Tafelmitschriften zur Vorlesung „Automatentheorie und ihre Anwendungen“ im Wintersemester 2018/19

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

Stand: 12. Januar 2019

Dieses Dokument ist noch unvollständig und wird regelmäßig aktualisiert.

Inhaltsverzeichnis

Teil I.

Endliche Automaten auf endlichen Wörtern

Teil II.

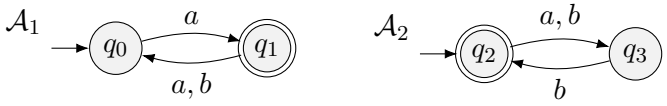
Endliche Automaten auf endlichen Bäumen

Teil III.

Endliche Automaten auf unendlichen Wörtern

T3.1 Produktkonstruktion für NEAs ist für NBAs nicht korrekt

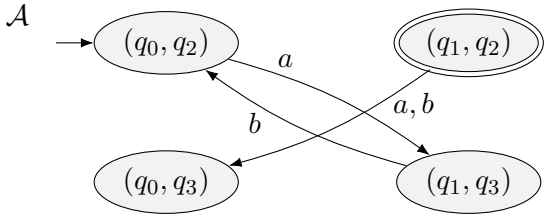
Sei $\Sigma = \{a, b\}$. Wir betrachten folgende **NEAs** $\mathcal{A}_1, \mathcal{A}_2$.



Dann gilt:

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

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



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

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

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

also ist jetzt

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

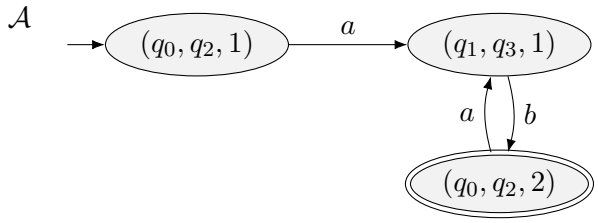
aber nach wie vor ist

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

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

T3.2 Produktkonstruktion für NBAs: Beispiel und Korrektheit

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



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

Korrektheitsbeweis. Zu zeigen ist:

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

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

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

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

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

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

also enthält r auch unendlich viele Zustände der Form

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

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

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

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

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

Wir betrachten die Folge

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

wobei die n_i induktiv wie folgt definiert sind:

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

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

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

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

„⊆“ 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 α, d. h. $q_0 = q_I$ (der einzige Anfangszustand von $\mathcal{A}_2$), und q_I kommt unendlich oft in r vor (weil es auch der einzige akzeptierende Zustand von $\mathcal{A}_2$ ist).

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

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

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

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

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 $\alpha \in L(\mathcal{A}_1)^\omega$.

„⊇“ 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 i. Also gibt es für jedes $j \geq 0$ einen erfolgreichen Run

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

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

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

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

T3.4 Beweis Charakterisierung NBA-erkennbarer Sprachen

Satz 3.9. Eine Sprache $L \subseteq \Sigma^\omega$ ist Büchi-erkennbar genau dann, wenn es reguläre Sprachen $V_1, W_1, \ldots, V_n, W_n$ gibt mit $n \geqslant 1$ und

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

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

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

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

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

❑

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

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

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

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

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

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

T3.6 Beweis Charakterisierung DBA-erkennbarer Sprachen

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

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

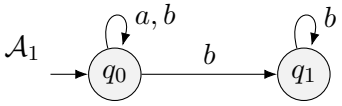
$$L_\omega(\mathcal{A}) = \overrightarrow{L(\mathcal{A})}$$

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

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

T3.7 Beispiele für Muller-Automaten

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



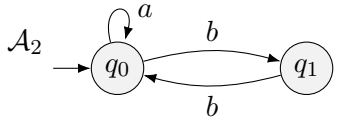
  (M1) Wenn $\mathcal{F}_1 = \{\{q_0\}\}$, dann $L_\omega(\mathcal{A}_1) = \Sigma^\omega$.

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

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

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

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



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

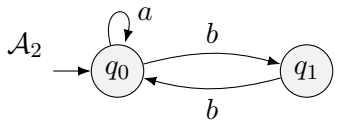
(M6) Wenn $\mathcal{F}_2 = \{\{q_1\}\}$, dann $L_\omega(\mathcal{A}_2) = \emptyset$ (denn wenn q_1 unendlich oft besucht wird, dann auch q_0).

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

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

T3.8 Beispiele für Rabin-Automaten

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



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

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

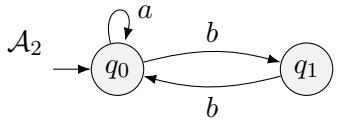
(R3) Wenn $\mathcal{P} = \{(\emptyset, \{q_1\})\}$, dann $L_\omega(\mathcal{A}) = L((a^*bb)^\omega)$ (dasselbe Akzeptanzverhalten wie NBA mit $\mathcal{F} = \{q_1\}$).

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

- Der Fall mehrerer Paare in der Akzeptanzkomponente $\mathcal{P}$ braucht nicht gesondert illustriert zu werden, denn für *alle* NRAs $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$ mit $\mathcal{P} = \bigcup_{i \leq n} \mathcal{P}_i$ 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 *nicht* ∞ oft vorkommen, also muss q_0 ∞ oft vorkommen $\rightsquigarrow$ wie M5).

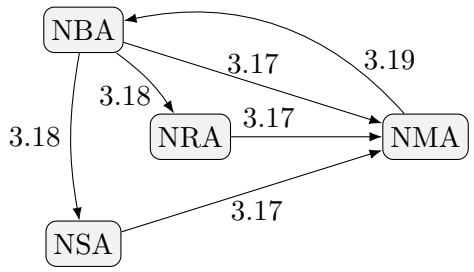
(S5) Wenn $\mathcal{P} = \{(\emptyset, \{q_0\})\}$, dann $L_\omega(\mathcal{A}) = \emptyset$ (denn q_0 kommt in jedem Run ∞ oft vor).

