Automatentheorie und ihre Anwendungen
Teil 5: Alternierung

Wintersemester 2017/18 Thomas Schneider

AG Theorie der künstlichen Intelligenz (TdKI)

http://tinyurl.com/ws1718-automaten

Warum Alternierung?

- Starke Beziehungen zwischen Logik und Automaten, z. B.:
  - NBAs ↔ LTL (Teil 3 dieser Vorlesung)
  - NEAs ↔ S1S (Satz von Büchi-Elgot-Trakhtenbrot, VL Logik)
- In Logiken kann man aber Sprachen oder Eigenschaften oft deutlich kürzer ausdrücken, z. B.:
  - LTL-Formel → NBA: exponentielle Explosion
  - S1S-Formel → NEA: sogar nicht-elementare Explosion
- Verkleinern dieser Lücke:

  Erlaube in Automaten nicht nur **existenzielle** (= nichtdeterm.) „Verzweigungen“, sondern auch **universelle**.

Warum Alternierung?

- „Alternierung“ heißt also, dass ein Maschinenmodell (abwechselnd) existenzielle und universelle Entscheidungen treffen kann.
- Alternierende Varianten gibt es für alle Automatentypen aus dieser Vorlesung (auf endlichen oder unendlichen Objekten, Wörtern oder Bäumen) und für andere Maschinenmodelle (z. B. Turingmaschinen).
- Für alternierende Automaten ist Komplementierung besonders leicht zu zeigen.
- Wir beschränken uns im Folgenden auf ω-**Wort**automaten, also auf alternierende Büchi-Automaten.

Überblick

Und nun ...

1. Einführung und Grundbegriffe
2. Von LTL zu alternierenden Automaten
3. Komplementierung
4. Übersetzung zu nichtdeterministischen Automaten

Alternierung: Grundidee

- Nichtdeterministischer Automat $\mathcal{A}$ akzeptiert eine Eingabe, wenn ein akzeptierender Run **existiert**.

  d. h.: falls (q, a, q'), $(q, a, q'') \in \Delta$, kann $\mathcal{A}$ in Situation (q, a) „entscheiden", wie der Run fortgesetzt wird.

  Mindestens eine dieser Entscheidungen muss zum Ziel führen.
- Alternierung erlaubt auch **universelle Entscheidungen**, in beliebiger Kombination mit existenziellen.
- „Beliebige Kombination" wird realisiert durch **positive Boolesche Formel**, d. h. aussagenlogische Formel ohne $\neg$.
- Statt eines Runs (Zustandsfolge) gibt es nun einen **Run-Baum**, der alle universellen Entscheidungen berücksichtigt.

Positive Boolesche Formeln

Definition 5.1 (Syntax)

Die Menge der **positiven Booleschen Formeln (PBFs)** über einer Menge X, geschrieben $\mathbf{B^+(X)}$, ist die kleinste Menge, für die gilt:

- Jedes Element $x \in X$ ist eine PBF.
- Die Konstanten $0, 1$ sind PBFs.
- Wenn φ, ψ PBFs sind, dann auch $\varphi \wedge \psi$ und $\varphi \vee \psi$.

Definition 5.2 (Semantik)

Jede Menge $Y \subseteq X$ definiert eine **Belegung** $\mathbf{V_Y} : X \to \{0, 1\}$:

$V(x) = 1$, falls $x \in Y$, und $V(x) = 0$ sonst.

Eine Menge $Y \subseteq X$ **erfüllt** eine PBF $\varphi \in \mathrm{B}^+(X)$, geschrieben $\mathbf{Y \models \varphi}$, wenn $V_Y \models \varphi$ (nach Standard-Semantik AL).

T 5.1

Alternierende Automaten

Definition 5.3

Ein **alternierender Büchi-Automat** auf ω-Wörtern **(ABA)** ist ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \delta, I, F)$, wobei

- Q eine endliche nichtleere **Zustandsmenge** ist,
- Σ eine **Alphabet** (endliche nichtleere Menge von Zeichen) ist,
- $\delta : Q \times \Sigma \to \mathrm{B}^+(Q)$ die **Überführungsfunktion** ist,
- $I \subseteq Q$ die Menge der **Anfangszustände** ist,
- $F \subseteq Q$ die Menge der **akzeptierenden Zustände** ist.

