\}\}$, dann $L_\omega(\mathcal{A}_1) = \{\alpha \in \Sigma^\omega \mid \#_a(\alpha) < \infty\}$.

(M3) Wenn $\mathcal{F}_1 = \{\{q_0, q_1\}\}$, dann $L_\omega(\mathcal{A}_1) = \emptyset$ (weil Wechsel von q_1 zu q_0 nicht möglich).

(M4) Wenn $\mathcal{F}_1 = \{\{q_0\}, \{q_1\}\}$, dann $L_\omega(\mathcal{A}_1) = \Sigma^\omega$ (Vereinigung der Fälle M1, M2).

- Betrachte nun den folgenden Muller-Automaten $\mathcal{A}_2 = (Q_2, \Sigma, \Delta_2, I_2, \mathcal{F}_2)$ über demselben Alphabet.



(M5) Wenn $\mathcal{F}_2 = \{\{q_0\}\}$, dann $L_\omega(\mathcal{A}_2) = L((a + bb)^* a^\omega)$ (Menge aller Wörter mit endlich vielen b’s, in denen zwischen je zwei a’s und vor dem ersten a eine gerade Anzahl von b’s steht).

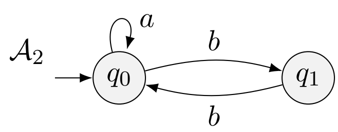
(M6) Wenn $\mathcal{F}_2 = \{\{q_1\}\}$, dann $L_\omega(\mathcal{A}_2) = \emptyset$ (denn wenn q_1 unendlich oft besucht wird, dann auch q_0).

(M7) Wenn $\mathcal{F}_2 = \{\{q_0, q_1\}\}$, dann $L_\omega(\mathcal{A}_2) = L((a^*bb)^\omega)$ (Menge aller Wörter wie in M5, aber mit *un*endlich vielen b's).

- Für *alle* Muller-Automaten $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ gilt: wenn $\mathcal{F} = \emptyset$ oder $\mathcal{F} = \{\emptyset\}$, dann $L_\omega(\mathcal{A}) = \emptyset$.

T3.8 Beispiele für Rabin-Automaten

- Betrachte den folgenden Rabin-Automaten $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$ über dem Alphabet $\Sigma = \{a, b\}$.



(R1) Wenn $\mathcal{P} = \{(\{q_0\}, \{q_1\})\}$, dann $L_\omega(\mathcal{A}) = \emptyset$ (Begründung wie bei M6).

(R2) Wenn $\mathcal{P} = \{(\{q_1\}, \{q_0\})\}$, dann $L_\omega(\mathcal{A}) = L((a + bb)^*a^\omega)$ (dasselbe Akzeptanzverhalten wie in M5).

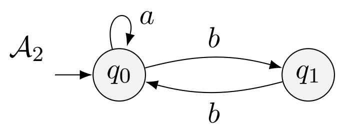
(R3) Wenn $\mathcal{P} = \{(\emptyset, \{q_1\})\}$, dann $L_\omega(\mathcal{A}) = L((a^*bb)^\omega)$ (dasselbe Akzeptanzverhalten wie NBA mit $\mathcal{F} = \{q_1\}$).

(R4) Wenn $\mathcal{P} = \{\{(S, \emptyset)\}\}$ für beliebiges $S \subseteq Q$, dann $L_\omega(\mathcal{A}) = \emptyset$ (folgt direkt aus Definition „erfolgreich“ für NRAs).

- Der Fall mehrerer Paare in der Akzeptanzkomponente $\mathcal{P}$ braucht nicht gesondert illustriert zu werden, denn für *alle* NRAs $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$ mit $\mathcal{P} = \bigcup_{i \le n} \mathcal{P}_i$ gilt: $L_\omega(\mathcal{A}) = \bigcup_{i \le n} L_\omega(Q, \Sigma, \Delta, I, \mathcal{P}_i)$.

T3.9 Beispiele für Streett-Automaten

- Betrachte den folgenden Streett-Automaten $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$ über dem Alphabet $\Sigma = \{a, b\}$.



(S1) Wenn $\mathcal{P} = \{(\{q_0\}, \{q_1\})\}$, dann $L_\omega(\mathcal{A}) = (a + bb)^\omega$ (denn jeder Run ist erfolgreich).

(S2) Wenn $\mathcal{P} = \{(\{q_0\}, \emptyset)\}$, dann $L_\omega(\mathcal{A}) = (a+bb)^\omega$ (denn jeder Run ist erfolgreich).

(S3) Wenn $\mathcal{P} = \{(\{q_1\}, \{q_0\})\}$, dann $L_\omega(\mathcal{A}) = (a * bb)^\omega$ (dasselbe Akzeptanzverhalten wie M7).

(S4) Wenn $\mathcal{P} = \{(\emptyset, \{q_1\})\}$, dann $L_\omega(\mathcal{A}) = (a + bb) * a^\omega$ (denn die Akzeptanzbedingung besagt: q_1 darf *nicht* ∞ oft vorkommen, also muss q_0 ∞ oft vorkommen $\rightsquigarrow$ wie M5).

(S5) Wenn $\mathcal{P} = \{(\emptyset, \{q_0\})\}$, dann $L_\omega(\mathcal{A}) = \emptyset$ (denn q_0 kommt in jedem Run ∞ oft vor).

(S6) Wenn $\mathcal{P} = \{(\emptyset, \{q_0, q_1\})\}$, dann $L_\omega(\mathcal{A}) = \emptyset$ (wie S5).

- Der Fall mehrerer Paare in der Akzeptanzkomponente $\mathcal{P}$ ist analog zu NRAs, aber mit einem entscheidenden Unterschied: für alle NSAs $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$ mit $\mathcal{P} = \bigcup_{i \le n} \mathcal{P}_i$ gilt: $L_\omega(\mathcal{A}) = \bigcap_{i \le n} L_\omega(Q, \Sigma, \Delta, I, \mathcal{P}_i)$.

T3.10 Beweis Korrektheit „von Muller- zu Büchi-Automaten“

Am Ende des Beweises von Lemma 3.19 ist zu zeigen: $L_\omega(\mathcal{A}') = L_\omega(\mathcal{A})$.

„$\supseteq$“ Sei $\alpha \in L_\omega(\mathcal{A})$ und $r = q_0 q_1 q_2 \cdots$ ein erfolgreicher Run von $\mathcal{A}$ (NMA!) auf α, also $q_0 \in I$ und $\mathsf{Inf}(r) = F = \{f_0, \dots, f_{n-1}\}$. Dann gibt es eine Position $i_0 \ge 1$, ab der nur noch akzeptierende Zustände auftreten, d. h. für alle $i \ge i_0$ ist $q_i = f_{j_i}$ für gewisse $j_i < n$. Aus r konstruieren wir wie folgt induktiv eine Folge $s = s_0 s_1 s_2 \cdots$ von Zuständen von $\mathcal{A}'$:

- Für alle $i < i_0$ setze $s_i = q_i$.
- $s_{i_0} = \langle f_j, j \rangle$ für $j = j_{i_0}$
- Für alle $i \ge i_0$ setze

$$s_{i+1} = \begin{cases} \langle f_{j_{i+1}}, k \rangle & \text{falls } s_i = \langle f_{j_i}, k \rangle \text{ und } j_i \ne k \\ \langle f_{j_{i+1}}, k \oplus_n 1 \rangle & \text{falls } s_i = \langle f_{j_i}, j_i \rangle \end{cases}$$

Diese Konstruktion von s stellt sicher:

- s ist Run von $\mathcal{A}'$ auf α (das ist leicht schrittweise anhand der Konstruktion von Δ' und von s nachvollziehbar).
- $s_0 \in I$ (nach Definition I').
- s ist erfolgreich, denn wegen $\mathsf{Inf}(r) = F$ gibt es ∞ viele Vorkommen von Zuständen der Form $\langle f_0, 0 \rangle$ in S.

Also ist $\alpha \in L_\omega(\mathcal{A}')$.

„$\subseteq$“ Sei $\alpha \in L_\omega(\mathcal{A}')$ und $s = s_0 s_1 s_2 \cdots$ ein erfolgreicher Run von $\mathcal{A}'$ (NBA!) auf α, also $s_0 \in I' = I$ und $\langle f_0, 0 \rangle$ kommt ∞ oft in s vor. Konstruiere daraus eine Folge $r = q_0 q_1 q_2 \cdots$ von Zuständen aus Q wie folgt:

- Wenn $s_i \in Q$, dann $q_i = s_i$.
- Wenn $s_i = \langle f_{j_i}, k \rangle$, dann $q_i = f_{j_i}$.

Diese Konstruktion stellt sicher:

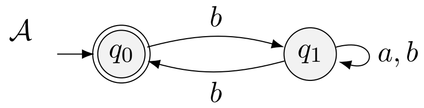
- r ist Run von $\mathcal{A}$ auf α (vgl. Konstruktion von Δ').

- $r_0 \in I$.
- r enthält nur endlich viele Zustände außerhalb F (weil in „Phase 2“ nur noch Zustände aus F vorkommen).
- Jeder Zustand aus F kommt in r unendlich oft vor (weil $\langle f_0, 0\rangle$ unendlich oft in s vorkommt und zwischen zwei solchen Vorkommen wegen des „Moduswechsels“ in Δ' jedes f_i, $0 < i < n$, mindestens einmal durchlaufen werden muss).

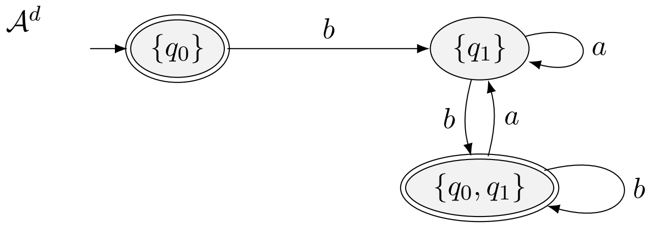
Damit ist r ein erfolgreicher Run des NMA $\mathcal{A}$ auf α, also $\alpha \in L_\omega(\mathcal{A})$. ❏

T3.11 Determinisierungsversuch mittels Potenzmengenkonstruktion

Wir betrachten folgenden NBA $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b\}$.



Die erkannte Sprache ist $L_\omega(\mathcal{A}) = L((b\Sigma^*b)^\omega) = \{\alpha \in \Sigma^\omega \mid a_0 = b \text{ und } \#_{bb}(\alpha) = \infty\}$. Mittels Potenzmengenkonstruktion erhalten wir folgenden DBA $\mathcal{A}^d$ (der Papierkorbzustand ist weggelassen).



Nun ist aber $(ba)^\omega \in L_\omega(\mathcal{A}^d) \setminus L_\omega(\mathcal{A})$. Der DBA $\mathcal{A}^d$ hat also auf dem Wort $(ba)^\omega$ einen *Bad Run* r, der keinem erfolgreichen Run von $\mathcal{A}$ auf $(ba)^\omega$ entspricht. Der Grund dafür ist, dass für jedes der unendlich vielen Präfixe $bab, babab, bababab, \dots$ von $(ba)^\omega$ das entsprechende Präfix von r zwar den akzeptierenden Zustand $\{q_0, q_1\}$ erreicht, aber der zugehörige in q_0 endende Teilrun von $\mathcal{A}$ nicht mehr zu einem erfolgreichen Run auf α fortgesetzt werden kann.

T3.12 Variation der Akzeptanzbedingung im vorigen Beispiel

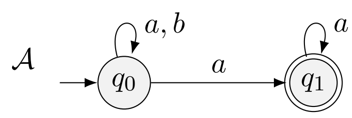
Wir betrachten dieselben Automaten $\mathcal{A}, \mathcal{A}^d$ wie im vorigen Beispiel. Man könnte versuchen, durch Variation der Akzeptanzbedingung von $\mathcal{A}^d$ einen zu $\mathcal{A}$ äquivalenten deterministischen Muller-, Rabin- oder Streett-Automaten zu erhalten, ohne die eigentliche Potenzmengenkonstruktion aufzugeben. Dieser Versuch muss aber scheitern, wovon man sich leicht überzeugt, wenn man alle möglichen Akzeptanzbedingungen systematisch durchgeht. Diese sind entweder trivial (d. h. führen offensichtlich zu $L_\omega(\mathcal{A}^d) = \emptyset$ oder $L_\omega(\mathcal{A}^d) = \Sigma^\omega$) oder laufen auf einen der folgenden Fälle hinaus: Erfolgreiche Runs …

1. … müssen $\{q_1\}$ und $\{q_0, q_1\}$ unendlich oft besuchen;
2. … dürfen nur $\{q_1\}$ unendlich oft besuchen;
3. … dürfen nur $\{q_0, q_1\}$ unendlich oft besuchen.

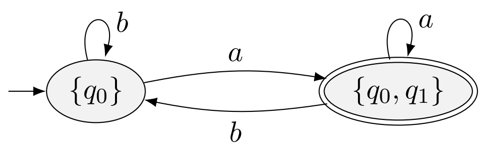
Im 1. und 2. Fall gibt es jedoch Bad Runs auf $(ba)^\omega$ bzw. ba^ω; im 3. Fall gibt es Wörter, die von $\mathcal{A}$ akzeptiert werden, aber nicht von $\mathcal{A}^d$, z. B. $(bba)^\omega$.

T3.13 Beispiel für Safras Trick 1

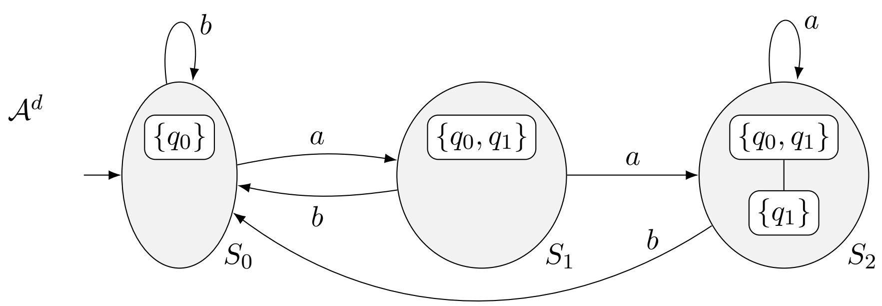
Betrachte folgenden NBA $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b\}$.



Dieser NBA akzeptiert genau die ω-Wörter mit endlich vielen b's. Die Potenzmengenkonstruktion liefert den DBA



mit einem Bad Run auf $(ab)^\omega$. Mittels Safras Trick 1 erhält man hingegen folgenden deterministischen Automaten $\mathcal{A}^d$.



Wenn man nun den Bad Run auf dem Wort $(ab)^\omega$ verhindern möchte, dann muss man die Akzeptanzbedingung so wählen, dass S_2 unendlich oft besucht werden muss, aber S_0 und S_1 nur endlich oft. Dies erreicht man z. B. durch die Rabin-Akzeptanzkomponente $\mathcal{P} = \{(\{S_0\}, \{S_2\})\}$.

T3.14 Anzahl der Knoten eines Safra-Baums

Wie auf Folie 3.66 halten wir die Zustandsmenge Q des ursprünglichen NBA und eine nichtleere, genügend große Menge von Knotennamen V fest. Ein Safra-Baum über Q, V ist ein geordneter Baum mit Knoten aus V, in dem jeder Knoten mit einem nichtleeren Makrozustand (MZ, Teilmenge $M \subseteq Q$) markiert ist, so dass gilt:

Wenn Knoten v mit M und v's Kinder mit $M_1, \ldots, M_n$ markiert sind, dann:

(1) $M_1 \cup \cdots \cup M_n \subsetneq M$

(2) M_i sind paarweise disjunkt

Behauptung. Jeder Safra-Baum über der Zustandsmenge Q hat höchstens $|Q|$ Knoten.

Beweis. Wir verwenden Induktion über die Höhe des Baums.

Induktionsanfang. Safra-Bäume der Höhe 0 (nur ein Wurzel-Knoten) erfüllen die Bedingung trivialerweise.

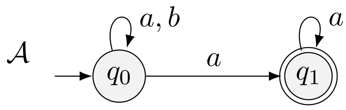
Induktionsschritt. Sei S ein beliebiger Safra-Baum der Höhe $n \geq 2$. Wir betrachten die Teilbäume $S_1, \ldots, S_k$ der Kinder der Wurzel. Wegen Eigenschaften (1) und (2) ist jeder S_i ein Safra-Baum über einer Teilmenge $Q_i \subseteq Q$, wobei die Q_i paarweise disjunkt sind mit

$$\biguplus_{i \leq k} Q_i \subsetneq Q. \tag{*}$$

Nach Induktionsvoraussetzung hat jeder S_i nur höchstens $|Q_i|$ Knoten. Folglich hat S maximal $\sum_{i \leq k} |Q_i| + 1$ Knoten, und wegen $(*)$ ist dieser Wert durch $|Q|$ beschränkt. ❏

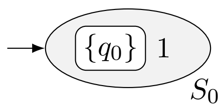
T3.15 Beispiel für die gesamte Safra-Konstruktion

Wir betrachten den NBA $\mathcal{A}$ aus T3.13:



Im Folgenden wird die Konstruktion des DRA $\mathcal{A}^d$ gemäß der Safra-Konstruktion schrittweise beschrieben. Dabei benennen wir die konstruierten Zustände (Safrabäume) der Reihe nach mit $S_0, S_1, S_2, \ldots$ und schreiben diese Namen jeweils rechts neben den entsprechenden Zustand. Außerdem verwenden wir innerhalb von Safrabäumen als Knotennamen die Zahlen $1, 2, 3, \ldots$ und schreiben sie rechts neben den jeweiligen Knoten. Markierte Knoten (Schritt 6) werden wie gehabt mit ! gekennzeichnet.

