Einführung LTL Komplementierung

## Automatentheorie und ihre Anwendungen Teil 5: Alternierung

Wintersemester 2018/19 Thomas Schneider

AG Theorie der künstlichen Intelligenz (TdKI)

http://tinyurl.com/ws1819-autom

Teil 5: Alternierung

Automatentheorie und ihre Anwendunger Teil 5: Alternierung

Wintersemsster 2018/19 Thomas Schneider

AC Thomas de kiestichen Intelligeer (1860)

http://tinyurl.com/ws1819-autom

8:30

### TODO:

- Übersetzung ABA → NBA besser verstehen (v. a. Beweisdetails in T5.8, 5.10 und Bsp.-Automat T5.11).
- Beweise sauberer führen (Finkbeiner-Skript z. T. lakonisch & hat Fehler)
- Intuition hinter Automatenkonstruktion klarer fassen (X, W)
- konkreten Automaten mit Bsp.-Run, dem kein Run-DAG entspricht, angeben

TODO: Weitere Themen siehe TODO.txt.

- 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.

Teil 5: Alternierung

└─Warum Alternierung?

Starke Beziehungen zwischen Logik und Automaten, z. B.:
 NBAs ++ LTL. (Teil 3 dieser Vorleuung)
 NEAs ++ SIS (Satz von Büchi-Elgot-Trakhtenbrot, VL Logik)

■ In Logikan kann man aber Sprachen oder Eigenschaften oft deutlich kürzer ausdrücken, z. B.:

• LTL-Formel → NBA: sogonentielle Explosion

• LTL-Formel → NBA:

Warum Alternierung?

S1S-Formel → NEA: sogar nicht-elementare Explosion
 Verkleinern dieser Lücke:
 Erlaube in Automaten nicht nur existenzielle (= nichtdeterm. Verzweisunsen": sondern auch univerzelle.

8:30

2019-02-01

Büchi-Elgot-Trakhtenbrot: Logik-VL. S1S = monad. SO auf lin. Strukt.

"nicht-elementar": jede Negation erfordert Potenzmengenkonstruktion, vergrößert Automaten exp.

 $\rightarrow$  "exp. Turm" unbeschränkt (Verschachtelungstiefe  $\neg$  (und  $\exists$ ))

Um die Lücke zu verkleinern, erweitert man das Automatenmodell so, dass es der Logik ähnlicher wird.

Einführung LTL Komplementierung

### 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  $\omega$ -Wortautomaten, also auf alternierende Büchi-Automaten.

Teil 5: Alternierung

└─Warum Alternierung?

"Alternierung" heißt also, dass ein Maschinenmodell
 (abwechselnd) existenzielle und universelle Entscheidunge

Warum Alternierung?

- treffen karn.

  Alterrierende Varianten gibt es für alle Automatentypen aus dieser Vorlesung (auf endlichen oder unendlichen Objekten, Wörtern oder Bäumen) und für andere Maschinammodelle (z. B.
  - Wörtern oder Bäumen) und für andere Maschinenmodelle (z. Turingmaschinen).

    • Für alternierende Automaten ist Komplementierung besonder

8:32

2019-02-01

"Abwechselnd" ist hier wichtig. Nur ex./nur univ. ist witzlos.

Für TMs: erlaubt z. B. feinere Komplexitätsanalyse

└─ Überblick

Überblick

Komplementierung

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

8:33

2019-02-01

Automatentheorie u. i. A. WiSe 2018/19

2 Von LTL zu alternierenden Automaten



## Alternierung: Grundidee

• Nichtdeterministischer Automat  $\mathcal{A}$  akzeptiert eine Eingabe, wenn ein erfolgreicher 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 ¬.
- Statt eines Runs (Zustandsfolge) gibt es nun einen Run-Baum, der alle universellen Entscheidungen berücksichtigt.

