8

Komplementierung von Büchi-Automaten

8.1 Zwei Sätze der unendlichen Kombinatorik

Die Komplementierung von Büchiautomaten nach Büchi verwendet zwei kombinatorische Sätze: Königs Lemma und den Satz von Ramsey, die wir hier der Vollständigkeit halber mit Beweis angeben.

Lemma 13 (Königs Lemma). *In einem Baum mit unendlich vielen Knoten, aber endlichem Verzweigungsgrad (kein Knoten hat unendlich viele Kinder) gibt es einen unendlichen Pfad.*

Beweis. Wir konstruieren induktiv eine Folge von Knoten $a_0, a_1, \ldots$, so dass a_{n+1} Kind von a_n ist und unterhalb a_n jeweils unendlich viele Knoten liegen, also a_n unendlich viele Nachkommen hat. Man nimmt für a_0 die Wurzel des Baumes; diese hat nach Voraussetzung unendlich viele Nachkommen. Sind $a_0, \ldots, a_n$ schon gefunden, so wählt man a_{n+1} als eines derjenigen Kinder von a_n, das unendlich viele Nachkommen hat. Ein solches muss vorhanden sein, denn sonst hätte a_n selbst nur endlich viele Nachkommen. In dieser Weise entsteht wie verlangt ein unendlicher Pfad.

Hier ist eine Anwendung von Königs Lemma. Gegeben sei ein Satz von quadratischen Kacheln mit gefärbten Kanten. Zwei Kanten passen zusammen, wenn sie dieselbe Farbe haben. Es ist unentscheidbar, ob man mit einem gegebenen Satz die Ebene pflastern kann. Immerhin gilt folgendes: kann man mit einem Satz von Kacheln einen Quadranten pflastern, so kann mit demselben Satz damit auch die ganze Ebene pflastern. Achtung: man kann nicht einfach die vier Quadranten pflastern; an den Nähten könnten ja dann die Farben nicht zusammenpassen.

Man arrangiert Pflasterungen von Quadraten der Kantenlänge $2i$ als Baum, indem die Kinder einer Pflasterung der Größe $2i \times 2i$ diejenigen der Größe $(2i+2) \times (2i+2)$ sind, die die gegebene um einen äußeren Ring erweitern. Die leere Pflasterung des 0×0 Quadrates bildet die Wurzel. Der

Baum hat unendlich viele Knoten, da nach Voraussetzung Quadrate beliebiger Größe pflasterbar sind. Ein unendlicher Pfad, der nach Königs Lemma existiert, definiert die gesuchte Pflasterung der Ebene.

Für eine Menge A bezeichne $\binom{A}{2}$ die Menge der zweielementigen Teilmengen von A.

Proposition 9 (Satz von Ramsey). *Sei A eine unendliche Menge, C eine endliche Menge von "Farben" und $f : \binom{A}{2} \to C$ eine "Färbung" der zweielementigen Teilmengen von A mit "Farben" aus C. Es existiert eine unendliche Teilmenge B von A, sodass f eingeschränkt auf $\binom{B}{2}$ konstant ist, d.h. es gibt eine feste Farbe $c \in C$, sodass $f(\{x, y\}) = c$ für alle $x, y \in B$.*

Beweis. Zunächst müssen wir A anordnen (wir können o.B.d.A. A mit $\mathbb{N}$ identifizieren und die normale Ordnung verwenden). Sollte A überabzählbar sein, dann schränken wir zunächst willkürlich auf eine abzählbare Teilmenge ein. Außerdem können wir annehmen, dass $C = \{1, \ldots, k\}$.

Wir konstruieren jetzt eine Folge $(a_0, A_0, c_0), (a_1, A_1, c_1), (a_2, A_2, c_2), \ldots$ sodass $a_0 < a_1 < a_2 < \ldots$ und $A_0 \supseteq A_1 \supseteq A_2 \supseteq \ldots$ und so dass die A_i alle unendlich sind und $f(\{a_i, x\}) = c_i$ für alle $x \in A_i$ und $a_{i+1} \in A_i$ für alle i.

Wir wählen a_0 beliebig und c_0 als eine Farbe, die unendlich viele Paare der Form $\{a_0, x\}$ färbt und dann $A_0 = \{x \mid f(\{a_0, x\}) = c_0$ Das geht, weil es nur endlich viele Farben gibt.

Ist die Folge schon bis n gebildet, dann nehmen wir aus A_n wiederum ein beliebiges Element a_{n+1} und verschaffen uns A_{n+1}, c_{n+1} wie bei (a_0, A_0, c_0).

Eine Farbe taucht nun unendlich oft auf in der Folge der c_i.

Die Menge derjenigen a_n's, die diese Farbe haben, leistet das Verlangte.

8.2 Büchi's Komplementierung

Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ ein Büchiautomat. Wir wollen einen Büchiautomaten $\mathcal{A}'$ konstruieren, sodass $L(\mathcal{A}) = \overline{L(\mathcal{A}')}$. Für Wort $u \in \Sigma^*$ führen wir folgende Notationen ein:

$u : q \to q'$ bedeutet: von Zustand q kommt man unter Abarbeitung von u im Automaten nach q'.

$u : q \to_F q'$ bedeutet: von Zustand q kommt man unter Abarbeitung von u im Automaten nach q' und durchläuft dabei einen Endzustand.

Wenn $u : q \to_F q'$ dann natürlich auch $u : q \to q'$. Wenn $u : q \to q_1$ und $v : q_1 \to q'$ und $q_1 \in F$, dann auch $uv : q \to_F q'$.

Wir definieren eine Äquivalenzrelation $\sim \subseteq \Sigma^* \times \Sigma^*$ wie folgt:

$$u \sim v \iff \forall q\ q'.(u : q \to q' \Leftrightarrow v : q \to q') \wedge (u : q \to_F q' \Leftrightarrow v : q \to_F q')$$

Lemma 14. *Die Äquivalenzrelation $\sim$ hat endlichen Index, das heißt, sie hat nur endlich viele Klassen.*

Beweis. Jede Klasse ist eindeutig bestimmt durch zwei Mengen von Zustandspaaren, also gibt es höchstens $2^{2|Q|^2}$ viele Klassen.

Lemma 15. *Jede Äquivalenzklasse der Relation $\sim$ ist regulär.*

Beweis. Für Zustände q, q' sei $L_{q,q'} = \{u \mid u : q \to q'\}$ (vgl. Beweis von Satz **??**) und $L^F_{q,q'} = \{u \mid q \to_F q'\}$. Man sieht leicht (Übung), dass jede Klasse regulärer Ausdruck in den (offensichtlich regulären) $L_{q,q'}$ und $L^F_{q,q'}$ ist.

Lemma 16. *Seien U, V zwei Klassen von $\sim$ und sei $w \in UV^\omega$. Wenn $w \in L(\mathcal{A})$, so ist sogar UV^ω eine Teilmenge von $L(\mathcal{A})$.*

Beweis. Wir zerlegen das Wort w als $w = uv_1v_2v_3v_4\ldots$ mit $u \in U$ und $v_i \in V$ und betrachten einen akzeptierenden Lauf. Aus der Annahme, dass U, V Äquivalenzklassen sind können wir dann einen akzeptierenden Lauf für jedes andere Wort aus UV^ω konstruieren.

Lemma 17. *Seien U, V zwei Klassen von $\sim$ und $w \in UV^\omega$. Wenn w in $\overline{L(\mathcal{A})}$, so ist sogar UV^ω eine Teilmenge von $\overline{L(\mathcal{A})}$.*

Beweis. Wäre $w' \in UV^\omega$ in $L(\mathcal{A})$, so auch w selbst wegen Lemma 16.

Lemma 18. *Für jedes beliebige Wort $w = a_0a_1a_2\cdots \in \Sigma^\omega$ existieren Äquivalenzklassen U, V von $\sim$, sodass $w \in UV^\omega$.*

Beweis. Jedes endliche Wort liegt in einer Klasse (nämlich "seiner" Klasse). Wir betrachten folgende Färbung von $\binom{\mathbb{N}}{2}$. Setze $f(\{i,j\})$ gleich der Äquivalenzklasse von $a_i \ldots a_{j-1}$, wobei o.B.d.A. $i < j$. Der Satz von Ramsey liefert eine Äquivalenzklasse V und eine unendliche Teilmenge $S \subseteq \mathbb{N}$, sodass $i, j \in S, i < j$ impliziert $a_i \ldots a_{j-1} \in V$. Wenn jetzt i_0 das kleinste Element von S ist und U die Äquivalenzklasse von $a_0 \ldots a_{i_0-1}$ ist, dann folgt $w \in UV^\omega$.

Proposition 10 (Büchi). *Sei $\mathcal{A}$ ein Büchiautomat. Die Sprache $\overline{L(\mathcal{A})}$ ist ω-regulär.*

Beweis. Aus dem vorher Gesagten folgt

$$\overline{L(\mathcal{A})} = \bigcup \{UV^w \mid U, V \in \Sigma^*/\sim \wedge\ UV^\omega \cap L(\mathcal{A}) = \emptyset\}$$

Damit aber liegt ein regulärer Ausdruck vor.