Wir nehmen wieder o. B. d. A. $I = \{q_I\}$ an.

Alternative Akzeptanzbedingungen (Muller, Parität usw.) sind auch möglich.

Run-Bäume

Betrachten **Baum mit Verzweigungsgrad $\leq n$**, für festes $n \in \mathbb{N}$

- Positionen: Menge $P \subseteq \{1, \dots, n\}^*$, präfix-abgeschlossen
- **Kinder** eines Knotens p: **Kinder**$(p) \subseteq \{p1, \dots, pn\}$
- **Tiefe, Ebene, Nachfolger, Pfad:** wie gehabt

Pfad in P: endliche oder unendliche Folge $\pi = \pi_0\pi_1\pi_2\cdots$ von **Positionen** $\pi_i \in P$ mit

- $\pi_0 = \varepsilon$ und
- $\pi_{i+1} \in \text{Kinder}(\pi_i)$ für alle $i \leq 0$

Σ-Baum (P, t) (Alphabet Σ):

- P wie oben
- $t : P \to \Sigma$ ist Markierungsfunktion **T 5.2**

Berechnungen und Akzeptanz

Definition 5.4

Ein **Run** eines ABA $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, F)$ auf einem Wort $\alpha = \alpha_0\alpha_1\alpha_2\cdots \in \Sigma^\omega$ ist ein ***Q*-Baum** (P, r), so dass:

- $r(\varepsilon) = q_I$
- für alle $p \in P$: wenn $r(p) = q$, dann

$$\{r(p') \mid p' \in \text{Kinder}(p)\} \models \delta(q, \alpha_{|p|}) \,. \qquad \textbf{T 5.3}$$

Run (P, r) ist **akzeptierend**, wenn für **jeden unendlichen** Pfad $\pi = \pi_0\pi_1\pi_2\dots$ in P gilt:

$$\text{Inf}(r(\pi_0)r(\pi_1)r(\pi_2)) \cap F \neq \emptyset \qquad \textbf{T 5.3 Forts.}$$

$L_\omega(\mathcal{A})$ $= \{\alpha \in \Sigma^\omega \mid \mathcal{A}$ hat einen akzept. Run auf $\alpha\}$ **T 5.3 Forts.**

(für andere Akzeptanzbedingungen analog)

Und nun . . .

Vorbetrachtungen

Übersetzung logischer Formeln in alternierende Automaten ist oft einfacher als in nichtdeterministische Automaten.

Hier am Beispiel LTL $\to$ ABA

Erinnerung an LTL: $\varphi ::= x \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid X\varphi \mid \varphi\, U\, \varphi$
mit $x \in$ AV (Aussagenvariablen)

$s, i \models \varphi\, U\, \psi$, falls $s, j \models \psi$ für ein $j \geqslant i$
und $s, k \models \varphi$ für alle k mit $i \leqslant k < j$

$F\varphi \equiv (x \vee \neg x)\, U\, \varphi$
$G\varphi \equiv \neg F \neg\varphi$

Expansionsgesetz:
$s, i \models \varphi\, U\, \psi$ **gdw.** $s, i \models \psi$ oder $(s, i \models \varphi$ und $s, i{+}1 \models \varphi\, U\, \psi)$

Intuitionen der Konstruktion

Seien φ eine LTL-Formel und ψ eine beliebige Teilformel.

$$\sim\!\psi = \begin{cases} \vartheta & \text{falls } \psi = \neg\vartheta \\ \neg\psi & \text{sonst} \end{cases}$$

cl(φ) $= \{\psi, \sim\!\psi \mid \psi$ ist Teilformel von $\varphi\}$

Bestandteile des ABA $\mathcal{A}_\varphi$

- Eingabealphabet: $\Sigma = 2^{\mathsf{AV}}$ wie gehabt
- Zustände: für jede Formel $\psi \in \mathrm{cl}(\varphi)$ ein q_ψ; Startzustand q_φ
- Übergänge:
  - für $\wedge, \vee$: mittels PBF
  - für $\neg$: per „Negation" der PBF
  - für $X\psi$: schicke q_ψ zur nächsten Position
  - für U: per Expansionsgesetz
- F verhindert unendliches „Aufschieben" von U-Teilformeln!