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

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

T3.10 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 $r = q_0 q_1 q_2 \cdots$ ein erfolgreicher Run von $\mathcal{A}$ (NMA!) auf α, also $q_0 \in I$ und $\mathsf{Inf}(r) = F$. Dann gibt es eine Position $i_0 \ge 1$, ab der nur noch akzeptierende Zustände auftreten, d. h. $q_i \in F$ für alle $i \ge i_0$. Daraus konstruieren wir wie folgt induktiv eine Folge $s = s_0 s_1 s_2 \cdots$ von Zuständen von $\mathcal{A}'$:

- Für alle $i < i_0$ setze $s_i = q_i$.
- $s_{i_0} = \langle q_{i_0}, \{q_{i_0}\}\rangle$
- 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 α (das ist leicht schrittweise anhand der Konstruktion von Δ' und von s nachvollziehbar).
- $s_0 \in I$ (nach Definition I').
- s ist erfolgreich, denn wegen $\mathsf{Inf}(r) = F$ gibt es ∞ viele Vorkommen von Zuständen der Form $\langle q_f, F\rangle$ in S.

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

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

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

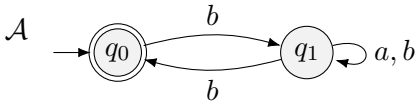
Diese Konstruktion stellt sicher:

- r ist Run von $\mathcal{A}$ auf α (vgl. Konstruktion von Δ').
- $r_0 \in I$.
- r enthält nur endlich viele Zustände außerhalb F (weil in „Phase 2“ nur noch Zustände aus F vorkommen).
- Jeder Zustand aus F kommt in r unendlich oft vor (weil $\langle q_f, F\rangle$ unendlich oft in s vorkommt).

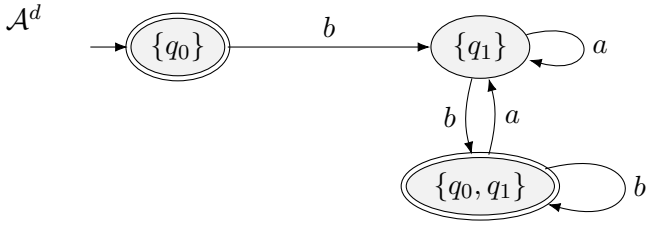
Damit ist r ein erfolgreicher Run des NMA $\mathcal{A}$ auf α, also $\alpha \in L_\omega(\mathcal{A})$. ❑

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 $bab, babab, bababab, \ldots$ von $(ba)^\omega$ das entsprechende Präfix von r zwar den akzeptierenden Zustand $\{q_0, q_1\}$ erreicht, aber der zugehörige in q_0 endende Teilrun von $\mathcal{A}$ nicht mehr zu einem erfolgreichen Run auf α fortgesetzt werden kann.

T3.13 Variation der Akzeptanzbedingung im vorigen Beispiel

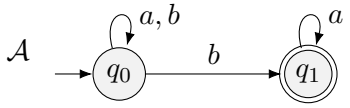
Wir betrachten dieselben Automaten $\mathcal{A}, \mathcal{A}^d$ wie im vorigen Beispiel. Man könnte versuchen, durch Variation der Akzeptanzbedingung von $\mathcal{A}^d$ einen zu $\mathcal{A}$ äquivalenten deterministischen Muller-, Rabin- oder Streett-Automaten zu erhalten, ohne die eigentliche Potenzmengenkonstruktion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

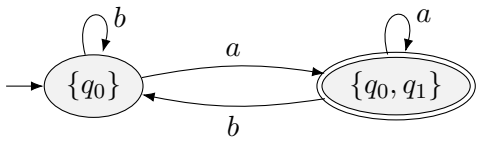
Im 1. und 2. Fall gibt es jedoch Bad Runs auf $(ba)^\omega$ bzw. ba^ω; im 3. Fall gibt es Wörter, die von $\mathcal{A}$ akzeptiert werden, aber nicht von $\mathcal{A}^d$, z. B. $(bba)^\omega$.

T3.14 Beispiel für Safras Trick 1

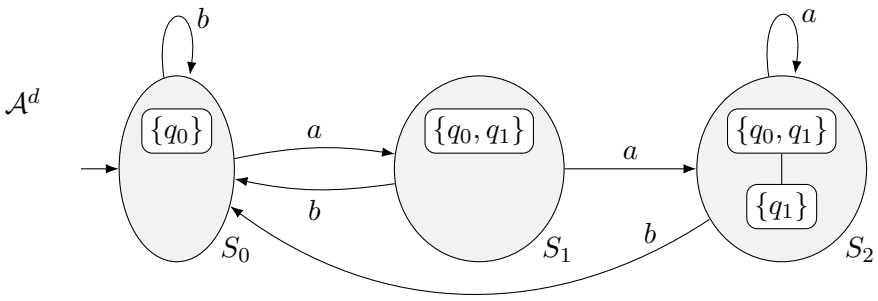
Betrachte folgenden NBA $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b\}$.



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



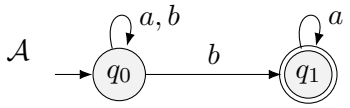
mit einem Bad Run auf $(ab)^\omega$. Mittels Safras Trick 1 erhält man hingegen folgenden deterministischen Automaten $\mathcal{A}^d$.



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

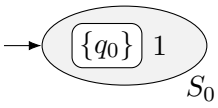
T3.15 Beispiel für die gesamte Safra-Konstruktion

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

Startzustand ist der folgende Safrabaum:

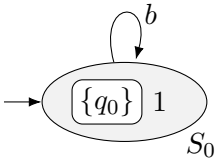


Folgezustand von S_0 mit b

- Schritt 1 der Safra-Konstruktion ist nicht anwendbar, da Knoten 1 nicht markiert ist.
- Schritt 2 ist nicht anwendbar, da Knoten 1 keine akzeptierenden Zustände enthält ($F = \{q_1\}$, siehe Bild).

- In Schritt 3 ändert sich der Makrozustand von Knoten 1 nicht, da der einzige b-Nachfolgezustand von q_0 wieder q_0 ist.
- Schritte 4–6 sind nicht anwendbar, da Knoten 1 noch keine Kinder hat.