 Nichtdeterministischer Automat A akzeptiert eine Eingabe, wenn ein erfolgreicher Run existiert.
 d. h.: falls (q. a, q²), (q. a, q²) ∈ Δ, kann A in Situation (q. a) "entscheiden", wie der Run fortgesetzt wird.

"entscheiden", wie der Run fortgesetzt wird.

Mindestens eine dieser Entscheidungen muss zum Ziel führen

Alternierung erlaubt auch universelle Entscheidungen,

in beliebiger Kombination mit existenziellen

Alternierung: Grundidee

 "Beliebige Kombination" wird realisiert durch positive Boolesche Formel, d.h. aussagenlogische Formel ohne –.
 Statt eines Runs (Zustandsfolge) gibt es nun einen Run-Baum, der alle universellen Entscheidungen berücksichtigt.

### 8:33

Positive Formeln werden manchmal auch monoton genannt. Hier egal. Wichtig: keine Negation!

Einführung Komplementierung

### 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  $\varphi \vee \psi$ .

Teil 5: Alternierung Einführung und Grundbegriffe

Die Menge der positiven Bookschen Formeln (PBFs) über einer Menge X, geschrieben B+(X), ist die kleinste Menge, für die gilt

Positive Boolesche Formeln

Wenn φ, ψ PBFs sind, dann auch φ ∧ ψ und φ ∨ ψ

Positive Boolesche Formeln

8:35

Positive Roolesche Formeln

### 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  $V_Y : X \to \{0,1\}$ :

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

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

T 5.1

Komplementierung

Teil 5: Alternierung Einführung und Grundbegriffe

Positive Boolesche Formeln

8:35

Automatentheorie u. i. A. WiSe 2018/19

Teil 5: Alternierung

### 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 **Zustandsmeng**e ist,
- $\bullet$   $\Sigma$  eine Alphabet (endliche nichtleere Menge von Zeichen) ist,
- $\delta: Q \times \Sigma \to \mathsf{B}^+(Q)$  die Überführungsfunktion ist,
- $I \subset Q$  die Menge der Anfangszustände ist,
- $F \subset Q$  die Menge der akzeptierenden Zustände ist.

Teil 5: Alternierung

Einführen, 5:

Einführen und Grundbegriffe

Einführeng und Grundbegriffe

Alternierende Automaten of und Antone und und Victoria (AMA) sit den Stiput 4 = (0.2.1.6.1; solution)

Omer underdie nielleben zeitzelnebenge ist.

2 con Alphate (nortiche nichten Many von Zinche) int 
1 con Alternierende Automaten

Alternierende Automaten

8:41

Überführungs**funktion:** weil die PBF bereits die nichtdeterministischen Entscheidungen enthält

### 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 **Zustandsmeng**e ist,
- $\bullet$   $\Sigma$  eine Alphabet (endliche nichtleere Menge von Zeichen) ist,
- $\delta: Q \times \Sigma \to B^+(Q)$  die Überführungsfunktion ist,
- $I \subset Q$  die Menge der Anfangszustände ist,
- $F \subset Q$  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.

8:41

Überführungs**funktion:** weil die PBF bereits die nichtdeterministischen Entscheidungen enthält

Alternierende Automaten

Alternative Akzeptanzbedingungen (Muller, Parität usw.) sind auci

### Run-Bäume

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

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

Teil 5: Alternierung Einführung und Grundbegriffe 2019-02-01 Run-Bäume

Run-Bäume

8:43

Runs sind jetzt Bäume mit endlichem Verzweigungsgrad; Bäume müssen nicht vollständig sein.

Präfix-Abg.: wie bei endlichen Bäumen; jedes Kind braucht sein Elter!

