

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

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

Stand: 11. November 2019

# **Inhaltsverzeichnis**

<b>I. Endliche Automaten auf endlichen Wörtern</b>	<b>3</b>
<b>II. Endliche Automaten auf endlichen Bäumen</b>	<b>8</b>
<b>III. Endliche Automaten auf unendlichen Wörtern</b>	<b>24</b>
<b>IV. Endliche Automaten auf unendlichen Bäumen</b>	<b>49</b>
<b>V. Alternierung</b>	<b>63</b>
<b>Literaturverzeichnis</b>	<b>71</b>

**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  $w_1 = \text{web}$  und  $w_2 = \text{ebay}$ , und sei  $\Sigma$  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  $\Sigma$  der Schleife am Zustand 0 steht für die Folge aller Symbole aus  $\Sigma$ .

## 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  $|w_1| + \dots + |w_n|$  Zustände, wenn  $w_1, \dots, w_n$  die Stichwörter sind.

## T1.5 Beispiel für die Anwendung des Pumping-Lemmas

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

**Vorbertragung 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 einzige 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.

$$M = \{\mathcal{A} \mid \mathcal{A} \text{ ist NEA}\}$$

$$X = \{\mathcal{A} \mid \mathcal{A} \text{ ist NEA, } L(\mathcal{A}) \neq \emptyset\}$$

$$M' = \{(G, s, t) \mid G = (V, E) \text{ ist gerichteter Graph und } s, t \in V\}$$

$$X' = \{(G, s, t) \mid (G, s, t) \in M' \text{ und es gibt Pfad von } s \text{ nach } t \text{ in } G\}$$

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

Es gilt  $X \leq_p X'$ , was durch folgende Polynomialzeitreduktion  $\pi$  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:

$$G_{\mathcal{A}} = (V, E) \quad \text{mit}$$

$$V = Q \uplus \{s, t\}$$

$$E = \{(q, q') \mid (q, a, q') \in \Delta, a \in \Sigma\} \cup \{(s, q) \mid q \in I\} \cup \{(q, t) \mid q \in F\}$$

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 einem  $q_f \in F$  enthalten; die Beschriftungen der entsprechenden Kanten in  $\mathcal{A}$  liefern ein Wort  $w \in L(\mathcal{A})$ , welches  $L(\mathcal{A}) \neq \emptyset$  bezeugt.

- $\pi$  lässt sich in Polynomialzeit berechnen:

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

**Teil II.**

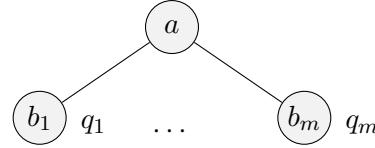
**Endliche Automaten**

**auf endlichen Bäumen**

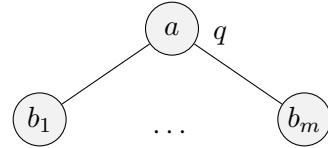
## T2.1 Skizze zur Intuition der Übergänge im Baumautomaten

Der Übergang  $a(q_1, \dots, q_m) \rightarrow q$  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.

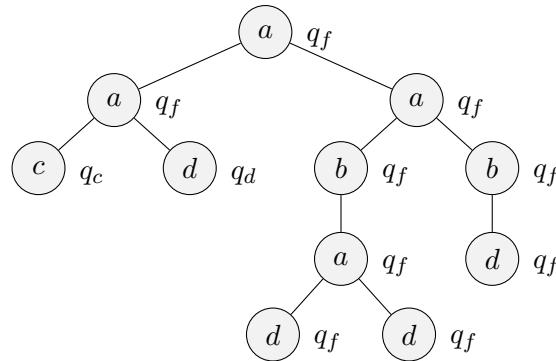


## T2.2 Beispiel-Run

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

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

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

$$a \rightarrow S \in \Delta^d \quad (*)$$

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

**Induktionsschritt.** Sei  $p$  eine Position der Höhe  $\geq 0$  mit Kindern  $p_1, \dots, p_m$ ,  $m > 0$ , und sei  $t(p) = a \in \Sigma_m$ . Weil die Höhe der  $p_i$  geringer ist als die von  $p$ , ist  $S_i := r_T^d(p_i)$  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

$$a(S_1, \dots, S_m) \rightarrow S \in \Delta^d \quad (**)$$

und setzen  $r_T^d(p) = S$ . Dann gilt (a) wegen IV und (\*), und (b) gilt wegen IV und weil  $S$  nach Definition von  $\Delta^d$  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  $a \rightarrow S \in \Delta^d$  gilt. Mit der Definition von  $\Delta^d$  erhalten wir daraus

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

Da die Bedingung  $a \rightarrow q \in \Delta$  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  $p_1, \dots, p_m$ ,  $m > 0$ , und sei  $t(p) = a \in \Sigma_m$ . Weil die Höhe der  $p_i$  geringer ist als die von  $p$ , ist die Induktionsvoraussetzung (IV) anwendbar und liefert:

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

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

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

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  $T_{p_i}$  mit  $r_i(\varepsilon) = q_i$ . 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  $(xx)$  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  $(xx)$ . 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  $T_{p_i}$  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.

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

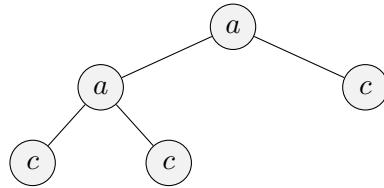
$$\begin{aligned}
 T \in L(\mathcal{A}^d) &\text{ gdw. } r_T^d(\varepsilon) \in F^d && (\text{Def. Akzeptanz, HA1}) \\
 &\text{ gdw. } r_T^d(\varepsilon) \cap F \neq \emptyset && (\text{Def. } F^d) \\
 &\text{ gdw. es gibt Run } r \text{ von } \mathcal{A} \text{ auf } T_\varepsilon \text{ mit } r(\varepsilon)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

## T2.4 Vorberichtigungen 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  $p_1, p_2$  und
- deren Teilbäume  $T_{p_1}, T_{p_2}$  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  $L(\mathcal{A}) = L_2$ .

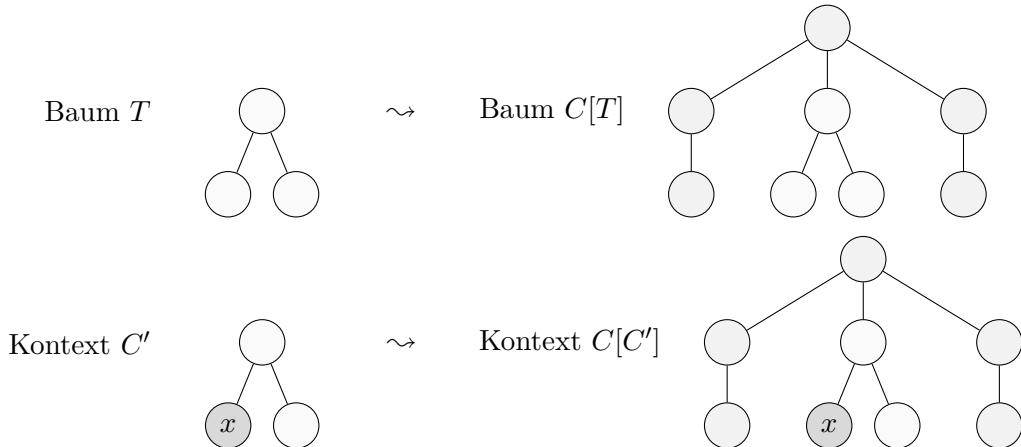


## T2.6 Beispiele für Kontexte

- Unärer Kontext, schematisch:



- Einsetzen:

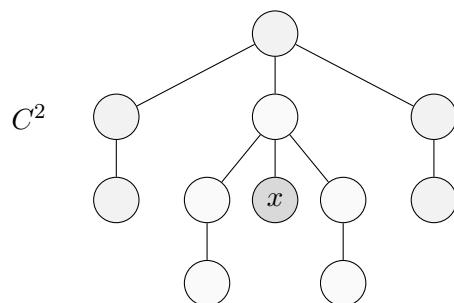


- Trivialer Kontext:

$C_0$

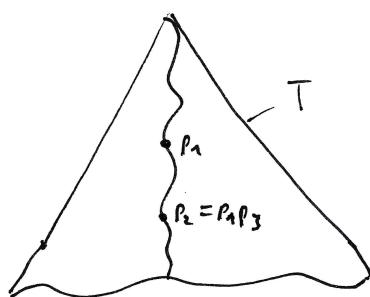


- Iterierte Kontexte:

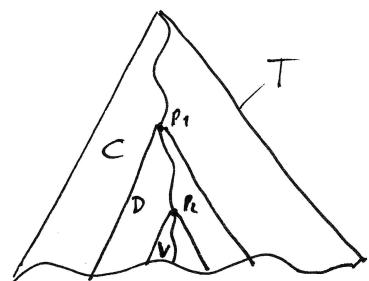


## T2.7 Illustrationen zum Beweis des Pumping-Lemmas

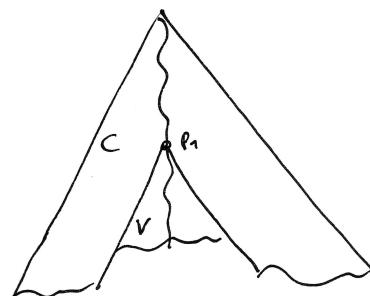
Folie 2.32:



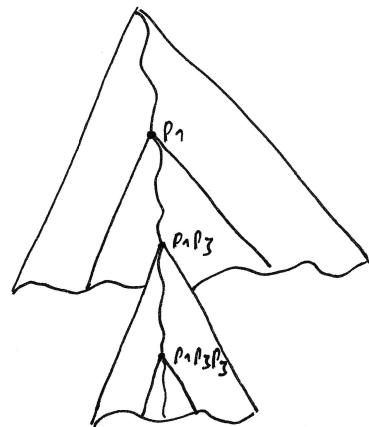
Folie 2.33:



Folie 2.34:



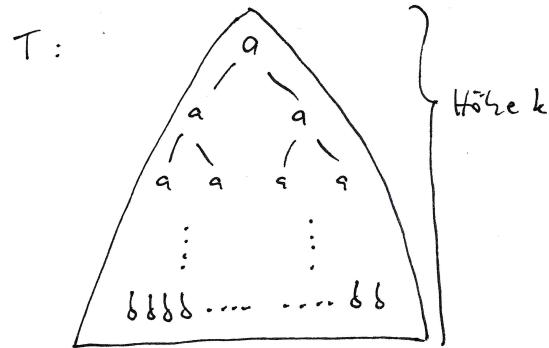
Folie 2.35:



## 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  $C[D^2[V]]$ :



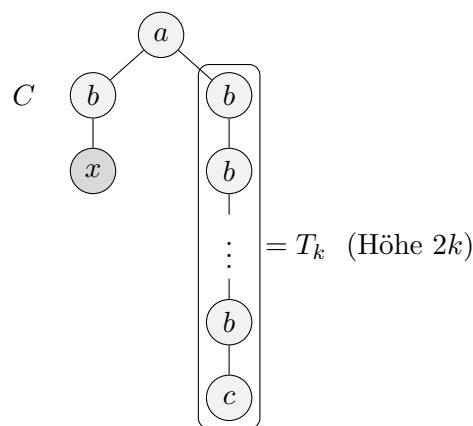
In diesem Baum gibt es einen Pfad der Länge  $d_1 + 2d_2 + d_3 > k$ , nämlich zur Position  $p_1p_2p_2p_3$ . Da  $T$  vollständig ist, hat  $p_1$  ein Kind, das nicht auf diesem Pfad liegt und dessen Teilbaum also nicht vom Pumpen betroffen ist. Deshalb gibt es in  $C[D^2[V]]$  auch einen Pfad der Länge  $k$ , 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  $\Sigma$  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  $(a, q_I) \rightarrow (q_1, q_2)$  mit  $(a, q_I)$  auf der linken Seite. Da  $a(bc), a(cb) \in L$ , muss  $\Delta$  auch die folgenden Übergänge enthalten:

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

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

## 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  $a(q_1, \dots, q_m) \rightarrow q$  der eindeutig bestimmte Übergang, der zu  $q \in R$  geführt hat. Setze:

- $P = \{\varepsilon, 1, \dots, m\}$

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

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

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

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

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

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

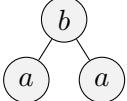
## T2.12 Beispiele für Hecken und Bäume

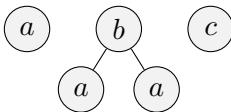
Sei  $\Sigma = \{a, b, c\}$ .

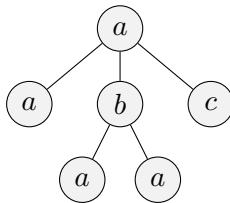
- $\varepsilon$  ist eine Hecke.