Es folgt, dass ein Automat $\mathcal{A}'$ existiert, der gerade das Komplement von $L(\mathcal{A})$ erkennt.

Will man diesen effektiv berechnen, so kann man wie folgt vorgehen: Die endlich vielen Äquivalenzklassen werden durch Repräsentanten dargestellt und aus dem Automaten abgelesen. Durch Tiefensuche im Automaten kann man dann diejenigen Klassen U, V identifizieren, für die UV^ω disjunkt von $L(\mathcal{A})$ ist. Wegen Lemma 16 und 17 ist das ja gleichbedeutend damit, dass uv^ω nicht in $L(\mathcal{A})$ ist, wobei $u \in U$ und $v \in V$ beliebige Repräsentanten sind.

Dies liefert einen Algorithmus zur Konstruktion von $\mathcal{A}'$. Allerdings hat der so gewonnene Automat $2^{2^{O(|\mathcal{A}|)}}$ Zustände.

8.3 Entscheidbarkeit

Wir erinnern uns, dass die monadische Logik zweiter Stufe (MSO) dieselbe Syntax wie wMSO hat, aber Mengenvariablen und damit auch Formeln beliebige also möglicherweise unendliche Teilmengen von $\mathbb{N}$ bezeichnen.

Proposition 11 (Büchi). *Es existiert ein effektives Verfahren, das zu gegebener MSO Formel ϕ entscheidet, ob sie erfüllbar ist.*

Beweis. Der Beweis des Satzes von Büchi erfolgt ganz analog zum Beweis des Satzes von Büchi-Elgot: zu gegebener MSO-Formel ϕ mit freien Mengenvariablen $X_0, \ldots, X_{n-1}$ (möglicherweise $n = 0$, also ϕ geschlossen) konstruiert man einen Büchiautomaten $\mathcal{A}_\phi$ über dem Alphabet $2^{\{0,\ldots,n-1\}}$ (also einelemeniges Alphabet wenn $n = 0$), derart, dass $L(\mathcal{A}_\phi) = \{w \mid w \models \phi\}$, wobei $w \models \phi$ bedeutet, dass ϕ wahr ist, wenn man die Variable X_i als die Menge $\{p \mid i \in a_p\}$ interpretiert, wobei $w = a_0 a_1 a_2 \ldots$. Man liest also das Wort w als unendliche Wertetabelle für die X_i.

Die Konstruktion des Automaten erfolgt durch Induktion über den Aufbau von ϕ. Die Boole'schen Operationen gehen wie üblich (Produktautomat, Komplementierung). Existenzquantoren behandelt man mit Satz **??** unter Einführung von Nichtdeterminismus.

Wie schon gesagt wird bei der Komplementierung die Zustandszahl (doppelt) exponentiell erhöht, also auch bei der Behandlung von Allquantoren, die man ja als $\neg\exists\neg$ auffasst. Dies führt wie bei Büchi-Elgot zu einem Verfahren der Komplexität $DTIME(2_{O(n)})$.

In der Praxis ist aber dieses Verfahren wesentlich schwieriger zu implementieren als das Verfahren für die wMSO und unseres Wissens existiert überhaupt keine allgemein verfügbare Implementierung. Unter anderem wird dies daran liegen, dass es für Büchiautomaten kein Minimierungsverfahren gibt. Allerdings gibt es in letzter Zeit einige erfolgversprechende Ansätze. So hat Thomas Wilke Arbeiten zur Minimierung von Büchiautomaten verfasst und Vardi-Kupferman haben einen alternativen Ansatz mit alternierenden Paritätsautomaten ins Spiel gebracht.

9

Weitere Akzeptanzbedingungen

Die Büchi-Akzeptanzbedingung — unendliches Auftreten von Zuständen aus einer Menge — ist zwar eine natürliche Erweiterung der Akzeptanz endlicher Automaten auf endlichen Wörtern, jedoch sicherlich nicht die einzige Möglichkeit, Akzeptanz auf unendlichen Wörtern zu definieren. In diesem Kapitel lernen wir weitere Akzeptanzbedingungen und die entsprechenden Automatentypen kennen.

9.1 Rabin-, Streett- und Paritätsautomaten

Läufe solcher Automaten sind wiederum unendliche Sequenzen $\rho = q_0, q_1, \ldots$ von Zuständen, die der Transitionsfunktion gehorchen und jeweils im Anfangszustand beginnen. Für solch einen Lauf ρ bezeichnen wir mit $Inf(\rho)$ die Menge derjenigen Zustände, die unendlich oft auf diesem Lauf vorkommen, d.h.

$$Inf(\rho) \;=\; \{q \mid q = q_i \text{ für unendlich viele } i\}$$

Definition 23. *Ein* Rabin-Automat *(NRA) ist ein Tupel* $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$*, wobei* Q, Σ, q_0, δ *wie bei NBA, bzw. NFA, definiert sind, und* $\mathcal{F} = \{(G_1, F_1), \ldots, (G_k, F_k)\}$ *mit* $G_i, F_i \subseteq Q$*.*

Solch ein NRA heißt deterministisch (DRA), falls $|\delta(q, a)| = 1$ *für alle* $q \in Q$ *und* $a \in \Sigma$*.*

Ein Lauf $\rho = q_0, q_1, \ldots$ *eines NRA heißt akzeptierend, falls es ein* $i \in \{1, \ldots, k\}$ *gibt, so dass* $Inf(\rho) \cap G_i \neq \emptyset$ *und* $Inf(\rho) \cap F_i = \emptyset$*. Die Sprache* $L(\mathcal{A})$ *ist wieder die Menge aller Wörter, für die es einen akzeptierenden Lauf gibt.*

Die Größe eines NRA ist die Anzahl $|Q|$ *seiner Zustände. Der* Index *eines Rabin-Automaten ist die Größe seiner Endzustandskomponente, also hier* k*.*

Die Endzustandskomponente enthält also Paare von im Unendlichen erlaubten und verbotenen Zuständen.

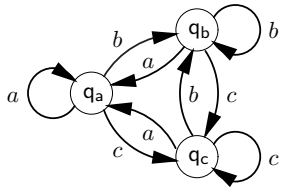


Abb. 9.1. Transitionsfunktion eines Automaten über dem Alphabet $\Sigma = \{a, b, c\}$.

Beispiel 11. Sei $\Sigma = \{a, b, c\}$. Betrachte die Sprache $L := \{w \in \Sigma^\omega \mid |w|_a = \infty \Rightarrow |w|_b = \infty\}$. Der NRA in Abb. 9.1 protokolliert in seiner Zustandsmenge einfach das zuletzt gelesene Zeichen eines Worts. Um ihn L erkennen zu lassen, reicht es aus sich zu überlegen, dass ein Wort w in L ist gdw. der eindeutige Lauf darauf entweder unendlich oft den Zustand $\mathsf{q_b}$ durchläuft oder nur endlich oft den Zustand $\mathsf{q_a}$. Ersteres wird durch das Rabin-Paar $(\{\mathsf{q_b}\}, \emptyset)$ modelliert. Letzteres wiederum entspricht dem Rabin Paar $(Q, \{\mathsf{q_a}\})$.

Somit wird der in Abb. 9.1 gezeigt Automat mit

$$\mathcal{F} \;=\; \{(\{\mathsf{q_b}\}, \emptyset), (Q, \{\mathsf{q_a}\})$$

zu einem NRA — bzw. sogar DRA — der L erkennt.

Die Streett-Bedingung ist das duale zu der Rabin-Bedingung. Syntaktisch ist ein Streett-Automat nicht von einem Rabin-Automat zu unterscheiden, seine Akzeptanz ist jedoch dual zu der eines Rabin-Automaten definiert.

Definition 24. *Ein* Streett-Automat *(NSA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$, wie bei einem NRA, also insbesondere mit $\mathcal{F} = \{(G_1, F_1), \ldots, (G_k, F_k)\}$. Deterministische Streett-Automaten (DSA) sind wie üblich definiert.*

Ein Lauf $\rho = q_0, q_1, \ldots$ eines NSA ist akzeptierend, falls für alle $i = 1, \ldots, k$ gilt: $Inf(\rho) \cap G_i \neq \emptyset$ impliziert $Inf(\rho) \cap F_i \neq \emptyset$. Die vom NSA $\mathcal{A}$ erkannte Sprache ist ebenfalls wie üblich definiert.

Genauso ist die Größe eines NSA wieder die Anzahl $|Q|$ seiner Zustände, und sein Index *ist die Größe seiner Endzustandskomponente, also k.*

Die Endzustandskomponente eines Streett-Automaten beschreibt also Abhängigkeiten der Form, dass ein bestimmter Zustand nur unendlich oft durchlaufen werden darf, wenn auch ein anderer unendlich oft durchlaufen wird.