### Run-Bäume

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

- Positionen: Menge  $P \subseteq \{1, ..., n\}^*$ , 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 > 0

Teil 5: Alternierung

Leinführung und Grundbegriffe

Run-Bäume

Run-Bäume

Run-Bäurne Baum ett Vorzedgespegad  $\leq n$ . für festes  $n \in \mathbb{N}$  Periotione Meng  $P \subseteq \{1, \ldots, n\}^n$ , prüs-dgeschoum n Ründer eines Kontens  $p \in \{1, \ldots, n\}^n$ , prüs-dgeschoum n Ründer konte Kontens  $p \in \text{Konte}(p) \subseteq \{p1, \ldots, np\}$  a Tind, Bench Kontelleyer, Pitte sin gehandt Pall in P -midfock and sensedfiche Folge  $n = n_0 n_1 n_2 \cdots n n_n$  and  $n_0 = n_0$  and

#### 8:43

2019-02-01

Runs sind jetzt Bäume mit endlichem Verzweigungsgrad; Bäume müssen nicht vollständig sein.

Präfix-Abg.: wie bei endlichen Bäumen; jedes Kind braucht sein Elter!

Einführung LTL Komplementierung

### Run-Bäume

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

- Positionen: Menge  $P \subseteq \{1, ..., n\}^*$ , präfix-abgeschlossen
- Kinder eines Knotens p: Kinder $(p) \subset \{p1, \ldots, pn\}$
- 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
- $\pi_{i+1} \in Kinder(\pi_i)$  für alle i > 0

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

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

T 5.2

Teil 5: Alternierung

LEInführung und Grundbegriffe

Run-Bäume

Run-Bäume

Plad in P: endicine oder unendliche Folge  $\pi = \pi_0\pi_1\pi_2 \cdots$  von Politikums  $\pi_1 \in P$  mix  $\bullet \in \mathbb{N}$  on  $\pi$  and  $\bullet \in \mathbb{N}$  in  $\mathbb{N}$  of  $\mathbb{N}$  in  $\mathbb{N}$  i

8:43

2019-02-01

Runs sind jetzt Bäume mit endlichem Verzweigungsgrad; Bäume müssen nicht vollständig sein.

Präfix-Abg.: wie bei endlichen Bäumen; jedes Kind braucht sein Elter!

## 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:

- $\bullet$   $r(\varepsilon) = q_1$
- für alle  $p \in P$ : wenn r(p) = q, dann

$$\{r(p') \mid p' \in \mathsf{Kinder}(p)\} \models \delta(q, \alpha_{|p|}).$$
 T5.3

(für andere Akzeptanzbedingungen analog)

Einführung und Grundbegriffe Berechnungen und Akzeptanz Berechnungen und Akzeptanz Ein Run eines ABA  $A = (Q, \Sigma, \delta, \{q_i\}, F)$  auf einem Wort  $\alpha = \alpha_1 \alpha_2 \alpha_2 \cdots \in \Sigma^{\omega}$  ist ein Q-Baum (P, r), so dass: • für alle  $p \in P$ : wenn r(p) = q, dann  $\{r(p') \mid p' \in Kinder(p)\} \models \delta(q, \alpha_{|p|}).$  T5.3 (für andere Akzeptanzbedingungen analog)

8:48 bis 8:59

Bedingung 2, Run:

Teil 5: Alternierung

die Kinder eines Knotens p enthalten eine Menge von Zuständen, die die PBF erfüllen, die  $\delta$  dem Paar (r(p), |p|-tes Zeichen von  $\alpha)$  zuweist

Ein Beispiel an Tafel; weiteres Beispiel im nächsten Abschnitt (LTL)!

#### 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:

- $\bullet$   $r(\varepsilon) = q_1$
- für alle  $p \in P$ : wenn r(p) = q, dann

$$\{r(p') \mid p' \in \mathsf{Kinder}(p)\} \models \delta(q, \alpha_{|p|}).$$
 T5.3

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

$$Inf(r,\pi) \cap F \neq \emptyset$$

T 5.3 Forts.