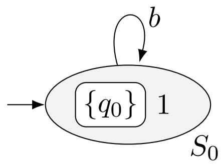
Startzustand ist der folgende Safrabaum:



Folgezustand von S_0 mit b

- Schritt 1 der Safra-Konstruktion ist nicht anwendbar, da Knoten 1 nicht markiert ist.
- Schritt 2 ist nicht anwendbar, da Knoten 1 keine akzeptierenden Zustände enthält ($F = \{q_1\}$, siehe Bild).
- In Schritt 3 ändert sich der Makrozustand von Knoten 1 nicht, da der einzige b-Nachfolgezustand von q_0 wieder q_0 ist.
- Schritte 4–6 sind nicht anwendbar, da Knoten 1 noch keine Kinder hat.

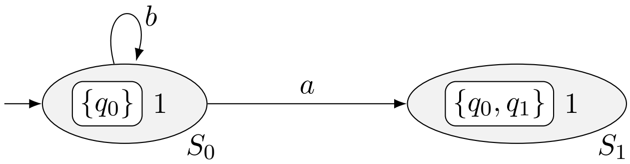
Also ist der b-Folgezustand von S_0 wieder S_0:



Folgezustand von S_0 mit a

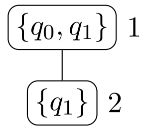
- Schritte 1–2 sind nicht anwendbar, siehe oben.
- In Schritt 3 ändert sich der Makrozustand von Knoten 1 zu $\{q_0, q_1\}$, da sowohl q_0 als auch q_1 von q_0 aus mit a erreicht werden können.
- Schritte 4–6 sind nicht anwendbar, da Knoten 1 noch keine Kinder hat.

Also ist der a-Folgezustand von S_0 ein neuer Safrabaum S_1, in dem Knoten 1 den Makrozustand $\{q_0, q_1\}$ hat:



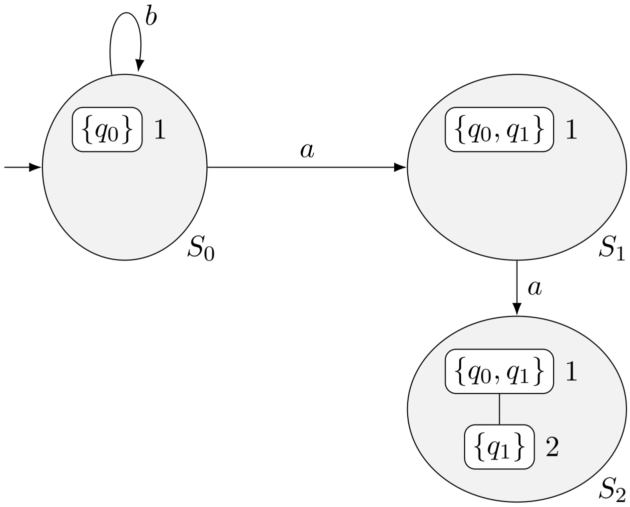
Folgezustand von S_1 mit a

- Schritt 1 ist nach wie vor nicht anwendbar (keine Markierung).
- In Schritt 2 wird ein neues Kind von Knoten 1 erzeugt, dessen Makrozustand aus dem akzeptierenden Zustand q_1 aus Knoten 1 besteht und das den Namen 2 bekommt:



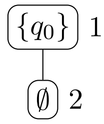
- In Schritt 3 wird auf beide Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit a ändert sich der Inhalt beider Makrozustände nicht.
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- Schritt 5 ist nicht anwendbar, da der Makrozustand von Knoten 2 nicht leer ist.
- Schritt 6 ist nicht anwendbar, da q_0 im Makrozustand von Knoten 1, aber nicht von Knoten 2 vorkommt.

Also ist der a-Folgezustand von S_1 ein neuer Safrabaum S_2 mit zwei Knoten wie folgt:



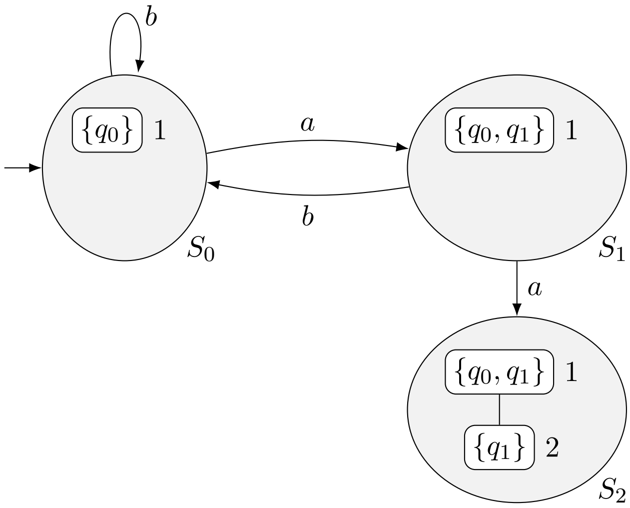
Folgezustand von S_1 mit b

- Schritte 1–2 wie oben.
- In Schritt 3 wird wieder auf beide Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit b ändern sich die Makrozustände wie folgt:



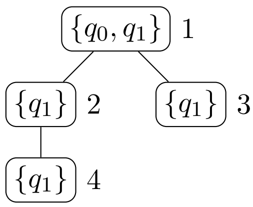
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- In Schritt 5 wird Knoten 2 gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

Damit ist der b-Folgezustand von S_1 der Safrabaum S_0:

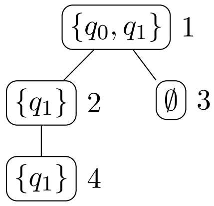


Folgezustand von S_2 mit a

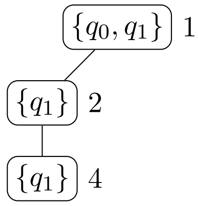
- Schritt 1 ist nach wie vor nicht anwendbar.
- In Schritt 2 wird je ein neues Kind von Knoten 1 und 2 erzeugt, da die Makrozustände beider Knoten akzeptierende Zustände enthalten:



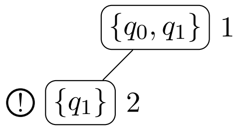
- In Schritt 3 wird auf alle Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit a ändert sich der Inhalt der Makrozustände nicht.
- In Schritt 4 wird q_1 aus Knoten 3 entfernt, da dieser Zustand im älteren Geschwister 2 enthalten ist:



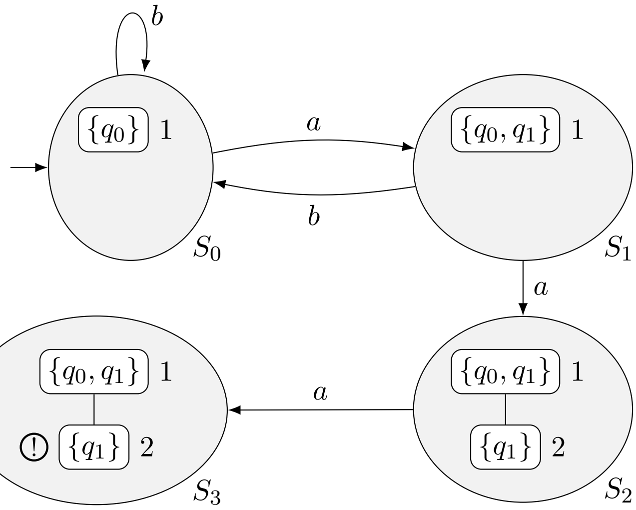
- In Schritt 5 wird Knoten 3 gelöscht:



- In Schritt 6 wird nun Knoten 4 gelöscht und Knoten 2 markiert:

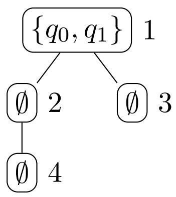


Der zuletzt abgebildete Safrabaum S_3 ist der a-Folgezustand von S_2:



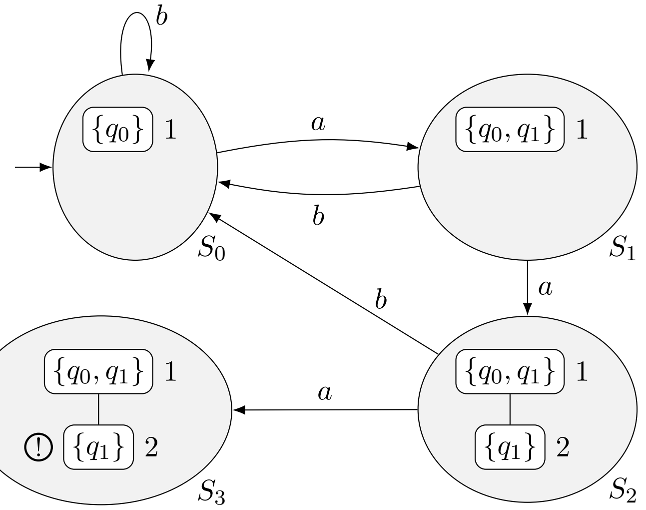
Folgezustand von S_2 mit b

- Schritte 1–2 wie oben.
- In Schritt 3 wird wieder auf alle Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit b ändern sich die Makrozustände wie folgt:



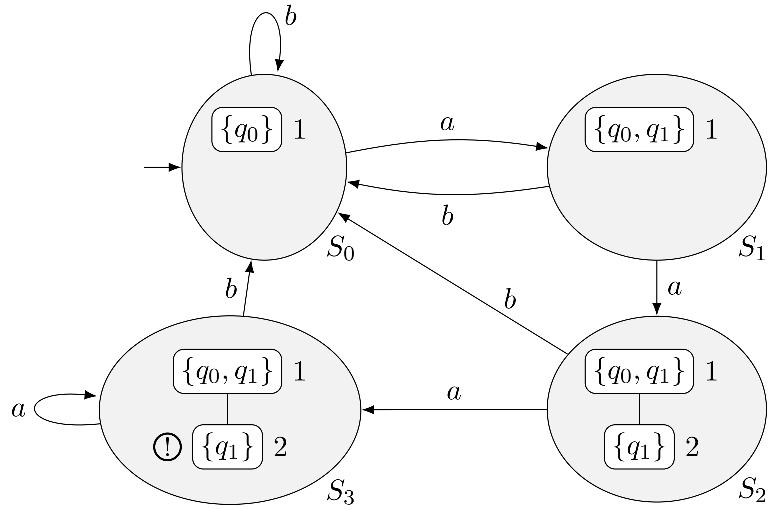
- Schritt 4 ist nicht anwendbar: alle Makrozsutände außer Knoten 1 sind leer.
- In Schritt 5 werden Knoten $2, 3, 4$ gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

Damit ist der b-Folgezustand von S_1 wieder der Safrabaum S_0:



Folgezustände von S_3

Da sich der Safrabaum S_3 von S_2 nur durch die Markierung des Knotens 2 unterscheidet, laufen die Schritte 2–6 genauso ab, nachdem in Schritt 1 die Markierung entfernt wurde. Damit hat S_3 dieselben Folgezustände wie S_2 (nämlich S_3 für a und S_0 für b):



Damit sind alle erreichbaren Zustände (Safrabäume) erzeugt.

Akzeptanzkomponente

Laut Folie 72 ist $\mathcal{P} = \{(E_1, F_1), (E_2, F_2)\} = \{(\emptyset, \emptyset), (\{S_1, S_1\}, \{S_3\})\}$.

T3.16 Korrektheitsbeweis der Safra-Konstruktion, Details

Hilfsaussage [HA]. Für alle T_i und alle Zustände p im Makrozustand **(MZ)** von v in T_{i+1} gibt es einen Zustand q im Makrozustand von v in T_i und einen endlichen Run $q \dots p$ von $\mathcal{A}$ auf dem zugehörigen Teilwort von α, der einen akzeptierenden Zustand enthält.

Skizze:

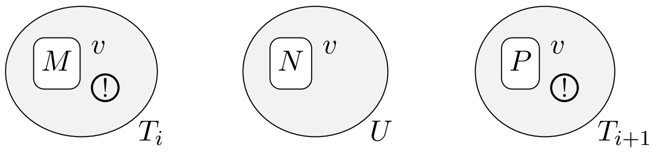
Teilrun: q $\cdots$ f $\cdots$ p

Beweis der HA. Damit v überhaupt mit $\textcircled{!}$ markiert werden kann, muss v direkt vor der entsprechenden Anwendung des Schrittes 6 Kinder haben. Diese wurden in irgendeiner vorigen Anwendung von Schritt 2 erzeugt. Damit dieser Schritt angewendet worden sein kann, muss gelten:

($*$) Zwischen T_i und T_{i+1} gibt es einen Zeitpunkt, zu dem der Makrozustand von v einen akzeptierenden Zustand $f \in F$ enthält.

Wir betrachten nun den Teilrun $T_i \dots T_{i+1}$ von s. Sei U ein Safra-Baum zwischen T_i und T_{i+1}, so dass Knoten v in U Bedingung ($*$) erfüllt, und seien $T_i = S_k$, $U = S_\ell$ und $T_{i+1} = S_m$ für entsprechende k, ℓ, m mit $0 \le k \le \ell < m$ (die Zeitpunkte des Vorkommens von T_i, U, T_{i+1} auf dem Run s; mit $k \le \ell$ ist also auch $U = T_i$ erlaubt).

Seien M, N, P die Makrozustände des Knotens v in T_i, U, T_{i+1}:



Mit den eingeführten Bezeichnungen lässt sich die zu zeigende HA nun so formulieren:

($**$) Für alle $p \in P$ gibt es ein $q \in M$ und einen endlichen Run $q \dots p$ von $\mathcal{A}$ auf $\alpha[k, m-1]$, der einen akzeptierenden Zustand enthält.

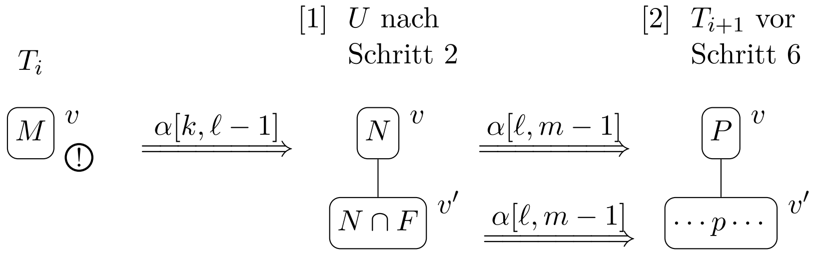
Um ($**$) zu beweisen, betrachten wir zunächst den Spezialfall, dass U der einzige Safra-Baum zwischen T_i und T_{i+1} ist, der ($*$) erfüllt. Im nächsten Schritt argumentieren wir dann für den allgemeinen Fall.

Im Spezialfall betrachten wir die endlichen Teilruns $T \dots U$ und $U \dots T_{i+1}$ von s auf den Teilwörtern $\alpha[k, \ell-1]$ bzw. $\alpha[\ell, m-1]$. Wir schauen uns dazu genauer [1] die Berechnung des Nachfolger-Baums von U und [2] die Berechnung von T_{i+1} aus seinem Vorgängerbaum X (evtl. ist $X = U$) an je einer bestimmten Stelle der Safra-Konstruktion von Δ^d an:

[1] Während der Berechnung des Übergangs $(U, \alpha_\ell, \cdot) \in \Delta^d$ wird in *Schritt 2* ein Kind v' von v mit Makrozustand $N \cap F$ erzeugt. Dieser Knoten v' bleibt in allen Safrabäumen vor T_{i+1} erhalten, weil kein Schritt 6 angewendet wird, denn v ist bis T_{i+1} nicht mit $\textcircled{!}$ markiert.

[2] Während der Berechnung des Übergangs $(X, \alpha_{m-1}, T_{i+1}) \in \Delta^d$ muss die Bedingung in *Schritt 6* erfüllt sein, da v in T_{i+1} markiert ist. Folglich haben direkt vorher die Knoten v und v' denselben Makrozustand (denn wir sind im Spezialfall; also werden keine weiteren Kinder von v erzeugt).

Mit diesen Erkenntnissen kann man den Teilrun $T_i \dots T_{i+1}$ schematisch so veranschaulichen:



Da neue Makrozustände in Schritt 3 (knotenweise Potenzmengenkonstruktion) erzeugt werden, bedeutet die „untere Zeile" des Schemas:

> Für alle $p \in P$ gibt es ein $p' \in N \cap F$ und einen endlichen Run $p' \dots p$ von $\mathcal{A}$ auf $\alpha[\ell, m - 1]$.