- 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  $\Sigma$ , 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

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

haben – damit wäre aber  $\Delta$  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

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

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  $\text{tgV}(p_1, p_2)$  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  $\varepsilon$  gemeinsam.

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



Nun betrachten wir die Sprache

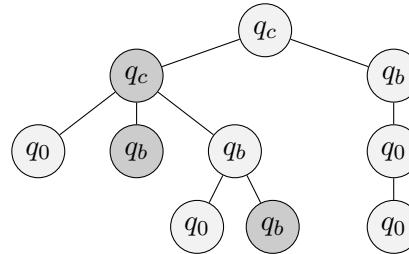
$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit } t(p_1) = t(p_2) = b \text{ und } t(\text{tgV}(p_1, p_2)) = c\}$$

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.

conference	$\rightarrow$	$track^+ + (session (break + \varepsilon))^+$
track	$\rightarrow$	$(session (break + \varepsilon))^+$
session	$\rightarrow$	chair talk <sup>+</sup>
talk	$\rightarrow$	(title authors) + (title speaker)
chair	$\rightarrow$	DATA
...		
title	$\rightarrow$	DATA

Eine mögliche Ableitung beginnt wie folgt.

- Beginne mit dem Startsymbol:



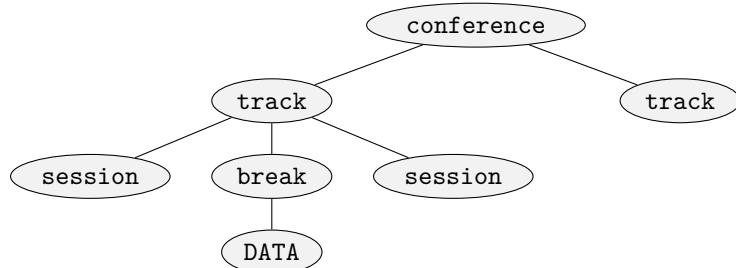
- Wende Regel 1 auf diesen Knoten an:



- 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:

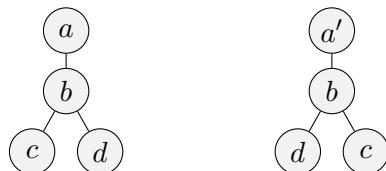


Man kann also mit DTDs nicht in Tiefe  $\geq 2$  zählen. Für unser Konferenz-Beispiel heißt das, dass man z. B. keine DTD schreiben kann, die erfordert, dass alle Konferenzen genau 5 Vorträge haben.

### 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

```
talk → (title authors) + (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:

```
talk → title (authors + 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, ... Schritten ( $\mathcal{A}_1$ ) bzw. 0, 2, 4, ... 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:

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

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

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

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  $s = s_0 s_1 s_2 \dots$ . Diese ist ein Run von  $\mathcal{A}_1$  auf  $\alpha$ : da  $r$  ein Run ist, folgt mit der Definition von  $I$  bzw.  $\Delta$ , 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  $\text{Inf}(r) \cap F \neq \emptyset$  (und der Definition von  $F$ ) enthält  $r$  unendlich viele Zustände der Form

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

also enthält  $r$  auch unendlich viele Zustände der Form

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

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  $\text{Inf}(s) \cap F_1 \neq \emptyset$  und damit  $s$  erfolgreich.

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

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

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

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  $I, F, \Delta$ , dass  $r$  ein erfolgreicher Run von  $\mathcal{A}$  auf  $\alpha$  ist. Folglich ist  $\alpha \in L_\omega(\mathcal{A})$ .  $\square$

### T3.3 Büchi-Erkennbarkeit von $W^\omega$ 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  $r = q_0 q_1 q_2 \cdots$  von  $\mathcal{A}_2$  auf  $\alpha$ , 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  $q_{i_0}, q_{i_1}, q_{i_2}, \dots$  alle Vorkommen von  $q_I$  in  $r$ . Für jedes  $j \geq 0$  betrachten wir die Folge

$$r_j := \underbrace{q_{i_j} q_{i_j+1} \cdots q_{i_{j+1}-1}}_{\substack{\parallel \\ q_I}} q_{i_{j+1}}.$$

Nach Konstruktion von  $\Delta_2$  und wegen der Annahmen über  $\mathcal{A}_1$  gibt es ein  $q_f \in F$ , so dass

$$\underbrace{q_{i_j} q_{i_j+1} \cdots q_{i_{j+1}-1}}_{\substack{\parallel \\ q_I}} q_f \in F$$

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  $\alpha = w_0 w_1 w_2 \cdots$  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  $j$ . 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  $\Delta_2$  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  $\alpha$ . Folglich ist  $\alpha \in L_\omega(\mathcal{A}_2)$ .  $\square$

## 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  $n \geq 1$  und

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

**Beweis.**

„ $\Leftarrow$ “ Folgt aus Lemmas 3.6, 3.8 und 3.9.

„ $\Rightarrow$ “ 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  $\omega$ -Wort  $\alpha \in L_\omega(\mathcal{A})$ , auf dem  $\mathcal{A}$  einen erfolgreichen Run  $r = q_0 q_1 q_2 \dots$  mit  $q_f \in \text{Inf}(r) \cap F$  hat, gibt es

- ein endliches Präfix von  $\alpha$ , 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 $\overrightarrow{W}$

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

$$\overrightarrow{W}_1 = \{a^n b^\omega \mid n \geq 0\} \cup \{a^\omega\}$$

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

$$\overrightarrow{W}_2 = \emptyset$$

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

$$\overrightarrow{W}_3 = \{a, b\}^\omega$$

$$W_4 = \{w \in \{a, b\}^* \mid \#_a(w) \text{ ist gerade}\}$$

$$\overrightarrow{W}_4 = \{\alpha \in \{a, b\}^\omega \mid \#_a(\alpha) = \infty \text{ oder } \alpha = wb^\omega \text{ mit } \#_a(w) \text{ gerade}\}$$

$$W_5 = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$

$$\overrightarrow{W}_5 = \left\{ \alpha \in \{a, b\}^\omega \mid \#\{i \mid \#_a(\alpha[0, i]) = \#_b(\alpha[0, i])\} = \infty \right\}$$

## T3.6 Beweis Charakterisierung DBA-erkennbarer Sprachen

**Satz 3.11.** Eine  $\omega$ -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 **DEA/DBA**  $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, F)$  gilt:

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

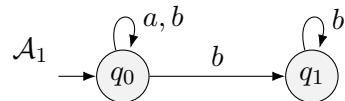
„ $\subseteq$ “ (Diese Richtung funktioniert sogar, wenn  $\mathcal{A}$  ein **NEA/NBA** ist.)

Sei  $\alpha \in L_\omega(\mathcal{A})$ . Dann gibt es einen erfolgreichen Run  $r = q_0 q_1 q_2 \dots$  von  $\mathcal{A}$  auf  $\alpha$ . Seien  $i_0, i_1, i_2, \dots \in \mathbb{N}$  die Positionen mit  $q_{i_j} \in F$ . Dann ist für jedes  $j \geq 0$  das Präfix  $q_0 \dots q_{i_j}$  von  $r$  ein erfolgreicher Run des **NEAs**  $\mathcal{A}$  auf  $\alpha_0 \dots \alpha_{i_j-1}$ . Damit gibt es unendlich viele Präfixe von  $\alpha$ , 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  $\alpha$  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  $\alpha \in L_\omega(\mathcal{A})$ .  $\square$

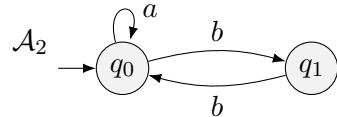
## T3.7 Beispiele für Muller-Automaten

- 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 denselben 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 *unendlich* 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  $\mathcal{P} = \bigcup_{i \leq n} \mathcal{P}_i$  gilt:  $L_\omega(\mathcal{A}) = \bigcup_{i \leq n} L_\omega(Q, \Sigma, \Delta, I, \mathcal{P}_i)$ .

### T3.9 Beispiele für Streett-Automaten

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

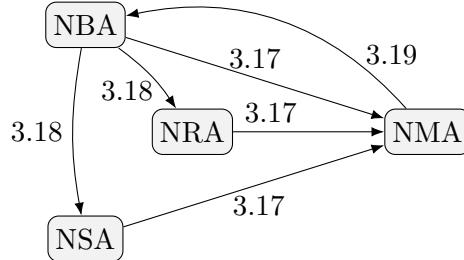


- (S1) Wenn  $\mathcal{P} = \{(\{q_0\}, \{q_1\})\}$ , dann  $L_\omega(\mathcal{A}) = (a + bb)^\omega$  (denn jeder Run ist erfolgreich).
- (S2) Wenn  $\mathcal{P} = \{(\{q_0\}, \emptyset)\}$ , dann  $L_\omega(\mathcal{A}) = (a + bb)^\omega$  (denn jeder Run ist erfolgreich).
- (S3) Wenn  $\mathcal{P} = \{(\{q_1\}, \{q_0\})\}$ , 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  $\infty$  oft vorkommen, also muss  $q_0 \in F$  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  $\infty$  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  $\mathcal{P} = \bigcup_{i \leq n} \mathcal{P}_i$  gilt:  $L_\omega(\mathcal{A}) = \bigcap_{i \leq n} L_\omega(Q, \Sigma, \Delta, I, \mathcal{P}_i)$ .

### T3.10 Überblick Beweis der Gleichmächtigkeit

Das folgende Bild illustriert, wie die Aussagen der Lemmata 3.17–3.19 zur Äquivalenz der vier Automatenmodelle (Satz 3.16) beitragen. Ein Pfeil vom Knoten NxA zum Knoten NyA bedeutet dabei: „jede NxA-erkennbare Sprache ist NyA-erkennbar“; die Beschriftung der Pfeile gibt die Nummer des Lemmas an.



### T3.11 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  $r = q_0 q_1 q_2 \dots$  ein erfolgreicher Run von  $\mathcal{A}$  (NMA!) auf  $\alpha$ , also  $q_0 \in I$  und  $\text{Inf}(r) = F$ . Dann gibt es eine Position  $i_0 \geq 1$ , ab der nur noch akzeptierende Zustände auftreten, d. h.  $q_i \in F$  für alle  $i \geq i_0$ . Daraus konstruieren wir wie folgt induktiv eine Folge  $s = s_0 s_1 s_2 \dots$  von Zuständen von  $\mathcal{A}'$ :

- Für alle  $i < i_0$  setze  $s_i = q_i$ .
- $s_{i_0} = \langle q_{i_0}, \{q_{i_0}\} \rangle$
- Für alle  $i > i_0$  setze

$$s_i = \begin{cases} \langle q_i, S \cup \{q_i\} \rangle & \text{falls } s_{i-1} = \langle q_{i-1}, S \rangle \text{ und } S \neq F \\ \langle q_i, \{q_i\} \rangle & \text{falls } s_{i-1} = \langle q_{i-1}, F \rangle \end{cases}$$

Diese Konstruktion von  $s$  stellt sicher:

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

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

„ $\subseteq$ “ Sei  $\alpha \in L_\omega(\mathcal{A}')$  und  $s = s_0s_1s_2\cdots$  ein erfolgreicher Run von  $\mathcal{A}'$  (NBA!) auf  $\alpha$ , also  $s_0 \in I'$  und ein  $\langle q_f, F \rangle$  kommt  $\infty$  oft in  $s$  vor. Konstruiere daraus eine Folge  $r = q_0q_1q_2\cdots$  von Zuständen aus  $Q$  wie folgt:

- Wenn  $s_i \in Q$ , dann  $q_i = s_i$ .
- Wenn  $s_i = \langle q_f, S \rangle$ , dann  $q_i = q_f$ .

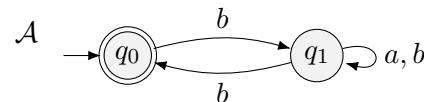
Diese Konstruktion stellt sicher:

- $r$  ist Run von  $\mathcal{A}$  auf  $\alpha$  (vgl. Konstruktion von  $\Delta'$ ).
- $r_0 \in I$ .
- $r$  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  $\langle q_f, F \rangle$  unendlich oft in  $s$  vorkommt).

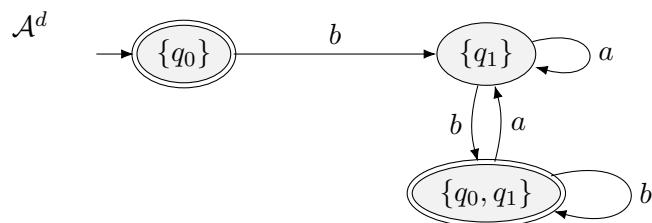
Damit ist  $r$  ein erfolgreicher Run des NMA  $\mathcal{A}$  auf  $\alpha$ , also  $\alpha \in L_\omega(\mathcal{A})$ .  $\square$

### T3.12 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  $\alpha$  fortgesetzt werden kann.

### T3.13 Variation der Akzeptanzbedingung im vorigen Beispiel

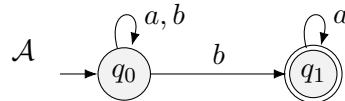
Wir betrachten dieselben Automaten  $\mathcal{A}, \mathcal{A}^d$  wie im vorigen Beispiel. Man könnte versuchen, durch Variation der Akzeptanzbedingung von  $\mathcal{A}^d$  einen zu  $\mathcal{A}$  äquivalenten deterministischen Muller-, Rabin- oder Streett-Automaten zu erhalten, ohne die eigentliche 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.

