Abschlusseig.

Büchi-Aut.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking Teil 3: unendliche Wörter

Automatentheorie und ihre Anwendungen Teil 3: endliche Automaten auf unendlichen Wörtern

> Wintersemester 2018/19 Thomas Schneider

> > AG Theorie der künstlichen Intelligenz (TdKI)

http://tinyurl.com/ws1819-autom

Di: S. 74 Mi: S. 111 8:40

**TODO**: Es ist nicht günstig, die ganzen anderen Automatentypen (Müller etc.) im Abschnitt "Determinisierung" zu haben.

→ Kapitel neu aufteilen, Tafelanschriebe umordnen, Nummern+Verweise im Skript anpassen:

- 1. Motivation
- 2. Büchi-Automaten (mit Def. DBA am Ende)
- 3. Abschlusseig. und Chrakterisierungen (einschl. Charakt. DBAs und deren "Schwäche")
- 4. Alternative Akzeptanzbedingungen
- 5. Determinisierung
- 6. Entscheidungsprobleme
- 7. Model-Checking

Außerdem: Safra-Konstruktion anders einführen/beweisen, s. TODO.txt

## Überblick

- Motivation
- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschaften
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierung
- 6 Entscheidungsprobleme
- Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Teil 3: unendliche Wörter

### Mattrazius

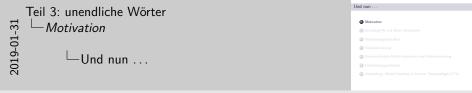
| Grandburgfff and Bick-Automates
| Abackburgffgerschaftes
| Charakburgffff and Bick-Automates
| Charakburgffff and Bick-Automates
| Charakburgfffff and Bick-Automates
| Charakburgffff and Bick-Automates and Outermittening
| Charakburgfffff and Bick-Automates and Outermittening
| Charakburgfffff and Bick-Automates and Outermittening
| Anneadoung Model Charking in Lineary Temporologis (LTL)

8:40

2019-01-31

## Und nun ...

- Motivation
- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschaften
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierung
- 6 Entscheidungsproblem
- 7 Anwendung: Model-Checking in Linearer Temporallogik (LTL)



## Terminierung

Terminierung von Algorithmen ist wichtig für Problemlösung.

#### Übliches Szenario:

- Eingabe: endliche Menge von Daten
- Lasse Programm *P* laufen, bis es terminiert
- Ausgabe: Ergebnis, das durch P berechnet wurde

Um Ausgabe zu erhalten, muss P für jede Eingabe terminieren.

Teil 3: unendliche Wörter

Motivation

Terminierung

## Terminierung

Motiv.

Terminierung von Algorithmen ist wichtig für Problemlösung.

#### Übliches Szenario:

- Eingabe: endliche Menge von Daten
- Lasse Programm *P* laufen, bis es terminiert
- ullet Ausgabe: Ergebnis, das durch P berechnet wurde

Um Ausgabe zu erhalten, muss P für jede Eingabe terminieren.

Beispiel: Validierung von XML-Dokumenten für gegebenes Schema

- Konstruiere Automaten für Schema und Dokument (terminiert)
- Reduziere auf Leerheitsproblem (terminiert)
- Löse Leerheitsproblem
   (sammle erreichbare Zustände terminiert)

Teil 3: unendliche Wörter

- Motivation

- Terminierung

Trensistering von Algorithmen int wichtig für Problemblung.

Übliches Stranslei:

6. Eingelse mildeliche Menge von Daten

6. Laus Programm P. Jaufen, bis a terminiset

6. Ausgalse Eppsich, aus der der Veranschund under

10m Ausgalse aus der der Veranschund under

10m Ausgalse ausgalse, mit der Veranschund under

10m Ausgalse ausgalse, mit der Veranschund und

10m Ausgalse ausgalse, mit der Veranschund und

10m Ausgalse ausgalsen, mit der Eingeleinen Schammen

10m Ausgalsen Aufgalsen und Gestalten und Gebaument (gemitter

10m Ausgalsen auf Leutenze der Veranschung)

10m Ausgalsen auf Leutenze der Veranschung und der

10m Ausgalsen auf Leutenze und Leutenze und der

10m Ausgalsen auf Leutenze und der

10m Ausgalsen auf Leutenze

10m Ausgalsen

10m Ausga

(sammle erreichbare Zustände - terminier

Ferminierung

## Terminierung unerwünscht

Von manchen Systemen/Programmen fordert man, dass sie nie terminieren.

### Beispiele:

- (Mehrbenutzer-)Betriebssysteme sollen beliebig lange laufen ohne abzustürzen, egal was Benutzer tun
- Bankautomaten, Flugsicherungssysteme, Netzwerkkommunikationssysteme, . . .



Büchi-Aut. Abschlusseig. Entscheidungsprobl. Model-Checking Determinismus

## Terminierung unerwünscht

Von manchen Systemen/Programmen fordert man, dass sie nie terminieren.

## Beispiele:

Motiv.

- (Mehrbenutzer-)Betriebssysteme sollen beliebig lange laufen ohne abzustürzen, egal was Benutzer tun
- Bankautomaten, Flugsicherungssysteme, Netzwerkkommunikationssysteme, . . .

#### Gängiges Berechnungsmodell:

- endliche Automaten mit nicht-terminierenden Berechnungen
- Terminierung wird als Nicht-Akzeptanz angesehen
- ursprünglich durch Büchi entwickelt (1960) Ziel: Algorithmen zur Entscheidung mathematischer Theorien

Teil 3: unendliche Wörter - Motivation

-Terminierung unerwünscht

sollen beliebig lange laufen ohne abzustürzen, egal was Benutzer tun Bankautomaten, Flursicherungssysteme. Natzwarkkommunikationssystema a endliche Automaten mit nicht-terminierenden Berechnungen

Von manchen Systemen/Programmen fordert man

u ursprünglich durch Büchi entwickelt (1960)

Ferminierung unerwünscht

a Terminierung wird als Nicht-Akzeptanz angesehe Ziel: Algorithmen zur Entscheidung mathematischer Theorie

# Ziel und Vorgehen dieses Kapitels

#### Ziel

Beschreibung von Automatenmodellen mit **unendlichen** Eingaben und **nicht-terminierenden** Berechnungen

#### Vorgehen

- Theorie: ausgiebiges Studium von Büchi-Automaten und der von ihnen erkannten Sprachen
  - Definition, Abschlusseigenschaften
  - Charakterisierung mittels regulärer Sprachen
  - Determinisierung
  - Entscheidungsprobleme
- Anwendung von Büchi-Automaten:
   Spezifikation & Verifikation in Linearer Temporallogik (LTL)

Teil 3: unendliche Wörter

Motivation

La de Businesse des Aspitels

La de Businesse von Annahmendelen mit semfliche Engaben

Worgen

Vorgen

La Ziel und Vorgehen dieses Kapitels

La Cautanamendelen en Rich Annahme

Onder on Neur Annahmendelen en Rich Annahmen

Onder on Neur Annahmendelen en Rich Annahmen

Onder on Neur Annahmen

Onder Annahmen

Onder Annahmen

Vorgen

Vorgen

Onder Annahmen

On

Ziel und Vorgehen dieses Kapitels

8:45

2019-01

Büchi-Aut. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

#### Beispiel: Philosophenproblem (Dining Philosophers Problem)

Erläutert Nebenläufigkeit und Verklemmung von Prozessen

Demonstriert auch unendliche Berechnungen

Hier: einfachste Version mit 3 Philosophen

Teil 3: unendliche Wörter - Motivation (Dining -Beispiel: Philosophenproblem

8:46

2019-01

Weil dieses Beispiel so schräg und archaisch ist, belasse ich es mal bei der männlichen Form.

Philosophers Problem)

Motiv.

Büchi-Aut. Abschlusseig. Motiv.

Determinismus

## Beispiel: Philosophenproblem

## (Dining Philosophers Problem)

Erläutert Nebenläufigkeit und Verklemmung von Prozessen

Demonstriert auch unendliche Berechnungen

Hier: einfachste Version mit 3 Philosophen

### Philosophenproblem

3 Philosophen  $P_1, P_2, P_3$ 

Für alle i gilt: entweder denkt  $P_i$ , oder  $P_i$  isst.

Alle  $P_i$  sitzen um einen runden Tisch.

Jeder  $P_i$  hat einen Teller mit Essen vor sich.

Zwischen je zwei Tellern liegt ein Essstäbchen.

Um zu essen, benötigt  $P_i$  beide Stäbchen neben seinem Teller.

Teil 3: unendliche Wörter - Motivation -Beispiel: Philosophenproblem Philosophers Problem)

#### 8:46

2019-01

Weil dieses Beispiel so schräg und archaisch ist, belasse ich es mal bei der männlichen Form.

## Beispiel: Philosophenproblem

## (Dining Philosophers Problem)

Model-Checking

Erläutert Nebenläufigkeit und Verklemmung von Prozessen

Demonstriert auch unendliche Berechnungen

Hier: einfachste Version mit 3 Philosophen

### Philosophenproblem

3 Philosophen  $P_1, P_2, P_3$ 

Für alle i gilt: entweder denkt  $P_i$ , oder  $P_i$  isst.