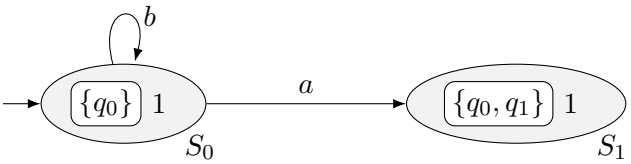
Also ist der b-Folgezustand von S_0 wieder S_0:



Folgezustand von S_0 mit a

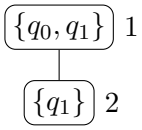
- Schritte 1–2 sind nicht anwendbar, siehe oben.
- In Schritt 3 ändert sich der Makrozustand von Knoten 1 zu $\{q_0, q_1\}$, da sowohl q_0 als auch q_1 von q_0 aus mit a erreicht werden können.
- Schritte 4–6 sind nicht anwendbar, da Knoten 1 noch keine Kinder hat.

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



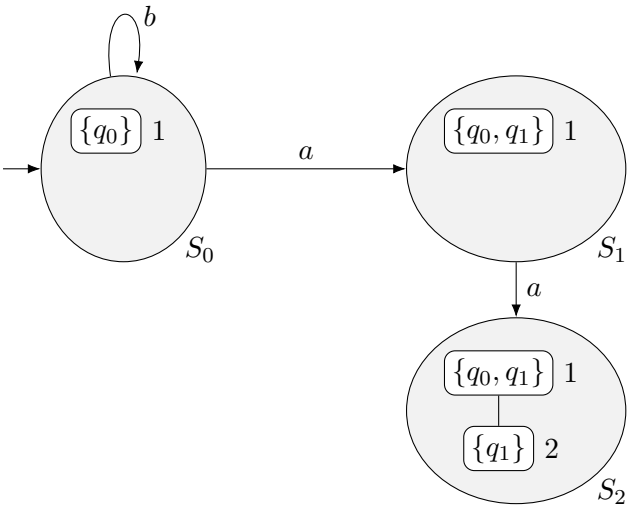
Folgezustand von S_1 mit a

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



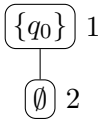
- In Schritt 3 wird auf beide Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit a ändert sich der Inhalt beider Makrozustände nicht.
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- Schritt 5 ist nicht anwendbar, da der Makrozustand von Knoten 2 nicht leer ist.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

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



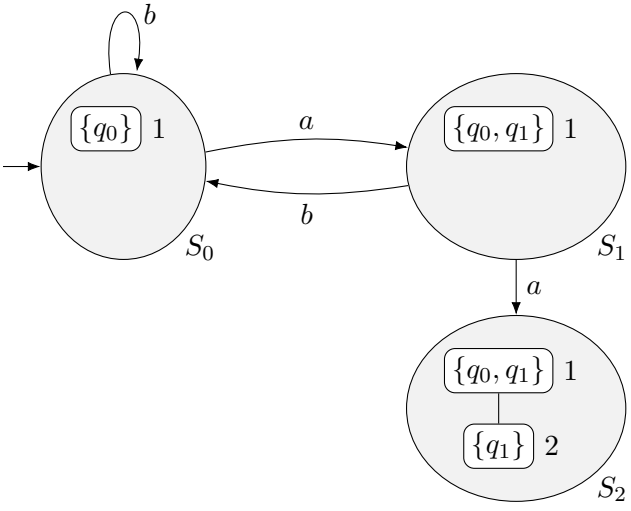
Folgezustand von S_1 mit b

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



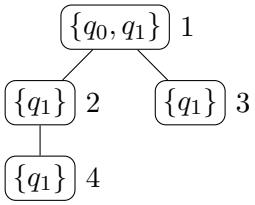
- Schritt 4 ist nicht anwendbar, da kein Knoten mehr als ein Kind hat.
- In Schritt 5 wird Knoten 2 gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

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

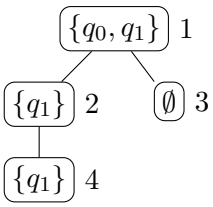


Folgezustand von S_2 mit a

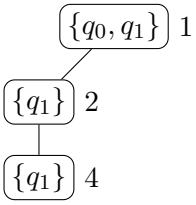
- Schritt 1 ist nach wie vor nicht anwendbar.
- In Schritt 2 wird je ein neues Kind von Knoten 1 und 2 erzeugt, da die Makrozustände beider Knoten akzeptierende Zustände enthalten:



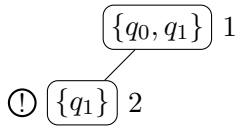
- In Schritt 3 wird auf alle Knoten die Potenzmengenkonstruktion angewendet; bei Übergang mit a ändert sich der Inhalt der Makrozustände nicht.
- In Schritt 4 wird q_1 aus Knoten 3 entfernt, da dieser Zustand im älteren Geschwister 2 enthalten ist:



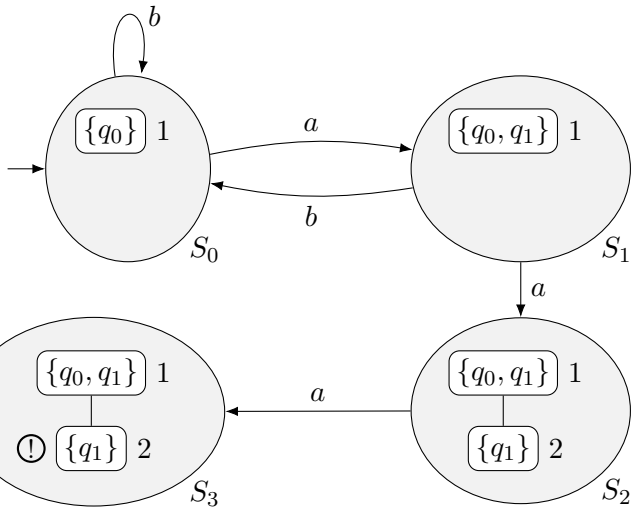
- In Schritt 5 wird Knoten 3 gelöscht:



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

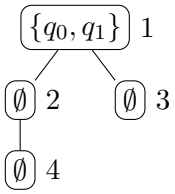


Der zuletzt abgebildete Safrabaum S_3 ist der a-Folgezustand von S_2:



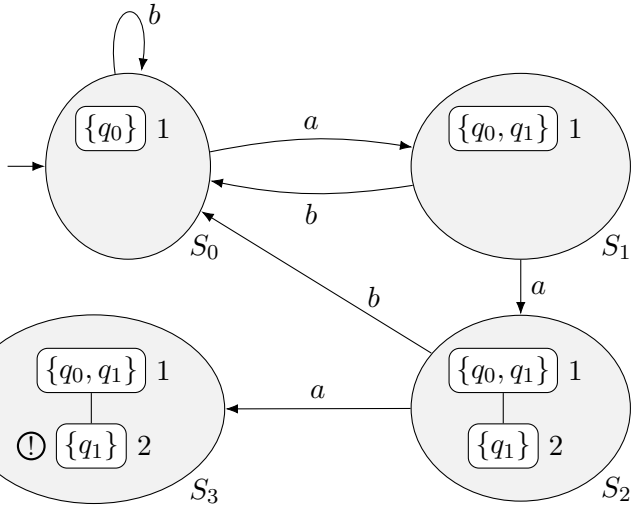
Folgezustand von S_2 mit b

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



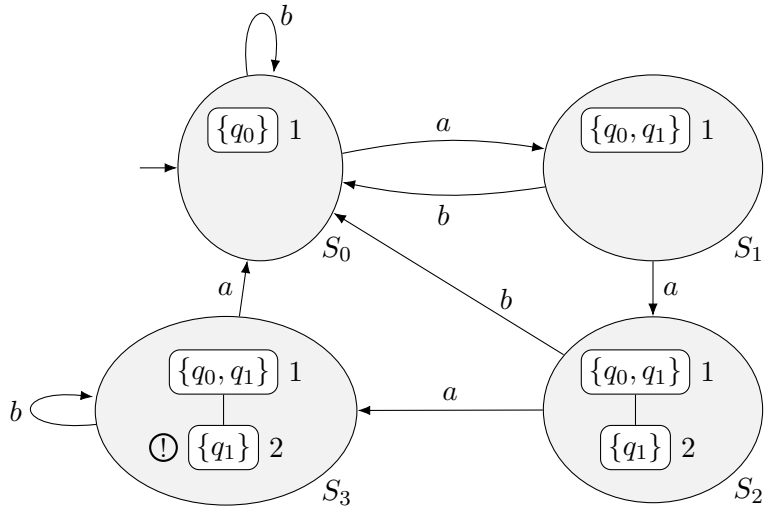
- Schritt 4 ist nicht anwendbar, da alle Makrozsutände außer dem von Knoten 1 leer sind.
- In Schritt 5 werden Knoten $2, 3, 4$ gelöscht.
- Schritt 6 ist nicht anwendbar, da Knoten 1 nun kein Kind mehr hat.

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.

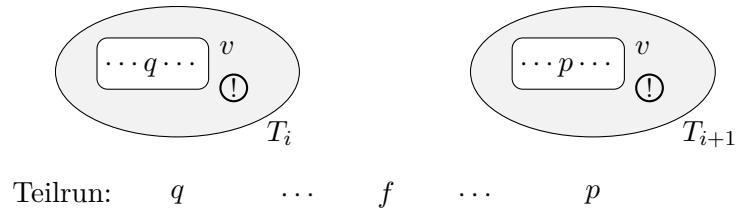
Akzeptanzkomponente

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 α, der einen akzeptierenden Zustand enthält.

Skizze:



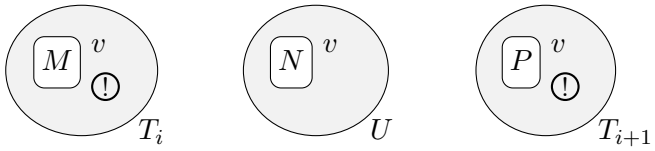
Beweis der HA. Damit v überhaupt mit ! markiert werden kann, muss v direkt vor der entsprechenden Anwendung des Schrittes 6 Kinder haben. Diese wurden in irgendeiner

vorigen Anwendung von Schritt 2 erzeugt. Damit dieser Schritt angewendet worden sein kann, muss gelten:

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

Wir betrachten nun den Teilrun $T_i \dots T_{i+1}$ von s. Sei U ein Safra-Baum zwischen T_i und T_{i+1}, so dass Knoten v in U Bedingung ($*$) erfüllt, und seien $T_i = S_k$, $U = S_\ell$ und $T_{i+1} = S_m$ für entsprechende k, ℓ, m mit $0 \leq k \leq \ell < m$ (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.

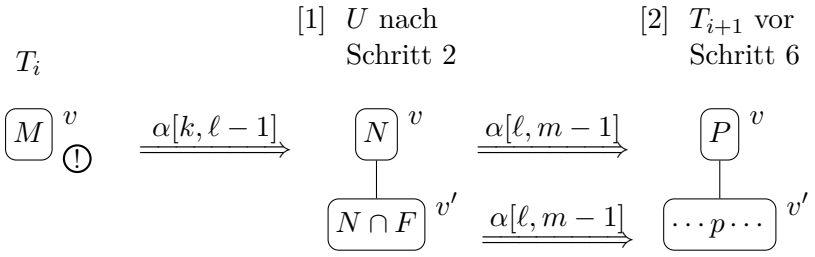
Um ($**$) zu beweisen, betrachten wir zunächst den Spezialfall, dass U der einzige Safra-Baum zwischen T_i und T_{i+1} ist, der ($*$) erfüllt. Im nächsten Schritt argumentieren wir dann für den allgemeinen Fall.