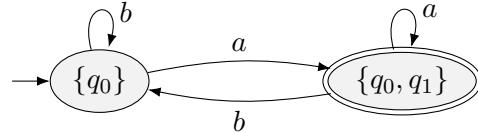
Im 1. und 2. Fall gibt es jedoch Bad Runs auf  $(ba)^\omega$  bzw.  $ba^\omega$ ; im 3. Fall gibt es Wörter, die von  $\mathcal{A}$  akzeptiert werden, aber nicht von  $\mathcal{A}^d$ , z. B.  $(bba)^\omega$ .

### T3.14 Beispiel für Safras Trick 1

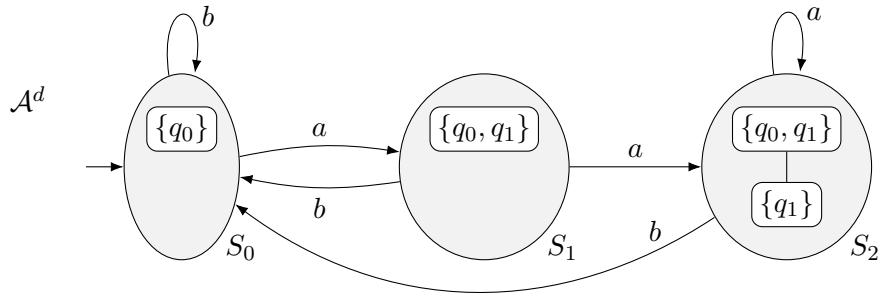
Betrachte folgenden NBA  $\mathcal{A}$  über dem Alphabet  $\Sigma = \{a, b\}$ .



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



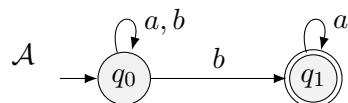
mit einem Bad Run auf  $(ab)^\omega$ . Mittels Safras Trick 1 erhält man hingegen folgenden deterministischen Automaten  $\mathcal{A}^d$ .



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  $\mathcal{P} = \{\{S_0\}, \{S_2\}\}$ .

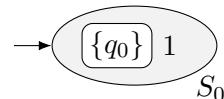
### T3.15 Beispiel für die gesamte Safra-Konstruktion

Wir betrachten den NBA  $\mathcal{A}$  aus dem vorigen Beispiel:



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  $S_0, S_1, S_2, \dots$  und schreiben diese Namen jeweils rechts neben den entsprechenden Zustand. Außerdem verwenden wir innerhalb von Safrabäumen als Knotennamen die Zahlen 1, 2, 3, ... und schreiben sie rechts neben den jeweiligen Knoten. Markierte Knoten (Schritt 6) werden wie gehabt mit  $\textcircled{1}$  gekennzeichnet.

**Startzustand** ist der folgende Safrabaum:

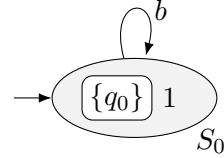


#### Folgezustand von $S_0$ mit $b$

- 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.

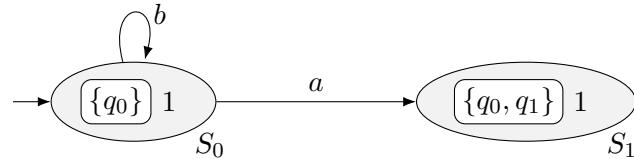
Also ist der  $b$ -Folgezustand von  $S_0$  wieder  $S_0$ :



### Folgezustand von $S_0$ mit $a$

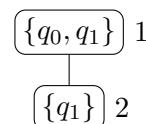
- Schritte 1–2 sind nicht anwendbar, siehe oben.
- In Schritt 3 ändert sich der Makrozustand von Knoten 1 zu  $\{q_0, q_1\}$ , da sowohl  $q_0$  als auch  $q_1$  von  $q_0$  aus mit  $a$  erreicht werden können.
- Schritte 4–6 sind nicht anwendbar, da Knoten 1 noch keine Kinder hat.

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



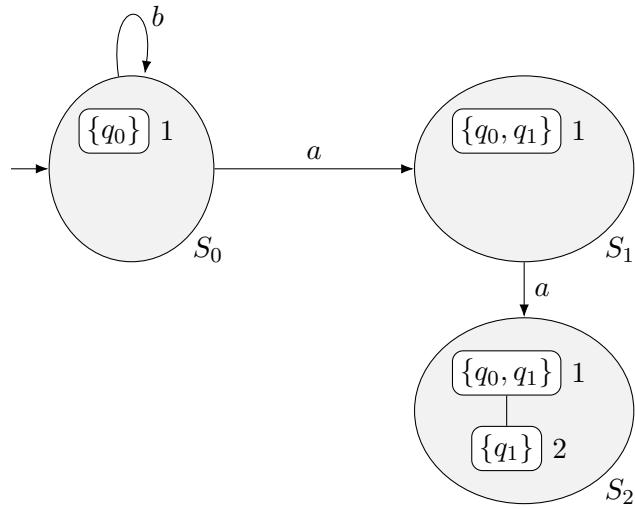
### Folgezustand von $S_1$ mit $a$

- 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:



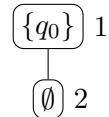
- In Schritt 3 wird auf beide Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit  $a$  ändert sich der Inhalt beider Makrozustände nicht.
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- Schritt 5 ist nicht anwendbar, da der Makrozustand von Knoten 2 nicht leer ist.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



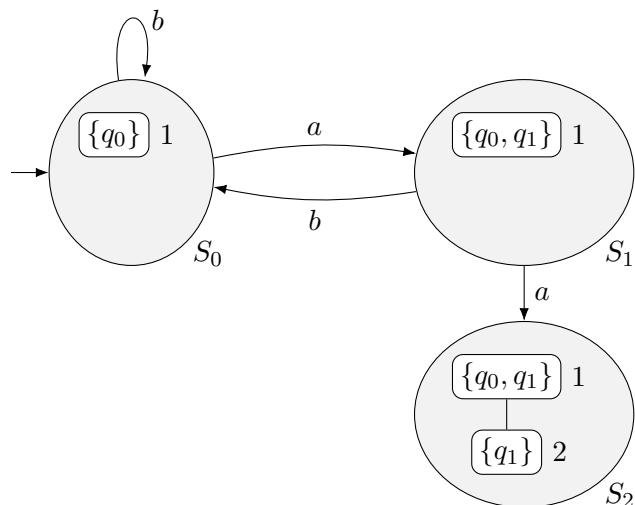
### Folgezustand von $S_1$ mit $b$

- 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:



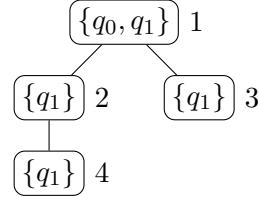
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- In Schritt 5 wird Knoten 2 gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

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

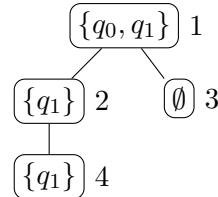


### Folgezustand von $S_2$ mit $a$

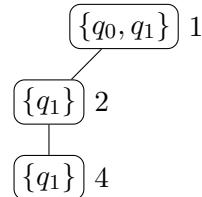
- Schritt 1 ist nach wie vor nicht anwendbar.
- In Schritt 2 wird je ein neues Kind von Knoten 1 und 2 erzeugt, da die Makrozustände beider Knoten akzeptierende Zustände enthalten:



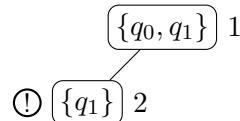
- In Schritt 3 wird auf alle Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit  $a$  ändert sich der Inhalt der Makrozustände nicht.
- In Schritt 4 wird  $q_1$  aus Knoten 3 entfernt, da dieser Zustand im älteren Geschwister 2 enthalten ist:



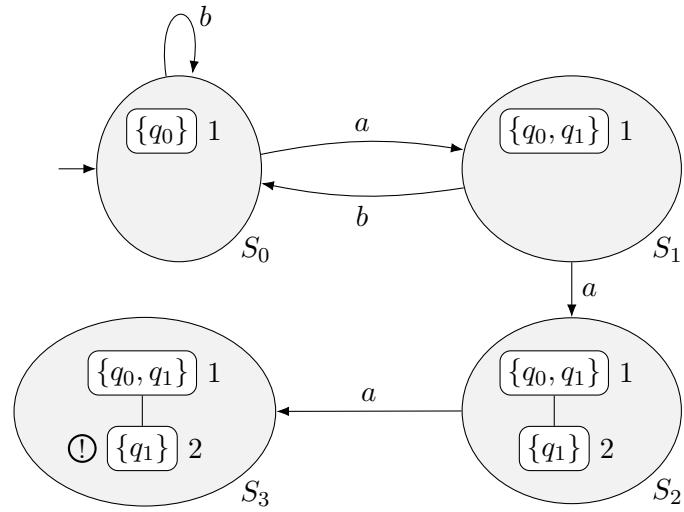
- In Schritt 5 wird Knoten 3 gelöscht:



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

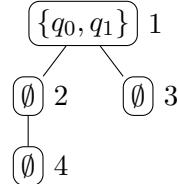


Der zuletzt abgebildete Safrabaum  $S_3$  ist der  $a$ -Folgezustand von  $S_2$ :



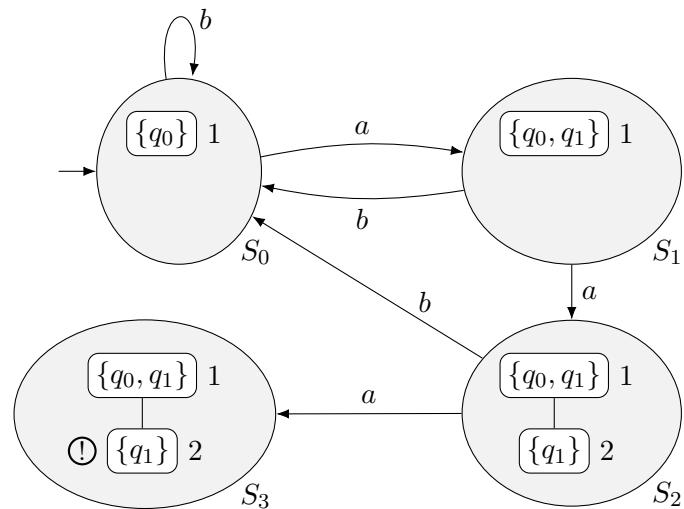
### Folgezustand von \$S\_2\$ mit \$b\$

- 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:



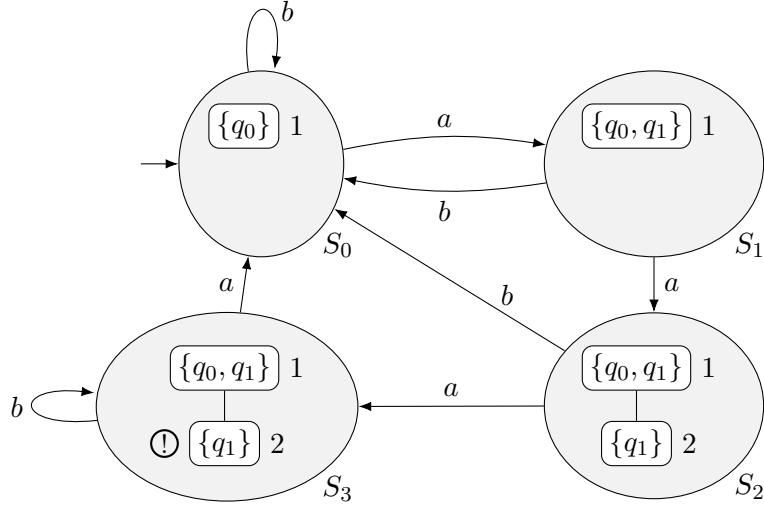
- Schritt 4 ist nicht anwendbar, da alle Makrozustände außer dem von Knoten 1 leer sind.
- In Schritt 5 werden Knoten 2, 3, 4 gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

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  $S_1$  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  $\alpha$ , der einen akzeptierenden Zustand enthält.