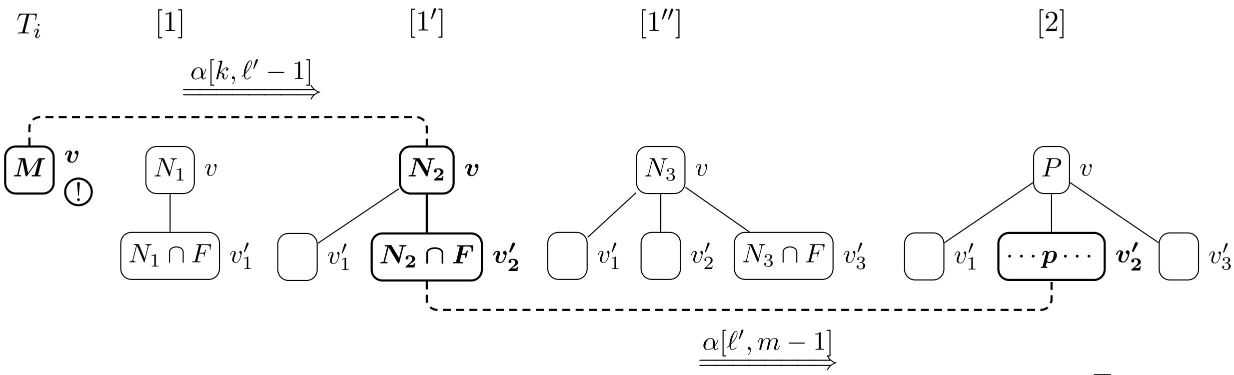
Diesen Run kann man durch die „obere Zeile" ergänzen:

> Für alle $p' \in N \cap F$ gibt es ein $q \in M$ und einen endlichen Run $q \dots p'$ von $\mathcal{A}$ auf $\alpha[k, \ell - 1]$.

Aus diesen beiden Aussagen erhält man wie gewünscht $(**)$.

Es kann natürlich mehrere solche Runs geben kann; es genügt aber zu wissen, dass es mindestens einen gibt. Es ist außerdem zu beachten, dass die „Leserichtung" von $(**)$ „rückwärts" ist, also „für alle $p \in P$ existiert ein $q \in M$ …" und *nicht* umgekehrt.

Nun betrachten wir den allgemeinen Fall, dass es mehrere Zwischenzustände der Art U mit Eigenschaft $(*)$ gibt. Dann gibt es auch mehrere Situationen [1], und in [2] werden die Makrozustände *aller* zugehörigen Kinder vereinigt. Um nun den gesuchten Run $q \dots p$ auf $\alpha[k, m - 1]$ zu erhalten, muss man nach der ersten „passenden" Situation [1] von der unteren zur oberen Zeile übergehen, d. h. wir erhalten $(**)$, indem wir für $p \in P$ in derjenigen Kopie von [1] in die obere Zeile gehen, in der dasjenige Kind v' von v erzeugt wird, das zu $p \in P$ führt.



❏

T3.17 Vollständigkeitsbeweis der Safra-Konstruktion, Details

Hilfsaussage [HA]. Es gibt einen Knotennamen v, für den gilt:

(a) $\exists m \geqslant 0 : S_i$ enthält Knoten v für alle $i \geqslant m$

(b) v ist in ∞ vielen S_i mit ! markiert

(Diese Aussage entspricht genau der Akzeptanzbedingung $\mathcal{P}^d$.)

Skizze: $S_0, S_1, \ldots, \underbrace{S_m, S_{m+1}, S_{m+2}, \ldots}_{\substack{v \text{ in allen } S_i \text{ enthalten} \\ \text{und unendlich oft markiert}}}$

Beweis der HA. Nach Konstruktion enthält der Makrozustand, der im Safrabaum S_i zu Knoten 1 gehört, alle Zustände von $\mathcal{A}$, die von einem Anfangszustand $q \in I$ aus erreicht werden können, indem die ersten i Zeichen von α gelesen werden (Potenzmengenkonstruktion). Deshalb enthält Knoten 1 in S_i immer den Zustand q_i aus dem Run r; somit hat Knoten 1 immer einen nichtleeren Makrozustand und wird nie in Schritt 5 entfernt. Damit erfüllt Knoten 1 die Bedingung (a) aus der Hilfsaussage. Wenn er auch Bedingung (b) erfüllt, ist der Beweis erbracht.

Anderenfalls gibt es einen Zeitpunkt $m' \geq 0$, so dass Knoten 1 in allen S_i *nicht* mit ! markiert ist:

$$S_0, S_1, \ldots, S_m, S_{m+1}, \ldots, \underbrace{S'_m, S_{m'+1}, \ldots, S_p, \ldots}_{\text{Knoten 1 ist in keinem } S_i \text{ markiert}}$$

Da der Run r erfolgreich ist, gibt es einen Zustand $f \in \mathsf{Inf}(r) \cap F$. Sei p der erste Zeitpunkt des Auftretens von f in r hinter m' (d. h. $q_p = f$ und $q_i \neq f$ für alle i mit $m < i < p$). Da $q_p = f$, tritt f im Makrozustand des Knotens 1 in S_p auf. Folglich wird in Schritt 2 der Berechnung von S_{p+1} ein neues jüngstes Kind zu Knoten 1 hinzugefügt, dessen Makrozustand f enthält. Da Knoten 1 für den Rest des Runs unmarkiert bleibt, wird q_i aus dem Run r für alle $i \geq p+1$ in einem *Kind* von 1 auftreten. Nach endlich vielen Anwendungen von Schritt 4 muss q_i dauerhaft in einem *festen* Kind c von 1 bleiben. Dieses Kind erfüllt also Bedingung (a) der HA.

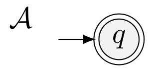
Nun kann man das bisherige Argument von 1 auf c übertragen: entweder erfüllt c auch Bedingung (b), oder es gibt ein Kind c', das (a) erfüllt. Diese Iteration kann man nicht beliebig oft fortsetzen, weil die Tiefe eines Safrabaums durch $|Q|$ beschränkt ist. Folglich muss es einen Nachfahren von 1 geben, der Bedingungen (a) und (b) erfüllt. ❏

T3.18 Anmerkungen zum Algorithmus fürs Leerheitsproblem

Es genügt nicht, in der **if**-Zeile auf die Bedingung

$$L(\mathcal{A}_{q_0,q_f}) = \emptyset \text{ oder } L(\mathcal{A}_{q_f,q_f}) = \emptyset$$

zu prüfen, denn dann würde der Algorithmus bei Eingabe des Automaten

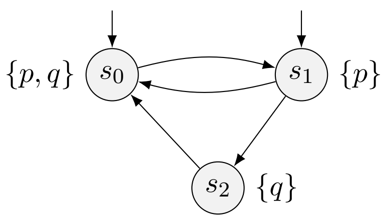


nach Raten von $q_0 = q_f = q$ „nicht leer“ ausgeben, obwohl $L_\omega(\mathcal{A}) = \emptyset$.

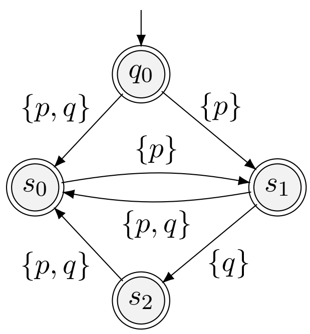
Der Test $L(\mathcal{A}_{q_f,q_f}) \subseteq \{\varepsilon\}$ lässt sich durch den Leerheitstest $L(\mathcal{A}_{q_f,q_f}) \cap (\Sigma^* \setminus \{\varepsilon\}) = \emptyset$ realisieren, wobei man für $\Sigma^* \setminus \{\varepsilon\}$ leicht einen (von $\mathcal{A}_{q_f,q_f}$ unabhängigen) NEA konstruieren kann. Der Produktautomat muss dabei „on the fly“ konstruiert werden, d. h. es werden immer nur die während des Leerheitstests aktuell erforderlichen Zustände und Übergänge im Speicher gehalten und niemals der gesamte Automat.

T3.19 NBA für eine Kripke-Struktur

Kripke-Struktur $\mathcal{S}$



zugehöriger NBA $\mathcal{A}_\mathcal{S}$

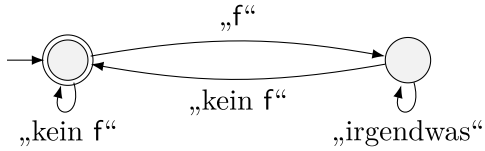


T3.20 NBAs für Beispiel-Eigenschaften

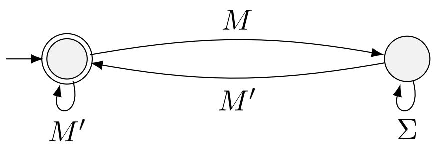
Im Mikrowellen-Beispiel ist das Alphabet $\Sigma = 2^{\{\mathsf{g},\mathsf{o},\mathsf{h},\mathsf{f}\}}$ (die Buchstaben stehen für gestartet, offen, heizt bzw. fehler).

(a) „Wenn ein Fehler auftritt, dann ist er nach endlicher Zeit behoben.“

Schematisch muss der Automat so aussehen:

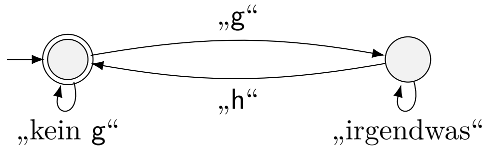


Dabei steht z. B. die Beschriftung „f“ für alle Alphabetzeichen (Teilmengen von $\{\mathsf{g},\mathsf{o},\mathsf{h},\mathsf{f}\}$), die f enthalten, und „kein f“ für alle Alphabetzeichen, die f nicht enthalten. Sei also $M = \{\{\mathsf{f}\},\{\mathsf{g},\mathsf{f}\},\{\mathsf{o},\mathsf{f}\},\ldots,\{\mathsf{g},\mathsf{o},\mathsf{h},\mathsf{f}\}\}$ und $M' = \Sigma \setminus M$. Dann sind die korrekten Kantenbeschriftungen im Automaten wie folgt.



(b) „Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen.“

Hier nur die schematische Repräsentation des Automaten; die korrekten Kantenbeschriftungen erhält man wie in (a).

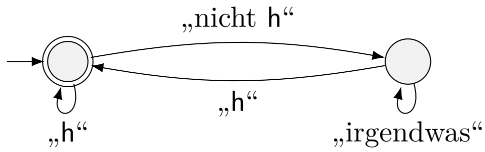


(c) „Wenn die Mikrowelle gestartet wird, ist es *möglich*, danach zu heizen.“

Der Automat ist derselbe wie in (b), nur muss man hier existenzielles Model-Checking statt universellem verwenden.

(d) „Die Mikrowelle heizt beliebig oft und ist beliebig oft offen.“

Der Automat $\mathcal{A}_1$, der beschreibt, dass die Mikrowelle beliebig oft heizt, ist dem in (a) sehr ähnlich (hier wieder nur die schematische Darstellung):



Der Automat $\mathcal{A}_2$ für „beliebig oft offen“ ist analog. Um zu beschreiben, dass die Mikrowelle *beides* beliebig oft tun, muss man den Produktautomaten von $\mathcal{A}_1$ und $\mathcal{A}_2$ bilden.

(e) „Die Mikrowelle *kann* beliebig oft heizen bzw. offen sein.“

Wie (d), aber mit existenziellem Model-Checking.

T3.21 Beispiele für LTL-Syntax und -Semantik

Betrachte folgenden Pfad π.

0	→ 1	→ 2	→ 3	→ 4	→ 5	→ 6	→ 7	→ ···		
	a	a	a	b		b	a	b	a	$\cdots$

Das heißt also, $\pi(0) = \emptyset$, $\pi(1) = \{a\}$ usw. Die Beschriftung „$b\ a\ \cdots$“ am rechten Rand bedeutet, dass für alle $i \geq 4$ gilt: $\pi(2i) = \{b\}$ und $\pi(2i+1) = \{a\}$. Dann gilt:

$\pi, 0 \not\models a$	$\pi, 0 \models \mathsf{F}a$	$\pi, 0 \not\models \mathsf{G}(a \vee b)$	$\pi, 1 \models a\ \mathsf{U}\, b$
$\pi, 0 \models \neg a$	$\pi, 0 \models \mathsf{F}b$	$\pi, 6 \models \mathsf{G}(a \vee b)$	$\pi, 0 \not\models a\ \mathsf{U}\, b$
$\pi, 0 \models \mathsf{X}a$	$\pi, 4 \models \mathsf{F}a$	$\pi, 5 \not\models \mathsf{G}(a \vee b)$	
$\pi, 0 \not\models \mathsf{X}\neg a$	$\pi, 0 \models \mathsf{X}(a \wedge \mathsf{F}b)$	$\pi, 0 \models \mathsf{GF}a$	

Dabei ist $\mathsf{GF}\varphi$ laut Semantik gleichbedeutend mit „unendlich oft φ“.

Betrachte nun den folgenden Pfad π'.

0	→ 1	→ 2	→ 3	→ 4	→ 5	→ 6	→ 7	→ ···
	a		a		a	b		

Dann gilt $\pi', 0 \not\models a\ \mathsf{U}\, b$, aber $\pi', 0 \models (\mathsf{X}a \vee \mathsf{XX}a)\ \mathsf{U}\, (\mathsf{X}b)$.

T3.22 Beispiele für die Erweiterung von Pfaden

Sei z. B. $\varphi_E = \mathsf{X}(a\ \mathsf{U}\, b)$ und der Pfad $\pi = s_0 s_1 s_2 \cdots$ wie folgt gegeben:

0	→ 1	→ 2	→ 3	→ 4	→ 5	→ 6	→ ···
a	a	b		a	a	b	

Das heißt also $s_0 = s_1 = \{a\}$, $s_2 = \{b\}$, $s_3 = \emptyset$ usw.; dieser Pfad entspricht somit dem Eingabewort, das aus den Zeichen $\{a\}, \{b\}, \emptyset, \{a\}, \{b\}, \emptyset, \ldots$ besteht (jede Teilmenge von AV ist ein Zeichen!).

Der zugehörige erweiterte Pfad $\overline{\pi} = t_0 t_1 t_2 \cdots$ ist der folgende.

0	→ 1	→ 2	→ 3	→ 4	→ 5	→ 6	→ ···
a	a	b		a	a	b	
$a\ \mathsf{U}\, b$	$a\ \mathsf{U}\, b$	$a\ \mathsf{U}\, b$	$\neg(a\ \mathsf{U}\, b)$	$a\ \mathsf{U}\, b$	$a\ \mathsf{U}\, b$	$a\ \mathsf{U}\, b$	
$\mathsf{X}(a\ \mathsf{U}\, b)$	$\mathsf{X}(a\ \mathsf{U}\, b)$	$\neg\mathsf{X}(a\ \mathsf{U}\, b)$	$\mathsf{X}(a\ \mathsf{U}\, b)$	$\mathsf{X}(a\ \mathsf{U}\, b)$	$\mathsf{X}(a\ \mathsf{U}\, b)$	$\neg\mathsf{X}(a\ \mathsf{U}\, b)$	
$\neg b$	$\neg b$	$\neg a$	$\neg a, \neg b$	$\neg b$	$\neg b$	$\neg a$	
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Das heißt also z. B.: t_0 ist die elementare Formelmenge $\{a,\ a\ \mathsf{U}\, b,\ \mathsf{X}(a\ \mathsf{U}\, b),\ \neg b\} \subseteq \mathsf{cl}(\varphi_E)$.

T3.23 Beispiele für elementare Formelmengen

Sei $\varphi_E = a\, \mathsf{U}\, (\neg a \wedge b)$. Dann ist $\mathsf{cl}(\varphi_E) = \{a,\ \neg a,\ b,\ \neg b,\ \neg a \wedge b,\ \neg(\neg a \wedge b),\ \varphi_E,\ \neg\varphi_E\}$.