Beispiel 12. Betrachte wieder die Sprache $L = \{w \in \{a, b, c\}^\omega \mid |w|_a = \infty \Rightarrow |w|_b = \infty\}$. Den in Abb. 9.1 gezeigten Automaten kann man auch als NSA bzw. DSA auffassen. Die Beschreibung der Sprache L lässt sich sehr leicht als Streett-Bedingung modellieren: $\mathcal{F} = \{(\{\mathsf{q_a}\}, \{\mathsf{q_b}\})\}$.

Die von Rabin- bzw. Streett-Automaten erkannte Sprachklasse enthält mindestens die ω-regulären Sprachen, weil die Büchi-Bedingung ein Spezialfall der Rabin- oder Streett-Bedingungen ist.

Theorem 20. *Für jeden NBA $\mathcal{A}$ der Größe n existiert ein NRA (NSA) $\mathcal{A}'$ der Größe n und vom Index* 1*, so dass $L(\mathcal{A}') = L(\mathcal{A})$.*

Beweis. Übung.

Die Dualität zwischen Rabin- und Streett-Bedingung wird durch das folgende Lemma formalisiert.

Lemma 19. *Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ ein NRA oder NSA und ρ ein Lauf dieses Automaten auf einem Wort $w \in \Sigma^\omega$. Dann ist ρ akzeptierend bzgl. der Rabin-Bedingung $\mathcal{F}$ gdw. ρ nicht akzeptierend bzgl. der Streett-Bedingung $\mathcal{F}$ ist.*

Beweis. Übung.

Daraus folgt sofort, dass sich ein DRA leicht zu einem DSA und umgekehrt komplementieren lässt.

Theorem 21. *Für jeden DRA (DSA) $\mathcal{A}$ der Größe n und Index k gibt es einen DSA (DRA) $\overline{\mathcal{A}}$ der Größe n und Index k, so dass $L(\overline{\mathcal{A}}) = \Sigma^\omega \setminus L(\mathcal{A})$.*

Beweis. Die Konstruktion ist trivial. Setze $\overline{\mathcal{A}} := \mathcal{A}$. Wichtig ist, dass dieser als Automat mit der jeweils anderen Akzeptanzbedingung aufgefasst wird. Die Korrektheit folgt mit Lemma 19 sofort aus der Tatsache, dass ein deterministischer Automat auf einem gegebenen Wort $w \in \Sigma^\omega$ genau einen Lauf hat. □

Beachte, dass der Determinismus eine wichtige Voraussetzung in diesem Theorem ist. Ein NRA lässt sich i.A. nicht so leicht in einen NSA komplementieren, da Nicht-Akzeptanz durch einen nicht-deterministischen Automaten eine universelle Aussage ist, die über alle Läufe quantifiziert. Dies wiederum kann aber nicht ohne weiteres von einem nicht-deterministischen Automaten modelliert werden.

Ein Nachteil von Rabin- und Streett-Automaten ist, dass ihre Akzeptanzbedingung jeweils nicht unter Komplement abgeschlossen ist. Beachte, dass sich dies auch bei der Büchi-Bedingung so verhält: "nicht unendlich oft" lässt sich nicht einfach wiederum als "unendlich oft" formulieren.

Wir definieren noch eine weitere Akzeptanzbedingung, welche dual zu sich selbst ist. Dabei werden den Zuständen eines Automaten Prioritäten zugeordnet, wobei gerade Prioritäten als gut und ungerade als schlecht anzusehen sind und ihr Betrag ausdrückt, wie wichtig der Zustand im Vergleich zu anderen ist.

Definition 25. *Ein* Paritätsautomat *(NPA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$, wobei Q, Σ, q_0, δ wie üblich definiert sind, und $\Omega : Q \to \mathbb{N}$. Ein deterministischer NPA (DPA) ist wie üblich definiert.*

Ein Lauf $\rho = q_0, q_1, \ldots$ eines Paritätsautomaten heißt akzeptierend, falls $\max\{\Omega(q) \mid q \in Inf(\rho)\}$ gerade ist.

Der Index *eines Paritätsautomaten misst die Komplexität seiner Akzeptanzbedingung und ist gegeben als (m, k), wobei*

$$\begin{aligned} m &= \min\{\Omega(q) \mid q \in Q\} \\ k &= \max\{\Omega(q) \mid q \in Q\} \end{aligned}$$

Beispiel 13. Wir betrachten wieder die Sprache $L = \{w \in \{a,b,c\}^\omega \mid |w|_a = \infty \Rightarrow |w|_b = \infty\}$ wie oben. Den in Abb. 9.1 gezeigten Automaten kann man ebenso als NPA bzw. DPA hernehmen, um damit L zu erkennen. Beachte, dass ein unendliches Durchlaufen des Zustands $\mathsf{q_b}$ offensichtlich gut in Bezug auf diese Sprache ist. Deswegen sollte dieser Zustand eine gerade Priorität haben. Ein unendliches Durchlaufen des Zustands $\mathsf{q_a}$ ist nur dann gut, wenn auch $\mathsf{q_b}$ unendlich oft durchlaufen wird, ansonsten schlecht. Deswegen sollte er eine ungerade Priorität, die kleiner ist als die von $\mathsf{q_b}$, haben. Schlussendlich kann Zustand $\mathsf{q_c}$ dann wieder eine noch kleinere gerade Priorität bekommen, denn diesen unendlich oft zu durchlaufen ist im Prinzip gut, es sei denn, $\mathsf{q_a}$ wird auch unendlich oft durchlaufen. Setze also

$$\Omega(\mathsf{q_a}) = 1 \ , \quad \Omega(\mathsf{q_b}) = 2 \ , \quad \Omega(\mathsf{q_c}) = 0$$

wodurch dieser Automat zu einem NPA mit Index $(0, 2)$ wird, der L erkennt.

Die Selbstdualität der Paritätsbedingung ist darin begründet, dass in einem Lauf ρ die maximale, unendlich oft auftretenden Priorität eindeutig ist (was die Endlichkeit der zugrundeliegenden Zustandsmenge benutzt). Diese ist nicht gerade gdw. sie ungerade ist, und "unendlich oft ungerade" lässt sich wiederum als "unendlich oft gerade" darstellen, wenn man alle Prioritäten um eine ungerade Zahl verschiebt. Solch eine Transformation erhält natürlich die totale Quasi-Ordnung auf den Zuständen, die durch Vergleich ihrer Prioritäten gegeben ist.

Beachte, dass dieses Prinzip leider wiederum nur zur Komplementierung deterministischer Automaten verwendet werden kann, weil bei solchen existentielle und universelle Quantifizierung über die Läufe auf einem gegebenen Wort dasselbe sind.

Theorem 22. *Für jeden DPA $\mathcal{A}$ der Größe n und vom Index (m, k) gibt es einen DPA $\overline{\mathcal{A}}$ der Größe n und vom Index $(m + 1, k + 1)$, so dass $L(\overline{\mathcal{A}}) = \Sigma^\omega \setminus L(\mathcal{A})$.*

Beweis. Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$ ein DPA. Definiere $\overline{\mathcal{A}} := (Q, \Sigma, q_0, \delta, \overline{\Omega})$, wobei für alle $q \in Q$:

$$\overline{\Omega}(q) \ := \ \Omega(q) + 1 \ .$$

Beachte, dass $\overline{\mathcal{A}}$ ebenfalls deterministisch ist. Die Abschätzung seiner Größe und seines Index ergeben sich direkt aus dieser Konstruktion.

Sei außerdem $w \in \Sigma^\omega$ und ρ der eindeutige Lauf von $\mathcal{A}$ auf w. Beachte, dass ρ dann auch der eindeutige Lauf von $\overline{\mathcal{A}}$ auf w ist und dass die maximale, unendliche oft auftretende Priorität darin bzgl. Ω gerade ist, gdw. sie bzgl. $\overline{\Omega}$ ungerade ist. □

Wie bei Rabin- und Streett-Automaten können Paritätsautomaten ebenfalls die Büchi-Bedingung leicht modellieren. Damit gilt also, dass Paritätsautomaten ebenfalls mindestens die Klasse der ω-regulären Sprachen erkennen.

Theorem 23. *Für jeden NBA $\mathcal{A}$ der Größe n existiert ein äquivalenter NPA $\mathcal{A}'$ der Größe n und vom Index $(1, 2)$.*

Beweis. Übung.

Andererseits gilt, dass NPAs höchstens die ω-regulären Sprachen erkennen. D.h. aus einem NPA lässt sich auch ein äquivalenter NBA gewinnen. Dies geht allerdings nicht ohne Vergrößerung der Zustandszahl, da die Paritätsbedingung offensichtlich mächtiger ist als die Büchi-Bedingung. Die Transformation beruht auf der Tatsache, dass die maximale, unendlich oft auftretende Priorität in einem Lauf gerade ist gdw. wenn es einen Moment in dem Lauf gibt, ab dem eine gerade Priorität unendlich oft auftritt und keine größere Priorität mehr vorkommt. Dies ist aber als Büchi-Bedingung formulierbar.