**Skizze:**



Teilrun:       $q$        $\dots$        $f$        $\dots$        $p$

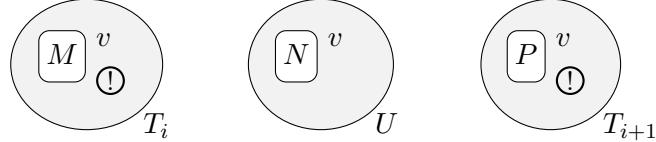
**Beweis der HA.** Damit  $v$  überhaupt mit  $\textcircled{1}$  markiert werden kann, muss  $v$  direkt vor der entsprechenden Anwendung des Schrittes 6 Kinder haben. Diese wurden in irgendeiner

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  $k, \ell, m$  mit  $0 \leq k \leq \ell < m$  (die Zeitpunkte des Vorkommens von  $T_i, U, T_{i+1}$  auf dem Run  $s$ ; mit  $k \leq \ell$  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:

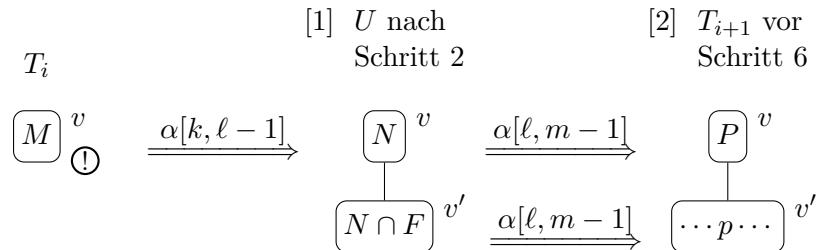
- (\*\*) Für alle  $p \in P$  gibt es ein  $q \in M$  und einen endlichen Run  $q \dots p$  von  $\mathcal{A}$  auf  $\alpha[k, m - 1]$ , der einen akzeptierenden Zustand enthält.

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 Teirläufe  $T \dots U$  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  $\Delta^d$  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  $\circled{1}$  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 Teillauf  $T_i \dots T_{i+1}$  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  $\alpha[\ell, m - 1]$ .

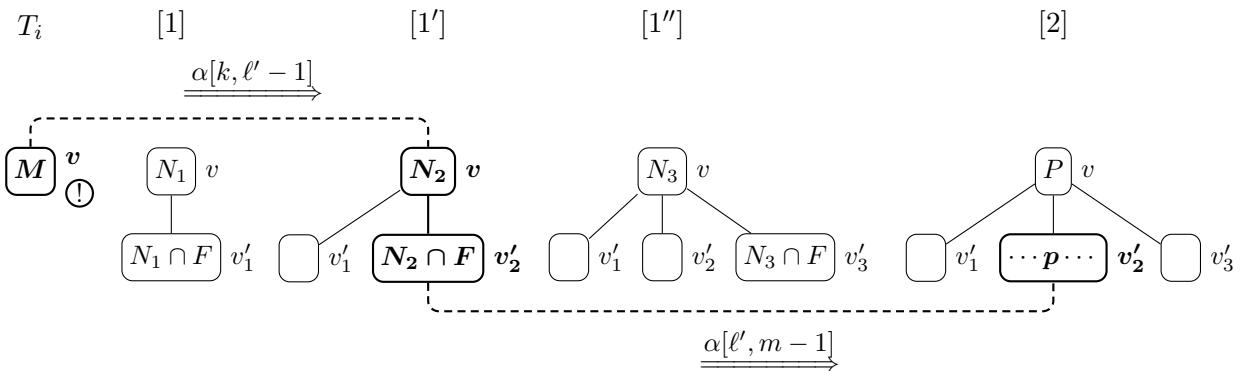
Diesen Run kann man durch die „obere Zeile“ ergänzen:

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  $q \in M \dots$ “ 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)  $\exists m \geq 0 : S_i$  enthält Knoten  $v$  für alle  $i \geq m$
- (b)  $v$  ist in  $\infty$  vielen  $S_i$  mit  $\text{①}$  markiert

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

**Skizze:**  $S_0, S_1, \dots, \underbrace{S_m, S_{m+1}, S_{m+2}, \dots}_{v \text{ in allen } S_i \text{ enthalten}}$

v in allen  $S_i$  enthalten  
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  $\alpha$  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.

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

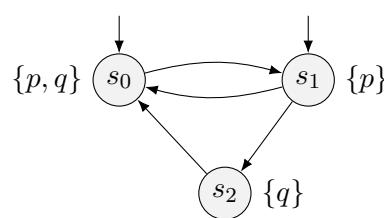
$S_0, S_1, \dots, S_m, S_{m+1}, \dots, \underbrace{S'_m, S_{m'+1}, \dots, S_p, \dots}_{\text{Knoten 1 ist in keinem } S_i \text{ markiert}}$

Da der Run  $r$  erfolgreich ist, gibt es einen Zustand  $f \in \text{Inf}(r) \cap F$ . 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.

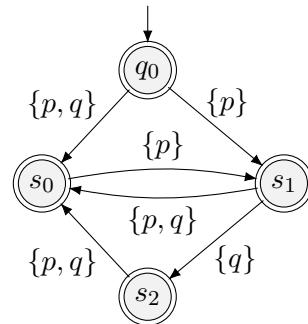
Nun kann man das bisherige Argument von 1 auf  $c$  übertragen: entweder erfüllt  $c$  auch Bedingung (b), oder es gibt ein Kind  $c'$ , das (a) erfüllt. Diese Iteration kann nicht beliebig oft fortsetzen, weil die Tiefe eines Safrabaums durch  $|Q|$  beschränkt ist. Folglich muss es einen Nachfahren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

### T3.18 NBA für eine Kripke-Struktur

Kripke-Struktur  $\mathcal{S}$



zugehöriger NBA  $\mathcal{A}_S$



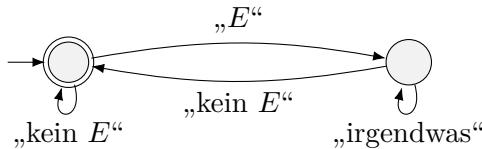
## T3.19 NBAs für Beispiel-Eigenschaften

### (1) Mikrowellen-Beispiel

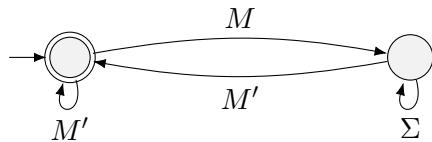
Hier ist das Alphabet  $\Sigma = 2^{\{S,C,H,E\}}$ .

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

Schematisch muss der Automat so aussehen:

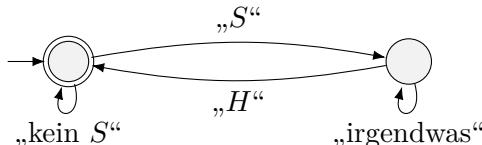


Dabei steht z. B. die Beschriftung „ $E$ “ für alle Alphabetzeichen (Teilmengen von  $\{S, C, H, E\}$ ), die  $E$  enthalten, und „kein  $E$ “ für alle Alphabetzeichen, die  $E$  nicht enthalten. Sei also  $M = \{\{E\}, \{S, E\}, \{C, E\}, \dots, \{S, C, H, E\}\}$  und  $M' = \Sigma \setminus M$ . Dann 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.“

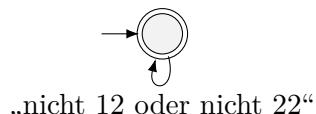
Der Automat ist derselbe wie in (b), nur muss man hier existenzielles Model-Checking statt universellem verwenden.

### (2) Nebenläufige Programme

Hier ist das Alphabet  $\Sigma = 2^{\{0,1,10,11,\dots,23\}}$ .

- (d) „Es kommt nie vor, dass beide Teilprogramme zugleich im kritischen Bereich sind.“

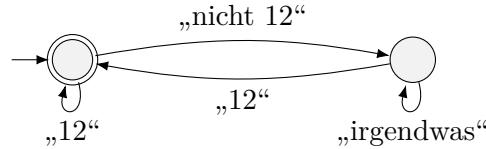
Schematisch muss der Automat so aussehen:



Dabei steht die Beschriftung „nicht 12 oder nicht 22“ der Schleife für alle Alphabetzeichen (Teilmengen von  $\{0, 1, 10, 11, \dots, 23\}$ ), die nicht gleichzeitig 12 und 22 enthalten. Damit ist die korrekte Kantenbeschriftung für die Schleife die Menge  $\{X \subseteq \{0, 1, 10, 11, \dots, 23\} \mid \{12, 22\} \not\subseteq X\}$ .

- (e) „Jedes Teilprogramm kommt beliebig oft in seinen kritischen Bereich.“

Der Automat  $\mathcal{A}_1$ , der beschreibt, dass  $P_1$  beliebig oft in seinen kritischen Bereich kommt, ist dem in (a) sehr ähnlich (hier wieder nur die schematische Darstellung):



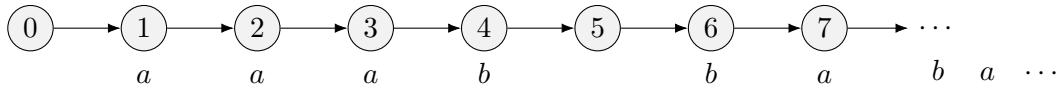
Der Automat  $\mathcal{A}_2$  für  $P_2$  ist analog. Um zu beschreiben, dass *beide* Programme beliebig oft in den jeweiligen kritischen Bereich kommen, muss man den Produktautomaten von  $\mathcal{A}_1$  und  $\mathcal{A}_2$  bilden.

- (e) „Jedes Teilprogramm kann beliebig oft in seinen kritischen Bereich gelangen.“

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

### T3.20 Beispiele für LTL-Syntax und Semantik

Betrachte folgenden Pfad  $\pi$ .

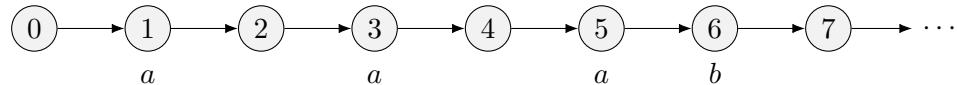


Das heißt also,  $\pi(0) = \emptyset$ ,  $\pi(1) = \{a\}$  usw. Die Beschriftung „ $b a \dots$ “ am rechten Rand bedeutet, dass für alle  $i \geq 4$  gilt:  $\pi(2i) = \{b\}$  und  $\pi(2i+1) = \{a\}$ . Dann gilt:

$$\begin{array}{llll} \pi, 0 \not\models a & \pi, 0 \models Fa & \pi, 0 \not\models G(a \vee b) & \pi, 1 \models a Ub \\ \pi, 0 \models \neg a & \pi, 0 \models Fb & \pi, 6 \models G(a \vee b) & \pi, 0 \not\models a Ub \\ \pi, 0 \models Xa & \pi, 4 \models Fa & \pi, 5 \not\models G(a \vee b) & \\ \pi, 0 \not\models X\neg a & \pi, 0 \models X(a \wedge Fb) & \pi, 0 \models GFa & \end{array}$$

Dabei ist  $GF\varphi$  laut Semantik gleichbedeutend mit „unendlich oft  $\varphi$ “.

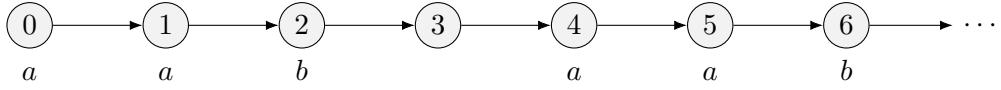
Betrachte nun den folgenden Pfad  $\pi'$ .



Dann gilt  $\pi', 0 \not\models a Ub$ , aber  $\pi', 0 \models (Xa \vee XXa) U (Xb)$ .

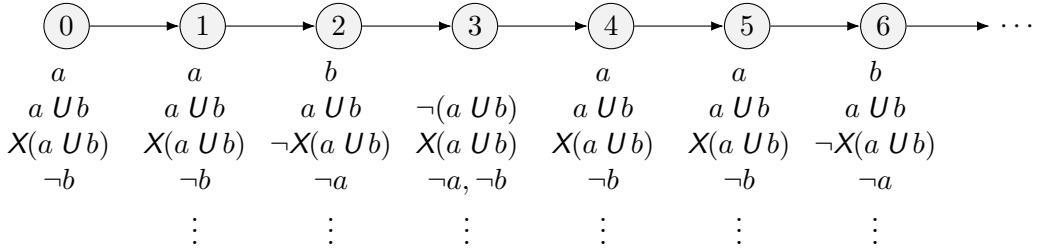
## T3.21 Beispiele für die Erweiterung von Pfaden

Sei z. B.  $\varphi_E = X(a \cup b)$  und der Pfad  $\pi = s_0s_1s_2 \dots$  wie folgt gegeben:



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  $\{a\}, \{b\}, \emptyset, \{a\}, \{b\}, \emptyset, \dots$  besteht (jede Teilmenge von  $\text{AV}$  ist ein Zeichen!).

Der zugehörige erweiterte Pfad  $\bar{\pi} = t_0t_1t_2 \dots$  ist der folgende.



