

# Automatentheorie und ihre Anwendungen

## Teil 5: Alternierung

Wintersemester 2018/19

Thomas Schneider

AG Theorie der künstlichen Intelligenz (TdKI)

<http://tinyurl.com/ws1819-autom>

# Warum Alternierung?

- Starke Beziehungen zwischen Logik und Automaten, z. B.:
  - NBAs  $\leftrightarrow$  LTL (Teil 3 dieser Vorlesung)
  - NEAs  $\leftrightarrow$  S1S (Satz von Büchi-Elgot-Trakhtenbrot, VL Logik)
- In Logiken kann man aber Sprachen oder Eigenschaften oft deutlich kürzer ausdrücken, z. B.:
  - LTL-Formel  $\rightarrow$  NBA: exponentielle Explosion
  - S1S-Formel  $\rightarrow$  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 erreichen.
- Wir beschränken uns im Folgenden auf  $\omega$ -**Wort**automaten, also auf alternierende Büchi-Automaten.

# Überblick

- 1 Einführung und Grundbegriffe
- 2 Von LTL zu alternierenden Automaten
- 3 Komplementierung

# Und nun ...

- 1 Einführung und Grundbegriffe
- 2 Von LTL zu alternierenden Automaten
- 3 Komplementierung

# 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  $\varphi, \psi$  PBFs sind, dann auch  $\varphi \wedge \psi$  und  $\varphi \vee \psi$ .

# Positive Boolesche Formeln

## Definition 5.1 (Syntax)

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

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

## Definition 5.2 (Semantik)

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

$$V_Y(x) = 1, \text{ falls } x \in Y; \quad V_Y(x) = 0 \text{ sonst.}$$

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

T 5.1



# Alternierende Automaten

## Definition 5.3

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

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

# Alternierende Automaten

## Definition 5.3

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

- $Q$  eine endliche nichtleere **Zustandsmenge** ist,
- $\Sigma$  eine **Alphabet** (endliche nichtleere Menge von Zeichen) ist,
- $\delta : Q \times \Sigma \rightarrow B^+(Q)$  die **Überföhrungsfunktion** 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  $p$ :  $\text{Kinder}(p) \subseteq \{p1, \dots, pn\}$
- **Tiefe, Ebene, Nachfolger, Pfad**: wie gehabt

# 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  $p$ :  $\text{Kinder}(p) \subseteq \{p1, \dots, pn\}$
- **Tiefe, Ebene, Nachfolger, Pfad**: wie gehabt

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

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

# 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  $p$ :  $\text{Kinder}(p) \subseteq \{p1, \dots, pn\}$
- **Tiefe, Ebene, Nachfolger, Pfad**: wie gehabt

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

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

**$\Sigma$ -Baum**  $(P, t)$  (Alphabet  $\Sigma$ ):

- $P$  wie oben
- $t : P \rightarrow \Sigma$  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

$$\{r(p') \mid p' \in \text{Kinder}(p)\} \models \delta(q, \alpha|_p) . \quad \mathbf{T\,5.3}$$

(für andere Akzeptanzbedingungen analog)

# 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

$$\{r(p') \mid p' \in \text{Kinder}(p)\} \models \delta(q, \alpha|_p). \quad \text{T 5.3}$$

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

$$\text{Inf}(r, \pi) \cap F \neq \emptyset \quad \text{T 5.3 Forts.}$$

(für andere Akzeptanzbedingungen analog)

# 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

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

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

$$\text{Inf}(r, \pi) \cap F \neq \emptyset \quad \text{T 5.3 Forts.}$$

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

(für andere Akzeptanzbedingungen analog)



# Und nun ...

- 1 Einführung und Grundbegriffe
- 2 Von LTL zu alternierenden Automaten**
- 3 Komplementierung

# Vorbetrachtungen

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

Hier am Beispiel  $\text{LTL} \rightarrow \text{ABA}$

# Vorbetrachtungen

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

Hier am Beispiel  $LTL \rightarrow ABA$

**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)

# Vorbetrachtungen

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

Hier am Beispiel  $LTL \rightarrow ABA$

**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  $j \geq i$   
und  $s, k \models \varphi$  für alle  $k$  mit  $i \leq k < j$

# Vorbetrachtungen

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

Hier am Beispiel  $LTL \rightarrow ABA$

**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  $j \geq i$   
und  $s, k \models \varphi$  für alle  $k$  mit  $i \leq k < j$

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

$$G\varphi \equiv \neg F\neg\varphi$$

# Vorbetrachtungen

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

Hier am Beispiel  $LTL \rightarrow ABA$

**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  $j \geq i$   
und  $s, k \models \varphi$  für alle  $k$  mit  $i \leq k < j$

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

$$G\varphi \equiv \neg F\neg\varphi$$

**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  $\varphi$  eine LTL-Formel und  $\psi$  eine beliebige Teilformel.

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

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

# Intuitionen der Konstruktion

Seien  $\varphi$  eine LTL-Formel und  $\psi$  eine beliebige Teilformel.

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

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

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

- Eingabealphabet:  $\Sigma = 2^{\text{AV}}$  wie gehabt
- Zustände: für jede Formel  $\psi \in \text{cl}(\varphi)$  ein  $q_\psi$ ; Startzustand  $q_\varphi$



# Intuitionen der Konstruktion

Seien  $\varphi$  eine LTL-Formel und  $\psi$  eine beliebige Teilformel.