„Negation von PBFs"

Idee: Nutzen stattdessen Dualität von $\wedge, \vee$ (de Morgan), um Negation nach innen zu ziehen.

Negation eines Atoms q_ψ ist dann $q_{\sim\psi}$.

Genauer: mittels Operator $\overline{}$ wie folgt:

$$\begin{aligned} \overline{\zeta_1 \wedge \zeta_2} &= \overline{\zeta_1} \vee \overline{\zeta_2} \\ \overline{\zeta_1 \vee \zeta_2} &= \overline{\zeta_1} \wedge \overline{\zeta_2} \\ \overline{q_\psi} &= q_{\sim\psi} \\ \overline{1} &= 0 \\ \overline{0} &= 1 \end{aligned}$$

Konstruktion des ABA

- $Q = \{q_\psi \mid \psi \in \mathrm{cl}(\varphi)\}$, $\quad q_I = q_\varphi$
- $\Sigma = 2^{\mathsf{AV}}$
- $\delta : Q \times \Sigma \to \mathsf{B}^+(Q)$ wie folgt:

$$\delta(q_x, a) = \begin{cases} 1 & \text{falls } x \in a \\ 0 & \text{sonst} \end{cases}$$

$$\begin{aligned} \delta(q_{\sim\psi}, a) &= \overline{\delta(q_\psi, a)} \\ \delta(q_{\psi\wedge\vartheta}, a) &= \delta(q_\psi) \wedge \delta(q_\vartheta) \\ \delta(q_{\psi\vee\vartheta}, a) &= \delta(q_\psi) \vee \delta(q_\vartheta) \\ \delta(q_{X\psi}, a) &= q_\psi \\ \delta(q_{\psi U\vartheta}, a) &= \delta(q_\vartheta, a) \vee (\delta(q_\psi, a) \wedge q_{\psi U\vartheta}) \end{aligned}$$

- $F = \{q_{\neg(\psi U \vartheta)} \mid \neg(\psi\ U\ \vartheta) \in \mathrm{cl}(\varphi)\}$ **T 5.4**

Vergleich mit Konstruktion LTL → (G)NBA aus Teil 3

Auffällige Unterschiede

- ABA hat linear viele Zustände, GNBA exponentiell viele.
- Hier wird die Bedeutung **aller** Operatoren in δ kodiert.

Gemeinsamkeiten

- Beide Konstruktionen verwenden das Expansionsgesetz.
- Beide Akzeptanzbedingungen verfolgen denselben Zweck: verbieten, die Erfüllung von U-Formeln ∞ weit hinauszuzögern.

1. Punkt bedeutet natürlich, dass es zu einem ABA im Allg. keinen polynomiell großen äquivalenten NBA geben kann.

(Mehr dazu später.)

Und nun . . .

Abschluss unter Komplement

. . . ist für ABA-erkennbare Sprachen besonders leicht zu zeigen.

Für eine PBF φ definieren wir **dual(φ)**
als die PBF, die durch „Umdrehen" von $\wedge$ und $\vee$ entsteht,
z. B.: $\text{dual}\big((q_1 \wedge q_2) \vee q_3\big) = (q_1 \vee q_2) \wedge q_3$

Wir betrachten zur weiteren Erleichterung jetzt A**M**As
(alternierende **Muller**-Aut., Akzeptanzkomp. $\mathcal{F} \subseteq 2^Q$ wie gehabt)

Abschluss unter Komplement

Satz 5.5
Die Klasse der MBA-erkennbaren ω-Sprachen ist unter Komplement abgeschlossen.