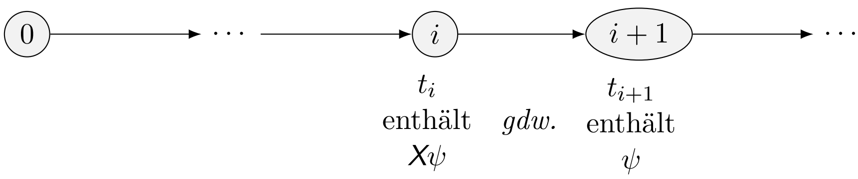
- $\{a, b, \varphi_E\}$ ist konsistent bezüglich der Aussagenlogik und lokal konsistent bzgl. des U-Operators, aber nicht maximal, weil sie weder $\neg a \wedge b$ noch $\neg(\neg a \wedge b)$ enthält.
- Fügt man $\neg a \wedge b$ hinzu, so ist die resultierende Menge $\{a,\ b,\ \neg a \wedge b,\ \varphi_E\}$ zwar maximal, aber nicht mehr konsistent bezüglich der Aussagenlogik, da sie wegen $\neg a \wedge b$ auch $\neg a$ enthalten müsste.
- Fügt man stattdessen $\neg(\neg a \wedge b)$ hinzu, so ist die neue Menge $\{a,\ b,\ \neg(\neg a \wedge b),\ \varphi_E\}$ maximal und konsistent bezüglich der Aussagenlogik, aber nicht mehr konsistent bezüglich U, weil sie nun zwar $\varphi_E = a\, \mathsf{U}\, (\neg a \wedge b)$, aber weder $\neg a \wedge b$ noch a enthält.
- Die elementaren Formelmengen sind folgende.

$$
\begin{array}{l}
\{\ \ a,\ \ \ b,\ \neg(\neg a \wedge b),\ \ \ \varphi_E\} \\
\{\ \ a,\ \ \ b,\ \neg(\neg a \wedge b),\ \neg\varphi_E\} \\
\{\ \ a,\ \neg b,\ \neg(\neg a \wedge b),\ \ \ \varphi_E\} \\
\{\ \ a,\ \neg b,\ \neg(\neg a \wedge b),\ \neg\varphi_E\} \\
\{\neg a,\ \neg b,\ \neg(\neg a \wedge b),\ \neg\varphi_E\} \\
\{\neg a,\ \ \ b,\ \ \ \ \ \ \neg a \wedge b,\ \ \ \varphi_E\}
\end{array}
$$

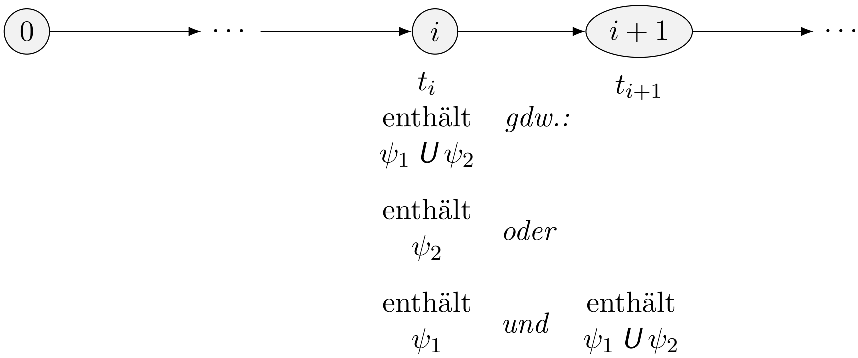
T3.24 Skizzen zur Def. der Übergangsrelation des GNBA

Bedingung ① besagt, dass im Beispiel in T3.22 *höchstens* die Übergänge $(t_0, \{a\}, t_1)$, $(t_1, \{a\}, t_2)$, $(t_2, \{b\}, t_3)$, $(t_3, \emptyset, t_4)$ usw. erlaubt sind (sofern jeweils ② und ③ auch erfüllt sind).

Bedingung ② kann man so veranschaulichen:



Bedingung ③ kann man so veranschaulichen:



Diese Bedingung nutzt die semantische Äquivalenz $\psi_1\, \mathsf{U}\, \psi_2 \equiv \psi_2 \vee (\psi_1 \wedge \mathsf{X}(\psi_1\, \mathsf{U}\, \psi_2))$ (weist diese selbst nach :–)).

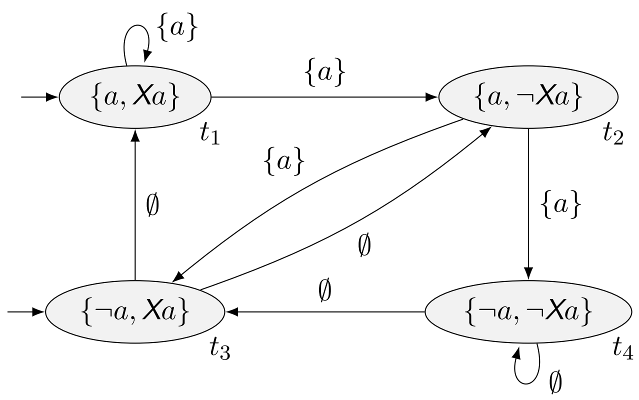
T3.25 GNBA für die Beispiel-Formel Xa

- $\varphi = Xa$
- $\mathsf{cl}(\varphi) = \{a, \neg a, Xa, \neg Xa\}$
- Zustände (elementare Formelmengen): da keine $\wedge$- oder U-Teilformel vorhanden ist, sind nur Konsistenz bezüglich $\neg$ und Maximalität relevant. Es gibt also folgende vier elementare Formelmengen:

$$t_1 = \{a, Xa\}$$
$$t_2 = \{a, \neg Xa\}$$
$$t_3 = \{\neg a, Xa\}$$
$$t_4 = \{\neg a, \neg Xa\}$$

- Übergänge: es genügt, Bedingung ② für X zu überprüfen, also gibt es Übergange
  - von t_1 mit $\{a\}$ zu t_1 und t_2
  - von t_2 mit $\{a\}$ zu t_3 und t_4
  - von t_3 mit $\emptyset$ zu t_1 und t_2
  - von t_4 mit $\emptyset$ zu t_3 und t_4
- Anfangszustände: t_1, t_3 (diese enthalten φ)
- Akzeptanzkomponente: $\mathcal{F} = \emptyset$, denn es gibt keine U-Teilformeln (intuitiv: also braucht auch kein unendlich langes „Aufschieben“ verhindert zu werden). Folglich sind *alle* Runs erfolgreich (vgl. Definition GNBA).

Graphische Darstellung:



T3.26 GNBA für die Beispiel-Formel $(\neg a)\, \mathsf{U}\, b$

- $\varphi = (\neg a)\, \mathsf{U}\, b$
- $\mathsf{cl}(\varphi) = \{a, \neg a, b, \neg b, (\neg a)\, \mathsf{U}\, b, \neg((\neg a)\, \mathsf{U}\, b)\}$
- Zustände (elementare Formelmengen): Wegen Konsistenz bezgüglich $\neg$ und Maximalität muss von $a, \neg a$ bzw. $b, \neg b$ bzw. $(\neg a)\, \mathsf{U}\, b, \neg((\neg a)\, \mathsf{U}\, b)$ jeweils genau eine Formel in der Menge enthalten sein. Damit gibt es höchstens $2^3 = 8$ elementare Formelmengen. Davon sind aber drei nicht konsistent bzgl. U:
  - $\{a,\ b,\ \neg((\neg a)\, \mathsf{U}\, b)\}$ und $\{\neg a,\ b,\ \neg((\neg a)\, \mathsf{U}\, b)\}$
    (denn wenn $b \in t$, dann muss auch $\neg((\neg a)\, \mathsf{U}\, b) \in t$ sein);
  - $\{a,\ \neg b,\ (\neg a)\, \mathsf{U}\, b\}$ (denn wenn $(\neg a)\, \mathsf{U}\, b \in t$ und $b \notin t$, dann muss $a \in t$ sein).

  Die verbleibenden fünf Formelmengen sind elementar:

  $$\begin{aligned} t_1 &= \{a,\ b,\ (\neg a)\, \mathsf{U}\, b\} & t_4 &= \{\neg a,\ \neg b,\ \neg((\neg a)\, \mathsf{U}\, b)\} \\ t_2 &= \{\neg a,\ b,\ (\neg a)\, \mathsf{U}\, b\} & t_5 &= \{a,\ \neg b,\ \neg((\neg a)\, \mathsf{U}\, b)\} \\ t_3 &= \{\neg a,\ \neg b,\ (\neg a)\, \mathsf{U}\, b\} \end{aligned}$$

- Übergänge: Bedingung ③ für U muss eingehalten werden, d. h.:

  $$(\neg a)\, \mathsf{U}\, b \in t \quad \text{gdw.} \quad b \in t \ \text{ oder } \ (\neg a \in t \ \text{ und } \ (\neg a)\, \mathsf{U}\, b \in t') \qquad (*)$$

  Nun kann man $(*)$ in folgende logisch äquivalente Aussage umformen, die nicht mehr „gdw." benutzt:

  $$\begin{aligned} & \big((\neg a)\, \mathsf{U}\, b \in t \ \text{ und } \ b \in t\big) && \text{(a)} \\ \text{oder } & \big((\neg a)\, \mathsf{U}\, b \in t \ \text{ und } \ \neg a \in t \ \text{ und } \ (\neg a)\, \mathsf{U}\, b \in t'\big) && \text{(b)} \\ \text{oder } & \big((\neg a)\, \mathsf{U}\, b \notin t \ \text{ und } \ b \notin t \ \text{ und } \ \neg a \notin t\big) && \text{(c)} \\ \text{oder } & \big((\neg a)\, \mathsf{U}\, b \notin t \ \text{ und } \ b \notin t \ \text{ und } \ (\neg a)\, \mathsf{U}\, b \notin t'\big) && \text{(d)} \end{aligned}$$

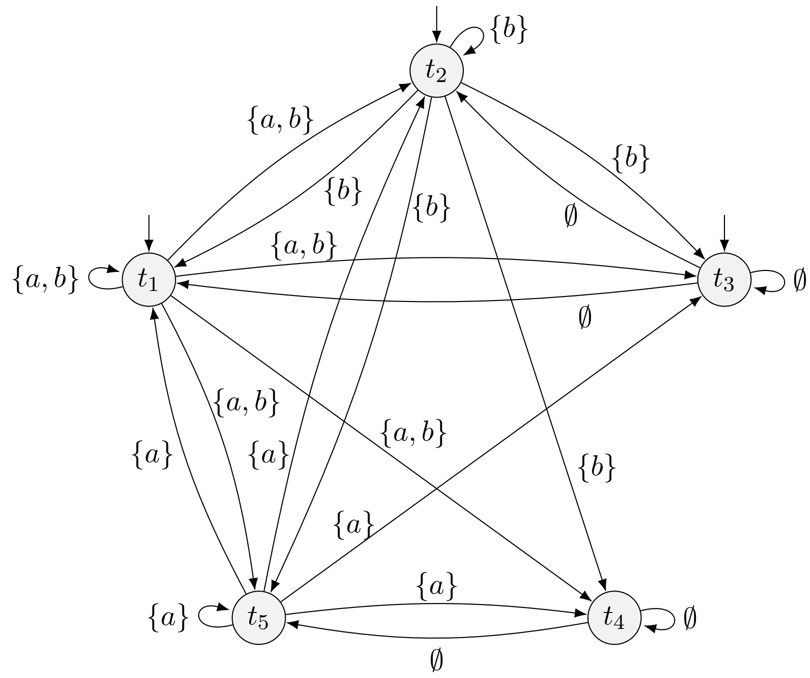
  Folglich gibt es Übergänge

  (a) von t_1 mit $\{a, b\}$ zu t_1, t_2, t_3, t_4, t_5 und
  von t_2 mit $\{b\}$ zu t_1, t_2, t_3, t_4, t_5;
  (b) von t_3 mit $\emptyset$ zu t_1, t_2, t_3;
  (c) von t_5 mit $\{a\}$ zu t_1, t_2, t_3, t_4, t_5;
  (d) von t_4 mit $\emptyset$ zu t_4, t_5.
- Anfangszustände: t_1, t_2, t_3 (diese enthalten φ)
- Akzeptanzkomponente:

  $$\begin{aligned} \mathcal{F} &= \left\{ M_{(\neg a)\mathsf{U}b} \right\} \\ &= \left\{ \{t \mid (\neg a)\, \mathsf{U}\, b \notin t \text{ oder } b \in t\} \right\} \\ &= \left\{ \{t_1, t_2, t_4, t_5\} \right\} \end{aligned}$$

  Es werden also Runs ausgeschlossen, die irgendwann nur noch durch t_3 gehen. Auf solchen Runs wird genau $(\neg a)\, \mathsf{U} b$ unendlich lange „aufgeschoben", denn $(\neg a)\, \mathsf{U} b \in t_3$ und $b \notin t_3$.

Graphische Darstellung:


$\{b\}$
t_2
$\{a,b\}$
$\{b\}$
$\{b\}$
$\{b\}$
$\emptyset$
$\{a,b\}$
$\{a,b\}$
t_1
t_3
$\emptyset$
$\emptyset$
$\{a,b\}$
$\{a,b\}$
$\{a\}$
$\{a\}$
$\{b\}$
$\{a\}$
$\{a\}$
$\{a\}$
t_5
t_4
$\emptyset$
$\emptyset$

Teil IV.

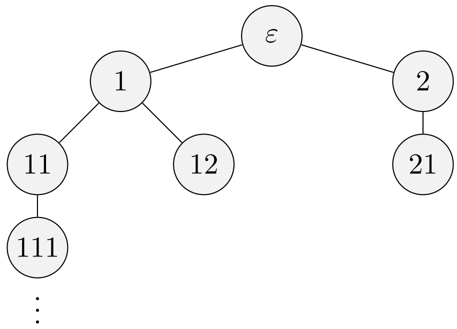
Alternierung

T4.1 Beispiel Semantik positiver Boolescher Formeln

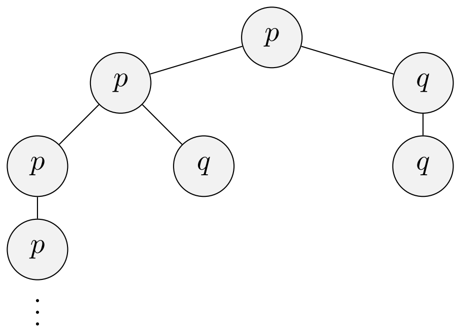
Wir betrachten die PBF $\varphi = (p \wedge q) \vee (r \wedge s)$ über der Variablenmenge $X = \{p, q, r, s\}$ und die Belegungen $Y_1 = \{p, q, r\}$ und $Y_2 = \{p, r\}$. Dann gilt $Y_1 \models \varphi$, aber $Y_2 \not\models \varphi$.

T4.2 Beispiel eines Run-Baums

Ein möglicher Baum mit Verzweigungsgrad $n = 2$ sieht so aus:



Ein möglicher Pfad ist $\pi = \varepsilon \cdot 2 \cdot 21$; dieser Pfad ist endlich. Wenn $\Sigma = \{p, q\}$, dann ist ein Σ-Baum z. B. folgender:



T4.3 Beispiel eines alternierenden Büchi-Automaten

Sei $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, F)$ der ABA mit

$$Q = \{p, q\}, \quad \Sigma = \{a, b\}, \quad F = \{p\}, \quad q_I = p,$$

$$\delta(p, a) = p \wedge q \tag{1}$$

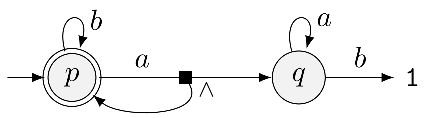
$$\delta(p, b) = p \tag{2}$$

$$\delta(q, a) = q \tag{3}$$

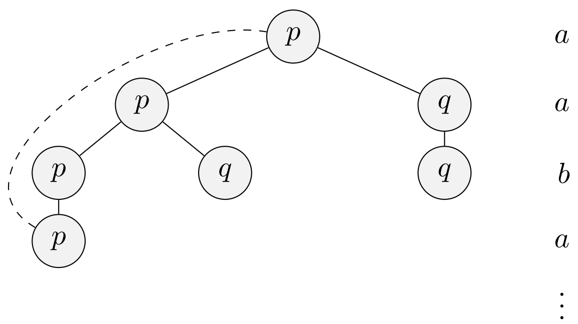
$$\delta(q, b) = \mathtt{1}. \tag{4}$$

Dabei bedeutet Übergang (1) von δ intuitiv: „schicke zwei Kopien von $\mathcal{A}$ in Zustand p bzw. q zum nächsten Zeichen". Übergänge (2) und (3) entsprechen den Übergängen eines DBA. Übergang (4) bedeutet: „schneide den Pfad ab; hier ist die Berechnung erfolgreich". Man beachte auch, dass der Fall $\delta(\cdot,\cdot) = \mathtt{0}$ einem erfolg*losen* Abbruch der Berechnung entspricht.

Diesen ABA kann man graphisch wie folgt darstellen:



Wir betrachten außerdem das Eingabewort $\alpha = (aab)^\omega$. Ein möglicher Run von $\mathcal{A}$ auf α ist der Baum aus dem vorigen Beispiel, wenn man ihn ab Position 111 so fortsetzt, dass sich der dargestellte Teilbaum unendlich oft wiederholt. Dies ist im folgenden Bild durch die gestrichelte Linie dargestellt; die zu jeder Ebene gehörigen Buchstaben von α stehen rechts daneben.



Dieser Run ist erfolgreich, denn der einzige unendliche Pfad ist $\pi = \varepsilon \cdot 1 \cdot 11 \cdot 111 \cdots$, welcher unendlich oft p enthält und damit der Büchi-Akzeptanzbedingung $F = \{q\}$ entspricht.

Die von $\mathcal{A}$ erkannte Sprache ist die Menge aller Wörter, die unendlich oft b enthalten, also $L_\omega(\mathcal{A}) = L(((a+b)^*b)^\omega)$. Dies lässt sich präzise so zeigen:

„$\supseteq$" Enthalte $\alpha \in \{a,b\}^\omega$ unendlich oft b. Um zu zeigen, dass α von $\mathcal{A}$ akzeptiert wird, betrachten wir den „minimalen" Run (P,t) von $\mathcal{A}$ auf α, also denjenigen Run, der bei Übergängen (2) und (3) nur ein Kind mit p bzw. q enthält und bei Übergang (4) keine Kinder.[1] Jeder Pfad π dieses Runs kann wegen $\delta(q,b) = \mathtt{1}$ nicht in einer Folge q^ω enden. Da außerdem δ keinen Übergang von q nach p erlaubt, kann π also nur entweder endlich sein oder aus nur p's bestehen. Im letzteren Fall ist die Akzeptanzbedingung erfüllt.

[1] Hier ist von einem „minimalen Run" die Rede, weil prinzipiell immer auch mehr Kinder erlaubt sind, denn wenn eine Belegung eine PBF erfüllt, dann auch jede Erweiterung (d. h. Obermenge). Deshalb nennt man PBFs auch manchmal *monotone* Boolesche Formeln.