Theorem 24. *Für jeden NPA $\mathcal{A}$ der Größe n und vom Index (m, k) existiert ein NBA $\mathcal{A}'$ der Größe $\leq n \cdot (k - m + 2)$ mit $L(\mathcal{A}') = L(\mathcal{A})$.*

Beweis. Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \Omega)$ ein NPA. Wir konstruieren einen äquivalenten NBA $\mathcal{A}'$, der intuitiv wie folgt arbeitet. Er besteht aus mehreren Komponenten, von der eine einfach eine Kopie von $\mathcal{A}$ ist, die auch den Anfangszustand enthält, aber keine Endzustände. In dieser Komponente kann $\mathcal{A}'$ jedes Präfix eines Laufs auf einem Wort in $L(\mathcal{A})$ verfolgen. In einem akzeptierenden Lauf von $\mathcal{A}$ auf einem Wort ist die größte, unendlich oft auftretende Priorität gerade. D.h. es gibt eine gerade Priorität, die ab einem gewissen Zeitpunkt in diesem Lauf unendlich oft auftritt, während ab dann keine größere Priorität mehr auftritt. Ersteres lässt sich durch eine Büchi-Bedingung modellieren. Zweiteres kann dadurch sichergestellt werden, dass $\mathcal{A}'$ für jede gerade Priorität i eine Komponente besitzt, in die er nichtdeterministisch wechseln kann und in der kein Zustand mit höherer Priorität als i vorkommt.

Formal wird $\mathcal{A}'$ wie folgt konstruiert. Sei $Q|_i = \{q \in Q \mid \Omega(q) \leq i\}$ für jedes i mit $m \leq i \leq k$. Dann ist $\mathcal{A}' = (Q', \Sigma, q_0, \delta', F)$, wobei

$$Q' \;:=\; Q \;\cup \bigcup_{\substack{i=m \\ i \text{ gerade}}}^{k} \{i\} \times Q|_i$$

Das jeweilige $\{i\}$ dient lediglich dazu, die einzelnen Zustände in den verschiedenen Komponenten zu markieren. Die Transitionsfunktion ist dann die folgende.

$$\begin{aligned} \delta'(q,a) &:= \delta(q,a) \cup \{(i,q') \mid m \leq i \leq k, i \text{ gerade }, q' \in \delta(q,a), \Omega(q') \leq i\} \\ \delta((i,q),a) &:= \{(i,q') \mid q' \in \delta(q,a), \Omega(q') \leq i\} \end{aligned}$$

Die Endzustände sind dann jeweils diejenigen mit ursprünglicher Priorität i in i-Komponenten.

$$F := \{(i, q) \mid \Omega(q) = i\}$$

Es sollte klar sein, dass $\mathcal{A}'$ höstens $|Q| + (k - m + 1) \cdot |Q|$ viele Zustände hat. Es bleibt noch die Korrektheit der Konstruktion zu zeigen.

"$\supseteq$" Sei $w \in L(\mathcal{A})$. Dann existiert ein akzeptierender Lauf $\rho = q_0, q_1, \ldots$ von $\mathcal{A}$ auf w, d.h. $\max\{\Omega(q) \mid q \in Inf(\rho)\}$ ist gerade. Also existiert ein gerades i mit $m \leq i \leq k$, so dass unendliche viele q_j die Priorität i haben, aber nur endliche viele haben jeweils eine Priorität i' mit $i' > i$. Also gibt es ein $n \in \mathbb{N}$, so dass für alle $j \geq n$ gilt: $\Omega(q_j) \leq i$. Zusätzlich gibt es unendlich viele $j \geq n$, so dass $\Omega(q_j) = i$. Man vergewissert sich leicht, dass dann

$$q_0, q_1, \ldots, q_{j-1}, (i, q_j), (i, q_{j+1}), (i, q_{j+2}), \ldots$$

ein akzeptierender Lauf von $\mathcal{A}'$ auf w ist.

"$\subseteq$" Analog. □

Auf eine sehr ähnliche Art und Weise lässt sich auch eine Rabin-Bedingung in eine Büchi-Bedingung übersetzen. Dies liegt daran, dass die Rabin-Bedingung existentiell über Paare von Mengen quantifiziert und ein nicht-deterministischer Büchi-Automat in seinem Zustandsraum diese Quantifizierung nachbauen kann. Eine Umwandlung eines Streett-Automaten in einen Büchi-Automaten ist dagegen schwieriger, aber ebenfalls möglich.

Theorem 25. *Für jeden NRA $\mathcal{A}$ der Größe n und vom Index k existiert ein äquivalenter NBA $\mathcal{A}'$ der Größe $\leq n \cdot (k + 1)$.*

Beweis. Übung.

9.2 Muller-Automaten

Nachdem die offensichtlich stärkeren Akzeptanzbedingungen Rabin, Streett und Parität dennoch nicht die Klasse der erkennbaren Sprachen im Vergleich zu Büchi vergrößern, stellt sich die Frage, ob es überhaupt eine Akzeptanzbedingung geben kann, die dies leistet. Wir betrachten dazu natürlich weiterhin nur vernünftige Bedingungen, d.h. solche, die wie die oben genannten über das unendliche Auftreten von Zuständen definiert sind. Dann gibt es eine offensichtlich mächtigste Akzeptanzbedingung, nämlich die sogenannte Muller-Bedingung.

Definition 26. *Ein* Muller-Automat *(NMA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$, wobei Q, Σ, q_0, δ wie üblich definiert sind und $\mathcal{F} \subseteq 2^Q$. Deterministische Muller-Automaten (DMA) sind ebenfalls wie üblich definiert.*

Die Größe eines NMA ist wieder die Anzahl $|Q|$ seiner Zustände, und sein Index *ist die Größe seiner Endzustandskomponente, also $|\mathcal{F}|$.*

Ein Lauf $\rho = q_0, q_1, \ldots$ eines NMA ist akzeptierend, falls $Inf(\rho) \in \mathcal{F}$. Wiederum ist $L(\mathcal{A})$ die vom NMA $\mathcal{A}$ erkannte Sprache.

Beispiel 14. Wir betrachten noch einmal die Sprache $L = \{w \in \{a,b,c\}^\omega \mid |w|_a = \infty \Rightarrow |w|_b = \infty\}$ und den in Abb. 9.1 gezeigten Automaten. Da dieser in seinem Zustand immer das zuletzt gelesene Zeichen reflektiert, kann man ihm leicht eine Mullerbedingung zuweisen, so dass er zu einem NMA bzw. sogar DMA wird, der L erkennt. Definiere dazu lediglich

$$\mathcal{F} \;:=\; \{S \subseteq \{\mathsf{q_a}, \mathsf{q_b}, \mathsf{q_c}\} \mid \mathsf{q_a} \in S \Rightarrow \mathsf{q_b} \in S\}$$

als Endzustandskomponente. Somit wird dieser Automat zu einem DMA vom Index 6, der L erkennt.

In einem Muller-Automaten wird also einfach explizit tabuliert, welche Mengen von Zuständen unendlich oft durchlaufen weren dürfen. Im Gegensatz dazu kann man also die anderen Akzeptanzbedingungen als symbolische Repräsentation von solchen Mengen auffassen. Es ist also nicht verwunderlich, dass man mithilfe von Muller-Automaten die anderen Automaten simulieren kann.

Theorem 26. *Für jeden NBA / NRA / NSA / NPA $\mathcal{A}$ der Größe n existiert ein äquivalenter NMA $\mathcal{A}'$ der Größe n.*

Beweis. Der Trick ist, den Automaten an sich beizubehalten und lediglich die Akzeptanzbedingung in der neuen Endzustandskomponente explizit zu machen. Wir zeigen dies hier exemplarisch für den Fall der Rabin-Automaten. Die anderen Fälle sind analog bzw. folgen gleich aus Thm. 20.

Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \{(G_1, F_1), \ldots, (G_k, F_k)\})$ ein NRA. Definiere einen NMA $\mathcal{A}' := (Q, \Sigma, q_0, \delta, \mathcal{F})$, wobei für alle $S \subseteq Q$

$$S \in \mathcal{F} \text{ gdw. } \exists i \in \{1, \ldots, k\} \text{ mit } S \cap G_i \neq \emptyset \text{ und } S \cap F_i = \emptyset$$

Sei nun $w \in \Sigma^\omega$. Dann ist $w \in L(\mathcal{A})$ gdw. es einen Lauf ρ gibt, der akzeptierend bzgl. der Rabin-Bedingung $\{(G_1, F_1), \ldots, (G_k, F_k)\}$ ist. Sei $S = \mathit{Inf}(\rho)$. Dies bedeutet, dass es ein $i \in \{1, \ldots, k\}$ gibt, so dass $S \cap G_i \neq \emptyset$ und $S \cap F_i = \emptyset$. Somit ist $S \in \mathcal{F}$ und ρ ist damit auch akzeptierender Lauf von $\mathcal{A}'$ auf w. Die Rückrichtung gilt genauso. □

Obwohl die Muller-Bedingung auf den ersten Blick viel stärker als die Büchi-Bedingung z.B. aussieht, erkennen Muller-Automaten auch nur die ω-regulären Sprachen.