(für andere Akzeptanzbedingungen analog)

Teil 5: Alternierung

Einführung und Grundbegriffe Berechnungen und Akzeptanz Berechnungen und Akzeptanz Ein Run eines ABA  $A = (Q, \Sigma, \delta, \{q_i\}, F)$  auf einem Wort  $\alpha = \alpha_1 \alpha_2 \cdots \in \Sigma^{\omega}$  ist ein Q-Baum (P, r), so dass • für alle  $p \in P$ : wenn r(p) = q, dann  $\{r(p') \mid p' \in Kinder(p)\} \models \delta(q, \alpha_{|p|}).$  T5.3 Run (P, r) ist erfolgreich, wenn für ieden unendlichen Pfad (für andere Akzeptanzbedingungen analog)

8:48 bis 8:59

Bedingung 2, Run:

die Kinder eines Knotens p enthalten eine Menge von Zuständen, die die PBF erfüllen, die  $\delta$  dem Paar (r(p), |p|-tes Zeichen von  $\alpha)$  zuweist

Ein Beispiel an Tafel; weiteres Beispiel im nächsten Abschnitt (LTL)!

## 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:

- $\bullet$   $r(\varepsilon) = q_1$
- für alle  $p \in P$ : wenn r(p) = q, dann

$$\{r(p') \mid p' \in \mathsf{Kinder}(p)\} \models \delta(q, \alpha_{|p|}).$$
 T5.3

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

$$Inf(r,\pi) \cap F \neq \emptyset$$

T 5.3 Forts.

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

(für andere Akzeptanzbedingungen analog)

Teil 5: Alternierung Einführung und Grundbegriffe

Berechnungen und Akzeptanz

Ein Run eines ABA  $A = (Q, \Sigma, \delta, \{q_i\}, F)$  auf einem Wort  $\alpha = \alpha_1 \alpha_2 \cdots \in \Sigma^{\omega}$  ist ein Q-Baum (P, r), so dass • für alle  $p \in P$ : wenn r(p) = q, dann  $\{r(p') \mid p' \in Kinder(p)\} \models \delta(q, \alpha_{|p|}).$  T5.3 Run (P, r) ist erfolgreich, wenn für ieden unendlichen Pfad  $L_{\omega}(A) = \{\alpha \in \Sigma^{\omega} \mid A \text{ hat einen erfolgr. Run auf } \alpha\}$  T5.3 Forts. (für andere Akzeptanzbedingungen analog)

Berechnungen und Akzeptanz

8:48 bis 8:59

Bedingung 2, Run: die Kinder eines Knotens p enthalten eine Menge von Zuständen, die die PBF erfüllen, die  $\delta$  dem Paar (r(p), |p|-tes Zeichen von  $\alpha)$  zuweist

Ein Beispiel an Tafel; weiteres Beispiel im nächsten Abschnitt (LTL)!

2 Von LTL zu alternierenden Automaten



Einführung LTL Komplementierung

## Vorbetrachtungen

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

Hier am Beispiel LTL  $\rightarrow$  ABA

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Vorbetrachtungen

Obersetzung logischer Formeln in alternierende Automaten ist oft einfacher als in nichtdeterministische Automaten. Hier am Beispiel LTL -> ABA

Vorbetrachtungen

#### 8:59

2019-02-01

Formeln hier als Grammatik angegeben.

- Variablen als x, nicht p (schon für Positionen vergeben).
- V ist eigentlich überflüssig, kann man aber gleich ganz bequem behandeln.
- F, G sind nur Abkürzungen mittels U.

Expansionsgesetz: kann man leicht semantisch überprüfen. Haben wir damals auch im (G)NBA kodiert. Allerdings jetzt einfacher mit ABA modellierbar!