„$\subseteq$“ Per Kontraposition: enthalte $\alpha \in \{a,b\}^\omega$ endlich oft b. Dann gibt es eine Position m, ab der α nur noch aus a's besteht. Sei (P,t) ein beliebiger Run von $\mathcal{A}$ auf α. Wir müssen zeigen, dass dieser einen unendlichen Pfad π enthält, der nicht die Akzeptanzbedingung von $\mathcal{A}$ erfüllt, der also nur endlich oft p enthält. Da der Anfangszustand p ist, gibt es wegen Übergängen (1) und (2) einen unendlichen Pfad in (P,t), der nur aus p's besteht. Die Position in Tiefe m auf diesem Pfad hat aber wegen Übergang (1) ein mit q markiertes Kind. In diesem Kind entspringt nun wegen Übergang (3) ein unendlicher Pfad, der nur noch aus q's besteht. Damit ist der gewünschte Pfad π gefunden.

T4.4 Beispiele für die Übersetzung LTL $\rightarrow$ ABA

Beispiel 1. Sei $\varphi = a\, U\, b$.
Dann ist $\mathsf{cl}(\varphi) = \{a,\, \neg a,\, b,\, \neg b,\, a\, U b,\, \neg(a\, U b)\}$, und der ABA $\mathcal{A}_\varphi = (Q, \Sigma, \delta, \{q_I\}, F)$ hat folgende Bestandteile.

$$\begin{aligned}
Q &= \{q_a,\, q_{\neg a},\, q_b,\, q_{\neg b},\, q_{a U b},\, q_{\neg(a U b)}\}\\
\Sigma &= \{\emptyset,\, \{a\},\, \{b\},\, \{a,b\}\}\\
q_I &= q_{a U b}\\
F &= \{q_{\neg(a U b)}\}
\end{aligned}$$

Die Übergangsfunktion δ ist durch die folgende Tabelle gegeben.

Zustand \ Zeichen	$\emptyset$	$\{a\}$	$\{b\}$	$\{a,b\}$
q_a	0	1	0	1
$q_{\neg a}$	1	0	1	0
q_b	0	0	1	1
$q_{\neg b}$	1	1	0	0
$q_{a U b}$	0	$q_{a U b}$	1	1
$q_{\neg(a U b)}$	1	$q_{\neg(a U b)}$	0	0

Die erste Zeile (q_a) ist durch die erste Gleichung für δ auf Folie 4.15 gegeben (Fall q_x), die zweite Zeile durch die folgende Gleichung (Fall $q_{\sim\psi}$). Die nächsten beiden Zeilen ($q_b, q_{\neg b}$) sind analog. Für die fünfte Zeile ($q_{a U b}$) betrachten wir die Spalte $\{a\}$. Die letzte Gleichung für δ auf Folie 4.15 lautet

$$\delta(q_{\psi U \vartheta}, a) = \delta(q_\vartheta, a) \vee \big(\delta(q_\psi, a) \wedge q_{\psi U \vartheta}\big),$$

also hier

$$\delta(q_{a U b}, \{a\}) = \delta(q_b, \{a\}) \vee \big(\delta(q_a, \{a\}) \wedge q_{a U b}\big). \tag{$*$}$$

Die Teile $\delta(q_b, \{a\})$ und $(\delta(q_a, \{a\})$ der Formel auf der rechten Seite von $(*)$ kann man aus der Tabelle ablesen: sie haben die Werte 0 bzw. 1. Man erhält also

$$\delta(q_{a\,\mathsf{U} b}, \{a\}) = \mathtt{0} \vee (\mathtt{1} \wedge q_{a\,\mathsf{U} b})$$

und mit den standardmäßigen Äquivalenzen der Aussagenlogik wird daraus

$$\delta(q_{a\,\mathsf{U} b}, \{a\}) = q_{a\,\mathsf{U} b}.$$

Für die anderen Spalten der 5. Zeile ergibt sich durch die geänderten Werte von $\delta(q_b, \{a\})$ und $\delta(q_a, \{a\})$ der Wert 0 bzw. 1. Die letzte Zeile ist wieder durch die vorhergehende Zeile und den Fall $q_{\sim\psi}$ bestimmt.

Wegen der Form von δ benutzt $\mathcal{A}_\varphi$ nicht die „volle Mächtigkeit" der Alternierung, sondern nur die Möglichkeit, Pfade erfolgreich oder erfolglos „abzubrechen" (Werte 1 bzw. 0).

Läufe des Automaten auf einigen Eingaben. Sei $\alpha = \{a\}\,\{a\}\,\{a\}\,\{b\}\,\emptyset^\omega$ das Wort, das dem LTL-Pfad entspricht, auf dem in den ersten 3 Zuständen die Aussagenvariablen a, im 4. Zustand b und in den übrigen Zuständen keine Aussagenvariable „gesetzt" ist. Dieser Pfad erfüllt im Anfangszustand die Formel $a\,\mathsf{U}\,b$, also muss $\mathcal{A}_\varphi$ auch α akzeptieren. Letzteres ist tatsächlich der Fall, und zwar mittels des Runs, der aus einem einzigen endlichen Pfad der Länge 4 besteht, wobei alle Positionen mit $q_{a\,\mathsf{U} b}$ markiert sind (überprüft es selbst mit der obigen Tabelle). Analog wird das Wort $\{b\}\,\emptyset^\omega$ von $\mathcal{A}_\varphi$ akzeptiert, und zwar auf einem einelementigen Run mit der Markierung $q_{a\,\mathsf{U} b}$. Im Gegensatz dazu wird $\{a\}^\omega$ nicht akzeptiert, denn hier besteht der einzig mögliche Run aus einem *unendlichen* mit $q_{a\,\mathsf{U} b}$ markierten Pfad, welcher die Akzeptanzbedingung $F = \{q_{\neg(a\,\mathsf{U} b)}\}$ verletzt und genau der zu vermeidenden Situation entspricht, dass $\neg(a\,\mathsf{U}\,b)$ unendlich lange hinausgezögert wird.

Beispiel 2. Sei $\varphi = \mathsf{GF}a$. Die Standardvorgehensweise gemäß der Folien ist, die Formel so umzuformen, dass sie nur noch U verwendet (siehe Folie 4.12). Die resultierende Formel wäre jedoch deutlich größer, was wiederum zu einer unhandlich großen Zustandsmenge und Übergangsfunktion von $\mathcal{A}_\varphi$ führen würde. Es bietet sich deshalb hier an, die Konstruktion von $\mathcal{A}_\varphi$ von Folie 4.15 um die offensichtlichen Fälle für die Operatoren G und F zu erweitern, und zwar in der Übergangsfunktion und der Akzeptanzbedingung

$$\begin{aligned}
&\delta(q_{\mathsf{F}\psi}, a) = \delta(q_\psi, a) \vee q_{\mathsf{F}\psi}\\
&\delta(q_{\mathsf{G}\psi}, a) = \delta(q_\psi, a) \wedge q_{\mathsf{G}\psi} \qquad (*)
\end{aligned}$$

$$F = \{q_{\neg(\psi\,\mathsf{U}\vartheta)} \mid \neg(\psi\,\mathsf{U}\,\vartheta) \in \mathsf{cl}(\varphi)\} \;\cup\; \{q_{\neg\mathsf{F}\psi} \mid \neg\mathsf{F}\psi \in \mathsf{cl}(\varphi)\} \;\cup\; \{q_{\mathsf{G}\psi} \mid \mathsf{G}\psi \in \mathsf{cl}(\varphi)\}$$

Dann bleibt der ABA überschaubar: $\mathsf{cl}(\varphi) = \{a,\ \neg a,\ \mathsf{F}a,\ \neg\mathsf{F}a,\ \mathsf{GF}a,\ \neg\mathsf{GF}a\}$, und $\mathcal{A}_\varphi = (Q, \Sigma, \delta, \{q_I\}, F)$ hat folgende Bestandteile.

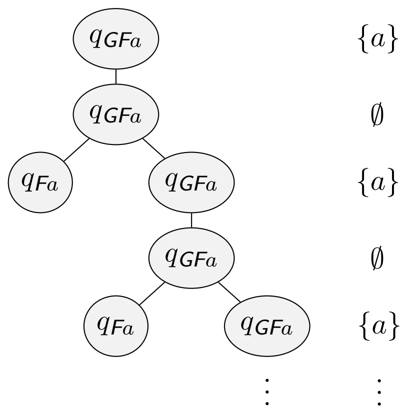
$$\begin{aligned}
Q &= \{q_a,\ q_{\neg a},\ q_{\mathsf{F}a},\ q_{\neg\mathsf{F}a},\ q_{\mathsf{GF}a},\ q_{\neg\mathsf{GF}a}\}\\
\Sigma &= \{\emptyset,\ \{a\}\}\\
q_I &= q_{\mathsf{GF}a}\\
F &= \{q_{\neg\mathsf{F}a}, q_{\mathsf{GF}a}\}
\end{aligned}$$

Die Übergangsfunktion δ ist durch die folgende Tabelle gegeben.

Zeichen / Zustand	$\emptyset$	$\{a\}$
q_a	$\mathsf{0}$	$\mathsf{1}$
$q_{\neg a}$	$\mathsf{1}$	$\mathsf{0}$
$q_{\mathsf{F}a}$	$q_{\mathsf{F}a}$	$\mathsf{1}$
$q_{\neg \mathsf{F}a}$	$q_{\neg \mathsf{F}a}$	$\mathsf{0}$
$q_{\mathsf{GF}a}$	$q_{\mathsf{F}a} \wedge q_{\mathsf{GF}a}$	$q_{\mathsf{GF}a}$
$q_{\neg \mathsf{GF}a}$	$q_{\neg \mathsf{F}a} \vee q_{\neg \mathsf{GF}a}$	$q_{\neg \mathsf{GF}a}$

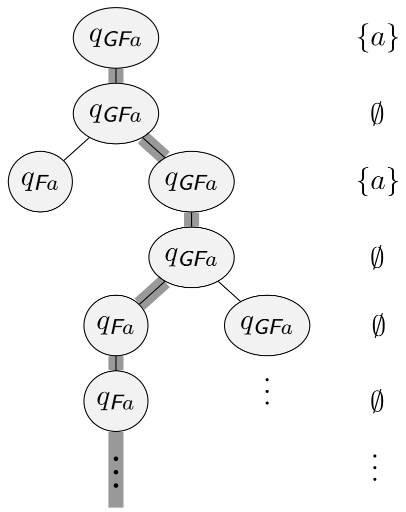
Die ersten 2 Zeilen werden wieder wie im vorigen Beispiel erzeugt; die dritte und vierte Zeile sind analog zu den Fällen $q_{a\,\mathsf{U}b}$ bzw. $q_{\neg(a\,\mathsf{U}b)}$. Die letzten beiden Zeilen bringen „echte" Alternierung; Zeile 5 sie kann man direkt aus Gleichung $(*)$ für δ und der 3. Tabellenzeile ablesen; Zeile 6 erhält man wieder durch „Negation" (Fall $q_{\sim\psi}$ auf Folie 4.15).

Läufe des Automaten auf einigen Eingaben. Sei $\alpha = (\{a\}\,\emptyset)^\omega$ das Wort, das dem LTL-Pfad entspricht, auf dem die Aussagenvariable a abwechselnd „gesetzt" und „nicht gesetzt" ist. Dieser LTL-Pfad erfüllt im Anfangszustand die Formel $\mathsf{GF}a$, also muss $\mathcal{A}_\varphi$ auch α akzeptieren. Letzteres ist tatsächlich der Fall, und zwar mittels des nebenstehenden Runs, dessen einziger unendlicher Pfad nur mit $q_{\mathsf{GF}a}$ markiert ist, also die Akzeptanzbedingung F erfüllt.



Im Gegensatz dazu wird $\{a\}\,\emptyset\,\{a\}\,\emptyset^\omega$ nicht akzeptiert, denn der einzig mögliche Run hat einen unendlichen Pfad, der auf $q_{\mathsf{F}a}^\omega$ endet und somit die Akzeptanzbedingung verletzt, siehe nebenstehendes Bild (besagter Pfad ist markiert). Dieser Pfad entspricht genau der zu vermeidenden Situation, dass $\mathsf{F}a$ unendlich lange hinausgezögert wird.

Es gibt hier nur diesen einzigen Run, weil die 6. Zeile der Tabelle nie zum Einsatz kommt.



T4.5-T4.11

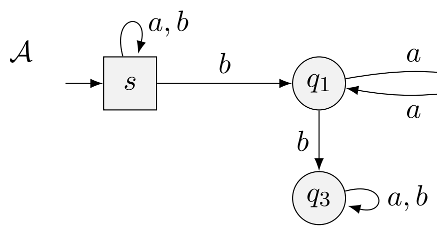
... folgen ...

T4.12 Beispiel für Paritätsautomat mit klass. Alternierung

Wir betrachten den KAPA $\mathcal{A} = (Q_\exists, Q_\forall, \Sigma, \Delta, \{s\}, c)$ mit

$$\begin{aligned} Q_\exists &= \{q_1, q_2, q_3\} & c(s) = c(q_3) &= 0 \\ Q_\forall &= \{s\} & c(q_1) = c(q_2) &= 1, \\ \Sigma &= \{a, b\} \end{aligned}$$

dessen Übergangsrelation Δ in folgendem Bild gegeben ist (existentielle Zustände sind rund, universelle eckig):



Die intuitive Idee hinter diesem Automaten ist, dass es in jedem Run (auf einem beliebigen ω-Wort) einen Pfad gibt, der nur den Zustand s besucht (wegen der a, b-Schleife), wobei bei jedem b ein weiterer Pfad abzweigt, der in q_1 beginnt und nur die q_i besucht. Es wird also dafür gesorgt, dass bei jedem b eine „Kopie“ des q_i-Teils von $\mathcal{A}$ auf dem Restwort läuft. Diese Kopien (bzw. die entsprechenden Pfade im Run) sind genau dann erfolgreich, wenn eine gerade Zahl a's und dann ein b gelesen werden. Damit akzeptiert $\mathcal{A}$ genau die ω-Wörter, die

- kein b enthalten oder
- unendlich viele b's sowie zwischen je zwei b's eine gerade Anzahl von a's enthalten,

d. h.:

$$L_\omega(\mathcal{A}) = a^\omega + a^*(b(aa)^*)^\omega$$

Beispiel-Runs von $\mathcal{A}$:

- auf $a(baa)^\omega$:

  ... Bild folgt ...

- auf aba^ω:

  ... Bild folgt ...

- auf $abaaa(baa)^\omega$:

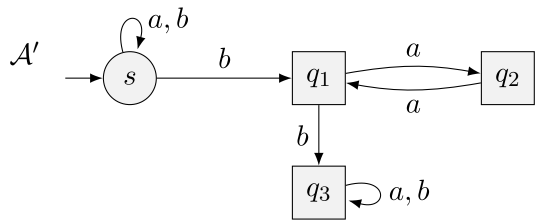
  ... Bild folgt ...

T4.13 Beispiel Komplementierung KAPA

Wir betrachten den KAPA $\mathcal{A}$ aus T4.12. Der Komplementautomat gemäß der Konstruktion ist $\mathcal{A}' = (Q'_\exists, Q'_\forall, \Sigma, \Delta, \{s\}, c')$ mit

$$Q'_\exists = \{s\} \qquad c'(s) = c'(q_3) = 1$$
$$Q'_\forall = \{q_1, q_2, q_3\} \qquad c'(q_1) = c'(q_2) = 2$$

und der folgenden graphischen Darstellung:



Wegen der geänderten Paritäten in c' akzeptiert $\mathcal{A}'$ nur, wenn s irgendwann über ein b verlassen wird und q_3 nie erreicht wird. Das heißt, es muss mindestens ein b in der Eingabe auftreten, und es gibt nur endlich viele b's (dann bleibt der Run in der q_1-q_2-Schleife) oder zwischen zwei b's stehen ungerade viele a's (dann endet der Pfad des Runs im universellen Zustand q_2, was auch zur Akzeptanz führt). Damit ist tatsächlich

$$L_\omega(\mathcal{A}') = \overline{L_\omega(\mathcal{A})}.$$

T4.14 Beispiel eines Spielverlaufs für $G_{\mathcal{A},\alpha}$

Wir betrachten wieder den KAPA $\mathcal{A}$ aus T4.12 und die Eingabe $\alpha = a(baa)^\omega$.

Die anfängliche Spielposition ist $(0, s)$.

Da $s \in Q_\forall$, ist **A** am Zug. Sie kann nur den Übergang (s, a, s) wählen, also ist die nächste Spielposition $(1, s)$.

Damit ist **A** wieder am Zug. Sie kann jetzt (s, b, s) oder (s, b, q_1) wählen. Wenn sie immer bei (s, b, s) bzw. (s, a, s) bleibt, wird eine unendliche Folge von s erzeugt, die **E** gewinnen lässt. Also betrachten wir den Fall, dass sie (s, b, q_1) wählt. Die Spielposition ist nun $(2, q_1)$.

Da $q_1 \in Q_\exists$, ist nun **E** am Zug. Sie kann nur (q_1, a, q_2) wählen; die Spielposition ist nun $(3, q_2)$.