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, \mathcal{F})$ ein AMA.

Konstruiere AMA $\mathcal{A}' = (Q, \Sigma, \delta', \{q_I\}, \mathcal{F}')$ wie folgt:

- Für alle $q \in Q$ und $a \in \Sigma$, setze
  $\delta'(q, a) = \text{dual}(\delta(q, a))$.
- $\mathcal{F}' = 2^Q \setminus \mathcal{F}$

Dann gilt: $L_\omega(\mathcal{A}') = \overline{L_\omega(\mathcal{A})}$ **T 5.5** □

Insbesondere ist $\mathcal{A}'$ (bis auf $\mathcal{F}'$) nicht größer als $\mathcal{A}$!

Und nun . . .

Ziel

Satz 5.6 (Miyano & Hayashi 1984)

Für jeden ABA $\mathcal{A}$ gibt es einen NBA $\mathcal{A}'$ mit $L(\mathcal{A}) = L(\mathcal{A}')$.

Alternierende und nichtdeterministische Büchi-Automaten sind also **gleichmächtig.**

Vorgehensweise

(1) Repräsentieren Runs von $\mathcal{A}$ (Bäume) als DAGs (gerichtete azyklische Graphen).

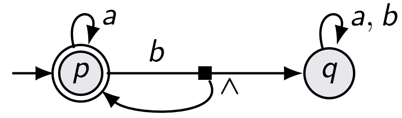
Run-DAGs entsprechen genau den **gedächtnislosen** Runs.

(2) Zeigen: $\mathcal{A}$ akzeptiert Eingabe α gdw. es einen **gedächtnislosen** Run-Baum (und damit einen Run-DAG) von $\mathcal{A}$ auf α gibt.

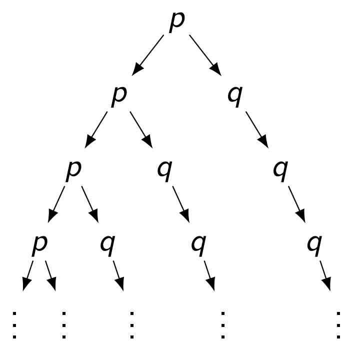
(3) Konstruktion des NBAs $\mathcal{A}'$ mit Anleihen von der Potenzmengenkonstruktion; Korrektheitsbeweis verwendet **(2)**.

Runs als DAGs

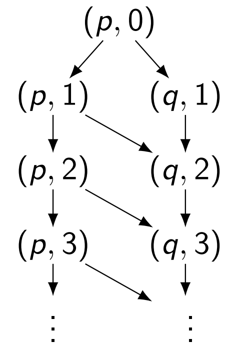
Betrachte den ABA $\mathcal{A}$:



Möglicher Run auf dem Wort b^ω :



Repräsentation als DAG:



Offenbar entspricht jeder Pfad im Run einem Pfad im DAG & umgekehrt.

Run-DAGs

Definition 5.7

Sei $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, F)$ ein ABA und $\alpha = \alpha_0\alpha_1\alpha_2 \cdots \in \Sigma^\omega$.

Ein **Run-DAG** von $\mathcal{A}$ auf α ist ein gerichteter azyklischer Graph (DAG) $\boldsymbol{G = (V, E)}$ mit den folgenden Eigenschaften.

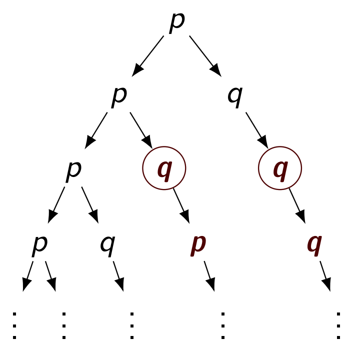
- $V \subseteq Q \times \mathbb{N}$ und $\langle q_I, 0\rangle \in V$
- Wenn $(v, v') \in E$, dann $v = \langle q, \ell\rangle$ und $v' = \langle q', \ell+1\rangle$ mit $q, q' \in Q$ und $\ell \in \mathbb{N}$.
- Wenn $\langle q, \ell\rangle \in V$, dann gibt es $X \subseteq Q$ mit $X \models \delta(q, \alpha_\ell)$ und:
  - $\langle q', \ell+1\rangle \in V$ für alle $q' \in X$
  - $(\langle q, \ell\rangle, \langle q', \ell+1\rangle) \in E$ für alle $q' \in X$

Run-DAG $G = (V, E)$ ist **akzeptierend**, wenn auf jedem Pfad unendlich viele Knoten aus $F \times \mathbb{N}$ vorkommen.