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

## T3.22 Beispiele für elementare Formelmengen

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

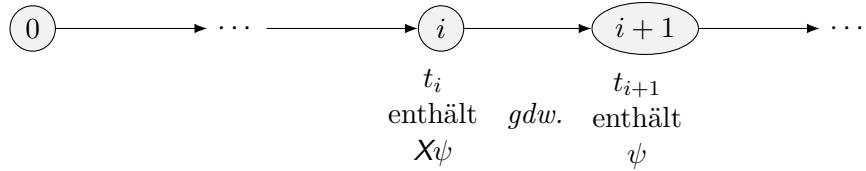
- $\{a, b, \varphi_E\}$  ist konsistent bezüglich der Aussagenlogik und lokal konsistent bezüglich des  $U$ -Operators, aber nicht maximal, weil weder  $\neg a \wedge b$  noch  $\neg(\neg a \wedge b)$  enthalten ist.
- Fügt man  $\neg a \wedge b$  hinzu, so ist die resultierende Menge  $\{a, b, \neg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

$$\begin{aligned} & \{a, b, \neg(\neg a \wedge b), \varphi_E\} \\ & \{a, b, \neg(\neg a \wedge b), \neg \varphi_E\} \\ & \{a, \neg b, \neg(\neg a \wedge b), \varphi_E\} \\ & \{a, \neg b, \neg(\neg a \wedge b), \neg \varphi_E\} \\ & \{\neg a, \neg b, \neg(\neg a \wedge b), \neg \varphi_E\} \\ & \{\neg a, b, \neg a \wedge b, \varphi_E\} \end{aligned}$$

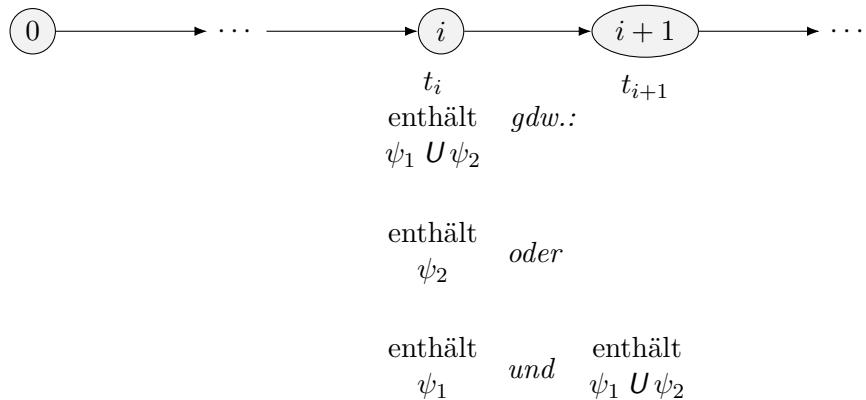
### T3.23 Skizzen zur Def. der Überführungsrelation des GNBA

Bedingung ① besagt, dass im Beispiel in T3.21 höchstens die Transitionen  $(t_0, \{a\}, t_1)$ ,  $(t_1, \{a\}, t_2)$ ,  $(t_2, \{b\}, t_3)$ ,  $(t_3, \emptyset, t_4)$  usw. erlaubt sind (sofern jeweils Bedingungen ② und ③ auch erfüllt sind).

Bedingung ② kann man so veranschaulichen:



Bedingung ③ kann man so veranschaulichen:



Diese Bedingung nutzt die semantische Äquivalenz  $\psi_1 U \psi_2 \equiv \psi_2 \vee (\psi_1 \wedge X(\psi_1 U \psi_2))$  (weist diese selbst nach :-)).

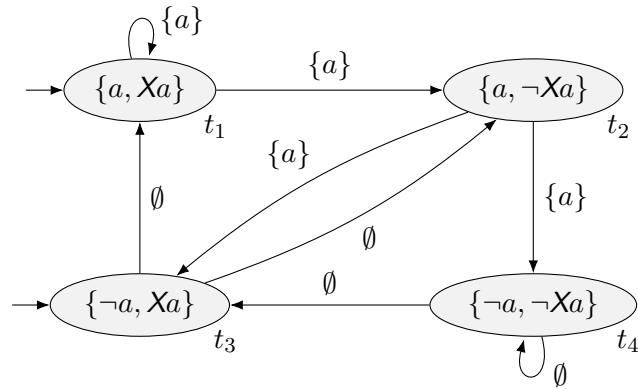
### T3.24 GNBA für die Beispiel-Formel Xa

- $\varphi = Xa$
- $\text{cl}(\varphi) = \{a, \neg a, Xa, \neg Xa\}$
- 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:

$$\begin{aligned} t_1 &= \{a, Xa\} \\ t_2 &= \{a, \neg Xa\} \\ t_3 &= \{\neg a, Xa\} \\ t_4 &= \{\neg a, \neg Xa\} \end{aligned}$$

- Übergänge: es genügt, Bedingung ② für  $X$  zu überprüfen, also gibt es Übergänge
  - 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  $\varphi$ )
- 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:



### T3.25 GNBA für die Beispiel-Formel $(\neg a) Ub$

- $\varphi = (\neg a) Ub$
- $\text{cl}(\varphi) = \{a, \neg a, b, \neg b, (\neg a) Ub, \neg((\neg a) Ub)\}$
- Zustände (elementare Formelmengen): Wegen Konsistenz bezüglich  $\neg$  und Maximalität muss von  $a, \neg a$  bzw.  $b, \neg b$  bzw.  $(\neg a) Ub, \neg((\neg a) Ub)$  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.  $U$ :
  - $\{a, b, \neg((\neg a) Ub)\}$  und  $\{\neg a, b, \neg((\neg a) Ub)\}$   
(denn wenn  $b \in t$ , dann muss auch  $\neg((\neg a) Ub) \in t$  sein);
  - $\{a, \neg b, (\neg a) Ub\}$  (denn wenn  $(\neg a) Ub \in t$  und  $b \notin t$ , dann muss  $a \in t$  sein).

Die verbleibenden fünf Formelmengen sind elementar:

$$\begin{array}{ll}
 t_1 = \{a, b, (\neg a) Ub\} & t_4 = \{\neg a, \neg b, \neg((\neg a) Ub)\} \\
 t_2 = \{\neg a, b, (\neg a) Ub\} & t_5 = \{a, \neg b, \neg((\neg a) Ub)\} \\
 t_3 = \{\neg a, \neg b, (\neg a) Ub\} &
 \end{array}$$

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

$$(\neg a) \cup b \in t \text{ gdw. } b \in t \text{ oder } (\neg a \in t \text{ und } (\neg a) \cup b \in t') \quad (*)$$

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

- (a)  $((\neg a) \cup b \in t \text{ und } b \in t)$
- oder  $((\neg a) \cup b \in t \text{ und } \neg a \in t \text{ und } (\neg a) \cup b \in t')$  (b)
- oder  $((\neg a) \cup b \notin t \text{ und } b \notin t \text{ und } \neg a \notin t)$  (c)
- oder  $((\neg a) \cup b \notin t \text{ und } b \notin t \text{ und } (\neg a) \cup b \notin t')$  (d)

Folglich gibt es Übergänge

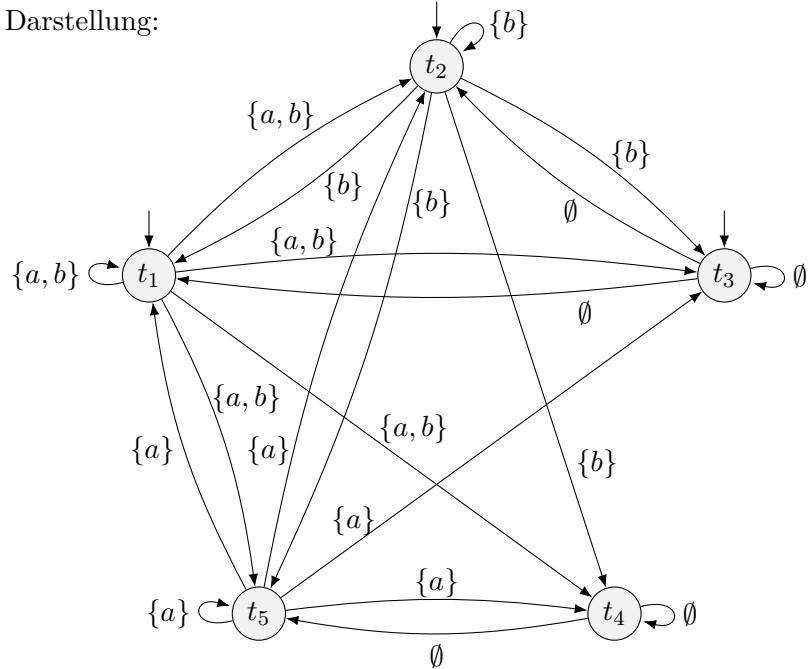
- (a) von  $t_1$  mit  $\{a, b\}$  zu  $t_1, t_2, t_3, t_4, t_5$  und von  $t_2$  mit  $\{b\}$  zu  $t_1, t_2, t_3, t_4, t_5$ ;
- (b) von  $t_3$  mit  $\emptyset$  zu  $t_1, t_2, t_3$ ;
- (c) von  $t_5$  mit  $\{a\}$  zu  $t_1, t_2, t_3, t_4, t_5$ ;
- (d) von  $t_4$  mit  $\emptyset$  zu  $t_4, t_5$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

- Anfangszustände:  $t_1, t_2, t_3$  (diese enthalten  $\varphi$ )
- Akzeptanzkomponente:

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

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

Graphische Darstellung:



**Teil IV.**

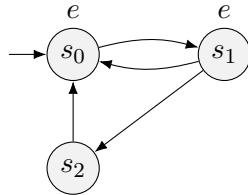
**Endliche Automaten**

**auf unendlichen Bäumen**

## T4.1 LTL-Formeln „zu stark/schwach“

Sei  $\varphi_1 := G(e \rightarrow F\neg e)$  und  $\varphi_2 := GF\neg e$ .

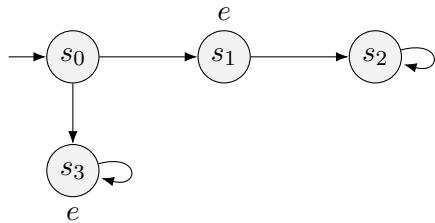
$\varphi_1, \varphi_2$  „zu stark für universelles Model Checking“. Wir betrachten folgende Kripke-Struktur  $\mathcal{S}$ :



Intuitiv gesprochen, erfüllt  $\mathcal{S}$  die Eigenschaft, die mit  $\varphi_1$  bzw.  $\varphi_2$  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  $\pi', 0 \models \varphi_i, i = 1, 2$ ).

$\varphi_1, \varphi_2$  „zu schwach für existenzielles Model Checking“. Wir betrachten folgende Kripke-Struktur  $\mathcal{S}'$ :

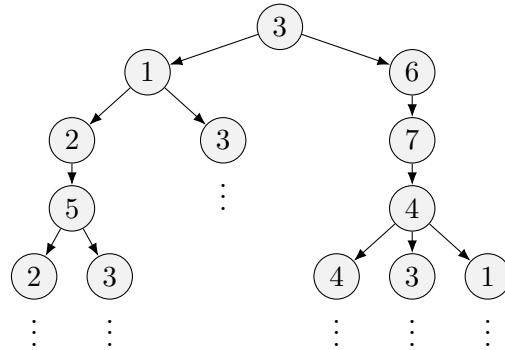


Intuitiv gesprochen, erfüllt  $\mathcal{S}'$  die gewünschte Eigenschaft *nicht*: der Pfad  $s_0 s_3$  kann nicht mehr wie gewünscht fortgesetzt werden.

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

## T4.2 Beispiel-Berechnungsbaum

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

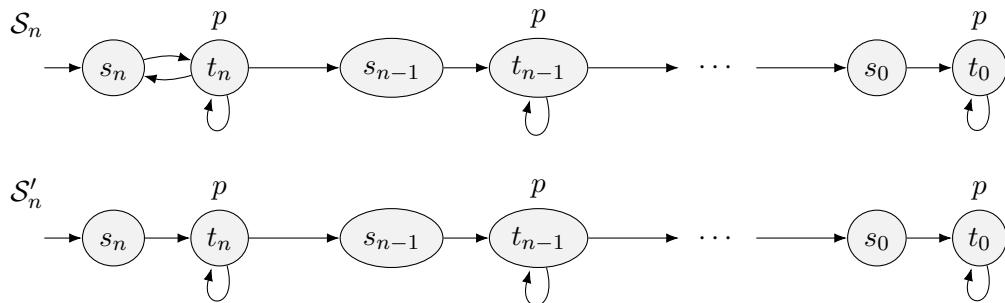


Die gegebene Struktur  $\mathcal{S}$  ist eine endliche Repräsentation dieses unendlichen Baums (und vieler weiterer Bäume).