Theorem 27. *Für jeden NMA $\mathcal{A}$ der Größe n und vom Index k gibt es einen NBA $\mathcal{A}'$ der Größe höchstens $n + k \cdot n \cdot 2^n$ mit $L(\mathcal{A}') = L(\mathcal{A})$.*

Beweis. Sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ ein NMA mit $|Q| = n$ und $\mathcal{F} = \{F_1, \ldots, F_k\}$. Wir konstruieren einen äquivalenten NBA, der intuitiv wie folgt arbeitet. Er besteht aus $k + 1$ vielen Komponenten, die jeweils einen Teilgraphen von $\mathcal{A}$ darstellen. Eine dieser Komponenten ist lediglich eine Kopie von $\mathcal{A}$ und enthält

den Anfangszustand des NBA aber kein Endzustände. In dieser Komponente kann $\mathcal{A}'$ ein beliebiges Präfix eines Wort in $L(\mathcal{A})$ erkennen. Ab irgendeinem Zeitpunkt werden in einem akzeptierenden Lauf von $\mathcal{A}$ jedoch genau die Zustände in einem F_i durchlaufen. Dazu kann $\mathcal{A}'$ zu jedem Zeitpunkt nicht-deterministisch in eine der k unabhängigen Komponenten wechseln, die aus Kopien von $\mathcal{A}$, eingeschränkt auf ein jeweiliges F_i, bestehen. Dadurch ist gewährleistet, dass höchstens die Zustände aus einem F_i unendlich oft in einem Lauf vorkommen. Es muss aber auch noch sichergestellt werden, dass diese alle unendlich oft vorkommen. Dazu enthält jede dieser Komponenten einen Zähler in Form einer Zustandsmenge, in der die in dieser Komponente durchlaufenen Zustände protokolliert werden. Enthält dieses Protokoll das gesamte F_i, so signalisiert der NBA dies mit einem Endzustand wird und setzt es wieder auf $\emptyset$ zurück.

Formal sei $\mathcal{A}' := (Q', \Sigma, q_0, \delta', F)$ mit

$$Q' \;:=\; Q \cup \bigcup_{i=1}^{k} \{i\} \times F_i \times 2^{F_i} \;.$$

Das jeweilige $\{i\}$ ist natürlich unnötig, erleichtern jedoch die Notation der Transitionsfunktion.

$$\begin{aligned} \delta'(q,a) \;&:=\; \delta(q,a) \cup \{(i,q',\emptyset) \mid i \in \{1,\ldots,k\}, q' \in \delta(q,a) \cap F_i\} \\ \delta'((i,q,S),a) \;&:=\; \begin{cases} \{(i,q',S \cup \{q\}) \mid q' \in \delta(q,a) \cap F_i\} & \text{, falls } S \neq F_i \\ \{(i,q',\emptyset) \mid q' \in \delta(q,a) \cap F_i\} & \text{, falls } S = F_i \end{cases} \end{aligned}$$

Endzustände sind alle solche, in denen in einer der F_i-Komponenten protokolliert wurde, dass alle Zustände aus F_i durchlaufen wurden.

$$F \;:=\; \{(i,q,F_i) \mid i \in \{1,\ldots,k\}, q \in F_i\}$$

Es bleibt noch die Korrektheit dieser Konstruktion zu zeigen.

"$\supseteq$" Angenommen, $w \in L(\mathcal{A})$. Dann existiert ein Lauf $\rho = q_0, q_1, \ldots$ so dass $Inf(\rho) \in \mathcal{F}$, also $Inf(\rho) = F_i$ für ein $i \in \{1,\ldots,k\}$. D.h. es gibt ein $j \in \mathbb{N}$, so dass $q_h \in F_i$ für alle $h \geq j$ und zusätzlich gibt es für jedes $q \in F_i$ unendlich viele h mit $q_h = q$. Man vergewissere sich, dass

$$\rho' \;=\; q_0, q_1, \ldots, q_{j-1}, (i,q_j,\emptyset), (i,q_{j+1},\{q_j\}), (i,q_{j+2},\{q_{j+1},q_{j+2}\}), \ldots$$

ein Lauf von $\mathcal{A}'$ auf w ist. Dieser ist außerdem akzeptierend, weil er unendlich viele Zustände der Form (i,q',F_i) enthalten muss, da nach Voraussetzung alle Zustände aus F_i unendlich oft in ρ vorkommen und somit die Protokoll-Komponente der Zustände unendlich oft F_i sein muss.

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

Wenn man die Theoreme 20, 23, 26 und 27 und die Definition von ω-Regularität zusammennimmt, so ergibt sich, dass alle hier vorgestellten Automatentypen genau die ω-regulären Sprachen erkennen.

Proposition 12. *Für eine Sprache $L \subseteq \Sigma^\omega$ sind die folgenden Aussagen äquivalent.*

a) L ist ω-regulär.
b) L wird von einem NBA erkannt.
c) L wird von einem NRA erkannt.
d) L wird von einem NSA erkannt.
e) L wird von einem NPA erkannt.
f) L wird von einem NMA erkannt.

9.3 Co-Büchi-Automaten

Zum Abschluss dieses Kapitels betrachten wir noch das Komplement der Büchi-Bedingung als Akzeptanzbedingung.

Definition 27. *Ein* co-Büchi-Automat *(NcoBA) ist ein Tupel $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$, wie bei einem NFA, insbesondere mit $F \subseteq Q$ einer Menge von Endzuständen. Solch ein NcoBA heißt wiederum deterministisch (DcoBA), falls $|\delta(q, a)| = 1$ für alle $q \in Q$ und $a \in \Sigma$. Die Größe eines NcoBA ist ebenfalls die Anzahl $|Q|$ seiner Zustände.*

Ein Lauf $\rho = q_0, q_1, \ldots$ eines NcoBA heißt akzeptierend, falls es ein $i \in \{1, \ldots, k\}$ gibt, so dass $q_j \in F$ für alle $j \geq i$. Die Sprache $L(\mathcal{A})$ ist wieder die Menge aller Wörter, für die es einen akzeptierenden Lauf gibt.

Der Name co-Büchi-Automat stammt daher, dass ein Lauf $\rho = q_0, q_1, \ldots$ die co-Büchi-Bedingung erfüllt, also ab einem gewissen Moment nur noch Endzustände durchläuft, gdw. wenn dieser Lauf nicht-akzeptierend bzgl. der Büchi-Bedingung für die Menge $Q \setminus F$ ist. Das bedeutet, dass sich zumindest die deterministischen Varianten von Büchi und co-Büchi-Automaten leicht ineinander komplementieren lassen.

Theorem 28. *Für jeden DBA / DcoBA $\mathcal{A}$ der Größe n existiert ein DcoBA / DBA $\overline{\mathcal{A}}$ der Größe n, so dass $L(\overline{\mathcal{A}}) = \Sigma^\omega \setminus L(\mathcal{A})$.*

Beweis. Übung.

Ein wesentlicher Unterschied zwischen Büchi- und co-Büchi-Automaten besteht darin, dass sich letztere determinisieren lassen.

Theorem 29. *Für jeden NcoBA $\mathcal{A}$ mit n Zuständen existiert ein DcoBA $\mathcal{A}'$ mit höchstens 3^n vielen Zuständen, so dass $L(\mathcal{A}') = L(\mathcal{A})$.*

Wir präsentieren den Beweis an späterer Stelle in einem anderen Kontext. Interessanterweise lässt sich die sogenannte Miyano-Hayashi-Konstruktion, die ursprünglich dafür konzipiert wurde, einen alternierenden Büchi-Automaten in einen nicht-deterministischen Büchi-Automaten zu überführen, auch dazu verwenden, co-Büchi-Automaten zu determinisieren.

Es stellt sich die Frage nach dem relativen Zusammenhang zwischen der Klasse der von co-Büchi erkannten bzw. ω-regulären Sprachen. Man erkennt recht leicht, dass letztere mindestens so mächtig wie erstere sein muss, da sich die co-Büchi-Bedingung leicht als Muller-Bedingung formulieren lässt. Dies führt aber zu einem unnötigen Blow-Up bei der weiteren Übersetzung in einen NBA. Andererseits kann man aber auch mit demselben Trick wie im Beweis von Thm. 27 einen NcoBA direkt in einen NBA übersetzen.

Theorem 30. *Für jeden NcoBA $\mathcal{A}$ der Größe n existiert ein NBA $\mathcal{A}'$ der Größe höchstens $2n$, so dass $L(\mathcal{A}') = L(\mathcal{A})$.*

Beweis. Übung.

Andererseits lässt sich nicht jeder NBA äquivalent in einen NcoBA umwandeln.