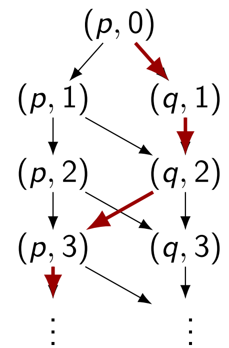
Runs ohne DAGs

Nicht jeder Run eines ABA kann als Run-DAG repräsentiert werden:

Betrachte folgenden Run eines ABA:



Zugehöriger DAG wäre:



Markierter DAG-Pfad entspricht **keinem** Run-Pfad.

Ursache: Teilbäume unter den zwei q in Ebene 2 sind verschieden, hängen von **„Vorgeschichte"** (Vorgängern auf Ebenen 0,1) ab.

Gedächtnislose Runs

Sei (P, r) ein Run von $\mathcal{A}$ mit n Zuständen.
Zur Erinnerung: $\text{Kinder}(p) \subseteq \{p1, \ldots, pn\}$ für alle $p \in P$.

Zwei Positionen p_1 und p_2 heißen **gleichartig**,
wenn $|p_1| = |p_2|$ und $r(p_1) = r(p_2)$.

In gedächtnislosen Runs hängen die Inhalte der Teilbäume unter gleichartigen Positionen nicht von der „Vorgeschichte" ab:

Definition 5.8

Ein Run (P, r) ist **gedächtnislos**, wenn für je zwei gleichartige Positionen p_1, p_2 und alle $i \in \{1, \ldots, n\}^*$ gilt:

- $p_1 i \in P$ gdw. $p_2 i \in P$
- $r(p_1 i) = r(p_2 i)$

Existenz gedächtnisloser Runs

Satz 5.9

Sei $\mathcal{A}$ ein ABA und $\alpha \in L(\mathcal{A})$.
Dann gibt es einen **gedächtnislosen** akzeptierenden Run von $\mathcal{A}$ auf α.

Beweis. Sei (P, r) ein akzeptierender Run von $\mathcal{A}$ auf α.
Konstruieren gedächtnislosen Run (P', r') durch gezieltes Kopieren.

Intuition:
wenn Zustand q in Ebene ℓ von (P, r) mehrmals vorkommt,
kopiere nur das Vorkommen, bei dem der **letzte akzeptierende** Zustand **am weitesten zurückliegt.** **T 5.6**

Definiere dazu Funktion $\gamma : P \to \mathbb{N}$ wie folgt:

$$\gamma(\varepsilon) = 0$$

$$\gamma(pi) = \begin{cases} \gamma(p) + 1 & \text{wenn } r(p) \notin F \\ 0 & \text{sonst} \end{cases}$$

T 5.6 Forts.

Existenz gedächtnisloser Runs

Definieren nun Abbildung **pos** : $Q \times \mathbb{N} \to P$, die jedem Paar $\langle q, \ell \rangle$ eine eindeutige Position in (P, r) zuweist:

- Wenn q nicht in Ebene ℓ von (P, r) vorkommt, dann ist $\text{pos}(q, \ell)$ undefiniert.
- Sonst ist $\text{pos}(q, \ell)$ die am weitesten links liegende Position p in Ebene ℓ mit
  - $r(p) = q$
  - Für alle p' auf Ebene ℓ mit $r(p') = q$ gilt $\gamma(p') \leq \gamma(p)$

T 5.6 Forts.