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

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

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

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



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

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

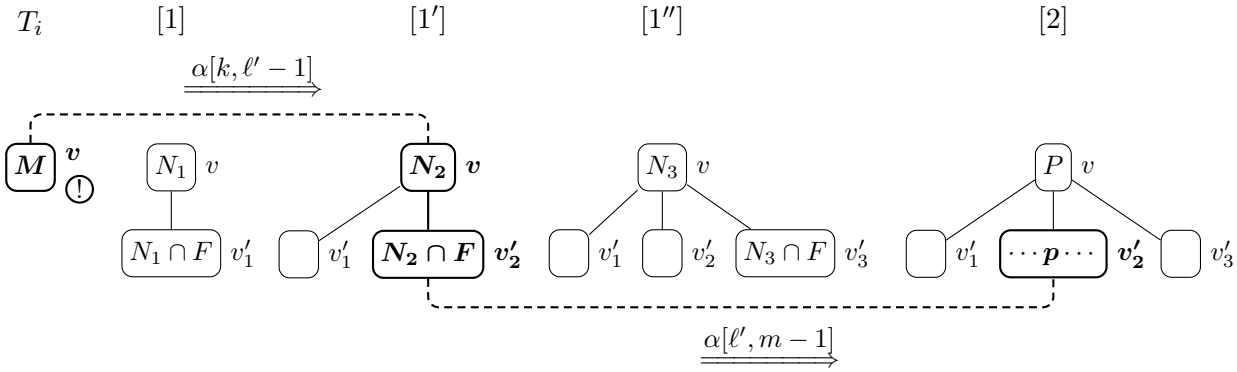
Diesen Run kann man durch die „obere Zeile“ ergänzen:

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

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

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

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



❑

T3.17 Vollständigkeitsbeweis der Safra-Konstruktion, Details

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

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

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

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

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

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

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

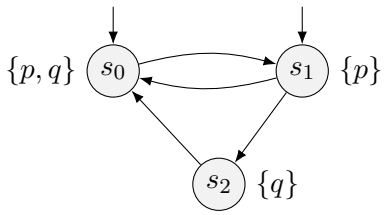
$$S_0, S_1, \ldots, S_m, S_{m+1}, \ldots, \underbrace{S'_m, S_{m'+1}, \ldots, S_p, \ldots}_{\text{Knoten 1 ist in keinem } S_i \text{ markiert}}$$

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

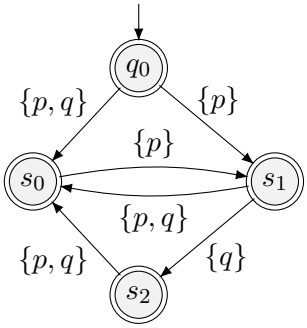
Nun kann man das bisherige Argument von 1 auf c übertragen: entweder erfüllt c auch Bedingung (b), oder es gibt ein Kind c', das (a) erfüllt. Diese Iteration kann man nicht beliebig oft fortsetzen, weil die Tiefe eines Safrabaums durch $|Q|$ beschränkt ist. Folglich muss es einen Nachfahren von 1 geben,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

T3.18 NBA für eine Kripke-Struktur

Kripke-Struktur $\mathcal{S}$



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



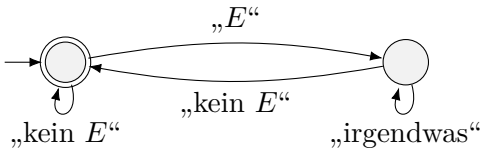
T3.19 NBAs für Beispiel-Eigenschaften

(1) Mikrowellen-Beispiel

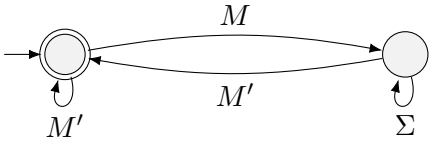
Hier ist das Alphabet $\Sigma = 2^{\{\mathsf{S},\mathsf{C},\mathsf{H},\mathsf{E}\}}$.

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

Schematisch muss der Automat so aussehen:

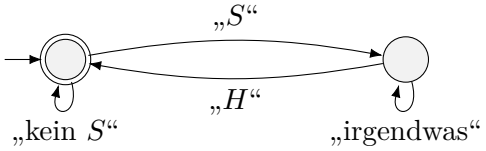


Dabei steht z. B. die Beschriftung „E“ für alle Alphabetzeichen (Teilmengen von $\{\mathsf{S},\mathsf{C},\mathsf{H},\mathsf{E}\}$), die E enthalten, und „kein E“ für alle Alphabetzeichen, die E nicht enthalten. Sei also $M = \{\{E\},\{S,E\},\{C,E\},\ldots,\{S,C,H,E\}\}$ und $M' = \Sigma \setminus M$. 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 $\Sigma = 2^{\{0,1,10,11,\ldots,23\}}$.

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

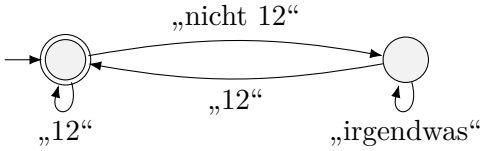
Schematisch muss der Automat so aussehen:

„nicht 12 oder nicht 22“

Dabei steht die Beschriftung „nicht 12 oder nicht 22“ der Schleife für alle Alphabetzeichen (Teilmengen von $\{0, 1, 10, 11, \ldots, 23\}$), die nicht gleichzeitig 12 und 22 enthalten. Damit ist die korrekte Kantenbeschriftung für die Schleife die Menge $\{X \subseteq \{0, 1, 10, 11, \ldots, 23\} \mid \{12, 22\} \nsubseteq X\}$.

(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):



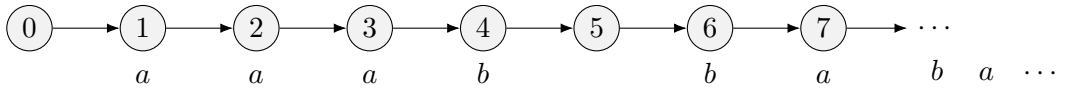
Der Automat $\mathcal{A}_2$ für P_2 ist analog. Um zu beschreiben, dass *beide* Programme beliebig oft in den jeweiligen kritischen Bereich kommen, muss man den Produktautomaten von $\mathcal{A}_1$ und $\mathcal{A}_2$ bilden.