Theorem 31. *Es gibt ω-reguläre Sprachen, die nicht von einem NcoBA erkannt werden.*

Beweis. Betrachte die Sprache $L = \{w \in \{a,b\}^\omega \mid w \text{ enthält unendlich viele } b\}$. Diese ist offensichtlich ω-regulär. Sei andererseits angenommen, dass diese auch von einem *NcoBA* erkannt würde. Dann würde sich laut Thm. 29 auch von einem DcoBA erkannt, und nach Thm. 28 würde ihr Komplement $\overline{L}$ auch von einem DBA erkannt. Dies ist aber laut Thm. 17 nicht der Fall. □

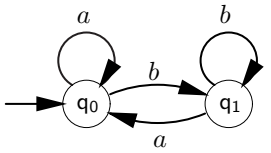
10

Determinisierung von Büchiautomaten

10.1 Das fundamentale Problem der Determinisierung

Der Titel dieses Kapitels klingt paradox; denn in Abschnitt 7.2 haben wir gesehen, dass deterministische Büchi-Automaten echt schwächer als nichtdeterministische sind. Es gibt ω-reguläre Sprachen, die nicht von einem DBA erkannt werden können. Somit muss es also NBA geben, die sich nicht determinisieren lassen. Die Antwort liegt in der Akzeptanzbedingung. Wie wir sehen werden, sind deterministische Rabin-, Muller oder Paritätsautomaten ausdrucksstark genug für diese Aufgabe. Dasselbe gilt zwar auch für Streett-Automaten. Wir präsentieren hier jedoch zunächst die berühmt gewordene — weil erste — Determinisierungsprozedur von Safra [**?**], die aus einem NBA einen DMA macht, den man auch leicht als DRA auffassen kann. Danach zeigen wir, wie man durch leichte Modifikation dessen sogar gleich einen DPA aus einem NBA machen kann. Dies ist die sogenannte Piterman-Konstruktion [**?**].

Beispiel 15. Sei $L = \{w \in \{a,b\}^\omega \mid |w|_b < \infty\}$ die aus Thm. 17 bekannte Sprache, welche nicht von einem DBA erkannt werden kann. Sie wird jedoch von dem Automaten



erkannt, wenn man als Akzeptanzbedingung formuliert, dass ein unendlicher Lauf den Zustand q_0 unendlich oft besucht, den Zustand q_1 jedoch nur endlich oft. Dies ist leicht als Rabin-Bedingung $\mathcal{F} = \{(\{\mathsf{q}_0\}, \{\mathsf{q}_1\})\}$ oder auch als Paritätsbedingung mit $\Omega(\mathsf{q}_0) = 0$ und $\Omega(\mathsf{q}_1) = 1$ formalisierbar.

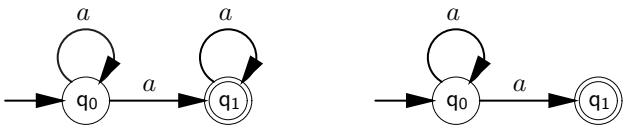
Bevor wir die *Safra-Konstruktion* kennenlernen, beantworten wir noch eine offensichtliche Frage: Kann man nicht die bei endlichen Automaten auf

endlichen Wörtern wohlbekannte Potenzmengenkonstruktion, die einen NFA in einen DFA umwandelt, bei NBA ebenso anwenden? Die Antwort ist nein. In naivster Variante würde diese zu einem gegebenen NBA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ einen deterministischen Automaten konstruieren, dessen Zustandsmenge 2^Q ist und dessen Transitionsfunktion definiert ist als

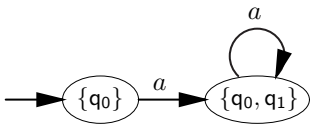
$$\delta'(S, a) \;:=\; \bigcup_{q \in S} \delta(q, a)$$

für jedes $a \in \Sigma$. Es stellt sich die Frage, wie man geeignet die Akzeptanzbedingung festelegen kann, so dass der resultierende Automat genau $L(\mathcal{A})$ erkennt. Das dies nicht geht, zeigt folgendes Beispiel.

Betrachte die beiden NBAs $\mathcal{A}_1$ (links) und $\mathcal{A}_2$ (rechts) über dem einelementigen Alphabet $\Sigma = \{a\}$.



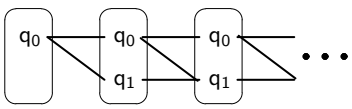
Es gilt offensichtlich $L(\mathcal{A}_1) = \{a^\omega\}$ und $L(\mathcal{A}_2) = \emptyset$, da in $\mathcal{A}_2$ kein Lauf unendlich oft einen Endzustand enthalten kann. Dennoch entsteht bei der Potenzmengenkonstruktion aus beiden Automaten derselbe folgende Automat.



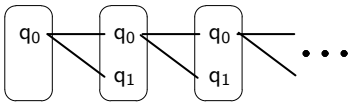
Es ist klar, dass sich auf diesem keine Akzeptanzbedingung definieren lässt — egal ob dies eine Rabin- oder Paritäts- oder sonstige Form hätte – so dass der resultierende Automat einerseits $\{a^\omega\}$, andererseits aber auch $\emptyset$ erkennt.

Dieses Beispiel eignet sich nicht nur gut dazu zu zeigen, dass Potenzmengenkonstruktion für NBAs nicht funktionieren kann, sondern auch, um das fundamentale Problem der Determinisierung von Büchi-Automaten zu verdeutlichen. Wir verfolgen einmal die Zustände q_0 und q_1 durch die Makrozustände des Potenzmengenautomaten auf dem einzigen Wort a^ω. Es ist klar, dass — mit Ausnahme des Anfangszustands — jeder Makrozustand einen der Mikrozustände q_0 oder q_1 nur enthält, weil der Zustand davor einen gewissen enthalten hat. In anderen Worten: Kein Zustand tritt urplötzlich in einem Makrozustand auf, ohne dass es nicht einen Grund dafür gegeben hätte in Form eines Vorgängers im Vorgängermakrozustand.

Für den aus $\mathcal{A}_1$ entstehenden Potenzmengenautomaten ergibt sich das folgende Diagramm.



Man erkennt, dass der Zustand q_1 im jeweiligen Makrozustand enthalten ist, weil er sowohl wegen q_0 als auch q_1 im Makrozustand davor hineinkommt. Für den aus $\mathcal{A}_2$ entstehenden Potenzmengenautomaten ergibt sich jedoch das folgende Diagramm.



Der Unterschied ist der folgende. Der erste Lauf der Makrozustände enthält einen internen Pfad, auf dem unendlich oft der Endzustand q_1 vorkommt. Der zweite Lauf enthält dies nicht. Daher sollte der erste akzeptierend sein, der zweite jedoch nicht. Da beide jedoch dieselben Makrozustände durchlaufen, muss eine Akzeptanzbedingung entweder verfeinert anhand der Übergänge zwischen den Makrozuständen definniert werden, oder die Determinisierungsprozedur muss andere Makrozustände verwenden, die diesen Unterschied deutlich und anhand einer der bekannten Akzeptanzbedingungen definierbar machen. Das fundamentale Problem bei der Determinisierung von Büchi-Automaten besteht also darin, akzeptierende Läufe des nicht-deterministischen Büchi-Automaten in Form von internen Pfaden o.ä. aufzufinden.

10.2 Die Safra-Konstruktion

Für den Rest dieses Abschnitts sei $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ ein NBA. Wir wollen einen deterministischen Mullerautomaten $\mathcal{M}$ konstruieren, sodass $L(\mathcal{A}) = L(\mathcal{M})$. Es ist nützlich, sich den Mullerautomaten $\mathcal{M}$ imperativ vorzustellen: In diesem Sinne ist der momentane Zustand von $\mathcal{M}$ ein Baum von Prozessen, wobei ein Prozess besteht aus

- einer eindeutigen Nummer PID,
- einer Menge von $\mathcal{A}$-Zuständen,
- einem Zeiger auf den Vaterprozess (außer beim Wurzelprozess) und Zeigern auf die Kinderprozesse, falls vorhanden,
- einer Lampe.

Zusätzlich wird eine lineare Ordnung aller momentan aktiven Prozesse mitgeführt, sodass man feststellen kann, welcher zuerst erzeugt wurde.

Am Anfang gibt es nur einen Prozess. Dieser ist beschriftet mit $\{q_0\}$. Das Fortschalten beim Lesen eines Symbols a erfolgt durch sukzessives Ausführen der folgenden Schritte:

1. Sollte ein Prozess Endzustände enthalten, so wird ein neuer Kindprozess erzeugt. Dieser ist beschriftet mit diesen Endzuständen. Die Endzustände verbleiben aber auch beim Elternprozess.
2. Jeder Prozess (einschließlich der neu erzeugten) wird gemäß der Potenzmengenkonstruktion weitergeschaltet. Man ersetzt also seine Beschriftung B durch $\bigcup_{q \in B} \delta(q, a)$.

3. Die Beschriftungen der Kinder sollen disjunkt sein: kommt ein Zustand in beiden vor, so muss der jüngere Prozess verzichten. Sollte dabei die Beschriftung leer werden, so wird der Prozess samt seinen (ebenfalls leeren) Kindern und Kindeskindern entfernt.
4. Sollten die Beschriftungen aller Kinder die Beschriftung des Elternknotens ergeben, so werden alle Kinder und Kindeskinder entfernt und die Lampe des Elternknotens kurz aufgeblitzt.

Ein Wort ist akzeptiert, wenn im entsprechenden Lauf ein Knotenname ab einem gewissen Zeitpunkt immer vorhanden ist (nicht entfernt wird) und immer wieder blitzt.