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

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

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

- Eingabealphabet:  $\Sigma = 2^{\text{AV}}$  wie gehabt
- Zustände: für jede Formel  $\psi \in \text{cl}(\varphi)$  ein  $q_\psi$ ; Startzustand  $q_\varphi$
- Übergänge:
  - für  $\wedge, \vee$ : mittels PBF
  - für  $\neg$ : per „Negation“ der PBF
  - für  $X\psi$ : schicke  $q_\psi$  zur nächsten Position
  - für  $U$ : per Expansionsgesetz

# Intuitionen der Konstruktion

Seien  $\varphi$  eine LTL-Formel und  $\psi$  eine beliebige Teilformel.

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

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

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

- Eingabealphabet:  $\Sigma = 2^{\text{AV}}$  wie gehabt
- Zustände: für jede Formel  $\psi \in \text{cl}(\varphi)$  ein  $q_\psi$ ; Startzustand  $q_\varphi$
- Übergänge:
  - für  $\wedge, \vee$ : mittels PBF
  - für  $\neg$ : per „Negation“ der PBF
  - für  $X\psi$ : schicke  $q_\psi$  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  $q_\psi$  ist dann  $q_{\sim\psi}$ .

**Genauer:** mittels Operator  $\overline{\phantom{x}}$  wie folgt:

$$\overline{\zeta_1 \wedge \zeta_2} = \overline{\zeta_1} \vee \overline{\zeta_2}$$

$$\overline{\zeta_1 \vee \zeta_2} = \overline{\zeta_1} \wedge \overline{\zeta_2}$$

$$\overline{q_\psi} = q_{\sim\psi}$$

$$\overline{1} = 0$$

$$\overline{0} = 1$$

# Konstruktion des ABA

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

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

$$\delta(q_{\sim\psi}, a) = \overline{\delta(q_\psi, a)}$$

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

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

$$\delta(q_{X\psi}, a) = q_\psi$$

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

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

**T 5.4**

# Vergleich mit Konstruktion $LTL \rightarrow (G)NBA$ aus Teil 3

## Auffällige Unterschiede

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

## Gemeinsamkeiten

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

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

# Und nun ...

- 1 Einführung und Grundbegriffe
- 2 Von LTL zu alternierenden Automaten
- 3 Komplementierung**

# Abschluss unter Komplement

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

Für eine PBF  $\varphi$  definieren wir **dual**( $\varphi$ )

als die PBF, die durch „Umdrehen“ von  $\wedge$  und  $\vee$  entsteht,

$$\text{z. B.: } \text{dual}((q_1 \wedge q_2) \vee q_3) = (q_1 \vee q_2) \wedge q_3$$

Wir betrachten zur weiteren Erleichterung jetzt **AMAs**

(alternierende **Muller**-Aut., Akzeptanzkomp.  $\mathcal{F} \subseteq 2^Q$  wie gehabt)

# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -Sprachen ist unter Komplement abgeschlossen.



# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -Sprachen ist unter Komplement abgeschlossen.

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

# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -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  $\delta'(q, a) = \text{dual}(\delta(q, a))$ .

# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -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  $\delta'(q, a) = \text{dual}(\delta(q, a))$ .
- $\mathcal{F}' = 2^Q \setminus \mathcal{F}$

**T 5.5**

# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -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  $\delta'(q, a) = \text{dual}(\delta(q, a))$ .
- $\mathcal{F}' = 2^Q \setminus \mathcal{F}$

**T 5.5**

Dann gilt:  $L_\omega(\mathcal{A}') = \overline{L_\omega(\mathcal{A})}$  (Beweis mittels Spielen)



# Abschluss unter Komplement

## Satz 5.5

Die Klasse der AMA-erkennbaren  $\omega$ -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  $\delta'(q, a) = \text{dual}(\delta(q, a))$ .
- $\mathcal{F}' = 2^Q \setminus \mathcal{F}$

**T 5.5**

Dann gilt:  $L_\omega(\mathcal{A}') = \overline{L_\omega(\mathcal{A})}$  (Beweis mittels Spielen)



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

# Alternierende vs. nichtdeterministische Automaten

Satz 5.6 (Miyano & Hayashi 1984)

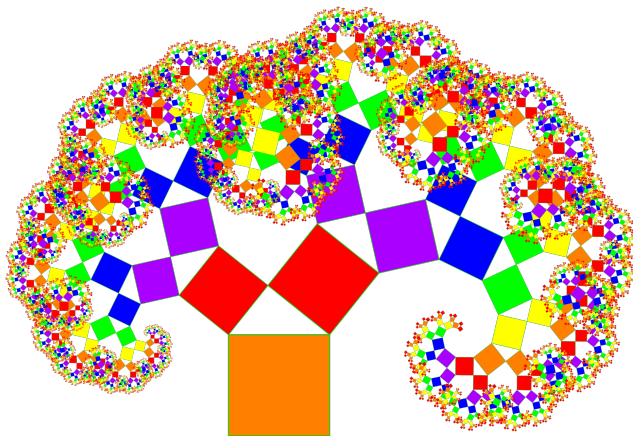
Für jeden ABA  $\mathcal{A}$  gibt es einen NBA  $\mathcal{A}'$  mit  $L_\omega(\mathcal{A}) = L_\omega(\mathcal{A}')$ .

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

**Beweisskizze:** Siehe Folien aus dem letzten Jahr

<http://tinyurl.com/ws1718-automaten>

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/lecture-notes.html)

[https://www.react.uni-saarland.de/teaching/  
automata-games-verification-15/downloads/notes.pdf](https://www.react.uni-saarland.de/teaching/automata-games-verification-15/downloads/notes.pdf)