Da s nicht mehr erreichbar ist und der q_i-Teil von $\mathcal{A}$ deterministisch ist, bleibt immer **E** am Zug, und die folgenden Spielpositionen sind $(4, q_1)$, $(5, q_3)$; danach bleibt das Spiel in q_3, und **E** gewinnt.

Wir beobachten: Je nachdem, ob und wann (bei welchem b in α) **A** den Übergang (s, b, q_1) wählt, macht der Spielverlauf einen Pfad im (einzigen) möglichen Run von $\mathcal{A}$ auf α sichtbar.

T4.15 Beispiele für Gewinnstrategien

Wir betrachten wieder den KAPA $\mathcal{A}$ aus T4.12.

- Wenn die Eingabe $\alpha = a(baa)^\omega$ ist, hat **E** eine Gewinnstrategie in $G_{\mathcal{A},\alpha}$:

  Wann immer **E** am Zug ist (Zustände q_i), wählt sie den durch das gelesene Zeichen eindeutig bestimmten Nachfolgezustand q_i. Die Form von α stellt sicher, dass nach einer geraden Anzahl von a's ein b folgt und damit von q_2 in q_3 übergegangen werden kann. Mit dieser (trivialen) Gewinnstrategie gewinnt **E** immer.

- Wenn die Eingabe $\alpha = aba^\omega$ ist, hat **A** eine Gewinnstrategie in $G_{\mathcal{A},\alpha}$:

  Beim ersten a in α wähle die einzig mögliche Transition (s, a, s). Beim folgenden b wähle (s, b, q_1). Dann ist **E** immer am Zug und wechselt nur noch zwischen q_1 und q_2, erreicht also nie q_3 und verliert damit. Dies entspricht genau dem rechten Pfad im entsprechenden Run in T4.12.

- Wenn die Eingabe $\alpha = abaaa(ba)^\omega$ ist, hat **A** eine Gewinnstrategie in $G_{\mathcal{A},\alpha}$:

  Wähle wieder beim ersten a in α den Übergang (s, a, s). und beim folgenden b dann (s, b, q_1). Ab jetzt ist wieder **E** immer am Zug und muss wegen der folgenden drei a's in q_2, q_1, q_2 wechseln. Von q_2 aus gibt es aber keinen Übergang mit b mehr, also kann **E** nicht mehr spielen und verliert damit. Dies entspricht wieder genau dem rechten Pfad im entsprechenden Teilrun in T4.12.

T4.16 Beweis von Lemma 4.17

Lemma 4.17. Seien $\mathcal{A} = (Q_\exists, Q_\forall, \Sigma, \Delta, \{q_I\}, c)$ ein KAPA und α ein ω-Wort. Dann gilt:

$$\alpha \in L_\omega(\mathcal{A}) \;\Leftrightarrow\; \textsf{\textbf{E}} \text{ hat Gewinnstrategie in } G_{\mathcal{A},\alpha} \text{ ab Position } (0, q_I)$$

Beweis.

„⇒“ Sei $\alpha \in L_\omega(\mathcal{A})$ und r ein erfolgreicher Run von $\mathcal{A}$ auf α. Wir beschreiben, wie man aus r eine Gewinnstrategie von **E** konstruieren kann. Dazu speichert sich **E** zu jedem Zeitpunkt i eine Position p_i in r. Für $i = 0$ ist $p_i = \varepsilon$ (die Wurzel des Runs). **E** geht nun so vor:

- In Spielposition (i, q) mit $q \in Q_\forall$ (d. h. **A** ist am Zug) und gespeicherter r-Position p_i macht **E** folgendes:

  Wenn **A** den Übergang (q, α_i, q') spielt, dann wähle eine Nachfolgerposition p' von p_i in r mit $r(p') = q'$ (die gibt es nach Def. Run) und setze $p_{i+1} = p'$.

- In Spielposition (i, q) mit $q \in Q_\exists$ (d. h. **E** ist am Zug) und gespeicherter r-Position p_i macht **E** folgendes:

  Wähle einen Übergang (q, α, q') und ein Kind p' von p mit $r(p') = q'$ (beides existiert nach Def. Run). Setze $p_{i+1} = p'$.

Die Folge der p_i bestimmt also einen Pfad π, der im Laufe des Spiels in r sichtbar gemacht wird.

Wenn π endlich ist, dann nur, weil irgendwann **A** nicht ziehen konnte, also gewinnt **E**. Wenn π unendlich ist, dann ist die Akzeptanzbedingung c erfüllt, weil r erfolgreich ist. Also gewinnt auch dann **E**.

Das Vorgehen beschreibt also eine Gewinnstrategie **E** in $G_{\mathcal{A},\alpha}$ ab Position $(0, q_I)$.

„$\Leftarrow$" Habe **E** eine Gewinnstrategie in $G_{\mathcal{A},\alpha}$ ab Position $(0, q_I)$. Wir konstruieren daraus einen erfolgreichen Run (P, r) von $\mathcal{A}$ auf α induktiv über die Ebenen des Baums:

- Anfangs setzen wir $P = \{\varepsilon\}$ und $r(\varepsilon) = q_I$.
- Wenn $p \in P$ auf Ebene i liegt und noch keine Nachfolger hat, dann:
  - Wenn $r(p) \in Q_\forall$, dann füge für jede Transition $(r(p), \alpha_i, q')$ einen Nachfolger p' von p zu P hinzu und setze $r(p') = q'$.
  - Wenn $r(p) \in Q_\exists$, dann betrachte diejenige Transition $(r(p), \alpha_i, q')$ die durch die Gewinnstrategie gegeben ist, füge einen Nachfolger p' von p zu P hinzu und setze $r(p') = q'$.

Die Konstruktion stellt sicher, dass r ein Run von $\mathcal{A}$ auf α ist.

Außerdem ist r erfolgreich, denn sonst gäbe es einen unendlichen Pfad in r, der die Akzeptanzbedingung c verletzt. Dann könnte aber **A** so spielen, dass dieser Pfad aufgedeckt wird, und **E** würde verlieren, ein Widerspruch zur Annahme.

Wir haben also den gewünschten Run gefunden, und somit gilt $\alpha \in L_\omega(\mathcal{A})$. ❑

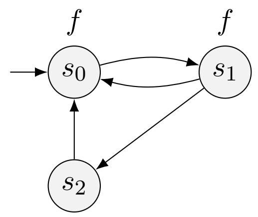
Teil V.

Endliche Automaten auf unendlichen Bäumen

T5.1 LTL-Formeln „zu stark/schwach“

Sei $\varphi_1 := \mathsf{G}(f \to \mathsf{F}\neg f)$ und $\varphi_2 := \mathsf{GF}\neg f$.

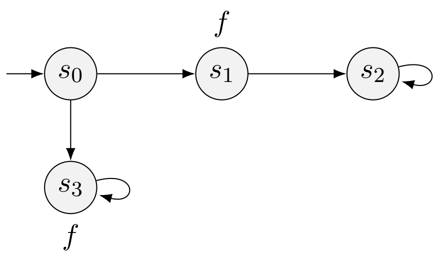
φ_1, φ_2 „zu stark für universelles Model Checking“. Wir betrachten folgende Kripke-Struktur $\mathcal{S}$:



Intuitiv gesprochen, erfüllt $\mathcal{S}$ die Eigenschaft, die mit φ_1 bzw. φ_2 ausgedrückt werden soll: es ist *möglich*, nach Besuchen von s_0 bzw. s_1 den Pfad so fortzusetzen, dass nach endlich vielen Schritten s_2 besucht wird.

Nach der Semantik von LTL wird jedoch ein Pfad „festgehalten“. Wenn wir also z. B. den Pfad $\pi = (s_0 s_1)^\omega$ betrachten, dann gilt $\pi, 0 \not\models \varphi_i$, $i = 1, 2$ (aber für den Pfad $\pi' = (s_0 s_1 s_2)^\omega$ gilt $\pi', 0 \models \varphi_i$, $i = 1, 2$).

φ_1, φ_2 „zu schwach für existenzielles Model Checking“. Wir betrachten folgende Kripke-Struktur $\mathcal{S}'$:

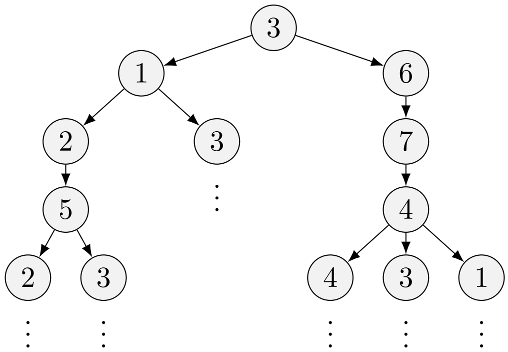


Intuitiv gesprochen, erfüllt $\mathcal{S}'$ die gewünschte Eigenschaft *nicht*: der Pfad $s_0 s_3$ kann nicht mehr wie gewünscht fortgesetzt werden.

Nach der Semantik von LTL genügt es jedoch, einen Pfad π zu finden, für den $\pi, 0 \models \varphi_i$ gilt, und das ist z. B. $\pi = s_1 s_1 s_2^\omega$.

T5.2 Beispiel-Berechnungsbaum

Wenn man die Beispielstruktur Mikrowelle im Zustand 3 auffaltet, erhält man:

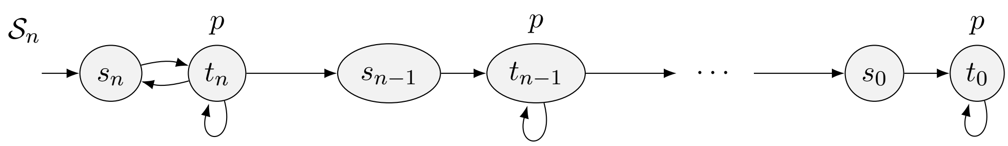


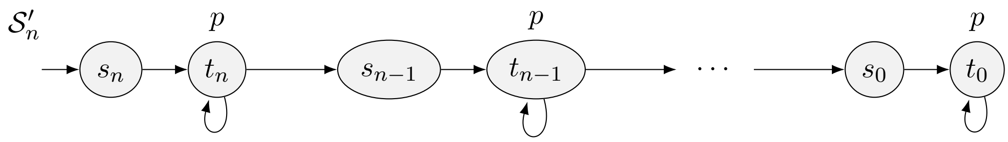
Die gegebene Struktur $\mathcal{S}$ ist eine endliche Repräsentation dieses unendlichen Baums (und vieler weiterer Bäume).

T5.x Beweis Teil 2 des Ausdrucksstärke-Lemmas (fakultativ)

Lemma 5.5 (2). Es gibt keine zu $\mathsf{FG}p$ äquivalente CTL-Zustandsformel.

Beweis. Betrachte zwei *Folgen* $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \ldots$ und $\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_2, \ldots$ von Kripke-Strukturen, die wie folgt aufgebaut sind. Für $n \geq 0$ ist





Insbesondere unterscheidet sich $\mathcal{S}'_n$ von $\mathcal{S}_n$ nur durch die fehlende Kante von t_n nach s_n. Nun gilt für alle $n \geq 0$:

(i) $\mathcal{S}_n \not\models \mathsf{FG}p$ (wegen Pfad $(s_n t_n)^\omega$ ab $s_n \in \S_0$)
(ii) $\mathcal{S}'_n \models \mathsf{FG}p$ (weil jeder Pfad auf ein $(t_i)^\omega$ enden muss, für ein $i \leq n$)

Außerdem zeigt man leicht per Induktion über n die folgende Aussage: Für alle $n \geq 0$ und alle CTL-Zustandsformeln ζ der Länge $\leq n$ gilt:

(iii) $\mathcal{S}_n \models \zeta$ gdw. $\mathcal{S}'_n \models \zeta$

Angenommen, es gebe eine CTL-Zustandsformel ζ mit $\zeta \equiv \mathsf{FG}p$. Sei $n := |\zeta|$. Dann gilt wegen (i) und (ii): $\mathcal{S}_n \not\models \zeta$ und $\mathcal{S}'_n \models \zeta$. Das widerspricht aber (iii). ❏

T5.3 Beispiel für den Model-Checking-Algorithmus

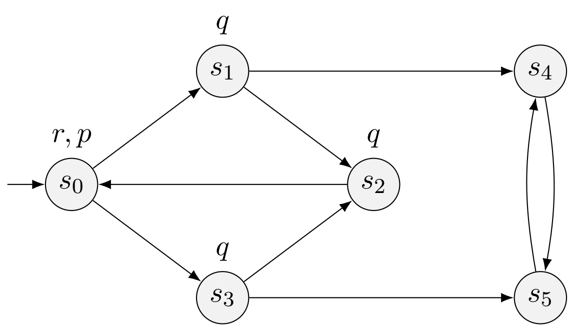
Wir betrachten die Zustandsformel $p \wedge \mathsf{AXE}(q \,\mathsf{U}\, r)$ und wandeln sie in ENF, wobei wir alle Zustands-Teilformeln in aufsteigender Größe benennen:

$$\zeta = \underbrace{p}_{\zeta_1} \wedge \underbrace{\neg\, \underbrace{\mathsf{EX} \underbrace{\neg\, \underbrace{\mathsf{E}(\underbrace{q}_{\zeta_2} \,\mathsf{U}\, \underbrace{r}_{\zeta_3})}_{\zeta_4}}_{\zeta_5}}_{\zeta_6}}_{\zeta_7}$$

Das heißt:

$$\begin{aligned}
\zeta_1 &= p \\
\zeta_2 &= q \\
\zeta_3 &= r \\
\zeta_4 &= \mathsf{E}(\zeta_2 \,\mathsf{U}\, \zeta_3) \\
\zeta_5 &= \neg\zeta_4 \\
\zeta_6 &= \mathsf{EX}\zeta_5 \\
\zeta_7 &= \neg\zeta_6 \\
\zeta &= \zeta_1 \wedge \zeta_7
\end{aligned}$$

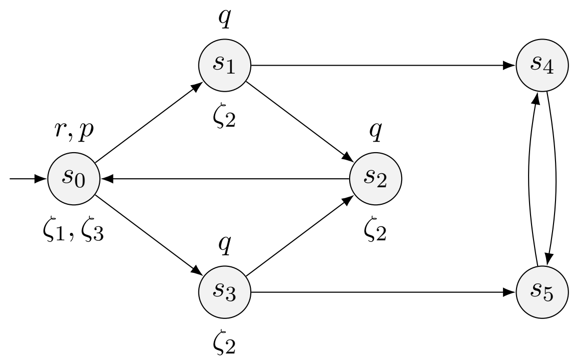
Die Formel ζ soll nun auf folgender Kripke-Struktur $\mathcal{S}$ ausgewertet werden.



Dazu berechnen wir gemäß Algorithmus `CTL-MC` die Mengen $\mathrm{Sat}(\zeta_i)$ für $i = 1, \ldots, 7$ sowie $\mathrm{Sat}(\zeta)$:

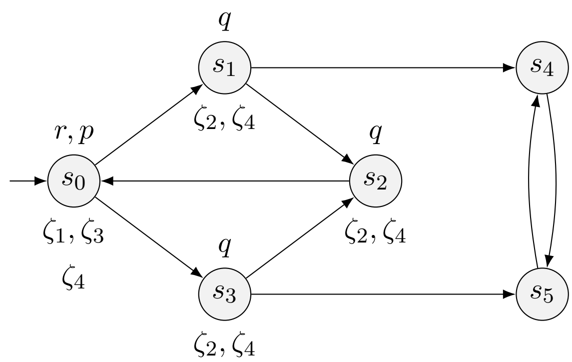
- $\mathrm{Sat}(\zeta_1) = \{s_0\}$
- $\mathrm{Sat}(\zeta_2) = \{s_1, s_2, s_3\}$
- $\mathrm{Sat}(\zeta_3) = \{s_0\}$

  Das heißt, dass in den ersten drei Schritten genau die Zustände mit den ζ_i „markiert" werden, die in der Eingabestruktur $\mathcal{S}$ mit der jeweiligen Aussagenvariable markiert sind:

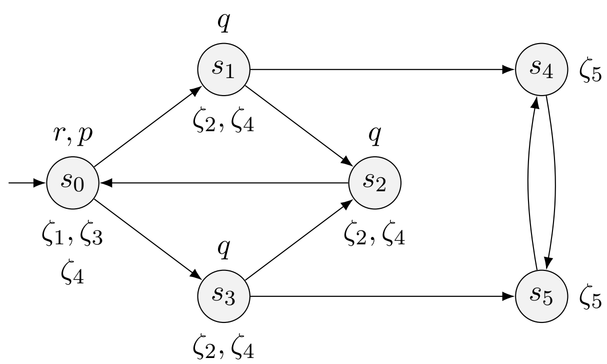


- $\mathrm{Sat}(\zeta_4)$ wird nun mittels des Subalgorithms `SatEU` aus $\mathrm{Sat}(\zeta_2)$ und $\mathrm{Sat}(\zeta_3)$ berechnet. Dazu wird S' mit $\mathrm{Sat}(\zeta_3) = \{s_0\}$ initialisiert; dann werden sukzessive s_2, s_1, s_3 hinzugefügt. Wir erhalten $\mathrm{Sat}(\zeta_4) = \{s_0, s_1, s_2, s_3\}$.