Die folgenden Invarianten lassen sich beobachten:

1. Die Beschriftungen der Kinder bilden zusammen immer eine echte Teilmenge der Beschriftung des Elternknotens.
2. Die Wurzelbeschriftung entspricht der naiven Potenzmengenkonstruktion.

Formal ist die Zustandsmenge von $\mathcal{M}$ als die Menge der *Safrabäume* erklärt:

Definition 28. *Ein Safrabaum über Q ist ein Tupel $(K, <, p, l, !)$, wobei folgendes gilt.*

- *Für die Knotenmenge K gilt $K \subseteq \{1, \ldots, 2|Q|\}$.*
- *Die Funktion $p : K \to K$ ordnet jedem Knoten x außer der Wurzel seinen Vaterknoten $p(x)$.*
- *"$<$" ist eine lineare Ordnung auf K. ($k_1 < k_2$ bedeutet, dass k_1 älter als k_2 ist.) Der Wurzelnoten r ist immer der älteste (kleinste) bzgl. $<$.*
- *Die Beschriftung der Knoten ist gegeben als Funktion $l : K \to 2^Q$, so dass für alle $k, k' \in K$ und Wurzel r gilt:*
  - *Wenn $p(k) = p(k')$ dann $l(k_1) \cap l(k_2) = \emptyset$. D.h. Bruderknoten haben disjunkte Beschriftungen.*
  - *Wenn $k \neq r$, so ist $l(k) \subseteq l(p(k))$. D.h. die Beschriftung eines Knotens ist eine Teilmenge der Beschriftung seines Vaters.*
  - *Wenn $k \neq r$, so ist $\bigcup_{k':p(k')=k} l(k') \subsetneq l(k)$. D.h. ein Vaterknoten enthält mehr Zustände in seiner Beschriftung als alle seine Söhne zusammen.*
- *Die Funktion $! : K \to \{\mathtt{tt}, \mathtt{ff}\}$ zeichnet einzelne Knoten aus, so dass für alle $k \in K$ gilt: Wenn $!k$, dann hat k keine Kinder (es gibt kein k' mit $p(k') = k$).*

Zuerst beobachten wir, dass ein Safrabaum nicht alle $2|Q|$ zur Verfügung stehenden Knotennamen ausschöpfen kann. Dies liegt daran, dass Vaterknoten in ihrer Beschriftung immer mehr Zustände tragen müssen als die Vereinigung aller ihrer Sohnknoten hat. Somit gibt es zu jedem $q \in Q$ höchstens einen (und damit eindeutigen) tiefsten Knoten in einem Safrabaum, der q enthält.

Lemma 20. *Sei t ein Safrabaum über Q. Dann hat t höchstens $|Q|$ Knoten.*

Beweis. Übung.

Eine Konsequenz daraus ist, dass also stets mindestens $|Q|$ Knotennamen unbenutzt sind.

Mit $S(Q)$ bezeichnen wir die Menge der Safrabäume über Q. Als nächstes schätzen wir deren Anzahl nach oben hin ab.

Lemma 21. *Sei Q Zustandsmenge mit $|Q| = n$. Dann ist $|S(Q)| = 2^{\mathcal{O}(n \log n)}$.*

Beweis. Ein Safrabaum ist eindeutig bestimmt durch

- die Menge $K \subseteq \{1, \ldots, 2n\}$ mit $|K| \leq n$.
- eine Funktion $f : Q \to \{\bot, 1, \ldots, 2n\}$, die jedem Zustand einen eindeutigen Knoten zuordnet, in dem er vorkommt,
- die Elternfunktion $p : \{1, \ldots, n\} \to \{\bot, 1, \ldots, 2n\}$,
- die Bruderfunktion $s : \{1, \ldots, n\} \to \{\bot, 1, \ldots, 2n\}$,

Beachte, dass Eltern- und Bruderfunktion nur auf K definiert sein müssen. Da es hier nur um die Abschätzung der Anzahl verschiedener Bäume geht, können wir davon ausgehen, dass diese lediglich auf den Knoten $\{1, \ldots, n\}$ definiert sind. Außerdem kann man sich auch auf den Fall beschränken, dass K genau n statt nur höchstens n viele Knoten hat, denn unbenutzte Knotennamen kann man ja als Dummyknoten sich irgendwo im Baum vorstellen.

Die Anzahl der möglichen Safrabäume über Q ist somit beschränkt durch

$$\binom{2n}{n} \cdot ((2n+1)^n)^3 \quad = \quad \frac{(2n)!}{(n!)^2} \cdot (2n+1)^{3n} \quad = \quad 2^{O(n \log n)}$$

□

Die Safra-Konstruktion verwendet Safra-Bäume als Zustände eines deterministischen Automaten. Wir müssen also erklären, wie man von einem Safra-Baum unter Einlesen eines Alphabetsymbols zum eindeutigen nächsten Zustand kommt. Wir halten weiterhin den NBA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ fest und definieren nun eine Transitionsfunktion auf Safra-Bäumen über Q.

Definition 29. *Die Übergangsfunktion $\Delta : S(Q) \times \Sigma \to S(Q)$ ist wie folgt definiert. Sei t ein Safrabaum, a ein Eingabesymbol. Der Safrabaum $t' = (K', p', l', !') = \Delta(t, a)$ wird aus $t = (K, p, l, !)$ gemäß der folgenden Schritte in der hier gegebenen Reihenfolge berechnet:*

1. *Ist $l(k) \cap F \neq \emptyset$ für einen Knoten k, so führe ein neues Kind mit frischem Namen ein und beschrifte es mit $l(k) \cap F$. Adjustiere die Altersrelation, sodass das neue Kind der jüngste Knoten wird.*
2. *Wende die Potenzmengenkonstruktion auf alle Knotenbeschriftungen an, d.h., ersetze $l(k)$ durch $\bigcup_{q \in l(k)} \delta(q, a)$ für alle Knoten k inklusive der in Schritt 1 erzeugten neuen Knoten.*

3. *Erscheint ein Zustand in mehreren Kindern desselben Knotens, so belasse ihn nur im ältesten derselben, d.h. entferene ihn aus allen anderen und damit auch aus deren Kindern, falls er dort vorhanden ist. Entferne Knoten, die dabei leer werden.*
4. *Entsprechen die Beschriftungen aller Kinder eines Knotens k der Beschriftung von k und ist diese nicht leer, so entferne die Kinder und setze $!'(k) = \mathtt{tt}$.*
5. *Für alle anderen Knoten setze $!'(k) = \mathtt{ff}$, selbst wenn $!(k) = \mathtt{tt}$.*

Damit können wir nun den deterministischen Muller-Automaten $\mathcal{M}$, der dieselbe Sprache wie $\mathcal{A}$ erkennt, definieren. Die Korrektheit der Konstruktion ist natürlich noch zu zeigen.

Definition 30. *Wir definieren den zu $\mathcal{A}$ gehörigen deterministischen Muller-Automaten $\mathcal{M}$ als $(S(Q), \Sigma, t_0, \Delta, \mathcal{F})$, wobei*

- *$S(Q)$ die Menge der Safrabäume über Q wie oben definiert ist,*
- *der Safrabaum $t_0 = (\{1\}, l, p, !)$ mit $l(1) = \{q_0\}$, $p(1) = 1$, $!(1) = \mathtt{ff}$ den Anfangszustand darstellt,*
- *die deterministische Transitionsfunktion Δ wie oben definiert ist,*
- *$\mathcal{F}$ aus der Menge aller Mengen M von Safrabäumen besteht, so dass es einen Knotennamen k gibt, der in allen Bäumen aus M vorhanden ist und ein Baum in M ist, in dem $!(k) = \mathtt{tt}$ gilt.*

Für beliebige Zustände q, q' eines Automaten und ein endliches Wort w über dem gleichen Alphabet schreiben wir $q \overset{w}{\Longrightarrow} q'$, falls der Automat unter Lesen des Wortes w vom Zustand q in $|w|$ vielen Schritten in den Zustand q' übergehen kann. Dabei gehen wir davon aus, dass der Automat aus dem Kontext ersichtlich ist. Handelt es sich dabei um einen NBA, so schreiben wir zusätzlich noch $q \overset{w}{\Longrightarrow}_{\mathsf{fin}} q'$, falls solch ein Übergang möglich ist, wobei zwischendurch mindestens ein Endzustand durchlaufen wird.

Als nächstes wollen wir zeigen, dass die Safra-Konstruktion korrekt ist, d.h. dass $L(\mathcal{M}) = L(\mathcal{A})$ gilt. Dazu machen wir zunächst drei wichtige Beobachtungen.

1. Gilt $w : t_0 \to t$ im Mullerautomaten, so auch $w : q_0 \to q$ für jeden Zustand q der in t erwähnt wird.
2. Gilt $w : t \to t'$ in $\mathcal{M}$ und hat k in t keine Kinder und bleibt k im gesamten Lauf bis t' am Leben und hat k in t' dann ein Kind n, so existiert für jeden Zustand $q' \in l(n)$ ein Zustand $q \in l(k)$ mit $w : q \to_F q'$.
3. Gilt $w : t \to t'$ in $\mathcal{M}$ und hat k in t keine Kinder und bleibt k im gesamten Lauf bis t' am Leben und ist $!(k) = \mathtt{tt}$ in t', so existiert für jeden Zustand $q' \in l(k)$ ein Zustand $q \in l(k)$ mit $w : q \to_F q'$.