## Vorbetrachtungen

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

Hier am Beispiel LTL  $\rightarrow$  ABA

Erinnerung an LTL: 
$$\varphi ::= x \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid X\varphi \mid \varphi U \varphi$$
  
mit  $x \in AV$  (Aussagenvariablen)

Teil 5: Alternierung Von LTL zu alternierenden Automaten └─Vorbetrachtungen

Vorbetrachtungen

Erinnerung an LTL:  $\varphi := x \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid X\varphi \mid \varphi U \varphi$ mit x ∈ AV (Aussagenvariables

#### 8:59

2019-02-01

Formeln hier als Grammatik angegeben.

- Variablen als x, nicht p (schon für Positionen vergeben).
- V ist eigentlich überflüssig, kann man aber gleich ganz bequem behandeln.
- F, G sind nur Abkürzungen mittels U.

Expansionsgesetz: kann man leicht semantisch überprüfen. Haben wir damals auch im (G)NBA kodiert. Allerdings jetzt einfacher mit ABA modellierbar!

└─Vorbetrachtungen

und s.  $k \models \varphi$  für alle k mit  $i \le k \le i$ 

## Vorbetrachtungen

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

Hier am Beispiel LTL  $\rightarrow$  ABA

Erinnerung an LTL: 
$$\varphi ::= x \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid X\varphi \mid \varphi U \varphi$$
  
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  $i \leqslant k < j$ 

#### 8:59

2019-02-01

Formeln hier als Grammatik angegeben.

- Variablen als x, nicht p (schon für Positionen vergeben).
- V ist eigentlich überflüssig, kann man aber gleich ganz bequem behandeln.
- F, G sind nur Abkürzungen mittels U.

Expansionsgesetz: kann man leicht semantisch überprüfen. Haben wir damals auch im (G)NBA kodiert. Allerdings jetzt einfacher mit ABA modellierbar!

Komplementierung

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

Hier am Beispiel LTL  $\rightarrow$  ABA

Erinnerung an LTL: 
$$\varphi ::= x \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid X\varphi \mid \varphi U \varphi$$
  
mit  $x \in AV$  (Aussagenvariablen)

$$\begin{array}{ll} s,i \models \varphi \ U \ \psi, & \text{falls } s,j \models \psi \ \text{für ein } j \geqslant i \\ & \text{und } s,k \models \varphi \ \text{für alle } k \ \text{mit } i \leqslant k < j \end{array}$$

$$F\varphi \equiv (x \lor \neg x) U \varphi$$
$$G\varphi \equiv \neg F \neg \varphi$$

Teil 5: Alternierung Von LTL zu alternierenden Automaten Vorbetrachtungen  $\begin{array}{ll} \text{Obsetrachtungen} \\ \text{Obsetrachtungen} \\ \text{Obsetrachtungen} \\ \text{Spinson} \\ \text{Lie an Beispil II.} \rightarrow \Delta \text{RA} \\ \text{Mean an Beispil II.} \rightarrow \Delta \text{RA} \\ \text{Enterweap II.} \\ \text{User an Beispil II.} \rightarrow \Delta \text{RA} \\ \text{Enterweap II.} \\ \text{Lie and Beispil II.} \rightarrow \Delta \text{RA} \\ \text{Enterweap II.} \\ \text{Lie and Beispil II.}$ 

#### 8:59

2019-02-01

Formeln hier als Grammatik angegeben.

└─Vorbetrachtungen

- Variablen als x, nicht p (schon für Positionen vergeben).
- V ist eigentlich überflüssig, kann man aber gleich ganz bequem behandeln.
- F, G sind nur Abkürzungen mittels U.

Expansionsgesetz: kann man leicht semantisch überprüfen. Haben wir damals auch im (G)NBA kodiert. Allerdings jetzt einfacher mit ABA modellierbar!