Alle  $P_i$  sitzen um einen runden Tisch.

Jeder  $P_i$  hat einen Teller mit Essen vor sich.

Zwischen je zwei Tellern liegt ein Essstäbchen.

Um zu essen, benötigt  $P_i$  beide Stäbchen neben seinem Teller.

 $\Rightarrow$  Keine zwei  $P_i$ ,  $P_i$  können gleichzeitig essen.

Teil 3: unendliche Wörter

Motivation

Beispiel: Philosophen Problem

Demonstrat and voluntum que no Personal

Beispiel: Philosophen Problem

Philosophers Problem

(Dining

Philosopher Table ... des P. tott.

Ander P. hat dem Table ... des P. tott.

Ander P. hat dem Table ... des P. tott.

Ander P. hat dem Table ... des P. tott.

Ander P. hat dem Table ... des P. tott.

The state ander contact for the finance out to the contact part of the language of the state des the state out to the state out to the state out to the language of the state des the state out to the state out to the language of the state des the state out to the state out to the language of the state des the state out to the state out to the language of the state des the state out to the language of the lang

#### 8:46

2019-01

Weil dieses Beispiel so schräg und archaisch ist, belasse ich es mal bei der männlichen Form.

# Skizze zum Philosophenproblem

#### Zusammenfassung

- Für alle i: entweder denkt  $P_i$ , oder  $P_i$  isst.
- ullet Keine zwei  $P_i, P_j$  können gleichzeitig essen.

Teil 3: unendliche Wörter

Motivation

Skizze zum Philosophenproblem

Skizze zum Philosophenproblem

F
ür alle i: entweder denkt P: oder P: isst.

. Keine zwei Pi, Pi können gleichzeitig essen.

Motiv. Büchi-Aut.

Abschlusseig.

Skizze zum Philosophenproblem

Charakt.

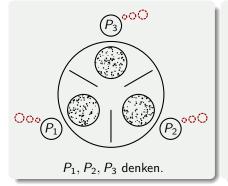
Determinismus

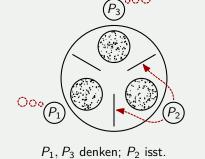
Entscheidungsprobl.

Model-Checking

## Zusammenfassung

- Für alle i: entweder denkt  $P_i$ , oder  $P_i$  isst.
- Keine zwei  $P_i$ ,  $P_i$  können gleichzeitig essen.





-Skizze zum Philosophenproblem

Teil 3: unendliche Wörter

- Motivation

# Modellierung durch endliches Transitionssystem

#### Annahmen

- Am Anfang denken (d) alle P<sub>i</sub>.
- Reihum können sich  $P_1, P_2, P_3$  entscheiden, ob sie denken oder essen (e) wollen.

#### Zustände des Systems

- Anfangszustand ddd1: alle  $P_i$  denken, und  $P_1$  trifft nächste Entscheidung.
- alle zulässigen Zustände:

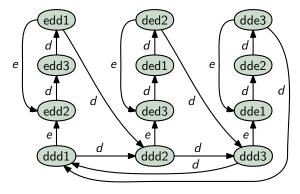
ddd1 edd1 ded1 dde1 ddd2 edd2 ded2 dde2 ddd3 edd3 ded3 dde3

## Zustandsüberführungen:

d oder e – je nach Entscheidung des  $P_i$ , der an der Reihe ist



## Das Transitionssystem

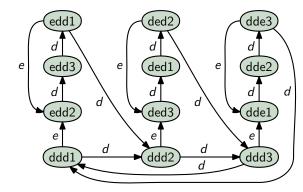


Teil 3: unendliche Wörter — *Motivation* 

□Das Transitionssystem

Das Transitionssystem

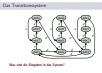
## Das Transitionssystem



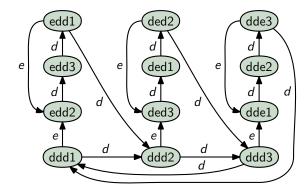
Was sind die Eingaben in das System?

Teil 3: unendliche Wörter — Motivation

└─Das Transitionssystem



## Das Transitionssystem

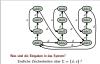


Was sind die Eingaben in das System?

Endliche Zeichenketten über  $\Sigma = \{d, e\}$ ?

Teil 3: unendliche Wörter *─ Motivation* 

☐Das Transitionssystem



Das Transitionssystem

8:51

Model-Checking

Motiv. Büchi-Aut.

Abschlusseig.

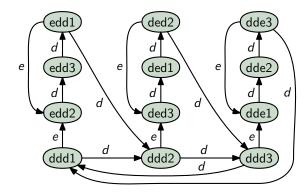
Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## Das Transitionssystem



#### Was sind die Eingaben in das System?

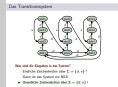
Endliche Zeichenketten über  $\Sigma = \{d, e\}$ ?

Dann ist das System ein NEA.

▶ Unendliche Zeichenketten über  $\Sigma = \{d, e\}!$ 

Teil 3: unendliche Wörter - Motivation

-Das Transitionssystem



## Warum unendliche Zeichenketten?

Nehmen an, jeder  $P_i$  möchte beliebig oft denken und essen.

System soll dazu beliebig lange ohne Terminierung laufen.

Philosoph  $P_i$  heißt zufrieden, wenn er währenddessen unendlich oft denkt und isst.

Teil 3: unendliche Wörter

-- Motivation

Warum unendliche Zeichenketten?

Nehmen an, jeder P; möchte beliebig oft denken und essen. System soll dazu beliebig lange ohne Terminierung laufen. Philosoph P; heißt zufrieden, wenn er währenddessen unendich oft denkt und isst.

Warum unendliche Zeichenketten?

Motiv. Büchi-Aut. Abschlusseig. Model-Checking Determinismus Entscheidungsprobl.

## Warum unendliche Zeichenketten?

Nehmen an, jeder  $P_i$  möchte beliebig oft denken und essen.

System soll dazu beliebig lange ohne Terminierung laufen.

Philosoph  $P_i$  heißt zufrieden, wenn er währenddessen unendlich oft denkt und isst.

#### → Mögliche Fragen:

- Kann das System überhaupt beliebig lange laufen?
- 2 lst es zusätzlich möglich, dass  $P_i$  zufrieden ist?
- $\odot$  Ist es möglich, dass  $P_1$ ,  $P_2$  zufrieden sind, aber  $P_3$  nicht?
- $\bullet$  Ist es möglich, dass alle  $P_i$  zufrieden sind?

Teil 3: unendliche Wörter - Motivation 2019-01

Warum unendliche Zeichenketten?

A Kann das System überhaupt beliebig lange laufen?

- - Milatirha Ernage

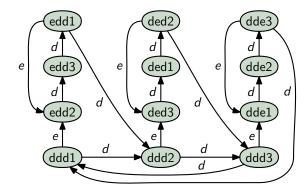
Warum unendliche Zeichenketten?

Philosoph P: heißt zufrieden, wenn er wührenddessen unendlich o Ist es zusätzlich möglich, dass P; zufrieden ist?

System soll dazu beliebig lange ohne Terminierung laufen

a lst es möglich, dass alle P: zufrieden sind

## Frage 1

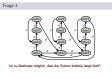


Ist es überhaupt möglich, dass das System beliebig lange läuft?

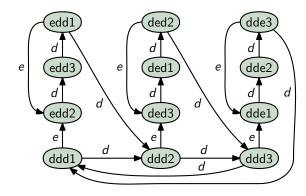
Teil 3: unendliche Wörter

Motivation

Frage 1

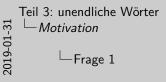


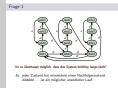
## Frage 1



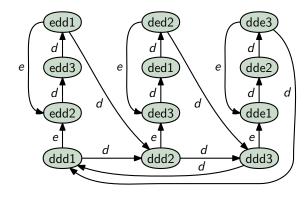
Ist es überhaupt möglich, dass das System beliebig lange läuft?

Ja: jeder Zustand hat mindestens einen Nachfolgerzustand. dddddd... ist ein möglicher unendlicher Lauf.

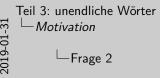




# Frage 2



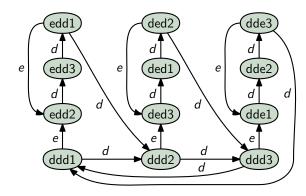
Ist es möglich, dass  $P_1$  zufrieden ist?



Frage 2

It is mightly, date P; addition int?

# Frage 2

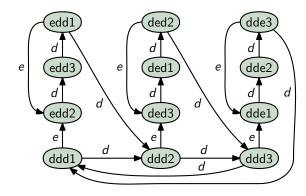


Ist es möglich, dass  $P_1$  zufrieden ist?

Ja: z. B. wenn ein Lauf ddd1 und edd1 unendlich oft durchläuft:  $ed^5ed^5...$ 



# Frage 3

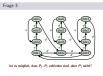


Ist es möglich, dass  $P_1$ ,  $P_2$  zufrieden sind, aber  $P_3$  nicht?