Die dritte Beobachtung folgt unmittelbar aus der zweiten.

Lemma 22 (Lauflemma). *Seien $U_0 = q_0, U_1, U_2, \ldots$ Teilmengen von Q, seien $u, v_1, v_2, v_3, \ldots$ endliche Wörter, sodass gilt:*

- *für alle* $q \in U_1$ *gilt* $u : q_0 \to q$
- *für alle* $q' \in U_{i+1}$ *existiert* $q \in U_i$*, sodass* $v_i : q \to_F q'$

Dann ist $uv_1v_2v_3 \cdots \in L(\mathcal{A})$.

Beweis. Konstruiere einen Baum, dessen Knoten mit Zuständen beschriftet sind. An der Wurzel steht q_0, die Knoten der Tiefe i sind mit den Zuständen aus U_i beschriftet. Zustand q' auf Niveau $i+1$ ist Kind von Zustand q auf Niveau i, wenn gilt $v_i : q \to_F q'$. Der Baum ist unendlich, da auf jedem Niveau Knoten vorhanden sind. Somit gibt es nach Königs Lemma einen unendlichen Pfad, der einen akzeptierenden Lauf in $\mathcal{A}$ definiert.

Proposition 13. $L(\mathcal{M}) \subseteq L(\mathcal{A})$.

Beweis. Sei $w \in \Sigma^\omega$ und $w \in L(\mathcal{M})$. Sei k ein Knotenname k, der im Lauf von $\mathcal{M}$ auf w ab einem gewissen Zeitpunkt immer präsent ist und unendlich oft markiert ist ("blitzt").

Sei t_i der Safrabaum beim i-ten Blitzen von k nach k's letztmaligem Verschwinden. Sei U_i die Beschriftung von k in t_i und v_i das Segment von w, das von t_i bis t_{i+1} abgearbeitet wird. Sei weiter u das Präfix von t, das bis zu t_1 abgearbeitet wird. Das Lauflemma zusammen mit der Folgerung liefert $w \in L(\mathcal{A})$.

Proposition 14. $L(\mathcal{A}) \subseteq L(\mathcal{M})$.

Beweis. Sei $w \in \Sigma^\omega$ und $r = q_0q_1q_2 \ldots$ ein akzeptierender Lauf von $\mathcal{A}$ auf w. Der Lauf wird auf jeden Fall in der Wurzel nachgebildet. Falls die Wurzel unendlich oft blitzt, dann sind wir fertig. Ansonsten sei f ein Endzustand, der unendlich oft in r vorkommt. Betrachte das erste Vorkommen von f in r nach dem letztmaligen Blitzen der Wurzel. Zu diesem Zeitpunkt wird ein neues Kind der Wurzel eingerichtet, dessen Beschriftung f enthält. Wir verfolgen den Lauf nun in diesem Kind. Es könnte sein, dass ein älteres Kind irgendwann (möglicherweise gleich am Anfang) den Lauf von diesem Kind übernimmt oder das Kind gar verschwindet. Dann konzentrieren wir uns entsprechend auf dieses ältere Kind usw. und finden auf diese Weise ein Kind der Wurzel, in dem der restliche Lauf nachgebildet wird. Sollte auch dieses nicht immer wieder blitzen, so finden wir mit demselben Argument unterhalb von ihm ein Kind, das nicht mehr verschwindet und den gesamten Restlauf nachbildet etc. Nachdem aber die Safrabäume eine beschränkte Tiefe haben, kann das nicht ewig so weitergehen und wir finden einen Knoten, der am Leben bleibt und immer wieder blitzt.

Etwas formaler argumentieren wir wie folgt: mit $t_0, t_1, t_2, \ldots$ bezeichnen wir die Folge der Safrabäume im (eindeutig bestimmten) Lauf von $\mathcal{M}$ auf w. Wir schreiben $t_i = (K_i, <_i, p_i, l_i, !_i)$. Ein Knoten k heiße *persistent*, wenn er ab einem gewissen Zeitpunkt i nicht mehr verschwindet, also $k \in K_j$ für alle $j \geq i$. Der *Beginn* $B(k)$ eines persistenten Knotens k ist das kleinste i, sodass $k \in K_j$ für alle $j \geq i$. Sind k, k' persistent und gilt $k <_j k'$ für ein

$j \geq \max(B(k), B(k'))$, so folgt $k <_j k'$ für alle $j \geq \max(B(k), B(k'))$ und wir schreiben dann $k < k'$. Hierdurch werden die persistenten Knoten total geordnet. Es gilt im übrigen $k < k' \iff B(k) < B(k')$.

Sind k, k' persistent und gilt $k = p_j(k')$ für ein $j \geq \max(B(k), B(k'))$, so folgt $k = p_j(k')$ für alle $j \geq \max(B(k), B(k'))$ und wir schreiben dann $k = p(k')$. Hierdurch bilden die persistenten Knoten einen (endlichen!) Baum. Die Tiefe $H(k)$ eines persistenten Knotens ist die Tiefe von k in diesem Baum, also sein Abstand von der Wurzel.

Ein persistenter Knoten k *trifft den Lauf*, wenn $j \geq B(k)$ existiert mit $q_j \in l_j(k)$. Wir bemerken, dass die Wurzel 1 den Lauf trifft.

Unter allen persistenten Knoten, die den Lauf treffen, suchen wir nun einen mit maximaler Tiefe und unter allen, die diese maximale Tiefe erreichen, bestimmen wir den ältesten und bezeichnen ihn mit k_0. Wir behaupten nun: k_0 blitzt unendlich oft, also $!_i(k_0) = \mathtt{tt}$ für unendlich viele i.

Sei $q_i \in l_i(k_0)$. Nachdem k_0 der älteste Knoten auf diesem Niveau ist, gilt auch $q_j \in l_j(k_0)$ für alle $j \geq i$. (Man muss sich hier kurz überlegen, dass es nicht möglich ist, dass k_0 von einem nicht-persistenten Knoten etwas weggenommen wird.)

Zu jedem $j_1 \geq i$ findet man $j_2 \geq j_1$ mit $q_{j_2} = f \in l_{j_2}(k_0)$, da ja f immer wieder kommt. Zu diesem Zeitpunkt wird ein Kind von k_0 erzeugt, wessen Beschriftung f enthält. Würde k_0 ab diesem Zeitpunkt nicht mehr blitzen, so bliebe dieses oder ein älteres Kind am Leben und träfe den Lauf im Widerspruch zur Extremalität von k_0.

Zusammengefasst haben wir nun bewiesen:

Proposition 15 (Safra). *Zu jedem Büchiautomaten $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ mit $|Q| = n$ existiert ein deterministischer Mullerautomat $\mathcal{M}$ mit $2^{O(n \log n)}$ Zuständen mit $L(\mathcal{A}) = L(\mathcal{M})$.*

Dieser Satz liefert einen alternativen Zugang zum Satz von Büchi (Entscheidbarkeit der MSO), da man einen deterministischen Mullerautomaten sehr leicht komplementieren kann: man ersetzt einfach die Akzeptanzbedingung $\mathcal{F}$ durch ihr Komplement. Man kann somit einer MSO-Formel ϕ induktiv einen deterministischen Mullerautomaten zuordnen. Beim Behandeln des Existenzquantors muss man dann aber den Umweg über einen nichtdeterministischen Büchiautomaten gehen, der dann wieder mithilfe der Safrakonstruktion zu determinisieren ist.

Für einen Rabinautomaten $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ mit $\mathcal{F} = \{(G_1, F_1), \ldots, (G_k, F_k)\}$ sei k der *Index* von $\mathcal{A}$.

Korollar 3. *Zu jedem Büchiautomaten $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ mit $|Q| = n$ existiert ein deterministischer Rabinautomat $\mathcal{M}$ mit $2^{O(n \log n)}$ Zuständen und Index höchstens $2n$, so dass $L(\mathcal{A}) = L(\mathcal{M})$ gilt.*

Beweis. Der deterministische Mullerautomat $\mathcal{M}$ aus der Safrakonstruktion läßt sich als Rabinautomat auffassen. Dazu sei lediglich

$$\begin{aligned} G_i &:= \{t \in S(Q) \mid \text{ der Knoten } i \text{ blitzt in } t \} \\ F_i &:= \{t \in S(Q) \mid \ t \text{ enthält den Knoten } i \text{ nicht } \} \end{aligned}$$

für $i = 1, \ldots, 2n$.

Proposition 16 (Safra). *Es gibt Büchi-Automaten $\mathcal{A}_n$ mit $O(n)$ vielen Zuständen, für die es keine deterministischen Rabinautomaten mit nur $2^{O(n)}$ vielen Zuständen und Index $O(n)$ gibt.*