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

Hier am Beispiel LTL  $\rightarrow$  ABA

Erinnerung an LTL: 
$$\varphi ::= x \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid X\varphi \mid \varphi U \varphi$$
  
mit  $x \in AV$  (Aussagenvariablen)

$$\begin{array}{ll} s,i \models \varphi \ U \ \psi, & \text{falls } s,j \models \psi \ \text{für ein } j \geqslant i \\ & \text{und } s,k \models \varphi \ \text{für alle } k \ \text{mit } i \leqslant k < j \end{array}$$

$$F\varphi \equiv (x \lor \neg x) U \varphi$$
$$G\varphi \equiv \neg F \neg \varphi$$

## Expansionsgesetz:

$$s, i \models \varphi \ U \psi \ \text{qdw}. \ s, i \models \psi \ \text{oder} \ (s, i \models \varphi \ \text{und} \ s, i+1 \models \varphi \ U \psi)$$

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Unsetzung leginber Formen in alternierunden Automaten sind erfelber den ein alternierunden Automaten sind erfelber den ein alternierunden Automaten. Hur zur Bespect 17. → ABS.

Emmeng at Ett. → att → att

#### 8:59

2019-02-01

Formeln hier als Grammatik angegeben.

- Variablen als x, nicht p (schon für Positionen vergeben).
- V ist eigentlich überflüssig, kann man aber gleich ganz bequem behandeln.
- F, G sind nur Abkürzungen mittels U.

Expansionsgesetz: kann man leicht semantisch überprüfen. Haben wir damals auch im (G)NBA kodiert. Allerdings jetzt einfacher mit ABA modellierbar!

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

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

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Intuitionen der Konstruktion

Seien  $\varphi$  eine LTL-Formel und  $\psi$  eine beliebige Tei $\sim \psi = \begin{cases} \vartheta & \text{falls } \psi = \neg \vartheta \\ \neg \psi & \text{sonst} \end{cases}$   $d(\varphi) = \{ \psi, \sim \psi \mid \psi \text{ ist Teilformel von } \varphi \}$ 

Intuitionen der Konstruktion

9:03

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

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

Bestandteile des ABA  ${\cal A}_{\scriptscriptstyle arphi}$ 

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

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Intuitionen der Konstruktion

Intuitionen der Konstruktion

9:03

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

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

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

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

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Intuitionen der Konstruktion

Soins  $\varphi$  view LTL-Formed and  $\psi$  sine buildingly Tailformed:  $\varphi \varphi = \begin{cases} \varphi & \text{fill } \varphi & \text{ord} \\ -\varphi & \text{soint} \end{cases}$   $G(\varphi) = \{ \varphi, \neg \psi \mid \varphi \text{ in Tailformed von } \varphi \}$ Bestanderis den ABA  $A_{\varphi}$   $\bullet \text{ Engelskeigheiden} \quad \Sigma = 2^{N_{\varphi}} \text{ wis gehale}$   $\bullet \text{ Zustische fils plus formul } \psi \in G(\psi) \text{ ain } q_{\varphi} \text{ Startnatztard } q_{\varphi}$ 

> für ∧, ∨: mittels PBF für ¬: per "Negation" der PBF für Xsi: schicke α., zur nächsten Position

Intuitionen der Konstruktion

#### 9:03

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

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

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

- Eingabealphabet:  $\Sigma = 2^{AV}$  wie gehabt
- Zustände: für jede Formel  $\psi \in \operatorname{cl}(\varphi)$  ein  $q_{\psi}$ ; Startzustand  $q_{\varphi}$
- Übergänge:
  - für ∧, ∨: mittels PBF
  - für ¬: per "Negation" der PBF
  - für  $X\psi$ : schicke  $q_{\psi}$  zur nächsten Position
  - für U: per Expansionsgesetz
- *F* verhindert unendliches "Aufschieben" von *U*-Teilformeln!