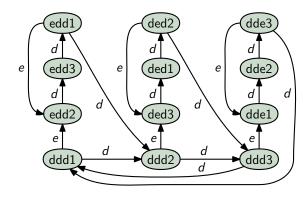
Teil 3: unendliche Wörter

Motivation

Frage 3

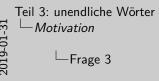


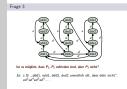
# Frage 3



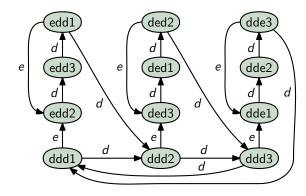
Ist es möglich, dass  $P_1$ ,  $P_2$  zufrieden sind, aber  $P_3$  nicht?

Ja: z. B. "ddd1, edd1, ddd2, ded2 unendlich oft, aber ddei nicht":  $ed^3ed^4ed^3ed^4\dots$ 





# Frage 4

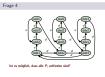


Ist es möglich, dass alle  $P_i$  zufrieden sind?

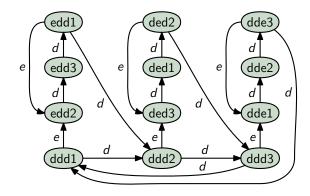
Teil 3: unendliche Wörter

- Motivation

- Frage 4

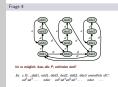


## Frage 4



Ist es möglich, dass alle  $P_i$  zufrieden sind?

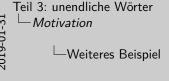
Ja: z. B. "ddd1, edd1, ddd2, ded2, ddd3, dde3 unendlich oft":  $ed^3ed^3\dots$  oder  $ed^2ed^3ed^2ed^3\dots$  oder ...



# Weiteres Beispiel

Motiv.

... siehe Anhang, Folie 122 ...



... siehe Anhang, Folie 122 .

Weiteres Beispiel

Und nun ...

- 2 Grundbegriffe und Büchi-Automaten
- Abschlusseigenschaften
- 5 Deterministische Büchi-Automaten und Determinisierung

Und nun . Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten 2019-01-31 Grundbegriffe und Büchi-Automaten └─Und nun ...

# Grundbegriffe

Motiv.

Unendliches Wort über Alphabet  $\Sigma$ 

- ist Funktion  $\alpha: \mathbb{N} \to \Sigma$
- $\alpha(n)$ : Symbol an *n*-ter Stelle (auch:  $\alpha_n$ )
- wird oft geschrieben als  $\alpha = \alpha_0 \alpha_1 \alpha_2 \dots$



Abschlusseig. Büchi-Aut.

Charakt. Determinismus

Entscheidungsprobl.

# Grundbegriffe

## Unendliches Wort über Alphabet $\Sigma$

- ist Funktion  $\alpha: \mathbb{N} \to \Sigma$
- $\alpha(n)$ : Symbol an *n*-ter Stelle (auch:  $\alpha_n$ )
- wird oft geschrieben als  $\alpha = \alpha_0 \alpha_1 \alpha_2 \dots$

#### Weitere Notation

- $\alpha[m, n]$ : endliche Teilfolge  $\alpha_m \alpha_{m+1} \dots \alpha_n$
- $\#_w(\alpha)$ : Anzahl der Vorkommen von w als Teilwort in  $\alpha$  $= \#\{(m, n) \mid \alpha[m, n] = w\}$
- $\mathbf{w}^{\omega}$ : unendliche Verkettung von w  $(\alpha \text{ mit } \alpha[i \cdot n, (i+1)n-1] = w \text{ f. alle } i \geqslant 0, n = |w|)$

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten -Grundbegriffe

Grundbegriffe

Unendliches Wort über Alphabet X  $\bullet$  ist Funktion  $\alpha:\mathbb{N}\to\Sigma$ 

u α(n): Symbol an n-ter Stelle (auch: α<sub>n</sub>)  $\mathbf{v}$  wird oft geschrieben als  $\alpha = \alpha_0\alpha_1\alpha_2$ 

#### Weitere Notation

- $\alpha[m,n]$ : endliche Teilfolge  $\alpha_m\alpha_{m+1}\dots\alpha_n$  #<sub>w</sub>(α): Anzahl der Vorkommen von w als Teilwort in α  $= \#\{(m, n) \mid \alpha(m, n) = w\}$
- w": unendliche Verkettung von w  $(\alpha \text{ mit } \alpha[i \cdot n, (i+1)n-1] = w \text{ f. alle } i \ge 0, n = |w|)$

9:02

2019-01-31

# Grundbegriffe

## Unendliches Wort über Alphabet $\Sigma$

- ist Funktion  $\alpha: \mathbb{N} \to \Sigma$
- $\alpha(n)$ : Symbol an *n*-ter Stelle (auch:  $\alpha_n$ )
- wird oft geschrieben als  $\alpha = \alpha_0 \alpha_1 \alpha_2 \dots$

#### Weitere Notation

- $\alpha[m, n]$ : endliche Teilfolge  $\alpha_m \alpha_{m+1} \dots \alpha_n$
- $\#_w(\alpha)$ : Anzahl der Vorkommen von w als Teilwort in  $\alpha$ =  $\#\{(m,n) \mid \alpha[m,n] = w\}$
- $\mathbf{w}^{\omega}$ : unendliche Verkettung von w  $(\alpha \text{ mit } \alpha[i \cdot n, (i+1)n-1] = w \text{ f. alle } i \geq 0, n = |w|)$

 $\Sigma^{\omega}$ : Menge aller unendlichen Wörter

$$ω$$
-Sprache:  $L ⊂ Σ^ω$ 

Teil 3: unendliche Wörter

Grundbegriffe und Büchi-Automaten

Grundbegriffe und Büchi-Automaten

Grundbegriffe

Grundbegriffe

Grundbegriffe

Grundbegriffe

Grundbegriffe

Grundbegriffe

Grundbegriffe

9:02

2019-01-31

Model-Checking Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

#### Büchi-Automaten

#### Definition 3.1

Ein nichtdeterministischer Büchi-Automat (NBA) über einem **Alphab**et  $\Sigma$  ist ein 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ , wobei

- Q eine endliche nichtleere Zustandsmenge ist.
- $\bullet$   $\Sigma$  eine endliche nichtleere Menge von Zeichen ist,
- $\Delta \subset Q \times \Sigma \times Q$  die Überführungsrelation ist,
- $I \subseteq Q$  die Menge der Anfangszustände ist,
- $F \subseteq Q$  die Menge der akzeptierenden Zustände ist.

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten

-Büchi-Automaten

u Q eine endliche nichtleere Zustandsmenge ist  $\bullet$   $\Sigma$  eine endliche nichtleere Menge von Zeichen ist Δ ⊂ O × Σ × O die Überführungsrelation ist. u / ⊂ O die Menee der Anfangszustände ist.

v F ⊂ O die Menge der akzeptierenden Zustände is

Büchi-Automaten

9:04

Automatentheorie u. i. A. WiSe 2018/19

#### Büchi-Automaten

#### Definition 3.1

Ein nichtdeterministischer Büchi-Automat (NBA) über einem Alphabet  $\Sigma$  ist ein 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ , wobei

- Q eine endliche nichtleere Zustandsmenge ist,
- $\bullet$   $\Sigma$  eine endliche nichtleere Menge von Zeichen ist,
- $\Delta \subseteq Q \times \Sigma \times Q$  die Überführungsrelation ist,
- $I \subset Q$  die Menge der Anfangszustände ist,
- $F \subset Q$  die Menge der akzeptierenden Zustände ist.

Bisher kein Unterschied zu NEAs, aber ...

Teil 3: unendliche Wörter

Grundbegriffe und Büchi-Automaten

Büchi-Automaten

Büchi-Automaten

Büchi-Automaten

Büchi-Automaten

Büchi-Automaten

Büchi-Automaten

Logical (2.5.4/1) webb (2.5.4/1) webb

Risher kein Unterschied zu NFAs aber

9:04

2019-01

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

# Berechnungen und Akzeptanz

#### Definition 3.2

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

• Ein Run von  $\mathcal{A}$  auf  $\omega$ -Wort  $\alpha$  ist eine Folge

$$r = q_0 q_1 q_2 \dots,$$

so dass für alle  $i \ge 0$  gilt:  $(q_i, \alpha_i, q_{i+1}) \in \Delta$ .

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten

Berechnungen und Akzeptanz

Berechnungen und Akzeptanz Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat. so dass für alle  $i \ge 0$  eilt:  $(\sigma_i, \alpha_i, \sigma_{i+1}) \in \Delta$ .

Büchi-Aut.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

# Berechnungen und Akzeptanz

## Definition 3.2

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

• Ein Run von  $\mathcal{A}$  auf  $\omega$ -Wort  $\alpha$  ist eine Folge

$$r = q_0 q_1 q_2 \dots$$

so dass für alle  $i \ge 0$  gilt:  $(q_i, \alpha_i, q_{i+1}) \in \Delta$ .

• Unendlichkeitsmenge Inf(r) von  $r = q_0q_1q_2...$ : Menge der Zustände, die unendlich oft in *r* vorkommen Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Berechnungen und Akzeptanz

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

Berechnungen und Akzeptanz

9:06

2019-01-31

# Berechnungen und Akzeptanz

## Definition 3.2

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

• Ein Run von  $\mathcal{A}$  auf  $\omega$ -Wort  $\alpha$  ist eine Folge

$$r = q_0 q_1 q_2 \dots$$

so dass für alle  $i \ge 0$  gilt:  $(q_i, \alpha_i, q_{i+1}) \in \Delta$ .