(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 π.

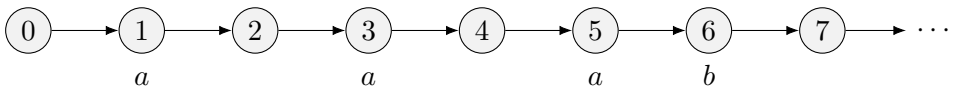


Das heißt also, $\pi(0) = \emptyset$, $\pi(1) = \{a\}$ usw. Die Beschriftung „$b\ a\ \cdots$“ am rechten Rand bedeutet, dass für alle $i \geq 4$ gilt: $\pi(2i) = \{b\}$ und $\pi(2i + 1) = \{a\}$. Dann gilt:

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

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

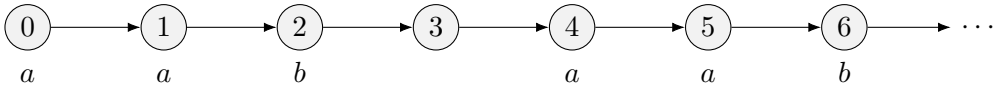
Betrachte nun den folgenden Pfad π'.



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

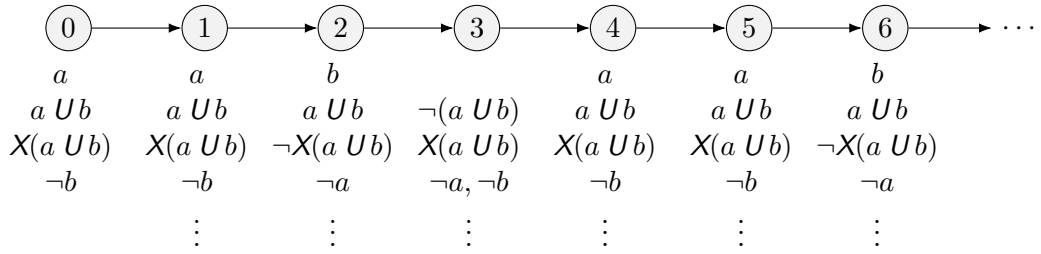
T3.21 Beispiele für die Erweiterung von Pfaden

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



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

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



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

T3.22 Beispiele für elementare Formelmengen

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

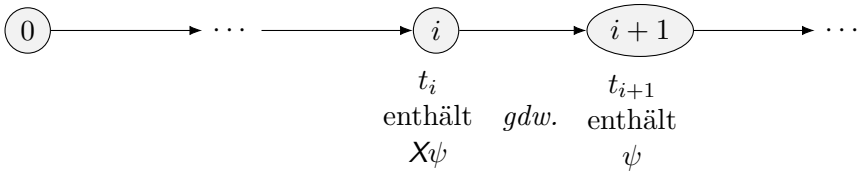
- $\{a, b, \varphi_E\}$ ist konsistent bezüglich der Aussagenlogik und lokal konsistent bezüglich des U-Operators, aber nicht maximal, weil weder $\neg a \wedge b$ noch $\neg(\neg a \wedge b)$ enthalten ist.
- Fügt man $\neg a \wedge b$ hinzu, so ist die resultierende Menge $\{a,\ b,\ \neg a \wedge b,\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

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

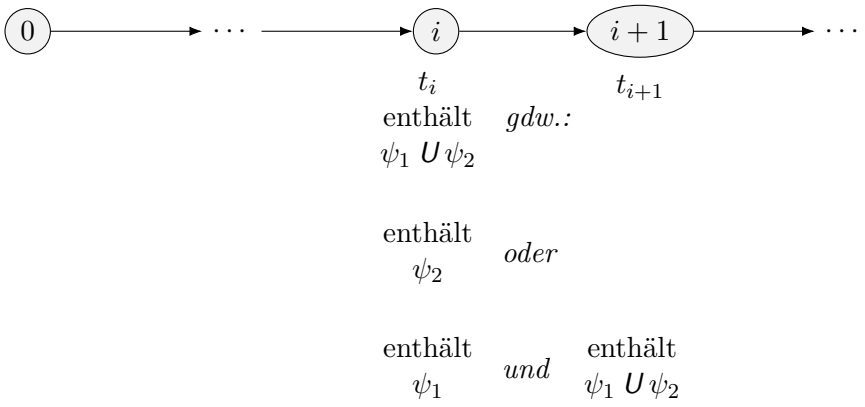
T3.23 Skizzen zur Def. der Überführungsrelation des GNBA

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 $\psi_1 \,\mathsf{U}\, \psi_2 \equiv \psi_2 \vee (\psi_1 \wedge \mathsf{X}(\psi_1 \,\mathsf{U}\, \psi_2))$ (weist diese selbst nach :–)).

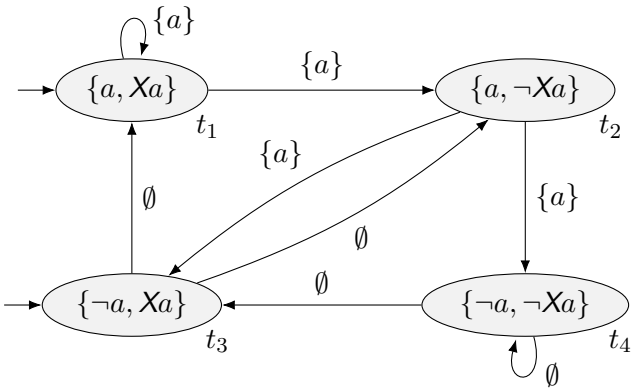
T3.24 GNBA für die Beispiel-Formel $\mathsf{X}a$

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

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

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

Graphische Darstellung:



T3.25 GNBA für die Beispiel-Formel $(\neg a)\ U\ b$

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

  Die verbleibenden fünf Formelmengen sind elementar:

  $t_1 = \{a,\ b,\ (\neg a)\ U\ b\}$ $\quad$ $t_4 = \{\neg a,\ \neg b,\ \neg((\neg a)\ U\ b)\}$
  $t_2 = \{\neg a,\ b,\ (\neg a)\ U\ b\}$ $\quad$ $t_5 = \{a,\ \neg b,\ \neg((\neg a)\ U\ b)\}$
  $t_3 = \{\neg a,\ \neg b,\ (\neg a)\ U\ b\}$

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

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

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

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

Folglich gibt es Übergänge

(a) von t_1 mit $\{a, b\}$ zu t_1, t_2, t_3, t_4, t_5 und
von t_2 mit $\{b\}$ zu t_1, t_2, t_3, t_4, t_5;

(b) von t_3 mit $\emptyset$ zu t_1, t_2, t_3;

(c) von t_5 mit $\{a\}$ zu t_1, t_2, t_3, t_4, t_5;

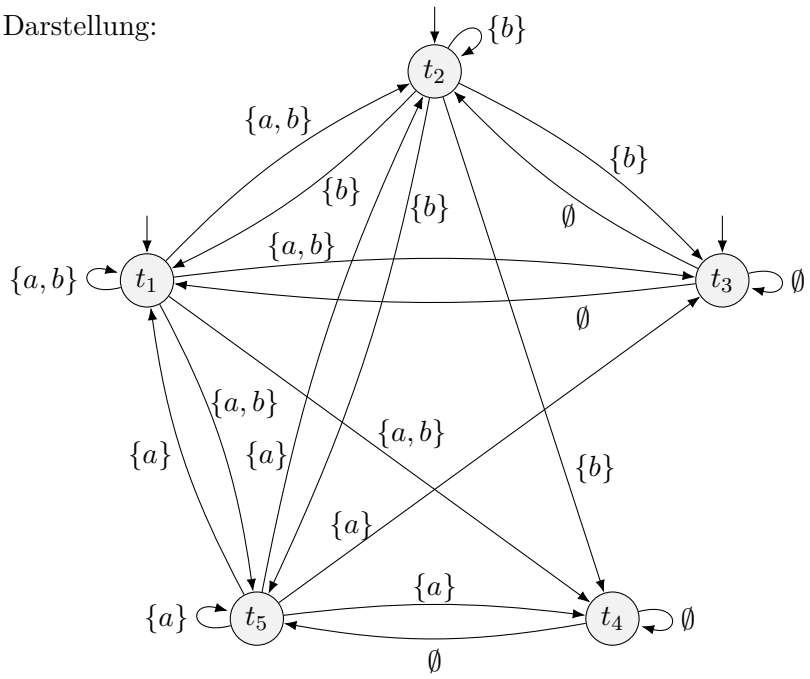
(d) von t_4 mit $\emptyset$ zu t_4, t_5.

- Anfangszustände: t_1, t_2, t_3 (diese enthalten φ)
- Akzeptanzkomponente:

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

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

Graphische Darstellung:



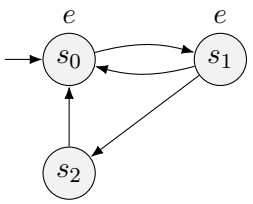
Teil IV.

Endliche Automaten auf unendlichen Bäumen

T4.1 LTL-Formeln „zu stark/schwach“

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

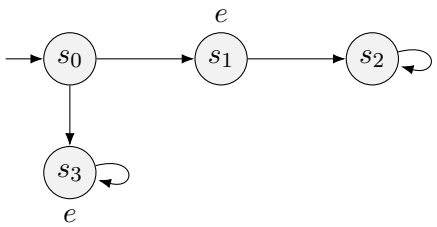
φ_1, φ_2 „zu stark für universelles Model Checking“. Wir betrachten folgende Kripke-Struktur $\mathcal{S}$:



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

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

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

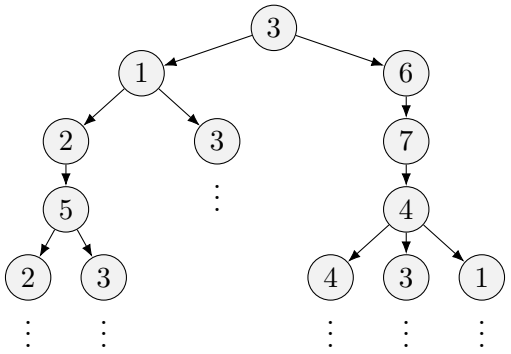


Intuitiv gesprochen, erfüllt $\mathcal{S}'$ die gewünschte Eigenschaft *nicht*: der Pfad $s_0 s_3$ kann nicht mehr wie gewünscht fortgesetzt werden.

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

T4.2 Beispiel-Berechnungsbaum

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

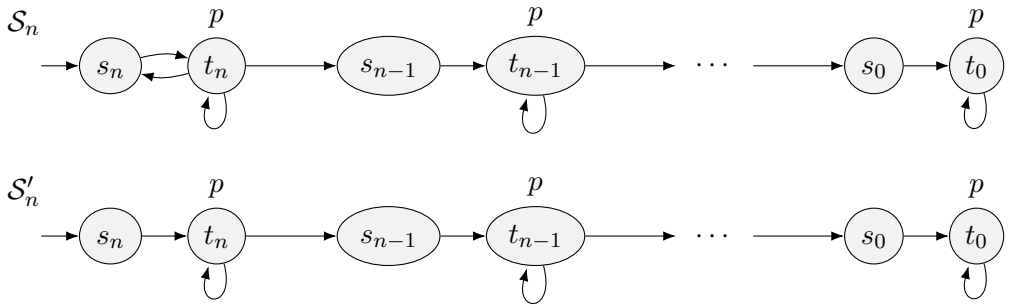


Die gegebene Struktur $\mathcal{S}$ ist eine endliche Repräsentation dieses unendlichen Baums (und vieler weiterer Bäume).

T4.3 Beweis Teil 2 des Ausdrucksstärke-Lemmas

Lemma 4.5 (2). Es gibt keine zu *FGp* äquivalente CTL-Zustandsformel.