### T4.3 Beweis Teil 2 des Ausdrucksstärke-Lemmas

**Lemma 4.5 (2).** Es gibt keine zu  $F G p$  äquivalente CTL-Zustandsformel.

**Beweis.** Betrachte zwei Folgen  $\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \dots$  und  $\mathcal{S}'_0, \mathcal{S}'_1, \mathcal{S}'_2, \dots$  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)  $\mathcal{S}_n \not\models FGp$  (wegen Pfad  $(s_n t_n)^\omega$  ab  $s_n \in \S_0$ )  
(ii)  $\mathcal{S}'_n \models FGp$  (weil jeder Pfad auf ein  $(t_i)^\omega$  enden muss, für ein  $i \leq n$ )

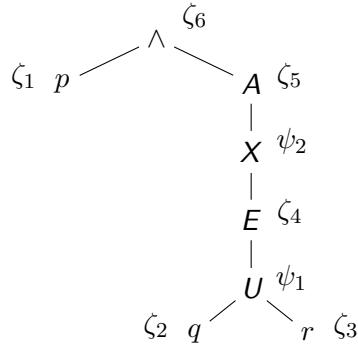
Außerdem zeigt man leicht per Induktion über  $n$  die folgende Aussage: Für alle  $n \geq 0$  und alle CTL-Zustandsformeln  $\zeta$  der Länge  $\leq n$  gilt:

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

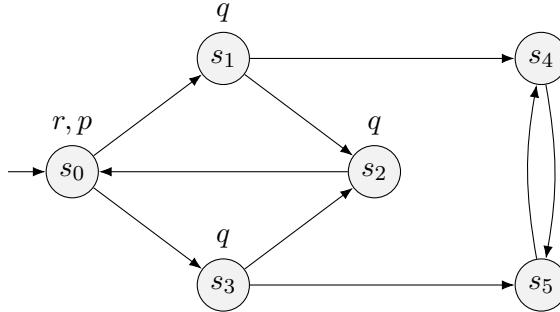
Angenommen, es gebe eine CTL-Zustandsformel  $\zeta$  mit  $\zeta \equiv FGp$ . Sei  $n := |\zeta|$ . Dann gilt wegen (i) und (ii):  $S_n \not\models \zeta$  und  $S'_n \models \zeta$ . Das widerspricht aber (iii).  $\square$

## T4.4 Beispiel für den Model-Checking-Algorithmus

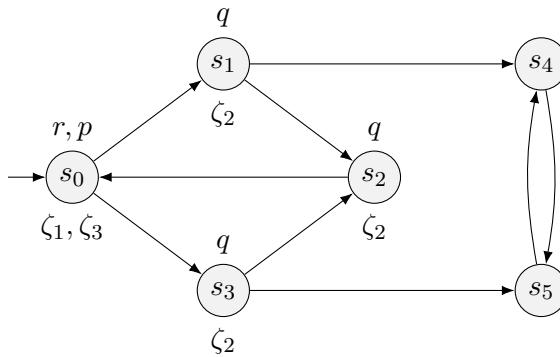
Wir betrachten die Zustandsformel  $\zeta = p \wedge A X E(q \ Ur)$ . Ihre Baumdarstellung ist wie folgt; die einzelnen Teilformeln sind mit  $\zeta_1, \dots, \zeta_6$  (Zustandsformeln) und  $\psi_1, \psi_2$  (Pfadformeln) markiert.



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

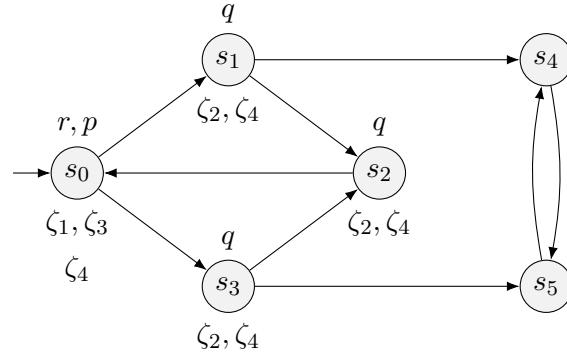


Zuerst werden für die Zustandsformeln  $\zeta_1, \zeta_2, \zeta_3$  aus den Blättern von  $\zeta$  alle Zustände markiert, die mit der jeweiligen Aussagenvariable markiert sind:

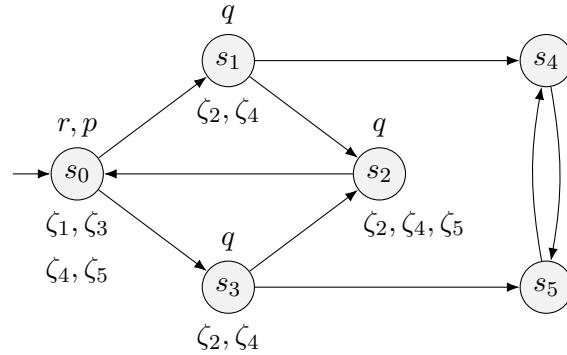


Die „nächsthöhere“ Zustandsformel ist  $\zeta_4 = E(\zeta_2 \ U \zeta_3)$ . Es werden also als nächstes alle Zustände mit  $\zeta_4$  markiert, die aufgrund der bisherigen Markierung  $E(\zeta_2 \ U \zeta_3)$  erfüllen,

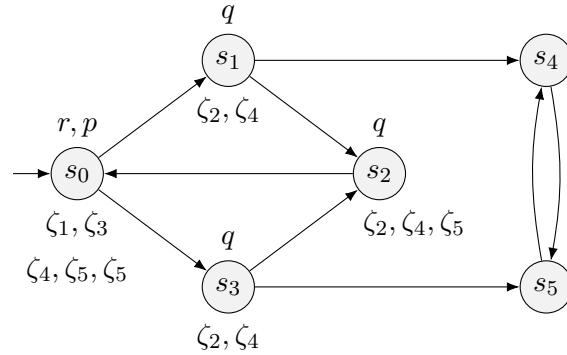
in denen also *mindestens ein* ( $E$ ) Pfad beginnt, der die Pfadformel  $\psi_1 = \zeta_2 \ U \zeta_3$  erfüllt. Dies sind  $s_0, s_1, s_2, s_3$ .



Nun wird die Zustandsformel  $\zeta_5 = A X \zeta_4$  behandelt, also werden alle diejenigen Zustände mit  $\zeta_5$  markiert, deren *alle* ( $A$ ) Nachfolger ( $X$ ) bereits mit  $\zeta_4$  markiert sind. Dies sind  $s_0, s_2$ .



Schließlich ist  $\zeta_6 = \zeta_1 \wedge \zeta_5$ ; also werden alle Zustände mit  $\zeta_6$  markiert, die bereits mit  $\zeta_1$  und  $\zeta_5$  markiert sind. Dies ist nur noch  $s_0$ .

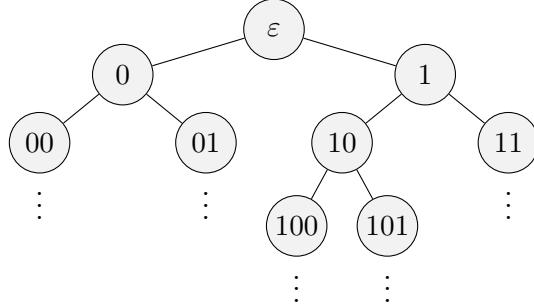


Da der einzige Startzustand  $s_0$  mit  $\zeta = \zeta_6$  markiert ist, gilt  $\mathcal{S} \models \zeta$ .

Der genaue Algorithmus kann z. B. in [BK08, Abschnitt 6] nachgelesen werden.

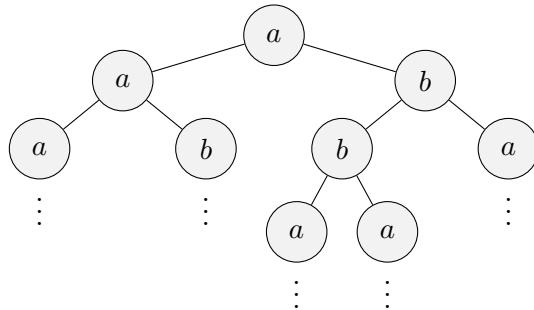
## T4.5 Veranschaulichung der Grundbegriffe

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, ...
- 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  **$\Sigma$ -Baum** z. B. folgender:



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

## T4.6 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  $1^\omega$  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:

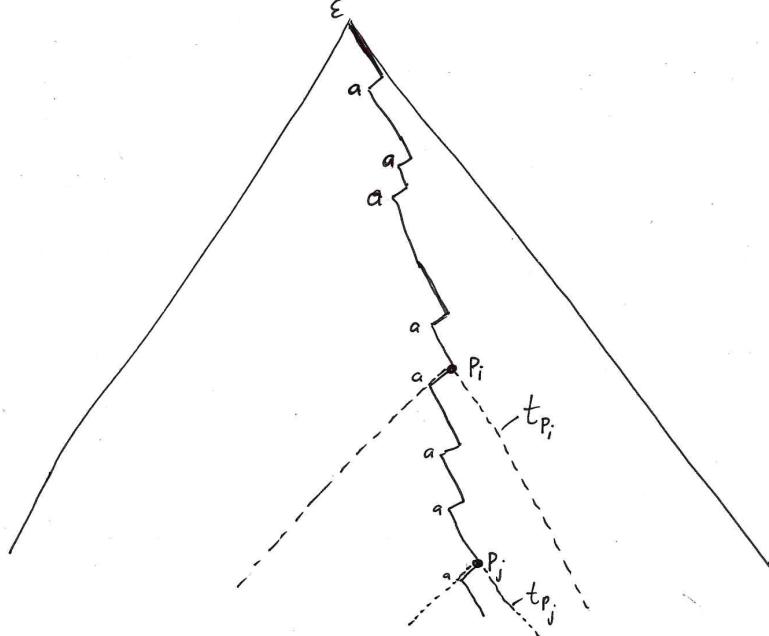
Es gibt  $m_0, m_1, \dots, m_n > 0$ , so dass  
 $r(1^{m_0}), r(1^{m_0}01^{m_1}), \dots, r(1^{m_0}01^{m_1}0 \dots 01^{m_n}) \in F$ .

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<sup>1</sup> und erhalten wieder einen erfolgreichen Run:

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

Aussage (\*) gilt, weil (a) wegen  $r(p_i) = r(p_j)$  weiterhin die Übergangsrelation  $\Delta$  respektiert wird und (b) weiterhin auf allen Pfaden ein akzeptierender 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.  $\square$

---

<sup>1</sup>Wie in Kapitel 2 bezeichnet  $t_p$  den Teilbaum des Baums  $t$ , dessen Wurzel  $p$  ist.

## T4.7 Beispiel für die Konstruktion „NMBA $\Rightarrow$ NPBA“

Seien  $Q = \{1, 2, 3\}$  und  $F = \{1, 2\}$ . Wir betrachten den Run  $r = 12131211\dots$  von  $\mathcal{A}$  auf einem Pfad  $\pi$ , 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  $\langle q_1 q_2 q_3, \ell \rangle$  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  $\langle q_1 q_2 q_3, \ell \rangle$  an den Positionen  $q_2, q_3$  immer die akzeptierenden Zustände 1, 2, und ab Position 7 ist  $\ell > |Q| - |F| = 1$ .

## T4.8 Beweis der Hilfsaussage für „NMBA $\Rightarrow$ NPBA“

„ $\Rightarrow$ “ Wegen  $\text{Inf}(q_0 q_1 q_2 \dots) = S$  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  $\text{perm}_i$  gerückt
- (ii') Bis  $m_2$  wurde jeder Zustand aus  $S$  mindestens einmal in die Endposition von  $\text{perm}_i$  gerückt.

Wegen (ii') müssen die letzten  $k$  Positionen von  $\text{perm}_{m_2}$  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  $\text{perm}_i$  gerückt; also ist  $\ell_i > n - k$  (was Teil ① der Hilfsaussage beweist), und die letzten  $k$  Positionen in  $\text{perm}_i$  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  $\text{perm}_i$  landet. Dies ist aber ein Widerspruch zu  $\text{Inf}(q_0 q_1 q_2 \dots) = S$ .

„ $\Leftarrow$ “ Kann mit einer ähnlichen Argumentation, die die Konstruktion analysiert, bewiesen werden.  $\square$

## T4.9 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.  $\text{Inf}(r, \pi) = F$  für alle Pfade  $\pi$  von  $t$ .

Sei  $s_0 = \langle t_1 \cdots t_{n-1} q_0, 1 \rangle \in I'$ , und sei  $s$  der gemäß  $\Delta'$  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  $\langle \text{perm}_p, \ell_p \rangle$ , wobei  $\text{perm}_p$  auf  $r(p)$  endet.