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

Von LTL zu alternierenden Automaten

Intuitionen der Konstruktion

Intuitionen der Konstruktion

F verhindert unendliches "Aufschieben" von U-Teilformein

9:03

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

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

Genauer: mittels Operator — wie folgt:

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

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

$$\overline{1} = 0$$

$$\overline{0} = 1$$

Teil 5: Alternierung

Von LTL zu alternierenden Automaten

"Negation von PBFs"

date. Mustan extendessen Dankildt van  $r_v \lor (dn \; Morgan)$ , un Nagatien extien inne zu zelmen. Nagatien et eine Atterm  $q_v$  itt dam  $r_v \leftrightarrow r_v$ . Germaner: mittals Operator  $\overline{\phantom{a}}$  wise folge:  $\begin{array}{c} Grmaner: \; mittals \; Operator & = viola \; folge: \\ \hline Grav G = G \lor G \\ \hline Grav G = G \lor G \\ \hline Grav G = G \lor G \\ \hline T = 0 \\ \hline G = 1 \\ \end{array}$ 

.Negation von PBFs"

#### 9:07

2019-02-01

Achtung: wir wollen *nur PBFs* "negieren", nicht LTL-Formeln. Deshalb brauchen wir nur Fälle für  $\land$ ,  $\lor$  und Atome.

Aussagenvariablen sind aber hier immer nur die  $q_{\psi}$ ; und die können wir negieren, indem wir  $\psi$  negieren.

• 
$$Q = \{q_{\psi} \mid \psi \in \mathsf{cl}(\varphi)\}, \quad q_I = q_{\varphi}$$

$$\Sigma = 2^{AV}$$

•  $\delta: Q \times \Sigma \to \mathsf{B}^+(Q)$  wie folgt:

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

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

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

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

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

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

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

T 5.4

9:09

2019-02-01

Erklären, 5min Pause, dann Beispiel vorrechnen → bis 9:30?

Man kann natürlich für die Zustände direkt  $\psi$  statt  $\underline{q_{\psi}}$  schreiben. Das sorgt aber für Verwirrung in der Definition von  $\overline{\phantom{a}}$ , weil man bei  $\wedge, \vee$  nicht mehr sieht, ob PBFs oder LTL-Formeln verknüpft werden.

### 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 äguivalenten NBA geben kann.

#### 9:30

2019-02

Teil 5: Alternierung

Teil 3

Letzter Satz: wenn ABA  $\rightarrow$  NBA wieder gemacht wird, dann "Mehr dazu später" in Folie wieder einkommentieren.

 $\vdash$  Vergleich mit Konstruktion LTL  $\rightarrow$  (G)NBA aus

2 Von LTL zu alternierenden Automaten

3 Komplementierung

Teil 5: Alternierung

Komplementierung

Und nun ...

2019-02-01

## Abschluss unter Komplement

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

Für eine PBF  $\varphi$  definieren wir  $\operatorname{dual}(\varphi)$  als die PBF, die durch "Umdrehen" von  $\wedge$  und  $\vee$  entsteht,

z. B.: dual
$$((q_1 \land q_2) \lor q_3) = (q_1 \lor q_2) \land q_3$$

Wir betrachten zur weiteren Erleichterung jetzt AMAs (alternierende Muller-Aut., Akzeptanzkomp.  $\mathcal{F} \subset 2^Q$  wie gehabt)



#### 9:31

Grund für Muller:

Akzeptanzbedingung des Komplementautomaten wird einfacher.

Einführung LTL Komplementierung

## Abschluss unter Komplement

#### Satz 5.5

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



#### 9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten → hier nicht.

**TODO:** Wenn Satz bewiesen werden soll, dann Beweis ganz neu ausarbeiten; man muss wohl Alternierung mittels existenzieller und universeller Zust. definieren; siehe Notizen zwischen T5.5 und T5.6.

2019-02-01

## 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.

Teil 5: Alternierung

Komplementierung

Abschluss unter Komplement

9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten → hier nicht.

#### 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  $\delta'(q, a) = \text{dual}(\delta(q, a))$ .

Teil 5: Alternierung

Komplementierung

Abschluss unter Komplement

Satz S.5

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

Brusik. Sei  $A = (Q, \Sigma, \delta, \{g_1\}, \mathcal{F})$  ein AMA.