- Unendlichkeitsmenge Inf(r) von  $r = q_0q_1q_2...$ : Menge der Zustände, die unendlich oft in r vorkommen
- Erfolgreicher Run  $r = q_0 q_1 q_2 \dots q_0 \in I$  und  $Inf(r) \cap F \neq \emptyset$

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Berechnungen und Akzeptanz

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat. Erfolgreicher Run  $r = a_1 a_2 a_3 \dots$   $a_r \in I$  und  $lnf(r) \cap F \neq \emptyset$ 

Berechnungen und Akzeptanz

9:06

## Definition 3.2

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

• Ein Run von  $\mathcal{A}$  auf  $\omega$ -Wort  $\alpha$  ist eine Folge

$$r = q_0 q_1 q_2 \dots,$$

so dass für alle  $i \ge 0$  gilt:  $(q_i, \alpha_i, q_{i+1}) \in \Delta$ .

- Unendlichkeitsmenge Inf(r) von  $r = q_0q_1q_2...$ Menge der Zustände, die unendlich oft in *r* vorkommen
- Erfolgreicher Run  $r = q_0 q_1 q_2 \dots q_0 \in I$  und  $Inf(r) \cap F \neq \emptyset$
- $\mathcal{A}$  akzeptiert  $\alpha$ , wenn es einen erfolgreichen Run von  $\mathcal{A}$  auf  $\alpha$  gibt.

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Berechnungen und Akzeptanz

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat Menge der Zustände, die unendlich oft in r vorkommer a Erfolgreicher Run  $r = q_0q_1q_2...$   $q_0 ∈ I$  und lnf(r) ∩ F ≠ ∅

wenn es einen erfolgreichen Run von A auf  $\alpha$  gibt.

Berechnungen und Akzeptanz

9:06

## Berechnungen und Akzeptanz

## Definition 3.2

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat.

• Ein Run von  $\mathcal{A}$  auf  $\omega$ -Wort  $\alpha$  ist eine Folge

$$r = q_0 q_1 q_2 \dots$$

so dass für alle  $i \ge 0$  gilt:  $(q_i, \alpha_i, q_{i+1}) \in \Delta$ .

- Unendlichkeitsmenge Inf(r) von  $r = q_0q_1q_2...$ : Menge der Zustände, die unendlich oft in *r* vorkommen
- Erfolgreicher Run  $r = q_0 q_1 q_2 \dots : q_0 \in I$  und  $Inf(r) \cap F \neq \emptyset$
- $\bullet$  A akzeptiert  $\alpha$ . wenn es einen erfolgreichen Run von  $\mathcal{A}$  auf  $\alpha$  gibt.
- Die von  $\mathcal{A}$  erkannte Sprache ist  $L_{\omega}(\mathcal{A}) = \{ \alpha \in \Sigma^{\omega} \mid \mathcal{A} \text{ akzeptiert } \alpha \}.$

Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Berechnungen und Akzeptanz

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein Büchi-Automat Menge der Zustände, die unendlich oft in r vorkommer a Erfolgreicher Run  $r = q_0q_1q_2...: q_0 \in I$  und  $lnf(r) \cap F \neq \emptyset$ wenn es einen erfolgreichen Run von A auf  $\alpha$  gibt

Berechnungen und Akzeptanz

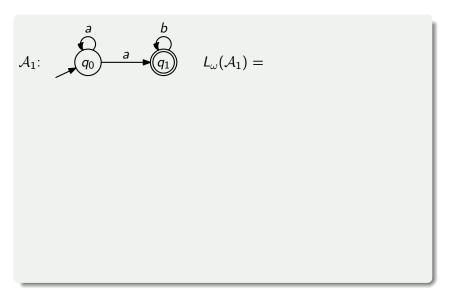
a Die von A erkannte Sprache ist  $L_{-}(A) = \{\alpha \in \Sigma^{\omega} \mid A \text{ akzentiert } \alpha\}$ 

9:06

2019-01-

# Beispiele

Motiv.



Teil 3: unendliche Wörter -Grundbegriffe und Büchi-Automaten 2019-01-31 └─Beispiele

Motiv.

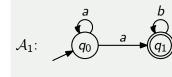
Beispiele

Charakt. Determinismus

Model-Checking

Entscheidungsprobl.





$$L_{\omega}(\mathcal{A}_1) = \{a^n b^{\omega} \mid n \geqslant 1\}$$

Teil 3: unendliche Wörter 2019-01-31 -Grundbegriffe und Büchi-Automaten └─Beispiele

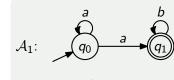


Charakt. Determinismus Entscheidungsprobl.

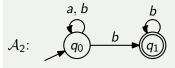
Model-Checking

## Beispiele

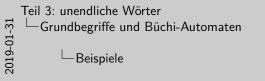
Motiv.



$$L_{\omega}(\mathcal{A}_1) = \{a^n b^{\omega} \mid n \geqslant 1\}$$



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



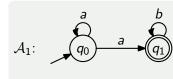
Beispiele  $A_1$ : b  $L_{\omega}(A_1) = \{s^ab^{\omega} \mid n \geqslant 1\}$ 

Charakt.

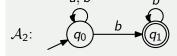
Determinismus

Entscheidungsprobl.

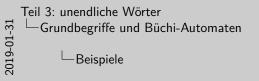
Motiv.



$$L_{\omega}(\mathcal{A}_1) = \{a^n b^{\omega} \mid n \geqslant 1\}$$

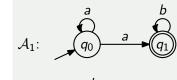


$$L_{\omega}(\mathcal{A}_2) = \{ \alpha \in \Sigma^{\omega} \mid \#_{a}(\alpha) < \infty \}$$

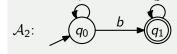




Motiv.



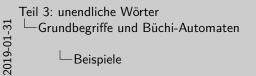
$$L_{\omega}(\mathcal{A}_1) = \{a^n b^{\omega} \mid n \geqslant 1\}$$

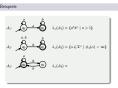


$$L_{\omega}(\mathcal{A}_2) = \{ \alpha \in \Sigma^{\omega} \mid \#_{a}(\alpha) < \infty \}$$

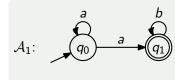
$$A_3$$
:  $q_0$   $b$   $q_1$ 

$$L_{\omega}(\mathcal{A}_3) =$$

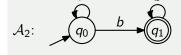




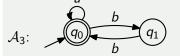
Motiv.



$$L_{\omega}(\mathcal{A}_1) = \{a^n b^{\omega} \mid n \geqslant 1\}$$

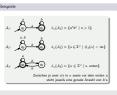


$$L_{\omega}(\mathcal{A}_2) = \{ \alpha \in \Sigma^{\omega} \mid \#_{a}(\alpha) < \infty \}$$



$$L_{\omega}(\mathcal{A}_3) = \{ \alpha \in \Sigma^{\omega} \mid \mathsf{s. unten} \}$$

Zwischen je zwei a's in  $\alpha$  sowie vor dem ersten a steht jeweils eine gerade Anzahl von b's. Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Beispiele



9:08 bis 9:14

2019-01-31

Büchi-Aut. Abschlusseig. Charakt.

# Mehr Beispiele

a, b

Motiv.

Determinismus

 $L_{\omega}(\mathcal{A}_4) =$ 



Entscheidungsprobl.

Model-Checking Teil 3: unendliche Wörter

2019-01-31

-Grundbegriffe und Büchi-Automaten

└─Mehr Beispiele

Mehr Beispiele

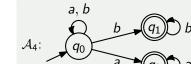
Charakt. Determinismus

Entscheidungsprobl.

Model-Checking

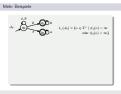
# Mehr Beispiele

Motiv.



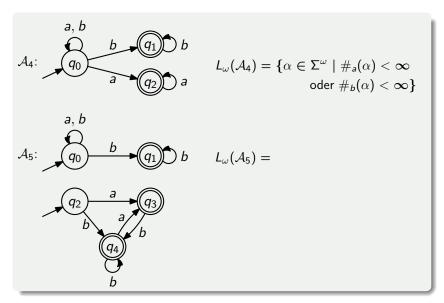
$$L_{\omega}(\mathcal{A}_4) = \{ \alpha \in \Sigma^{\omega} \mid \#_{\mathfrak{a}}(\alpha) < \infty \\ \text{oder } \#_{\mathfrak{b}}(\alpha) < \infty \}$$

Teil 3: unendliche Wörter -Grundbegriffe und Büchi-Automaten └─Mehr Beispiele



# Mehr Beispiele

Motiv.



Teil 3: unendliche Wörter

Grundbegriffe und Büchi-Automaten

Mehr Beispiele

Mehr Beispiele

Büchi-Aut.

a, b

*a*, *b* 

Mehr Beispiele

Motiv.

Abschlusseig.