Wir müssen noch zeigen, dass  $s$  erfolgreich ist. Dazu betrachten wir einen beliebigen Pfad  $\pi$  und die zugehörige Zustandsfolge  $s_0 s_1 s_2 \dots$  von  $s$  mit  $s_i = \langle \text{perm}_i, \ell_i \rangle$ . Da  $r$  erfolgreich ist, gilt  $\text{Inf}(r, \pi) = F$ . 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  $n - |F| + 1, \dots, n$  in  $\text{perm}_i$  ist  $F$ .

Wegen ② muss  $\text{Inf}(s, \pi)$  einen Zustand der Form  $\langle q_1 \cdots q_n, n - |F| + 1 \rangle$  mit  $\{q_{n-|F|+1}, \dots, q_n\} = F$  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  $\pi$  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  $s(p) = \langle \text{perm}_p, \ell_p \rangle$  und  $r(p)$  das letzte Element von  $\text{perm}_p$ , für alle  $p \in \{0, 1\}^*$ . Nach Definition von  $I'$  und  $\Delta'$  ist dann

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

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

- (i)  $\langle q_1 \cdots q_n, \ell \rangle$  unendlich oft in  $s$  auftritt, aber
- (ii) *kein*  $\langle \cdot, \ell' \rangle$  mit  $\ell' < \ell$  unendlich oft auftritt.

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

## T4.10 Beispiel eines Spielverlaufs für $G_{\mathcal{A},t}$

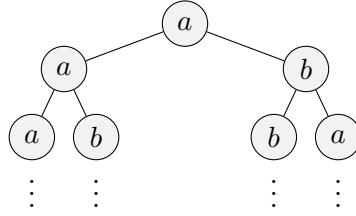
Sei  $\mathcal{A} = (\{q_1, q_2\}, \{a, b\}, \Delta, \{q_1\}, c)$  ein NPBA mit

$$\Delta = \{(q_1, a, q_1, q_2), (q_2, a, q_2, q_1), (q_1, b, q_2, q_1), (q_2, b, q_1, q_2)\} \quad \text{und}$$

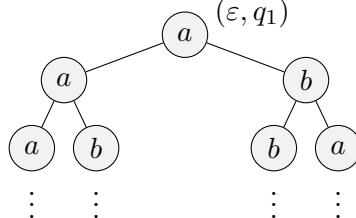
$$c : q_1 \mapsto 0, q_2 \mapsto 1.$$

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  $F = \{q_1\}$ .

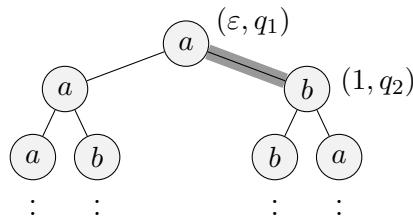
Wir betrachten folgenden Eingabebaum  $t$ .



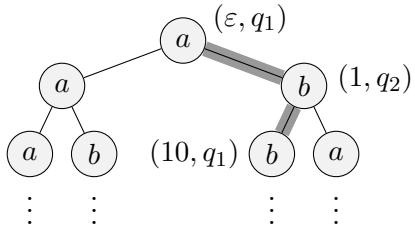
In der anfänglichen Spielposition ist nur die Wurzel mit  $q_1$  markiert, also ist der bisherige Spielverlauf die Folge, die nur aus dem Paar  $(\varepsilon, q_1)$  besteht. Zur Veranschaulichung wird der Spielverlauf im Folgenden an den von **PF** gewählten Pfad im Baum geschrieben, also zu Beginn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

### T4.11 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  $\Delta$  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 4.18 und Folgerung 4.19 zeigen, dass dies nicht an der Einfachheit von  $\mathcal{A}$  liegt.

### T4.12 Beweis der Rückrichtung von Lemma 4.17

**Behauptung.** **Aut** hat Gewinnstrategie in  $G_{\mathcal{A}, t}$  ab Position  $(\varepsilon, q_I) \Rightarrow t \in L_\omega(\mathcal{A})$

**Beweis.** Habe **Aut** eine Gewinnstrategie in  $G_{\mathcal{A}, t}$  ab Position  $(\varepsilon, q_I)$ . 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  $(r(p), t(p), q_0, q_1)$ , die durch die Gewinnstrategie der Position  $(p, r(p))$  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, was ein Widerspruch zur Annahme wäre, dass **Aut** eine Gewinnstrategie hat.  $\square$

## T4.13 Beispiel für die Sprache $L_{s,t}$

Ab jetzt bezeichnen wir Pfade im Baum als Wörter  $\pi \in \{0,1\}^*$ ; beispielsweise steht  $\pi = 0110 \dots$  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  $f_p : \Delta \rightarrow \{0,1\}$  enthält) und  $\pi = 0110 \dots$ . Dann wird durch  $\pi$  folgendes  $\omega$ -Wort  $\alpha \in L_{s,t}$  bestimmt:

$$\alpha = \begin{array}{cccccc} & f_\varepsilon & f_0 & f_{01} & f_{011} & f_{0110} & \dots \\ & t(\varepsilon) & t(0) & t(01) & t(011) & t(0110) & \\ & 0 & 1 & 1 & 0 & 0 & \end{array}$$

Dabei bezeichnet jede Box ein Symbol von  $\alpha$ , also gemäß Folie 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  $\alpha$  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  $\alpha$  gibt den Inhalt von  $s$  auf dem Pfad  $\pi$  wieder und die Folge aller zweiten Komponenten den Inhalt von  $t$  auf  $\pi$ . Die Folge der dritten Komponenten ist  $\pi$  selbst.

## T4.14 Beweis von Lemma 4.22

**Lemma 4.22.**  $s$  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  $\pi$ , so dass das  $\Sigma'$ -Wort

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

von  $\mathcal{A}'$  akzeptiert wird. Sei  $r = q_0 q_1 q_2 \dots$  ein erfolgreicher Run von  $\mathcal{A}'$  auf  $\alpha$ . Dann gibt es für jede Position  $j \geq 0$  in  $\alpha$  eine Transition

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

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

$$\delta_j = (q_j, t(\pi_1 \dots \pi_j), q'_0, q'_1) \in \Delta$$

mit  $s(\pi_1 \dots \pi_j)(\delta_j) = \pi_{j+1}$ , wobei  $s(\pi_1 \dots \pi_j)$  die Funktion  $f_{\pi_1 \dots \pi_j} \in F$  ist.

Wir betrachten nun denjenigen Spielverlauf von  $G_{\mathcal{A},t}$ , in dem **Aut** in Spielposition  $\pi_1 \dots \pi_j$  jeweils  $\delta_j$  wählt und daraufhin **PF** mit  $s(\pi_1 \dots \pi_j) = \pi_{j+1}$  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  $(\varepsilon, q_I)$ , 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

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

- **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  $r' = q_0 q_1 q_2 \cdots$  der während des Spielverlaufs besuchten Zustände. Nach Konstruktion von  $\Delta'$  ist  $r'$  ein Run von  $\mathcal{A}'$  auf dem zugehörigen  $\omega$ -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**-Gewinnabaum.  $\square$

## T4.15 Beweis von Lemma 4.23

**Lemma 4.23.**  $t \in L_\omega(\mathcal{B})$  gdw. es gibt  $F$ -Baum  $s$  mit  $L_{s,t} \subseteq L_\omega(\mathcal{A}'')$

**Beweis.**

„ $\Rightarrow$ “ 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  $\Delta^{\text{neu}}$  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  $\Delta^{\text{neu}}$  folgt, dass es für jedes  $p$  eine Funktion  $f_p \in F$  gibt mit

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

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

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

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

$$\alpha_\pi = \langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3 \rangle \dots \quad (*)$$

aus  $L_{s,t}$ . 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  $\alpha_\pi$  ist.

„ $\Leftarrow$ “ Seien  $t, s$  so, dass  $L_{s,t} \subseteq L_\omega(\mathcal{A}'')$ . 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  $L_{s,t} \subseteq L_\omega(\mathcal{A}'')$  gibt es für jeden Pfad  $\pi$  einen erfolgreichen Run von  $\mathcal{A}''$  auf dem Wort  $\alpha_\pi$ , das gemäß (\*) definiert ist. Da  $\mathcal{A}''$  deterministisch ist, sind seine Runs auf Wörtern aus  $L_{s,t}$  eindeutig bestimmt. Insbesondere gilt: wenn zwei Pfade  $\pi, \pi'$  dasselbe Präfix  $\pi_1 \dots \pi_k = \pi'_1 \dots \pi'_k$  haben, dann weisen die beiden eindeutig bestimmten Runs von  $\mathcal{A}''$  den ersten  $k$  Positionen der zugehörigen Wörter aus  $L_{s,t}$  dieselben Zustände zu. Das heißt, dass es für jede Position  $p$  einen *eindeutig bestimmten* Zustand  $q$  gibt, so dass für alle Pfade  $\pi$  mit  $p = \pi_1 \dots \pi_k$  der Run von  $\mathcal{A}''$  „auf  $\pi$ “ 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  $\pi$  in  $r$  dem Run  $r''$  von  $\mathcal{A}''$  auf dem zugehörigen Wort  $\alpha_\pi \in L_{s,t}$ . Da  $r''$  erfolgreich ist (siehe oben) und  $\mathcal{B}$  die Akzeptanzkomponente  $c''$  von  $\mathcal{A}''$  übernimmt, ist auch  $r$  erfolgreich; somit ist  $t \in L_\omega(\mathcal{B})$ .  $\square$

**Teil V.**

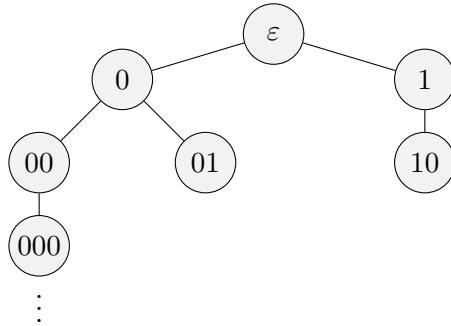
**Alternierung**

## T5.1 Beispiel Semantik positiver Boolescher Formeln

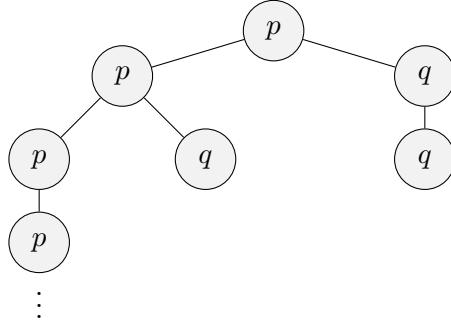
Wir betrachten die PBF  $\varphi = (p \wedge q) \vee (r \wedge s)$  über der Variablenmenge  $X = \{p, q, r, s\}$  und die Belegungen  $Y_1 = \{p, q, r\}$  und  $Y_2 = \{p, r\}$ . Dann gilt  $Y_1 \models \varphi$ , aber  $Y_2 \not\models \varphi$ .

## T5.2 Beispiel eines Run-Baums

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



Ein möglicher Pfad ist  $\pi = \varepsilon \cdot 1 \cdot 10$ ; dieser Pfad ist endlich. Wenn  $\Sigma = \{p, q\}$ , dann ist ein  $\Sigma$ -Baum z. B. folgender:



## T5.3 Beispiel eines alternierenden Büchi-Automaten

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

$$Q = \{p, q\}, \quad \Sigma = \{a, b\}, \quad F = \{p\}, \quad q_I = p, \quad (1)$$

$$\delta(p, a) = p \wedge q \quad (1)$$

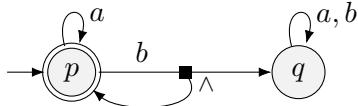
$$\delta(p, b) = p \quad (2)$$

$$\delta(q, a) = q \quad (3)$$

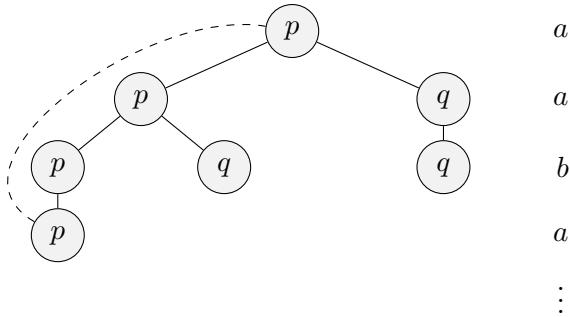
$$\delta(q, b) = 1. \quad (4)$$

Dabei bedeutet Übergang (1) von  $\delta$  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  $\delta(\cdot, \cdot) = 0$  einem erfolglosen 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  $\alpha$  ist der Baum aus dem vorigen Beispiel, wenn man ihn ab Position 000 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  $\alpha$  stehen rechts daneben.