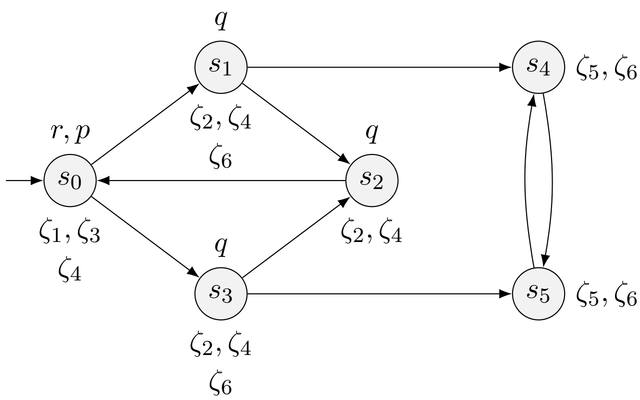
  Das heißt, in $\mathcal{S}$ werden diese vier Zustände mit ζ_4 „markiert“:



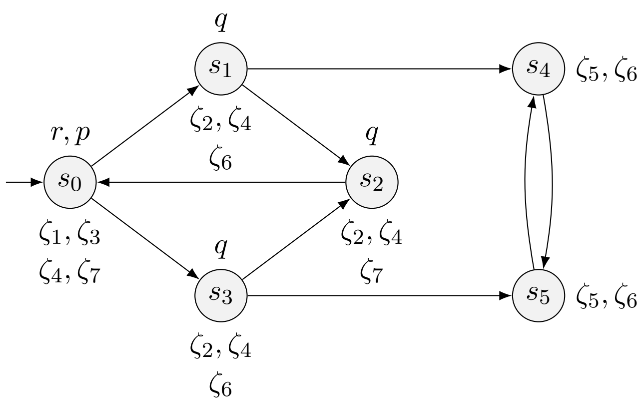
- $\mathrm{Sat}(\zeta_5) = S \setminus \mathrm{Sat}(\zeta_4) = \{s_4, s_5\}$:



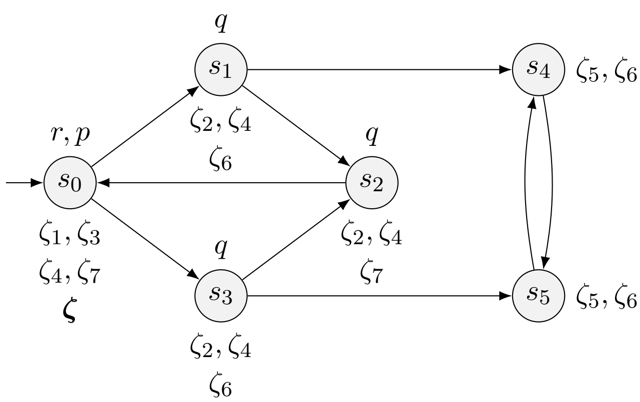
- $\mathrm{Sat}(\zeta_6)$ wird nun mittels des Subalgorithms `SatEX` aus $\mathrm{Sat}(\zeta_5)$ berechnet, d. h. $\mathrm{Sat}(\zeta_6) = \{s \in S \mid$ es gibt $s' \in \mathrm{Sat}(\zeta_5)$ mit $(s, s') \in R$. Damit ist $\zeta_6 = \{s_1, s_3, s_4, s_5\}$:



- $\mathrm{Sat}(\zeta_7) = S \setminus \mathrm{Sat}(\zeta_6) = \{s_0, s_2\}$:



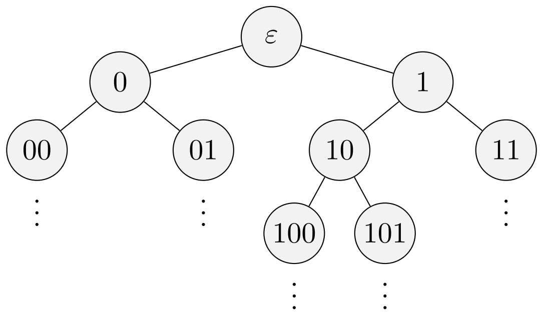
- $\mathrm{Sat}(\zeta) = \mathrm{Sat}(\zeta_1) \cap \mathrm{Sat}(\zeta_7) = \{s_0\}$:



Da $S_0 = \{s_0\}$ und $s_0 \in \mathrm{Sat}(\zeta)$, gibt der Algorithmus 1 aus, d. h. $\mathcal{S} \models \zeta$.

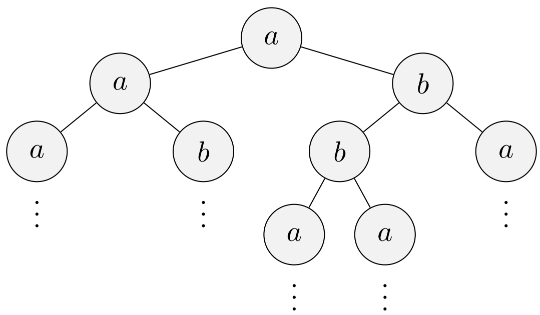
T5.4 Veranschaulichung der Grundbegriffe

Die **Positionen** im unendlichen vollständigen Binärbaum sehen so aus:



- Knoten 1 hat als **linkes Kind** 10 und als **rechtes Kind** 11.
- Knoten 1 hat **Nachfolger** $10, 11, 100, 101, 110, 111, 1000, \dots$.
- Knoten 11 hat **Tiefe** 2.
- Auf **Ebene 2** sind Knoten $00, 01, 10, 11$.
- Ein **Pfad** ist z. B. $\{\varepsilon, 1, 10, 101, \dots\}$.

Wenn $\Sigma = \{a, b\}$, dann ist ein **Σ-Baum** z. B. folgender:



Das heißt also $t(\varepsilon) = a$, $t(0) = a$, $t(1) = b$ usw.

T5.5 Details des „Pumpens“ in „Büchi- vs. Muller-Erkennbarkeit“

Wie betrachten einen erfolgreichen Run r auf t (Existenz dieses Runs: siehe Ende von Folie 38). Auf dem Pfad 1^ω wird ein $f \in F$ unendlich oft besucht. Insbesondere gibt es also ein $m_0 > 0$ mit $r(1^{m_0}) \in F$. Auf dem Pfad $1^{m_0}01^\omega$ wird ebenfalls ein $f \in F$ unendlich oft besucht; also gibt es ein $m_1 > 0$ mit $r(1^{m_0}01^{m_1}) \in F$. Diese Argumentation kann man jetzt noch weitere $(n-1)$ Mal iterieren, und man erhält:

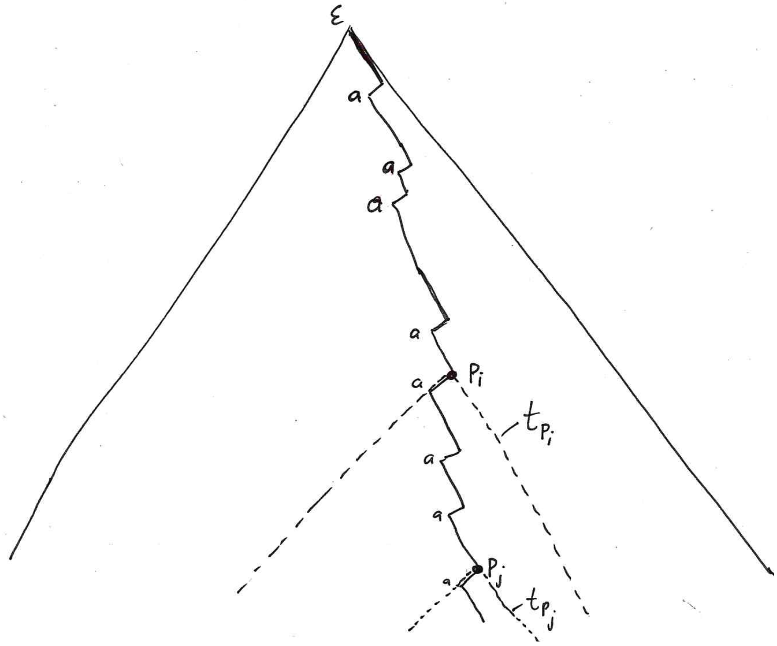
> Es gibt $m_0, m_1, \dots, m_n > 0$, so dass
> $r(1^{m_0})$, $r(1^{m_0}01^{m_1})$, $\dots$, $r(1^{m_0}01^{m_1}0\cdots01^{m_n}) \in F$.

Da $|F| = n$, gibt es laut Schubfachprinzip Indizes i, j mit $0 \leq i < j \leq n$ und

$$r(1^{m_0}0 \cdots 01^{m_i}) = r(1^{m_0}0 \cdots 01^{m_i}0 \cdots 01^{m_j}) \in F.$$

Wir nennen die beiden Positionen p_i, p_j, d. h. $r(p_i) = r(p_j) \in F$.

Skizze:



Nun gibt es in t auf dem Pfad von der Wurzel zur Position p_i genau $i < n$ „Linksschritte", also nach Definition von t genau i Positionen p mit $t(p) = a$. Da $i < j$, gibt es auf dem Pfad von p_i zu p_j mindestens einen weiteren „Linksschritt", also mindestens eine Position p mit $t(p) = a$. Da $r(p_i) = r(p_j) \in F$, können wir in t (und r) den Teilbaum t_{p_j} (r_{p_j}) durch t_{p_i} (r_{p_i}) ersetzen[2] und erhalten wieder einen erfolgreichen Run:

$$r[p_j \to r_{p_i}] \text{ ist ein erfolgreicher Run auf } t[p_j \to t_{p_i}]. \qquad (*)$$

Aussage $(*)$ gilt, weil (a) wegen $r(p_i) = r(p_j)$ weiterhin die Übergangsrelation Δ respektiert wird und (b) weiterhin auf allen Pfaden ein akzepzierender Zustand unendlich oft besucht werden muss.

Dieses Ersetzen kann nun unendlich oft iteriert werden. Der auf diese Weise aus r resultierende Baum ist ein erfolgreicher Run von $\mathcal{A}$ auf dem aus t resultierenden Baum. Letzterer hat jedoch einen Pfad mit unendlich vielen a's, ist also gar nicht in L enthalten; ein Widerspruch. ❏

[2] Wie in Kapitel 2 bezeichnet t_p den Teilbaum des Baums t, dessen Wurzel p ist.

T5.6 Beispiel für die Konstruktion „NMBA ⇒ NPBA“

Seien $Q = \{1, 2, 3\}$ und $F = \{1, 2\}$. Wir betrachten den Run $r = 12131211\dots$ von $\mathcal{A}$ auf einem Pfad π, bei dem ab der 5. Position nur noch Zustände aus F vorkommen. Ein zugehöriger Run r' von $\mathcal{A}'$ auf demselben Pfad ist:

$$\langle 231, 1\rangle \ \langle 312, 1\rangle \ \langle 321, 2\rangle \ \langle 213, 1\rangle \ \langle 231, 2\rangle \ \langle 312, 1\rangle \ \langle 321, 2\rangle \ \langle 312, 3\rangle$$

An der ersten Position sind alle Paare $\langle q_1q_2q_3, \ell\rangle$ möglich, in denen $q_3 = q_0$ ist, denn es gibt noch keine „Vergangenheit“, in der Zustände aufgetreten sein können. Das Paar $\langle 312, 1\rangle$ an der zweiten Position gibt durch seine zweite Komponente $\ell = 1$ an, dass Zustand 2 (Ende der ersten Komponente 312) in der vorangehenden Permutation (231) an 1. Stelle aufgetreten ist. Ab Position 6 haben die Paare $\langle q_1q_2q_3, \ell\rangle$ an den Positionen q_2, q_3 immer die akzeptierenden Zustände $1, 2$, und ab Position 7 ist $\ell > |Q| - |F| = 1$.

T5.7 Beweis der Hilfsaussage für „NMBA ⇒ NPBA“

„⇒“ Wegen $\mathsf{Inf}(q_0q_1q_2\cdots) = S$ gibt es Zeitpunkte

(i) m_1, ab dem nur noch Zustände aus S vorkommen, d. h. $q_i \in S$ für alle $i \geq m_1$;

(ii) $m_2 > m_1$ so, dass während der Zeitpunkte $m_1, m_1 + 1, \dots, m_2$ *jeder* Zustand aus S mindestens einmal vorkommt.

Daraus folgt

(i′) Für alle $i \geq m_1$ werden nur noch Zustände aus S in die Endposition von perm_i gerückt

(ii′) Bis m_2 wurde *jeder* Zustand aus S mindestens einmal in die Endposition von perm_i gerückt.

Wegen (ii′) müssen die letzten k Positionen von perm_{m_2} genau die k Zustände aus S enthalten (und die ersten $n - k$ Positionen die Zustände aus $Q \setminus S$). In zukünftigen Zuständen s_i mit $i > m_2$ wird nie ein Zustand aus $Q \setminus S$ in die Endposition von perm_i gerückt; also ist $\ell_i > n - k$ (was Teil (1) der Hilfsaussage beweist), und die letzten k Positionen in perm_i sind aus S (was Teil (2) (b) beweist).

Um Teil (2) (a) zu zeigen, nehmen wir an, es sei $\ell_i = n - k + 1$ für nur endlich viele i. Dann müsste es aber einen Zustand aus S geben, der nur endlich oft besucht wird und deshalb dauerhaft in Position $n - k + 1$ von perm_i landet. Dies ist aber ein Widerspruch zu $\mathsf{Inf}(q_0q_1q_2\cdots) = S$.

„⇐“ Kann mit einer ähnlichen Argumentation, die die Konstruktion analysiert, bewiesen werden. ❏

T5.8 Beweis der Korrektheit, „NMBA $\Rightarrow$ NPBA“

Es bleibt zu zeigen: $L_\omega(\mathcal{A}) = L_\omega(\mathcal{A}')$.

„$\subseteq$“ Sei $t \in L_\omega(\mathcal{A})$. Dann gibt es einen erfolgreichen Run r von $\mathcal{A}$ auf t, d. h. $\mathsf{Inf}(r, \pi) = F$ für alle Pfade π von t.

Sei $s_0 = \langle t_1 \cdots t_{n-1} q_0, 1\rangle \in I'$, und sei s der gemäß Δ' eindeutig bestimmte Run von $\mathcal{A}'$ auf t, der folgende Eigenschaften erfüllt.

(i) $s(\varepsilon) = s_0$

(ii) Für alle $p \in \{0,1\}^*$ hat $s(p)$ die Form $\langle \mathsf{perm}_p, \ell_p\rangle$, wobei perm_p auf $r(p)$ endet.

Wir müssen noch zeigen, dass s erfolgreich ist. Dazu betrachten wir einen beliebigen Pfad π und die zugehörige Zustandsfolge $s_0 s_1 s_2 \cdots$ von s mit $s_i = \langle \mathsf{perm}_i, \ell_i\rangle$. Da r erfolgreich ist, gilt $\mathsf{Inf}(r, \pi) = F$. Wegen der Hilfsaussage folgt daraus:

(1) Für endlich viele i ist $\ell_i \leq n - |F|$.

(2) Für unendlich viele i gilt:

  (a) $\ell_i = n - |F| + 1$

  (b) Die Menge der Zustände an den Positionen $n - |F| + 1, \ldots, n$ in perm_i ist F.

Wegen (2) muss $\mathsf{Inf}(s, \pi)$ einen Zustand der Form $\langle q_1 \cdots q_n, n - |F| + 1\rangle$ mit $\{q_{n-|F|+1}, \ldots, q_n\} = F$ enthalten, aber wegen (1) *keinen* Zustand der Form $\langle q_1 \cdots q_n, \ell\rangle$ für $\ell \leq n - |F|$. Damit erfüllt der Pfad π in s die Akzeptanzbedingung von $\mathcal{A}'$; also ist $t \in L_\omega(\mathcal{A}')$.

„$\supseteq$“ Sei $t \in L_\omega(\mathcal{A}')$. Dann gibt es einen erfolgreichen Run s von $\mathcal{A}'$ auf t. Sei $s(p) = \langle \mathsf{perm}_p, \ell_p\rangle$. Wir konstruieren einen Run r von $\mathcal{A}$ auf t, indem wir $r(p)$ auf das letzte Element von perm_p setzen, für alle $p \in \{0,1\}^*$. Nach Definition von I' und Δ' ist dann nämlich

- $r(\varepsilon) \in I'$ und
- $\big(r(p), t(p), r(p0), r(p1)\big) \in \Delta$ für alle $p \in \{0,1\}^*$,

also ist r tatsächlich ein Run von $\mathcal{A}$ auf t. Es bleibt zu zeigen, dass r erfolgreich ist. Dazu betrachten wir einen beliebigen Pfad π. Da s erfolgreich ist und wegen der Akzeptanzbedingung von $\mathcal{A}'$ gibt es einen Zustand $\langle q_1 \cdots q_n, \ell\rangle$ mit $\{q_\ell, \ldots, q_n\} = F$, so dass

(i) $\langle q_1 \cdots q_n, \ell\rangle$ unendlich oft in s auftritt,   aber

(ii) *kein* $\langle q'_1 \cdots q'_n, \ell'\rangle$ mit $\{q'_1, \ldots, q'_n\} \neq F$ und $\ell' < \ell$ unendlich oft auftritt.