 $L_{\omega}(\mathcal{A}_4) = \{ \alpha \in \Sigma^{\omega} \mid \#_{a}(\alpha) < \infty \}$ 

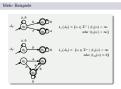
 $L_{\omega}(\mathcal{A}_5) = \{\alpha \in \Sigma^{\omega} \mid \#_{a}(\alpha) < \infty$ 

oder  $\#_b(\alpha) < \infty$ }

oder  $\#_{aa}(\alpha) = 0$ 

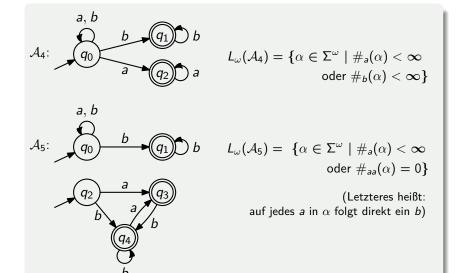
Grundbegriffe und Büchi-Automaten

Mehr Beispiele



Motiv.

2019-01-31



Teil 3: unendliche Wörter Grundbegriffe und Büchi-Automaten Mehr Beispiele

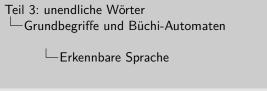
Mehr Beispiele  $L_{\omega}(A_t) = \{\alpha \in \Sigma^{\omega} \mid \#_{s}(\alpha) < \infty \}$ oder  $\#_{k}(\alpha) < \infty\}$ (Letzteres heißt: auf iedes a in o folgt direkt ein b)

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

# Erkennbare Sprache

## Definition 3.3

Eine Sprache  $L \subseteq \Sigma^{\omega}$  ist Büchi-erkennbar, wenn es einen NBA  $\mathcal{A}$  gibt mit  $L = L_{\omega}(\mathcal{A})$ .



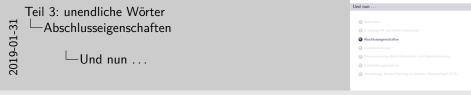
Definition 3.3 Eine Sprache  $L\subseteq \Sigma^\omega$  ist Büchi-erkennbar, winn as einen NBA  $\mathcal A$  gibt mit  $L=L_\omega(\mathcal A)$ .

Erkennbare Sprache

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Und nun ...

- Motivation
- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschaften
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierun,
- 6 Entscheidungsproblem
- Anwendung: Model-Checking in Linearer Temporallogik (LTL



Büchi-Aut. Abschlusseig. Determinismus Entscheidungsprobl. Model-Checking

# Operationen auf $\omega$ -Sprachen

Zur Erinnerung: die Menge der Büchi-erkennbaren Sprachen heißt abgeschlossen unter

- Vereiniqung, wenn gilt: Falls  $L_1$ ,  $L_2$  Büchi-erkennbar, so auch  $L_1 \cup L_2$ .
- Schnitt, wenn gilt: Falls  $L_1$ ,  $L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ .
- Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch  $\overline{L}$ .

Teil 3: unendliche Wörter Abschlusseigenschaften 2019-01

-Operationen auf  $\omega$ -Sprachen

a Vereinigung, wenn eilt:

Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ 

Operationen auf ω-Sprachen

a Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch T.

iv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Operationen auf $\omega$ -Sprachen

Zur Erinnerung: die Menge der Büchi-erkennbaren Sprachen heißt abgeschlossen unter

- Vereinigung, wenn gilt:
   Falls L<sub>1</sub>, L<sub>2</sub> Büchi-erkennbar, so auch L<sub>1</sub> ∪ L<sub>2</sub>.
  - Talls  $L_1$ ,  $L_2$  Ductin-erkellibar, so auch  $L_1$
- Schnitt, wenn gilt:
   Falls L<sub>1</sub>, L<sub>2</sub> Büchi-erkennbar, so auch L<sub>1</sub> ∩ L<sub>2</sub>.
- Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch  $\overline{L}$ .

## Quiz

Unter welchen Operationen sind die Büchi-erkennbaren Sprachen abgeschlossen, und wie leicht ist das zu zeigen?

Vereinigung? Schnitt? Komplement? Teil 3: unendliche Wörter

—Abschlusseigenschaften

-Operationen auf  $\omega$ -Sprachen

Falls L<sub>1</sub> L<sub>2</sub> Bloth-irkenthar, so auch L<sub>1</sub> ∩ L<sub>2</sub> • Komphament, were git: Falls L Bloth-irkenthar, so auch Z.

General Comparations sind die Bloth-irkentharen Sprachen begondhossen, with wis felicht ist das zu zeigen?

Verwingung?

Verwingung?

Komphament?

Operationen auf ω-Sprachen

a Vereinigung, wenn eilt:

9:21

Büchi-Aut. Abschlusseig. Determinismus Entscheidungsprobl. Model-Checking

# Operationen auf $\omega$ -Sprachen

Zur Erinnerung: die Menge der Büchi-erkennbaren Sprachen heißt abgeschlossen unter

- Vereinigung, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cup L_2$ .
- Schnitt, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ .
- Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch  $\overline{L}$ .

## Quiz

Unter welchen Operationen sind die Büchi-erkennbaren Sprachen abgeschlossen, und wie leicht ist das zu zeigen?

> Vereinigung? (leicht) Schnitt? Komplement?

Teil 3: unendliche Wörter -Abschlusseigenschaften 2019-01

-Operationen auf  $\omega$ -Sprachen



Operationen auf ω-Sprachen

a Vereinigung, wenn eilt:

Büchi-Aut. Abschlusseig. Determinismus Entscheidungsprobl. Model-Checking

# Operationen auf $\omega$ -Sprachen

Zur Erinnerung: die Menge der Büchi-erkennbaren Sprachen heißt abgeschlossen unter

- Vereinigung, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cup L_2$ .
- Schnitt, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ .
- Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch  $\overline{L}$ .

## Quiz

Unter welchen Operationen sind die Büchi-erkennbaren Sprachen abgeschlossen, und wie leicht ist das zu zeigen?

> Vereinigung? (leicht) Schnitt? (mittel) Komplement?

Teil 3: unendliche Wörter -Abschlusseigenschaften 2019-01

-Operationen auf  $\omega$ -Sprachen

Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ abgeschlossen, und wie leicht ist das zu zeigen?

Operationen auf ω-Sprachen

a Vereinigung, wenn eilt:

Büchi-Aut. Abschlusseig. Entscheidungsprobl. Model-Checking Determinismus

# Operationen auf $\omega$ -Sprachen

Zur Erinnerung: die Menge der Büchi-erkennbaren Sprachen heißt abgeschlossen unter

- Vereiniqung, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cup L_2$ .
- Schnitt, wenn gilt: Falls  $L_1, L_2$  Büchi-erkennbar, so auch  $L_1 \cap L_2$ .
- Komplement, wenn gilt: Falls L Büchi-erkennbar, so auch  $\overline{L}$ .

## Quiz

Unter welchen Operationen sind die Büchi-erkennbaren Sprachen abgeschlossen, und wie leicht ist das zu zeigen?

> Vereinigung? (leicht) Schnitt? (mittel) Komplement? (schwer)

Teil 3: unendliche Wörter -Abschlusseigenschaften 2019-01

-Operationen auf  $\omega$ -Sprachen



Operationen auf ω-Sprachen

a Vereinigung, wenn eilt:

tiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Abgeschlossenheit

## Satz 3.4

Die Menge der Büchi-erkennbaren Sprachen ist abgeschlossen unter den Operationen  $\cup$  und  $\cap$ .

Direkte Konsequenz aus den folgenden Lemmata.

Abgeschlossenheit unter —: siehe Abschnitt "Determinisierung"



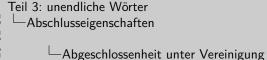
Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Abgeschlossenheit unter Vereinigung

## Lemma 3.5

Seien  $\mathcal{A}_1, \mathcal{A}_2$  NBAs über  $\Sigma.$ 

Dann gibt es einen NBA  $A_3$  mit  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ .



Lemma 3.5 Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ . Dann gibt as einen NBA  $A_3$  mit  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ .

Abgeschlossenheit unter Vereinigung

Büchi-Aut.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

# Abgeschlossenheit unter Vereinigung

## Lemma 3.5

Seien  $A_1, A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $A_3$  mit  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ .

Beweis. analog zu NEAs und NEBAs:

Seien 
$$A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$$
 für  $i = 1, 2$ .

O. B. d. A. gelte  $Q_1 \cap Q_2 = \emptyset$ .

Konstruieren  $A_3 = (Q_3, \Sigma, \Delta_3, I_3, F_3)$  wie folgt.

Teil 3: unendliche Wörter Abschlusseigenschaften

—Abgeschlossenheit unter Vereinigung

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ . Dann gibt es einen NBA  $A_3$  mit  $L_1(A_3) = L_1(A_1) \cup L_1(A_2)$ . Beweis, analog zu NEAs und NEBAs: O. B. d. A. selte Q₁ ∩ Q₂ = ∅.

Abgeschlossenheit unter Vereinigung

9:24

 $Q_1 = Q_1 \cup Q_2$ 

•  $\Delta_3 = \Delta_1 \cup \Delta_2$ •  $t_3 = t_1 \cup t_2$ •  $F_3 = F_1 \cup F_2$ 

# Abgeschlossenheit unter Vereinigung

## Lemma 3.5

Seien  $A_1, A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $A_3$  mit  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ .

Beweis. analog zu NEAs und NEBAs:

Seien 
$$A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$$
 für  $i = 1, 2$ .

O. B. d. A. gelte  $Q_1 \cap Q_2 = \emptyset$ .

Konstruieren  $A_3 = (Q_3, \Sigma, \Delta_3, I_3, F_3)$  wie folgt.

• 
$$Q_3 = Q_1 \cup Q_2$$

$$\bullet \ \Delta_3 = \Delta_1 \cup \Delta_2$$

• 
$$I_3 = I_1 \cup I_2$$

• 
$$F_3 = F_1 \cup F_2$$

Teil 3: unendliche Wörter

Abschlusseigenschaften

Abgeschlossenheit unter Vereinigung

[Lemma 3.5] Seise  $A_{k}$ ,  $A_{k}$  NBAs über  $\Sigma$ Dans gibt es einem NBA  $A_{k}$  mit  $L_{n}(A_{k}) = L_{n}(A_{k}) \cup L_{n}(A_{k})$ . Brunk. analog  $z_{k}$  NEAs und NEBAs: Seise  $A_{k} = (Q_{k}, \Sigma_{n}, L_{k}, L_{k})$  für i = 1, 2. O. B. d. A. gibt  $Q_{k} \cap Q_{k} = 0$ .

9:24

Büchi-Aut. Abschlusseig. Entscheidungsprobl. Model-Checking Charakt. Determinismus

# Abgeschlossenheit unter Vereinigung

## Lemma 3.5

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $A_3$  mit  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ .

Beweis. analog zu NEAs und NEBAs:

Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  für i = 1, 2.

O. B. d. A. gelte  $Q_1 \cap Q_2 = \emptyset$ .

Konstruieren  $A_3 = (Q_3, \Sigma, \Delta_3, I_3, F_3)$  wie folgt.

- $Q_3 = Q_1 \cup Q_2$
- $\bullet$   $\Delta_3 = \Delta_1 \cup \Delta_2$
- $I_3 = I_1 \cup I_2$
- $F_3 = F_1 \cup F_2$

Dann gilt:  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ 

Teil 3: unendliche Wörter -Abschlusseigenschaften

—Abgeschlossenheit unter Vereinigung

Abgeschlossenheit unter Vereinigung Seien A1, A2 NBAs über Σ. Dann gibt es einen NBA  $A_3$  mit  $L_1(A_3) = L_1(A_1) \cup L_1(A_2)$ . Beweis, analog zu NEAs und NEBAs: O. B. d. A. selte  $Q_1 \cap Q_2 = \emptyset$ .  $a O_1 = O_1 \cup O_2$  $\mathbf{v} \cdot \Delta_1 = \Delta_1 \cup \Delta_2$  $\bullet$   $I_1 = I_1 \cup I_2$  $a F_1 = F_1 \cup F_2$ Dann gilt:  $L_{\omega}(A_3) = L_{\omega}(A_1) \cup L_{\omega}(A_2)$ 

9:24

Büchi-Aut.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

### Für NEAs: Produktautomat

Idee: lasse  $A_1$  und  $A_2$  "gleichzeitig" auf Eingabewort laufen.

Gegeben  $A_1, A_2$ , konstruiere  $A_3$  mit  $L(A_3) = L(A_1) \cap L(A_2)$ :

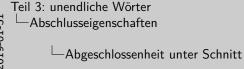
• 
$$Q_3 = Q_1 \times Q_2$$

• 
$$\Delta_3 = \{((p, p'), a, (q, q')) \mid (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2\}$$

• 
$$I_3 = I_1 \times I_2$$

• 
$$F_3 = F_1 \times F_2$$

T 3.1



Idee: lasse  $A_1$  und  $A_2$  "gleichzeitig" auf Eingabewort laufen. ■  $\Delta_1 = \{((\rho, \rho'), a, (q, q')) \mid (\rho, a, q) \in \Delta_1 \& (\rho', a, q') \in \Delta_2\}$  $\mathbf{v} h = h \times h$  $\bullet F_1 = F_1 \times F_2$ 

9:25 bis 9:35?

# Abgeschlossenheit unter Schnitt

### Für NEAs: Produktautomat

Idee: lasse  $A_1$  und  $A_2$  "gleichzeitig" auf Eingabewort laufen.

Gegeben  $A_1, A_2$ , konstruiere  $A_3$  mit  $L(A_3) = L(A_1) \cap L(A_2)$ :

$$Q_3 = Q_1 \times Q_2$$

• 
$$\Delta_3 = \{((p, p'), a, (q, q')) \mid (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2\}$$

• 
$$I_3 = I_1 \times I_2$$

• 
$$F_3 = F_1 \times F_2$$

T 3.1

Funktioniert das auch für Büchi-Automaten?

Teil 3: unendliche Wörter -Abschlusseigenschaften

—Abgeschlossenheit unter Schnitt

Idee: lasse  $A_1$  und  $A_2$  "gleichzeitig" auf Eingabewort laufen. ■  $\Delta_1 = \{((\rho, \rho'), a, (q, q')) \mid (\rho, a, q) \in \Delta_1 \& (\rho', a, q') \in \Delta_2\}$  $\mathbf{v} h = h \times h$ 

 $\phi F_1 = F_1 \times F_2$ Funktioniert das auch für Büchi-Automaten

Abgeschlossenheit unter Schnitt

9:25 bis 9:35?

## Abgeschlossenheit unter Schnitt

### Für NEAs: Produktautomat

Idee: lasse  $A_1$  und  $A_2$  "gleichzeitig" auf Eingabewort laufen.

Gegeben  $A_1, A_2$ , konstruiere  $A_3$  mit  $L(A_3) = L(A_1) \cap L(A_2)$ :

- $Q_3 = Q_1 \times Q_2$
- $\Delta_3 = \{((p, p'), a, (q, q')) \mid (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2\}$
- $l_3 = l_1 \times l_2$
- $F_3 = F_1 \times F_2$

T 3.1

### Funktioniert das auch für Büchi-Automaten?

**Nein**.  $A_1$  und  $A_2$  besuchen ihre akzeptierenden Zustände möglicherweise nicht synchron! T 3.1 Forts.

Teil 3: unendliche Wörter -Abschlusseigenschaften —Abgeschlossenheit unter Schnitt Abgeschlossenheit unter Schnitt Idee: lasse As und As "eleichzeitie" auf Einzabewort laufen ■  $\Delta_1 = \{((\rho, \rho'), a, (q, q')) \mid (\rho, a, q) \in \Delta_1 \& (\rho', a, q') \in \Delta_2\}$  $\mathbf{v} h = h \times h$  $\phi F_1 = F_1 \times F_2$ möelicherweise nicht synchron!

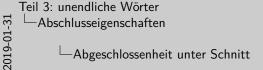
9:25 bis 9:35?

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Abgeschlossenheit unter Schnitt

## **Neue Idee** für Schnitt-Automat A:

- ullet  $\mathcal{A}$  simuliert  $\mathcal{A}_1, \mathcal{A}_2$  nach wie vor parallel, aber mit 2 Modi 1,2
- ullet Modus i bedeutet: warte auf einen akz. Zustand f von  $\mathcal{A}_i$
- Sobald so ein f erreicht ist, wechsle den Modus.
- Run von  $\mathcal{A}$  ist erfolgreich, wenn er  $\infty$  oft den Modus wechselt.
- $\rightarrow$  Es werden genau die Wörter akzeptiert, für die  $A_1, A_2$  jeweils einen erfolgreichen Run haben.



Nese Idee für Schnitt-Automat A:

• A simuliert A<sub>1</sub>, A<sub>2</sub> nach wie vor parallel, aber mit 2 Modi 1, 2

• Modus i bedeutet: warte auf einen akz. Zustand f von A:

Modus i bedeutet: warte auf einen akz. Zustand f von A;
 Sobald so ein f erreicht ist, wechsle den Modus.

■ Sobald so ein f erreicht ist, wechsle den Modus.
• Run von A ist erfolgreich, wenn er ∞ oft den Modus wechselt

Abgeschlossenheit unter Schnitt

Es werden genau die W\u00f6rter akzeptiert, f\u00fcr die A<sub>1</sub>, A<sub>2</sub> jewei\u00e4s einen erfolgreichen Run haben.

Motiv. Büchi-Aut. Abschlusseig.

Charakt.

akt. Determinismus

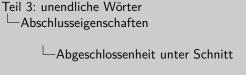
Entscheidungsprobl.

ann sibt es einen NBA A mit  $L \cdot (A) = L \cdot (A_1) \cap L \cdot (A_2)$ .

## Lemma 3.6

Seien  $A_1, A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .



**16:00** 9:37 bis 9:59?

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

Büchi-Aut.

Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

2019-01

## Abgeschlossenheit unter Schnitt

### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

Teil 3: unendliche Wörter -Abschlusseigenschaften

Abgeschlossenheit unter Schnitt

Abgeschlossenheit unter Schnitt ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$  $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2\}$ 

9:37 bis 9:59? 16:00

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

Büchi-Aut

Abschlusseig.

Charakt.

### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

Teil 3: unendliche Wörter -Abschlusseigenschaften

ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$ Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2 $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $\cup \{((p, q', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (q', a, q') \in \Delta_2\}$ 

Abgeschlossenheit unter Schnitt

Abgeschlossenheit unter Schnitt

9:37 bis 9:59? 16:00

2019-01

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

Büchi-Aut

Abschlusseig.

Charakt. Determinismus

Entscheidungsprobl.

## Abgeschlossenheit unter Schnitt

### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

Teil 3: unendliche Wörter -Abschlusseigenschaften Abgeschlossenheit unter Schnitt

ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$ leweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2 $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $\cup \{((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$ 

Abgeschlossenheit unter Schnitt

9:37 bis 9:59? 16:00

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 1)) \mid p' \in F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

-Abschlusseigenschaften

Abgeschlossenheit unter Schnitt

ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$ leweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2 $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $\cup \{((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2\}$  $\cup \{((p, p', 2), a, (p, p', 1)) \mid p' \in F, \& (p, a, p) \in \Delta_1 \& (p', a, p') \in \Delta_2\}$ 

9:37 bis 9:59? 16:00

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

## Abgeschlossenheit unter Schnitt

#### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 1)) \mid p' \in F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$I = I_1 \times I_2 \times \{1\}$$

$$F = Q_1 \times F_2 \times \{2\}$$

T 3.2

Teil 3: unendliche Wörter -Abschlusseigenschaften

Abgeschlossenheit unter Schnitt

ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$ leweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2 $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $\cup \{((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $F = Q_1 \times F_2 \times \{2\}$ 

Abgeschlossenheit unter Schnitt

9:37 bis 9:59? 16:00

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

Beweis bis 16:20

Motiv Büchi-Aut Abschlusseig. Charakt.

Determinismus

### Lemma 3.6

Seien  $A_1$ ,  $A_2$  NBAs über  $\Sigma$ .

Dann gibt es einen NBA  $\mathcal{A}$  mit  $L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}_1) \cap L_{\omega}(\mathcal{A}_2)$ .

Beweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2.

Konstruieren  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  wie folgt.

$$Q = Q_1 \times Q_2 \times \{1, 2\}$$

$$\Delta = \{ ((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 1), a, (q, q', 2)) \mid p \in F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\cup \{ ((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$$

$$\bigcup \{ ((p, p', 2), a, (q, q', 1)) \mid p' \in F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \} 
I = I_1 \times I_2 \times \{1\}$$

$$F = Q_1 \times F_2 \times \{2\}$$

T 3.2 Forts.

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

T 3.2

-Abschlusseigenschaften

Abgeschlossenheit unter Schnitt

ann sibt es einen NBA A mit  $L_{-}(A) = L_{-}(A_{1}) \cap L_{-}(A_{2})$ leweis: Seien  $A_i = (Q_i, \Sigma, \Delta_i, I_i, F_i)$  NBAs für i = 1, 2 $\Delta = \{((p, p', 1), a, (q, q', 1)) \mid p \notin F_1 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $\cup \{((p, p', 2), a, (q, q', 2)) \mid p' \notin F_2 \& (p, a, q) \in \Delta_1 \& (p', a, q') \in \Delta_2 \}$  $F = O_1 \times F_2 \times \{2\}$ Dann gilt  $L_{\omega}(A) = L_{\omega}(A_1) \cap L_{\omega}(A_2)$ . T 3.2 Forts.

Abgeschlossenheit unter Schnitt

9:37 bis 9:59? 16:00

Teil 3: unendliche Wörter

Kurze Wdhlg.: haben Büchi-Automaten eingeführt, Abschlusseigenschaften behandelt.

Vereinigung war einfach; Schnitt ist komplizierter:

2 Kopien des "alten" Produktaut., weil akz. Zust. asynchron auftreten können

Konstruktion auf Folie; jetzt noch 2. Richtung des Korrektheitsbeweises

Beweis bis 16:20

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Abgeschlossenheit unter Komplement

... siehe Abschnitt "Deterministische Büchi-Automaten und Determinisierung"



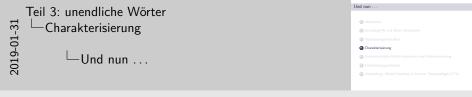
**16:20** 9:59–10:00, Ende

. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

### Und nun ...

Motivatio

- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschafter
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierung
- 6 Entscheidungsproblem
- Anwendung: Model-Checking in Linearer Temporallogik (LTL

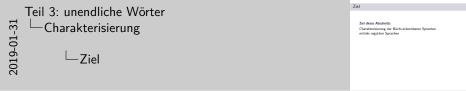


Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Motiv. Ziel

**Ziel dieses Abschnitts** 

Charakterisierung der Büchi-erkennbaren Sprachen mittels regulärer Sprachen



16:20

 $W^\omega$  entspricht dem Kleene-Stern bei Sprachen endlicher Wörter;  $W\!L$  entspricht der Konkatenation.

Büchi-Aut. Abschlusseig.

Charakt.

Determinismus

### Ziel dieses Abschnitts

Charakterisierung der Büchi-erkennbaren Sprachen mittels regulärer Sprachen

#### **Etwas Notation**

Seien  $W \subseteq \Sigma^*$  und  $L \subseteq \Sigma^\omega$ .

- $W^{\omega} = \{w_0 w_1 w_2 \cdots \mid w_i \in W \setminus \{\varepsilon\} \text{ für alle } i \geqslant 0\}$ (ist  $\omega$ -Sprache, weil  $\varepsilon$  ausgeschlossen wurde)
- $WL = \{ w\alpha \mid w \in W, \ \alpha \in L \}$ (ist  $\omega$ -Sprache)

Ziel Teil 3: unendliche Wörter 7iel dieses Aberbeitts Charakterisierung Charakterisierung der Büchi-erkennbaren Sprachen mittels regulärer Sprachen  $\mathbf{w} \ \mathbf{W}^{\omega} = \{w_0w_1w_2 \cdots \mid w_i \in W \setminus \{\varepsilon\} \text{ für alle } i \geqslant 0\}$ └ Ziel •  $WL = \{w\alpha \mid w \in W, \alpha \in L\}$ (ist ω-Sprache)

#### 16:20

2019-01

 $W^{\omega}$  entspricht dem Kleene-Stern bei Sprachen endlicher Wörter; WL entspricht der Konkatenation.

Entscheidungsprobl.

Model-Checking

## Von regulären zu Büchi-erkennbaren Sprachen (1)

Charakt.

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Teil 3: unendliche Wörter Charakterisierung

└Von regulären zu Büchi-erkennbaren Sprachen (1)

#### 16:22

(Idee: Füge neuen Startzustand hinzu; dupliziere alle Kanten, die von bisherigen SZen ausgehen.

Damit sind (1) und (2) erreicht. (3) ist korrekt, weil  $\varepsilon \notin L(A_1)$ .

$$W = \{a^n \mid n \text{ ist prim}\}\$$

$$\Rightarrow W^{\omega} = \{a^{\omega}\} \qquad (\text{und } W^* = \{a^n \mid n \ge 2\})$$

Model-Checking

## Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 1)

Sei  $\mathcal{A}$  ein **NEA** mit  $L(\mathcal{A}) = W$ .

Charakterisierung Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennb. Sei 4 ein NFA mit I (4) - W └Von regulären zu Büchi-erkennbaren Sprachen (1)

16:22

2019-01

(Idee: Füge neuen Startzustand hinzu; dupliziere alle Kanten, die von bisherigen SZen ausgehen.

Damit sind (1) und (2) erreicht. (3) ist korrekt, weil  $\varepsilon \notin L(A_1)$ .

$$W = \{a^n \mid n \text{ ist prim}\}$$
  

$$\Rightarrow W^{\omega} = \{a^{\omega}\} \qquad (\text{und } W^* = \{a^n \mid n \ge 2\})$$

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 1)

Sei A ein **NEA** mit L(A) = W.

Dann gibt es NEA  $A_1$  mit  $L(A_1) = W \setminus \{\varepsilon\}$  (Abschlusseig.!)

Teil 3: unendliche Wörter
Charakterisierung

Von regulären zu Büchi-erkennbaren Sprachen (1)

Lemma 3.7

Für irde reguläre Sorache W C X\* gilt: W~ ist Büchi-erkennbar.

Beseix. (Schritt 1)
Sei A ein NEA mit L(A) = W.

└Von regulären zu Büchi-erkennbaren Sprachen (1)

#### 16:22

2019-01

(Idee: Füge neuen Startzustand hinzu; dupliziere alle Kanten, die von bisherigen SZen ausgehen.

Damit sind (1) und (2) erreicht. (3) ist korrekt, weil  $\varepsilon \notin L(A_1)$ .

$$W = \{a^n \mid n \text{ ist prim}\}$$
  

$$\Rightarrow W^{\omega} = \{a^{\omega}\} \qquad (\text{und } W^* = \{a^n \mid n \ge 2\})$$

## Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 1)

Sei  $\mathcal{A}$  ein **NEA** mit  $L(\mathcal{A}) = W$ .

Dann gibt es NEA  $A_1$  mit  $L(A_1) = W \setminus \{\varepsilon\}$  (Abschlusseig.!)