Existenz gedächtnisloser Runs

Konstruieren (P', r') durch Kopieren aus (P, r) mithilfe von pos:

- $\varepsilon \in P'$ und $r'(\varepsilon) = r(\varepsilon)$
- Wenn p bereits in Ebene ℓ von P' und $i \leq n$ eine mögliche Kind-Nummer ist, dann:
  - $pi \in P'$ gdw. $\text{pos}(r'(p), \ell) \cdot i \in P$
  - Wenn $pi \in P'$, dann $r'(pi) = r\big(\text{pos}(r'(p), \ell) \cdot i\big)$

T 5.6 Forts.

Nun gilt:

- (P', r') ist ein Run von $\mathcal{A}$ auf α **T 5.7**
- (P', r') akzeptierend **T 5.8**
- (P', r') gedächtnislos **T 5.9**

□

Konstruktion des NBA

Sei $\mathcal{A} = (Q, \Sigma, \delta, \{q_I\}, F)$ ein ABA.
Konstruiere NBA $(Q', \Sigma, \Delta, I', F')$ wie folgt.

- $Q' = 2^Q \times 2^Q$
- $I' = \{\langle \{q_0\}, \emptyset \rangle\}$
- $$\Delta = \{(\langle X, \emptyset\rangle,\ a,\ \langle X', X' \setminus F\rangle) \mid X' \models \bigwedge_{q \in X} \delta(q,a)\} \cup \{(\langle X, W\rangle,\ a,\ \langle X', W' \setminus F\rangle) \mid W \neq \emptyset,\ W' \subseteq X',\ X' \models \bigwedge_{q \in X} \delta(q,a),\ W' \models \bigwedge_{q \in W} \delta(q,a)\}$$
- $F' = \{\langle X, \emptyset\rangle \mid X \subseteq Q\}$ **T 5.10**

Lemma 5.10

$L(\mathcal{A}') = L(\mathcal{A})$

Beweis: s. Tafel **T 5.11** □

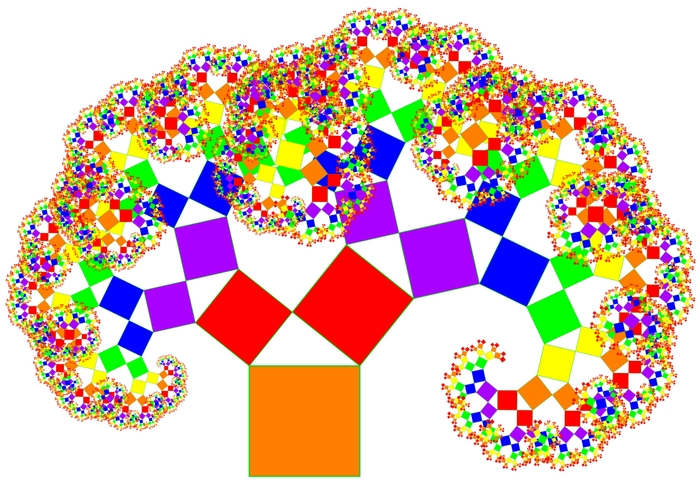
„Ernte“

Aus Lemma 5.10 folgt nun wie gewünscht:

Satz 5.6 (Miyano & Hayashi 1984)

Für jeden ABA $\mathcal{A}$ gibt es einen NBA $\mathcal{A}'$ mit $L(\mathcal{A}) = L(\mathcal{A}')$.

Fast fertig für dieses Semester ...



Pythagoras-Baum. Quelle: Wikipedia, User Gjacquenot (Lizenz CC BY-SA 3.0)

Danke für Eure Aufmerksamkeit!

Literatur für diesen Teil

Bernd Finkbeiner.
Automata, Games, and Verification.
Vorlesungsskript, Universität des Saarlandes, SoSe 2015.
Kap. 8: Alternating Büchi Automata.
https://www.react.uni-saarland.de/teaching/automata-games-verification-15/lecture-notes.html
https://www.react.uni-saarland.de/teaching/automata-games-verification-15/downloads/notes.pdf