Die Eigenschaften (i) und (ii) entsprechen aber genau den Teilen (2) bzw. (1) aus der Hilfsaussage, also gilt $\mathsf{Inf}(r, \pi) = \{q_\ell, \ldots, q_n\} = F$. Damit ist $t \in L_\omega(\mathcal{A})$. ❑

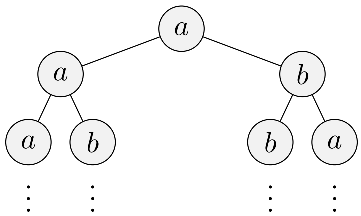
T5.9 Beispiel eines Spielverlaufs für $G_{\mathcal{A},t}$

Sei $\mathcal{A} = (\{q_1, q_2\},\ \{a, b\},\ \Delta,\ \{q_1\},\ c)$ ein NPBA mit

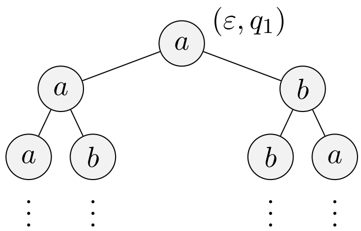
$$\Delta = \{(q_1, a, q_1, q_2),\ (q_2, a, q_2, q_1),\ (q_1, b, q_2, q_1),\ (q_2, b, q_1, q_2)\} \quad \text{und}$$
$$c\ :\quad q_1 \mapsto 0,\ q_2 \mapsto 1.$$

Die Übergangsrelation von $\mathcal{A}$ sorgt also dafür, dass beim Lesen eines a der aktuelle Zustand ans linke Kind und der andere ans rechte Kind weitergegeben wird und umgekehrt beim Lesen eines b. Die Akzeptanzbedingung fordert, dass q_1 unendlich oft vorkommen muss, und entspricht der Büchi-Akzeptanzkomponente $F = \{q_1\}$.

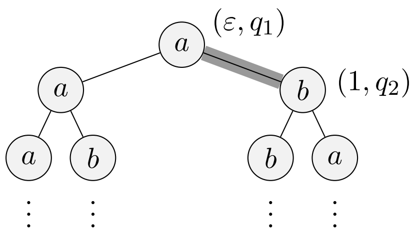
Wir betrachten folgenden Eingabebaum t.



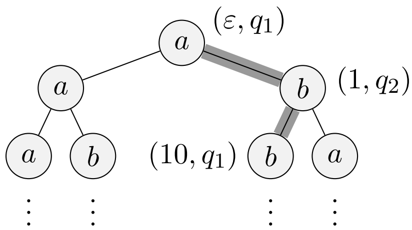
In der anfänglichen Spielposition ist nur die Wurzel mit q_1 markiert, also ist der bisherige Spielverlauf die Folge, die nur aus dem Paar (ε, q_1) besteht. Zur Veranschaulichung wird der Spielverlauf im Folgenden an den von **PF** gewählten Pfad im Baum geschrieben, also zu Beginn so:



In der ersten Runde wählt **Aut** die zum Zustand q_1 und Zeichen a passende Transition (q_1, a, q_1, q_2), und **PF** antwortet darauf mit der Wahl des rechten Kindknotens. Dadurch ist nun das rechte Kind der Wurzel mit q_2 markiert:



In der zweiten Runde wählt **Aut** die zum Zustand q_2 und Zeichen b passende Transition (q_2, b, q_1, q_2), und **PF** antwortet darauf mit der Wahl des linken Kindknotens. Dadurch ist nun das linke Kind der zuletzt betrachteten Position mit q_1 markiert:



usw.

T5.10 Beispiel für Gewinnstrategie

Wir betrachten den NPBA $\mathcal{A}$ aus dem letzten Beispiel. Eine Gewinnstrategie für **PF** ab Spielposition v ist folgende:

> In Spielposition v', die durch die Zugfolge $v \cdots v'$ bestimmt ist, wobei $v' = (q_i, t(p), q_j, q_k)$ mit $(i, j, k \in \{1, 2\})$,
> wähle das linke Kind, wenn $j = 2$ und das rechte Kind, wenn $k = 2$
> (wegen Δ muss immer einer dieser beiden Fälle eintreten).

Diese Strategie stellt eine Funktion f dar, die jeder Zugfolge $v \cdots v'$ einen eindeutig bestimmten Zug von **PF** (linkes/rechtes Kind) zuweist. Sie ist eine Gewinnstrategie, denn sie stellt sicher, dass unabhängig davon, wie **Aut** spielt, auf dem gesamten gespielten Pfad nur q_2 auftritt; also ist die Akzeptanzbedingung von $\mathcal{A}$ verletzt.

In dieser Gewinnstrategie für den sehr einfachen Automaten $\mathcal{A}$ hängt $f(v \cdots v')$ nur von v' ab und nicht von den vorhergehenden Spielpositionen in $v \cdots v'$. Die Strategie ist also *gedächtnislos*, was intuitiv so viel bedeutet wie: „der bisherige Spielverlauf kann ignoriert/vergessen werden“. Satz 5.18 und Folgerung 5.19 zeigen, dass dies nicht an der Einfachheit von $\mathcal{A}$ liegt.

T5.11 Beweis von Lemma 5.17

Lemma 5.17. Seien $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ-Baum. Dann gilt: $t \in L_\omega(\mathcal{A})$ **gdw.** **Aut** hat Gewinnstrategie in $G_{\mathcal{A},t}$ ab Position (ε, q_I)

Beweis. Wir wandeln erfolgreiche Runs direkt in Gewinnstrategien und umgekehrt.

„⇒" Gelte $t \in L_\omega(\mathcal{A})$ und sei r erfolgreicher Run von $\mathcal{A}$ auf t. Die Gewinnstrategie für **Aut** lässt sich leicht wie folgt aus r konstruieren.

- In Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$.
- In allen anderen Spielpositionen (p, q) wähle $(q, t(p), r(p0), r(p1))$.

Wenn **Aut** diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r. Da r erfolgreich ist, gewinnt **Aut** nach Def. von $G_{\mathcal{A},t}$.

„⇐" Habe **Aut** eine Gewinnstrategie in $G_{\mathcal{A},t}$ ab Position (ε, q_I). Wir konstruieren einen erfolgreichen Run r von $\mathcal{A}$ auf t induktiv über die Ebenen von t:

- $r(\varepsilon) = q_I$
- Wenn $r(p)$ definiert ist, dann betrachte die Transition $\big(r(p), t(p), q_0, q_1\big)$, die durch die Gewinnstrategie der Spielposition $\big(p, r(p)\big)$ zugewiesen wird. Setze $r(p0) = q_0$ und $r(p1) = q_1$.

Nach Definition der Spielzüge von **Aut** ist r ein Run von $\mathcal{A}$ auf t. Außerdem ist r erfolgreich, denn sonst gäbe es einen Pfad in t, auf dem die Akzeptanzbedingung von $\mathcal{A}$ nicht erfüllt wäre. Dann könnte aber **Aut** verlieren, indem **PF** diesen Pfad wählt – Widerspruch zur Annahme, dass **Aut** eine Gewinnstrategie hat. ❑

T5.12 Beispiel für die Sprache $L_{s,t}$

Ab jetzt bezeichnen wir Pfade im Baum als Wörter $\pi \in \{0,1\}^*$; beispielsweise steht $\pi = 0110\cdots$ für den Pfad $\{\varepsilon, 0, 01, 011, \dots\}$.

Sei t ein beliebiger Baum, s ein Strategiebaum (der nach Definition für jede Position $p \in \{0,1\}^*$ eine Funktion $f_p : \Delta \to \{0,1\}$ enthält) und $\pi = 0110\cdots$. Dann wird durch π folgendes ω-Wort $\alpha \in L_{s,t}$ bestimmt:

$$\alpha = \boxed{\begin{matrix} f_\varepsilon \\ t(\varepsilon) \\ 0 \end{matrix}} \; \boxed{\begin{matrix} f_0 \\ t(0) \\ 1 \end{matrix}} \; \boxed{\begin{matrix} f_{01} \\ t(01) \\ 1 \end{matrix}} \; \boxed{\begin{matrix} f_{011} \\ t(011) \\ 0 \end{matrix}} \; \boxed{\begin{matrix} f_{0110} \\ t(0110) \\ 0 \end{matrix}} \; \cdots$$

Dabei bezeichnet jede Box ein Symbol von α, also gemäß Folie 5.55 ein Tripel, und jede erste Komponente f_p eines Tripels steht wiederum für dasjenige $|\Delta|$-Tupel von Nullen und Einsen, welches f_p repräsentiert. Das erste Zeichen von α ist also $\langle f_\varepsilon, t(\varepsilon), 0 \rangle$.

Mit anderen Worten: die Folge $f_\varepsilon, f_0, f_{01}, \dots$ aller ersten Komponenten der Zeichen von α gibt den Inhalt von s auf dem Pfad π wieder und die Folge aller zweiten Komponenten den Inhalt von t auf π. Die Folge der dritten Komponenten ist π selbst.

T5.13 Beweis von Lemma 5.22

Lemma 5.22. s ist ein **PF**-Gewinnbaum für t gdw. $L_{s,t} \cap L_\omega(\mathcal{A}') = \emptyset$

Beweis.

„$\Rightarrow$" Wir beweisen die Kontraposition. Gelte $L_{s,t} \cap L_\omega(\mathcal{A}') \neq \emptyset$. Dann gibt es einen Pfad π, so dass das Σ'-Wort

$$\alpha = \langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \ \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \ \langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3 \rangle \ \cdots$$

von $\mathcal{A}'$ akzeptiert wird. Sei $r = q_0q_1q_2\cdots$ ein erfolgreicher Run von $\mathcal{A}'$ auf α. Dann gibt es für jede Position $j \geq 0$ in α eine Transition

$$\Big(q_j,\ \langle s(\pi_1\cdots\pi_j), t(\pi_1\cdots\pi_j), \pi_{j+1}\rangle,\ q_{j+1}\Big) \ \in \ \Delta'.$$

Nach Konstruktion von Δ' gibt es dann in $\mathcal{A}$ eine entsprechende Transition

$$\delta_j = \Big(q_j,\ t(\pi_1\cdots\pi_j),\ q_0',\ q_1'\Big) \ \in \ \Delta$$

mit $s(\pi_1\cdots\pi_j)(\delta_j) = \pi_{i+1}$, wobei $s(\pi_1\cdots\pi_j)$ die Funktion $f_{\pi_1\cdots\pi_j} \in F$ ist.
Wir betrachten nun denjenigen Spielverlauf von $G_{\mathcal{A},t}$, in dem **Aut** in Spielposition $\pi_1\cdots\pi_j$ jeweils δ_j wählt und daraufhin **PF** mit $s(\pi_1\cdots\pi_j) = \pi_{j+1}$ antwortet. Die während dieses Spiels erzeugte Zustandsfolge ist genau der Run r, also gewinnt **Aut** (denn $\mathcal{A}'$ hat nach Konstruktion dieselbe Akzeptanzkomponente c wie $\mathcal{A}$). In diesem Spielverlauf verliert also **PF**, obwohl sie nach Strategie s gespielt hat. Folglich ist s kein **PF**-Gewinnbaum für t.

„$\Leftarrow$" Gelte $L_{s,t} \cap L_\omega(\mathcal{A}') = \emptyset$. Wir betrachten einen beliebigen Spielverlauf von $G_{\mathcal{A},t}$ ab (ε, q_I), bei dem **PF** nach Strategie s spielt, d. h. für alle $j \geq 0$

- wählt **Aut** in Spielposition $(\pi_1\cdots\pi_j,\ q_j)$ eine beliebige Transition

$$\delta_j = \Big(q_j,\ t(\pi_1\cdots\pi_j),\ q_{j0}, q_{j1}\Big) \ \in \ \Delta \qquad \text{und}$$

- **PF** antwortet mit q_{j+1} gemäß s:

$$q_{j+1} = \begin{cases} q_{j0} & \text{falls } s(\pi_1\cdots\pi_j)(\delta_j) = 0 \\ q_{j1} & \text{sonst} \end{cases}$$

Wir betrachten nun die Folge $r' = q_0q_1q_2\cdots$ der während des Spielverlaufs besuchten Zustände. Nach Konstruktion von Δ' ist r' ein Run von $\mathcal{A}'$ auf dem zugehörigen ω-Wort

$$\alpha = \langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \ \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \ \langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3 \rangle \ \cdots \in L_{s,t}.$$

Da $L_{s,t} \cap L_\omega(\mathcal{A}') = \emptyset$, ist r' nicht erfolgreich. Der Run r' entspricht außerdem einem Pfad eines Runs des ursprünglichen $\mathcal{A}$ auf t, nämlich desjenigen Runs r, der durch die Entscheidungen von **Aut** bestimmt wird. Deshalb kann r nicht erfolgreich sein. Wir haben also für eine *beliebige* Zugfolge von **Aut** gezeigt, dass der zugehörige Run von $\mathcal{A}$ auf t nicht erfolgreich sein kann. Somit muss s eine Gewinnstrategie für **PF** kodieren, ist also ein **PF**-Gewinnbaum. ❏

T5.14 Beweis von Lemma 5.23

Lemma 5.23. $t \in L_\omega(\mathcal{B})$ gdw. es gibt F-Baum s mit $\mathsf{L}_{s,t} \subseteq L_\omega(\mathcal{A}'')$

Beweis.

„⇒“ Sei $t \in L_\omega(\mathcal{B})$ und r ein erfolgreicher Run von $\mathcal{B}$ auf t, d. h.

(i) für jede Baumposition $p \in \{0,1\}^*$ gibt es in Δ^{neu} einen entsprechenden Übergang $(r(p), t(p), r(p0), r(p1))$ und

(ii) r erfüllt auf jedem Pfad die Akzeptanzbedingung c'' von $\mathcal{A}''$ (diese hat ja $\mathcal{B}$ von $\mathcal{A}''$ übernommen).

Aus (i) und der Konstruktion von Δ^{neu} folgt, dass es für jedes p eine Funktion $f_p \in F$ gibt mit

$$\Big(r(p),\ \langle f_p, t(p), 0\rangle,\ r(p0)\Big) \in \Delta'' \qquad \text{und} \tag{iii}$$

$$\Big(r(p),\ \langle f_p, t(p), 1\rangle,\ r(p1)\Big) \in \Delta''. \tag{iv}$$

Sei nun s der durch diese f_p bestimmte F-Baum, d. h. $s(p) = f_p$.

Wir betrachten einen beliebigen Pfad $\pi = \pi_1\pi_2\pi_3\cdots$ und das zugehörige ω-Wort

$$\alpha_\pi = \big\langle s(\varepsilon), t(\varepsilon), \pi_1\big\rangle\ \big\langle s(\pi_1), t(\pi_1), \pi_2\big\rangle\ \big\langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3\big\rangle\ \cdots \tag{$*$}$$

aus $\mathsf{L}_{s,t}$. Es bleibt zu zeigen, dass $\alpha_\pi \in L_\omega(\mathcal{A}'')$ ist. Dies folgt, weil (i), (iii) und (iv) „bezeugen“, dass r ein erfolgreicher Run von $\mathcal{A}''$ auf α_π ist.

„⇐“ Seien t, s so, dass $\mathsf{L}_{s,t} \subseteq L_\omega(\mathcal{A}'')$. Um zu zeigen, dass $t \in L_\omega(\mathcal{B})$, konstruieren wir einen erfolgreichen Run aus allen Pfaden, die $\mathcal{A}''$ akzeptiert. Dabei benutzen wir, dass $\mathcal{A}''$ deterministisch ist.

Wegen $\mathsf{L}_{s,t} \subseteq L_\omega(\mathcal{A}'')$ gibt es für jeden Pfad π einen erfolgreichen Run von $\mathcal{A}''$ auf dem Wort α_π, das gemäß $(*)$ definiert ist. Da $\mathcal{A}''$ deterministisch ist, sind seine Runs auf Wörten aus $\mathsf{L}_{s,t}$ eindeutig bestimmt. Insbesondere gilt: wenn zwei Pfade π, π' dasselbe Präfix $\pi_1 \cdots \pi_k = \pi'_1 \cdots \pi'_k$ haben, dann weisen die beiden eindeutig bestimmten Runs von $\mathcal{A}''$ den ersten k Positionen der zugehörigen Wörter aus $\mathsf{L}_{s,t}$ dieselben Zustände zu. Das heißt, dass es für jede Position p einen *eindeutig bestimmten* Zustand q gibt, so dass für alle Pfade π mit $p = \pi_1 \cdots \pi_k$ der Run von $\mathcal{A}''$ „auf π“ der Position k den Zustand q zuweist. Mit dieser Erkenntnis können wir einen Run r von $\mathcal{B}$ auf t wie folgt definieren:

Für jede Position p setze $r(p) :=$ der beschriebene Zustand q.

Dadurch entspricht jeder Pfad π in r dem Run r'' von $\mathcal{A}''$ auf dem zugehörigen Wort $\alpha_\pi \in \mathsf{L}_{s,t}$. Da r'' erfolgreich ist (siehe oben) und $\mathcal{B}$ die Akzeptanzkomponente c'' von $\mathcal{A}''$ übernimmt, ist auch r erfolgreich; somit ist $t \in L_\omega(\mathcal{B})$. ❑