Dieser Run ist erfolgreich, denn der einzige unendliche Pfad ist  $\pi = \varepsilon \cdot 0 \cdot 00 \cdot 000 \dots$ , 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  $\alpha$  von  $\mathcal{A}$  akzeptiert wird, betrachten wir den „minimalen“ Run  $(P, t)$  von  $\mathcal{A}$  auf  $\alpha$ , also denjenigen Run, der bei Übergängen (2) und (3) nur ein Kind mit  $p$  bzw.  $q$  enthält und bei Übergang (4) keine Kinder.<sup>2</sup> Jeder Pfad  $\pi$  dieses Runs kann wegen  $\delta(q, b) = 1$  nicht in einer Folge  $q^\omega$  enden. Da außerdem  $\delta$  keinen Übergang von  $q$  nach  $p$  erlaubt, kann  $\pi$  also nur entweder endlich sein oder aus nur  $p$ 's bestehen. Im letzteren Fall ist die Akzeptanzbedingung erfüllt.

---

<sup>2</sup>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: enthält  $\alpha \in \{a, b\}^\omega$  endlich oft  $b$ . Dann gibt es eine Position  $m$ , ab der  $\alpha$  nur noch aus  $a$ 's besteht. Sei  $(P, t)$  ein beliebiger Run von  $\mathcal{A}$  auf  $\alpha$ . Wir müssen zeigen, dass dieser einen unendlichen Pfad  $\pi$  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  $\pi$  gefunden.

## T5.4 Beispiele für die Übersetzung LTL → ABA

**Beispiel 1.** Sei  $\varphi = a \mathbin{U} b$ .

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

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

Die Übergangsfunktion  $\delta$  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
$q_{a \mathbin{U} b}$		0	$q_{a \mathbin{U} b}$	1	1
$q_{\neg(a \mathbin{U} b)}$		1	$q_{\neg(a \mathbin{U} b)}$	0	0

Die erste Zeile ( $q_a$ ) ist durch die erste Gleichung für  $\delta$  auf Folie 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 ( $q_{a \mathbin{U} b}$ ) betrachten wir die Spalte  $\{a\}$ . Die letzte Gleichung für  $\delta$  auf Folie 15 lautet

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

also hier

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

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

$$\delta(q_{aUb}, \{a\}) = 0 \vee (1 \wedge q_{aUb})$$

und mit den standardmäßigen Äquivalenzen der Aussagenlogik wird daraus

$$\delta(q_{aUb}, \{a\}) = q_{aUb}.$$

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  $\delta$  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  $a U b$ , also muss  $\mathcal{A}_\varphi$  auch  $\alpha$  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  $q_{aUb}$  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  $q_{aUb}$ . Im Gegensatz dazu wird  $\{a\}^\omega$  nicht akzeptiert, denn hier besteht der einzige mögliche Run aus einem *unendlichen* mit  $q_{aUb}$  markierten Pfad, welcher die Akzeptanzbedingung  $F = \{q_{\neg(aUb)}\}$  verletzt und genau der zu vermeidenden Situation entspricht, dass  $\neg(a Ub)$  unendlich lange hinausgezögert wird.

**Beispiel 2.** Sei  $\varphi = GFa$ . Die Standardvorgehensweise gemäß der Folien ist, die Formel so umzuformen, dass sie nur noch  $U$  verwendet (siehe Folie 12 in Teil 5). 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 15 um die offensichtlichen Fälle für die Operatoren  $G$  und  $F$  zu erweitern, und zwar in der Übergangsfunktion und der Akzeptanzbedingung

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

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

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

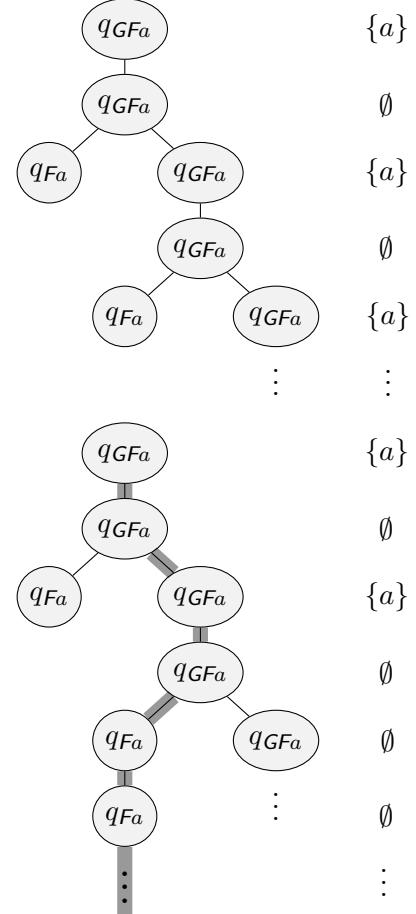
$$\begin{aligned}Q &= \{q_a, q_{\neg a}, q_{Fa}, q_{\neg Fa}, q_{GFa}, q_{\neg GFa}\} \\ \Sigma &= \{\emptyset, \{a\}\} \\ q_I &= q_{GFa} \\ F &= \{q_{\neg Fa}, q_{GFa}\}\end{aligned}$$

Die Übergangsfunktion  $\delta$  ist durch die folgende Tabelle gegeben.

Zustand	Zeichen	$\emptyset$	$\{a\}$
$q_a$		0	1
$q_{\neg a}$		1	0
$q_{Fa}$		$q_{Fa}$	1
$q_{\neg Fa}$		$q_{\neg Fa}$	0
$q_{GFa}$		$q_{Fa} \wedge q_{GFa}$	$q_{GFa}$
$q_{\neg GFa}$		$q_{Fa} \vee q_{\neg GFa}$	$q_{\neg GFa}$

Die ersten 2 Zeilen werden wieder wie im vorigen Beispiel erzeugt; die dritte und vierte Zeile sind analog zu den Fällen  $q_{aUb}$  bzw.  $q_{\neg(aUb)}$ . Die letzten beiden Zeilen bringen „echte“ Alternierung; Zeile 5 sie kann man direkt aus Gleichung (\*) für  $\delta$  und der 3. Tabellenzeile ablesen; Zeile 6 erhält man wieder durch „Negation“ (Fall  $q_{\sim\psi}$  auf Folie 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  $GFa$ , also muss  $\mathcal{A}_\varphi$  auch  $\alpha$  akzeptieren. Letzteres ist tatsächlich der Fall, und zwar mittels des nebenstehenden Runs, dessen einziger unendlicher Pfad nur mit  $q_{GFa}$  markiert ist, also die Akzeptanzbedingung  $F$  erfüllt.



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

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

## T5.5 Beispiel für die Komplementierung

Wir betrachten den Automaten aus T5.3, aber mit einer Müller-Akzeptanzbedingung, also den AMA  $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, \mathcal{F})$  mit

$$\begin{aligned} Q &= \{p, q\}, \quad \Sigma = \{a, b\}, \quad \mathcal{F} = \{\{p\}, \{p, q\}\}, \quad q_I = p, \\ \delta(p, a) &= p \wedge q \\ \delta(p, b) &= p \\ \delta(q, a) &= q \\ \delta(q, b) &= 1, \end{aligned}$$

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

Die Regeln zur Komplementierung auf Folie 19 liefern den AMA  $\mathcal{A}' = (Q, \Sigma, \delta', \{q_I\}, \mathcal{F}')$  mit

$$\begin{aligned} \mathcal{F}' &= \{\emptyset, \{q\}\} \\ \delta'(p, a) &= p \vee q \\ \delta'(p, b) &= p \\ \delta'(q, a) &= q \\ \delta'(q, b) &= 0. \end{aligned}$$

Da in keinem  $\delta(\cdot, \cdot)$  mehr eine Konjunktion auftritt, verhält sich  $\mathcal{A}'$  genauso wie ein NMA, der im Fall  $(p, a)$  (1. Zeile von  $\delta$ ) zwei mögliche Entscheidungen treffen kann und im Fall  $(q, b)$  „blockiert“, also keinen Übergang hat. Das heißt, man überzeugt sich also leicht, dass  $\mathcal{A}'$  äquivalent zum NMA  $\mathcal{A}'' = (Q, \Sigma, \Delta, \{q_I\}, \mathcal{F}')$  mit  $\Delta = \{(p, a, p), (p, a, q), (p, b, p), (q, a, q)\}$  ist und dass  $L_\omega(\mathcal{A}'') = \bar{L}_\omega(\mathcal{A})$  gilt.

## Beweisskizze für die Komplementierung (fakultativ)

Wir haben für die Komplementierung von AMAs (Satz 5.5 auf Folie 19) nicht die Korrektheit bewiesen. Falls es euch interessiert, wie man vorgehen muss, ist hier eine grobe Skizze – allerdings für den Fall von Paritätsautomaten (aber wir wissen ja bereits, wie man die Muller- und Paritätsbedingung ineinander übersetzt).

1. Wir betrachten eine „vereinfachte“ Variante der Alternierung, bei der die Zustandsmenge  $Q$  eines APA partitioniert ist in die Menge  $Q_{\exists}$  der *existentiellen Zustände* und die Menge  $Q_{\forall}$  der *universellen Zustände*. Für Zustände  $q \in Q_{\exists}$  ist  $\delta(q, a)$  für jedes Zeichen  $a$  eine Disjunktion von Zuständen; für Zustände  $q \in Q_{\forall}$  ist  $\delta(q, a)$  eine Konjunktion. Diese Variante entspricht der ursprünglichen Definition von Alternierung. Wir nennen solche Automaten hier  $A^-PAs$ .
2. Offensichtlich ist jeder  $A^-PA$  auch ein APA. Die Umkehrung gilt nicht; dennoch sind  $A^-PAs$  und APAs gleichmächtig: zum einen kann man jeden APA in einen

äquivalenten NPA umwandeln (dieses nichttriviale Resultat basiert auf Spielen und hat dieses Jahr nicht mehr in die Vorlesung gepasst); zum anderen ist natürlich jeder NPA ein A<sup>-</sup>PA.

3. Wenn man die „Konstruktion“ zur Komplementierung auf Folie 19 auf einen A<sup>-</sup>PA anwendet, dann werden dadurch lediglich die Mengen  $Q_{\exists}$  und  $Q_{\forall}$  miteinander vertauscht (und die Akzeptanzkomponente komplementiert). Sei  $\mathcal{A}'$  der resultierende A<sup>-</sup>PA.
4. Nun können wir wieder Spiele ähnlich zu denen aus Teil 4 betrachten: für jeden A<sup>-</sup>PA  $\mathcal{A}$  und jedes Eingabewort  $\alpha$  ist das Spiel  $G_{\mathcal{A}, \alpha}$  wie folgt definiert:
  - Die Spielpositionen sind Paare  $(q_i, \alpha_i)$  mit  $q_i \in Q$  und  $\alpha_i$  der  $i$ -te Buchstabe von  $\alpha$ .
  - Die Startposition ist  $(q_0, \alpha_0)$  mit  $q_0 = q_I$  (Startzustand von  $\mathcal{A}$ ).
  - Wenn  $q_i \in Q_{\exists}$ , dann wählt **Aut** einen Nachfolgerzustand  $q_{i+1}$ ; wenn  $q_i \in Q_{\forall}$ , dann wählt **PF** ihn.
  - **Aut** gewinnt, wenn (a) das Spiel unendlich lange läuft und die Akzeptanzbedingung von  $\mathcal{A}$  erfüllt oder (b) das Spiel endet, weil **PF** keinen Zug machen kann (d. h. in Position  $(q_i, \alpha_i)$  ist  $q_i \in Q_{\forall}$  und  $\delta(q_i, \alpha_i)$  ist die leere Konjunktion).

Ähnlich wie in Teil 4 ist nun wieder einfach zu sehen, dass  $\alpha \in L_{\omega}(\mathcal{A})$  genau dann, wenn **Aut** eine Gewinnstrategie in  $G_{\mathcal{A}, \alpha}$  hat.

Da es kein Unentschieden gibt, gilt auch:

$$\alpha \notin L_{\omega}(\mathcal{A}) \quad \text{gdw.} \quad \mathbf{PF} \text{ hat Gewinnstrategie in } G_{\mathcal{A}, \alpha} \quad (*)$$

Gedächtnislose Gewinnstrategien werden hier allerdings nicht benötigt.

5. Da sich durch die Konstruktion in Punkt 3 die Rollen von **Aut** und **PF** genau vertauschen, erhält man mit (\*):

$$\begin{aligned} \alpha \notin L_{\omega}(\mathcal{A}) &\quad \text{gdw.} \quad \mathbf{PF} \text{ hat Gewinnstrategie in } G_{\mathcal{A}, \alpha} \\ &\quad \text{gdw.} \quad \mathbf{Aut} \text{ hat Gewinnstrategie in } G_{\mathcal{A}', \alpha} \\ &\quad \text{gdw.} \quad \alpha \in L_{\omega}(\mathcal{A}') \end{aligned}$$

# Literaturverzeichnis

- [BK08] Baier, Christel und Joost-Pieter Katoen: *Principles of model checking*. MIT Press, 2008.