Konstraiere AMA  $A^{\mu} = (Q, \Sigma, \delta, \{g_2\}, \mathcal{F})$  wie folgt:

o Für alle  $q \in Q$  and  $a \in \Sigma$ . setze

Abschluss unter Komplement

9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten  $\rightarrow$  hier nicht.

## 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

Teil 5: Alternierung

Komplementierung

Abschluss unter Komplement

Absorbius unter Komplement

[Sail 5.5]
Dis Klaus der AMA-arkanshavan  $\omega$ -Sprachen ist unter Komplement allegachismen.

Bronk. Sid  $A = (0, \Sigma, \delta, (\phi), T)$  ein AMA.

Kontzarisa AMA  $X^{\mu} \in (0, \Sigma, F, (\phi), T)$  ein Moja Kontzarisa AMA  $X^{\mu} \in (0, \Sigma, F, (\phi), T)$  ein Moja  $X^{\mu} \in X^{\mu}$  and  $X^{\mu} \in X^{\mu}$  a

9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten → hier nicht.

## 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)

Teil 5: Alternierung

Komplementierung

Abschluss unter Komplement

Absorbiuss unter Komplement

Satz S.5

Die Nieue der AMA-erkentheurs un Sprachen ist unter Komplement abgenähmen. 
Bestens. Sir  $A = \{0, \Sigma, \delta, \{\phi\}, \mathcal{T}\}$  ein AMA.
Kontroller AMA,  $\mathcal{F} = \{0, \Sigma, \mathcal{F}, \{\phi\}, \mathcal{T}\}$  wie folgt:

\* First wird  $\phi \in \mathcal{G}$  and  $\phi \in \Sigma$  et autes  $\mathcal{F} = \mathcal{F} = \mathcal{F} = \mathcal{F}$ .

Dans  $\phi | \mathbf{f} \in \mathcal{K}_f | \mathbf{f} = \mathbb{L}_f \mathbf{f} |$  (Breasis noticels Spinles)

9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten → hier nicht.

#### 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  $\mathcal{A}' = (Q, \Sigma, \delta', \{q_l\}, \mathcal{F}')$  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)

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

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

Teil 5: Alternierung

Komplementierung

Bases.

De Klaine der AMA erkenderen "Sprachte ist enter Komplement

Bronk. Ser A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontrale AMA A = (0. T. A. (g.), 7) and AMA.

Kontral

9:32 bis 9:45 (nur Bsp. und kein Beweis)

Konstruktion: vergleiche mit Safra-Konstruktion

Beweis: braucht Spiele (sehr ähnlich zu denen aus Kap. 4) und alternative Def. alternierender Automaten → hier nicht.

## 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

Komplementierung

-Alternierende vs. nichtdeterministische Automaten

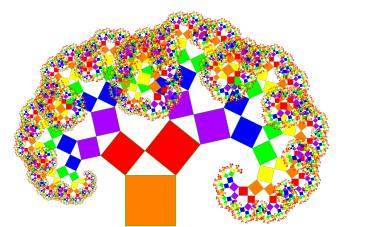
9:45

Teil 5: Alternierung

Noch ein wichtiges Resultat, aber ohne Beweis

Einführung LTL Komplementierung

## Fast fertig für dieses Semester . . .



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

# Danke für Eure Aufmerksamkeit!

Teil 5: Alternierung



Fast fertig für dieses Semester . . .

#### 9:46

- Literatur
- Eval.
- Prüfungshinweise?

Einführung LTL Komplementierung

### 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

Teil 5: Alternierung

2019-02-01

Literatur für diesen Teil

Literatur für diesen Teil

Bernd Finkbeiner.
Automata, Games, and Verification.

Vorleungsskript, Universität des Saarlandes, So Kap. 8: Alternating Büchi Automata.

Automatentheorie u. i. A. WiSe 2018/19

Teil 5: Alternierung

Teil 5: Alternierung