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



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

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

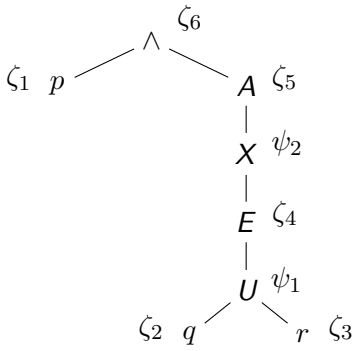
Außerdem zeigt man leicht per Induktion über n die folgende Aussage: Für alle $n \geq 0$ und alle CTL-Zustandsformeln ζ der Länge $\leq n$ gilt:

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

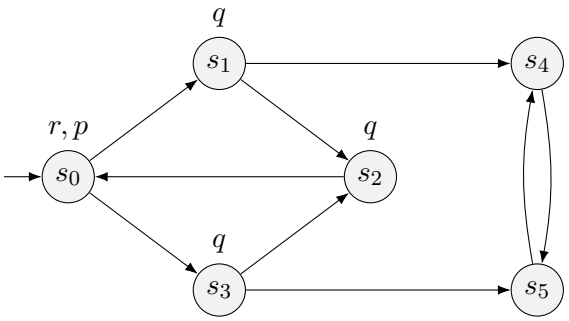
Angenommen, es gebe eine CTL-Zustandsformel ζ mit $\zeta \equiv \mathsf{FGp}$. Sei $n := |\zeta|$. Dann gilt wegen (i) und (ii): $\mathcal{S}_n \not\models \zeta$ und $\mathcal{S}'_n \models \zeta$. Das widerspricht aber (iii). ❏

T4.4 Beispiel für den Model-Checking-Algorithmus

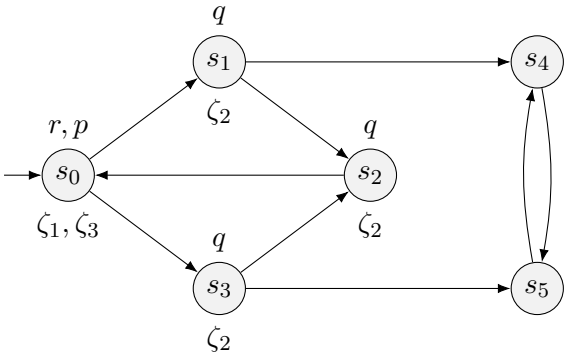
Wir betrachten die Zustandsformel $\zeta = p \wedge \mathsf{AXE}(q\,\mathsf{U}\,r)$. Ihre Baumdarstellung ist wie folgt; die einzelnen Teilformeln sind mit $\zeta_1, \ldots, \zeta_6$ (Zustandsformeln) und ψ_1, ψ_2 (Pfadformeln) markiert.



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

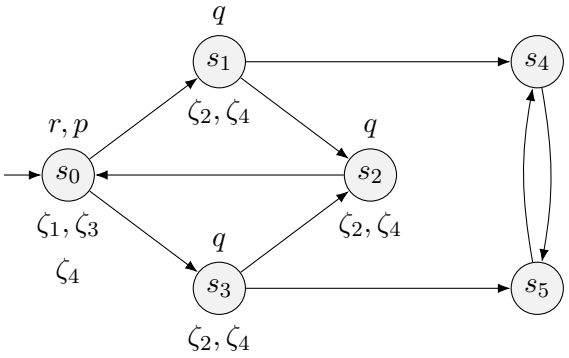


Zuerst werden für die Zustandsformeln $\zeta_1, \zeta_2, \zeta_3$ aus den Blättern von ζ alle Zustände markiert, die mit der jeweiligen Aussagenvariable markiert sind:

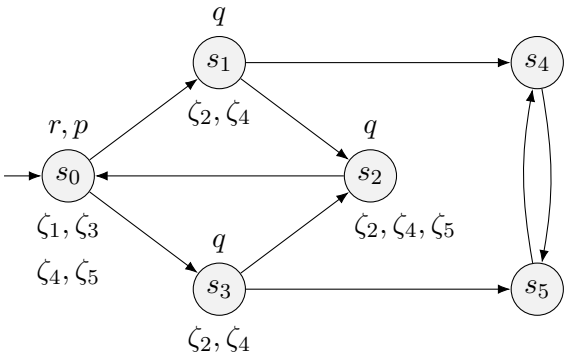


Die „nächsthöhere“ Zustandsformel ist $\zeta_4 = \mathsf{E}(\zeta_2\,\mathsf{U}\,\zeta_3)$. Es werden also als nächstes alle Zustände mit ζ_4 markiert, die aufgrund der bisherigen Markierung $\mathsf{E}(\zeta_2\,\mathsf{U}\,\zeta_3)$ erfüllen,

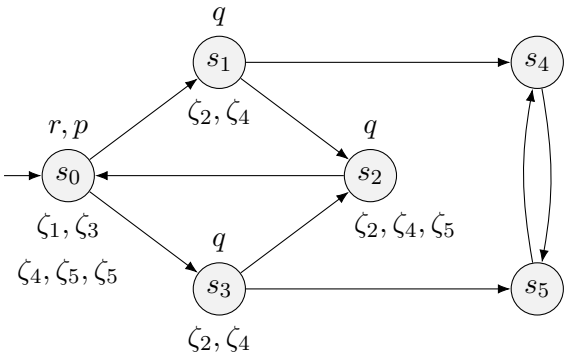
in denen also *mindestens ein* (E) Pfad beginnt, der die Pfadformel $\psi_1 = \zeta_2 \,\mathsf{U}\, \zeta_3$ erfüllt. Dies sind s_0, s_1, s_2, s_3.



Nun wird die Zustandsformel $\zeta_5 = \mathsf{AX}\zeta_4$ behandelt, also werden alle diejenigen Zustände mit ζ_5 markiert, deren *alle* (A) Nachfolger (X) bereits mit ζ_4 markiert sind. Dies sind s_0, s_2.



Schließlich ist $\zeta_6 = \zeta_1 \wedge \zeta_5$; also werden alle Zustände mit ζ_6 markiert, die bereits mit ζ_1 und ζ_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.

Literaturverzeichnis

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