- O. B. d. A. habe  $A_1$  . . .
- $\bullet$  einen einzigen Anfangszustand  $q_I$  und
- **2** keine in  $q_l$  eingehenden Kanten: keine Transitionen  $(\cdot, \cdot, q_l)$
- $\bigcirc$  und sei  $q_1 \notin F$ .

Teil 3: unendliche Wörter -Charakterisierung Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennb. └Von regulären zu Büchi-erkennbaren Sprachen (1

#### 16:22

2019-01

(Idee: Füge neuen Startzustand hinzu; dupliziere alle Kanten, die von bisherigen SZen ausgehen.

Damit sind (1) und (2) erreicht. (3) ist korrekt, weil  $\varepsilon \notin L(A_1)$ .

$$W = \{a^n \mid n \text{ ist prim}\}$$

$$\Rightarrow W^{\omega} = \{a^{\omega}\} \qquad (\text{und } W^* = \{a^n \mid n \ge 2\}))$$

## Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 1)

Sei  $\mathcal{A}$  ein **NEA** mit  $L(\mathcal{A}) = W$ .

Dann gibt es NEA  $A_1$  mit  $L(A_1) = W \setminus \{\varepsilon\}$  (Abschlusseig.!)

- O. B. d. A. habe  $A_1$  . . .
- einen einzigen Anfangszustand quund
- **2** keine in  $q_i$  eingehenden Kanten: keine Transitionen  $(\cdot, \cdot, q_i)$
- $\odot$  und sei  $q_i \notin F$ .

Diese Form lässt sich durch Hinzufügen eines frischen Anfangszustandes (und der entsprechenden Transitionen) erreichen! (Ü) Charakterisierung Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennba Von regulären zu Büchi-erkennbaren Sprachen (1 a keine in g, eingehenden Kanten: keine Transitionen (-, -, g Diese Form lässt sich durch Hinzufüren eines frischen Anfangs

#### 16:22

Idee: Füge neuen Startzustand hinzu; dupliziere alle Kanten, die von bisherigen SZen ausgehen.

Damit sind (1) und (2) erreicht. (3) ist korrekt, weil  $\varepsilon \notin L(A_1)$ .

$$W = \{a^n \mid n \text{ ist prim}\}$$

$$\Rightarrow W^{\omega} = \{a^{\omega}\} \qquad (\text{und } W^* = \{a^n \mid n \ge 2\}))$$

Büchi-Aut Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

## Charakt. Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2a)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Von regulären zu Büchi-erkennbaren Sprachen (1) Teil 3: unendliche Wörter Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennba Charakterisierung

16:24

2019-01

Letztlich ist das dieselbe Idee wie bei der Kleene-Abg. der NEA-erkennbaren Sprachen:

─Von regulären zu Büchi-erkennbaren Sprachen (1)

erzeuge Kreis, der ∞ oft durchlaufen werden kann.

Die kann man aber nicht so leicht auf Büchiaut. übertragen, denn sie führt  $\varepsilon$ -Kanten ein, und diese kann man innerhalb von Kreisen nicht so leicht eliminieren wie bei NEAs (ehemals akz. Zustände könnten keine ausgehenden Kanten mehr haben, also gehen erfolgr. Runs verloren ...)

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus

## Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2a)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Idee: konstruiere NBA  $\mathcal{A}_2$ , der

- ullet  $\mathcal{A}_1$  simuliert, bis ein akzeptierender Zustand erreicht ist und
- dann nichtdeterministisch entscheidet,
   ob die Simulation fortgesetzt wird
   oder eine neue Simulation von q<sub>0</sub> aus gestartet wird

Teil 3: unendliche Wörter — Charakterisierung

└Von regulären zu Büchi-erkennbaren Sprachen (1)

Für jiede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^-$  ist Büchi-erkennha Beweik. (Schritt 2a) Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_i\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{e\}$ . Idex: konstruiere WBA.  $A_2$ , der  $B_1$  der institution is howevergender F sortend erreicht ist und

ec: konstruiere NBA A<sub>2</sub>, der v A<sub>3</sub> simuliert, bis ein akzeptierender Zustand erreicht ist und o dann nichtdeterministisch entscheidet, ob die Simulation fortgesetzt wird order nien naus Simulation non n. aus eestartet wird

Von regulären zu Büchi-erkennbaren Sprachen (1)

#### 16:24

2019-01

Letztlich ist das dieselbe Idee wie bei der Kleene-Abg. der NEA-erkennbaren Sprachen:

erzeuge Kreis, der ∞ oft durchlaufen werden kann.

Die kann man aber nicht so leicht auf Büchiaut. übertragen, denn sie führt  $\varepsilon$ -Kanten ein, und diese kann man innerhalb von Kreisen nicht so leicht eliminieren wie bei NEAs (ehemals akz. Zustände könnten keine ausgehenden Kanten mehr haben, also gehen erfolgr. Runs verloren . . . )

Model-Checking

Entscheidungsprobl.

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

16:26 bis  $16:45 \rightarrow 5$ min Pause

Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2b)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Definiere NBA  $\mathcal{A}_2 = (Q_1, \Sigma, \Delta_2, \{q_I\}, \{q_I\})$  mit

└Von regulären zu Büchi-erkennbaren Sprachen (1)

2019-01

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

# Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennba

Charakterisierung

└Von regulären zu Büchi-erkennbaren Sprachen (1)

Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2b)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Definiere NBA  $\mathcal{A}_2 = (Q_1, \Sigma, \Delta_2, \{q_I\}, \{q_I\})$  mit

 $\Delta_2 = \Delta_1 \cup \{(q, a, q_I) \mid (q, a, q_f) \in \Delta_1 \text{ für ein } q_f \in F\}$ 

16:26 bis  $16:45 \rightarrow 5$ min Pause

## Von regulären zu Büchi-erkennbaren Sprachen (1)

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2b)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Definiere NBA  $\mathcal{A}_2 = (Q_1, \Sigma, \Delta_2, \{q_l\}, \{q_l\})$  mit

$$\Delta_2 = \Delta_1 \cup \{(q, a, q_I) \mid (q, a, q_f) \in \Delta_1 \text{ für ein } q_f \in F\}$$

(d. h. alle Kanten, die in  $A_1$  zu einem akz. Zustand führen, können in  $A_2$  zusätzlich zu  $g_I$  führen

- siehe "nichtdeterministisch entscheidet" auf voriger Folie!)

Teil 3: unendliche Wörter Charakterisierung

2019-01

Von regulären zu Büchi-erkennbaren Sprachen (1

Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennba Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_i\}, F)$  mit den genanntei Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ . Definiere NBA  $A_2 = (Q_1, \Sigma, \Delta_2, \{q_i\}, \{q_i\})$  mit alaba nicht daturministisch antschaldet" auf undere Folial

16:26 bis  $16:45 \rightarrow 5$ min Pause

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

2019-01

Teil 3: unendliche Wörter Charakterisierung

Von regulären zu Büchi-erkennbaren Sprachen (1

Für iede reguläre Sprache W C Σ\* gilt: W~ ist Büchi-erkennba Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

> önnen in Ag zusätzlich zu ig führen alaba nicht daturministisch antschaldet" auf undere Folial

16:26 bis  $16:45 \rightarrow 5$ min Pause

#### Lemma 3.7

Für jede reguläre Sprache  $W \subseteq \Sigma^*$  gilt:  $W^{\omega}$  ist Büchi-erkennbar.

Beweis. (Schritt 2b)

Sei also  $A_1 = (Q_1, \Sigma, \Delta_1, \{q_l\}, F)$  mit den genannten Eigenschaften und  $L(A_1) = W \setminus \{\varepsilon\}$ .

Von regulären zu Büchi-erkennbaren Sprachen (1)

Definiere NBA  $\mathcal{A}_2 = (Q_1, \Sigma, \Delta_2, \{q_l\}, \{q_l\})$  mit

$$\Delta_2 = \Delta_1 \cup \{(q, a, q_l) \mid (q, a, q_f) \in \Delta_1 \text{ für ein } q_f \in F\}$$

(d. h. alle Kanten, die in  $A_1$  zu einem akz. Zustand führen, können in  $A_2$  zusätzlich zu  $g_I$  führen

- siehe "nichtdeterministisch entscheidet" auf voriger Folie!)

Noch zu zeigen:  $L_{\omega}(A_2) = L(A_1)^{\omega}$ 

T 3.3 □

Automatentheorie u. i. A. WiSe 2018/19

Teil 3: unendliche Wörter

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Von regulären zu Büchi-erkennbaren Sprachen (2)

### Lemma 3.8

Für jede reguläre Sprache  $W\subseteq \Sigma^*$  und jede Büchi-erkennbare Sprache  $L\subseteq \Sigma^\omega$  gilt:

WL ist Büchi-erkennbar.

Teil 3: unendliche Wörter

Charakterisierung

Charakterisierung

Von regulären zu Büchi-erkennbaren Sprachen (2)

Lemma 28.

Für geld engelen Sprach IV C 2° gelt.

With at Block-industrials Sprache L C 2° gilt.

With at Block-industrials

Von regulären zu Büchi-erkennbaren Sprachen (2)

16:50

Büchi-Aut. Charakt. Entscheidungsprobl. Model-Checking Abschlusseig.

## Von regulären zu Büchi-erkennbaren Sprachen (2)

#### Lemma 3.8

Für jede reguläre Sprache  $W \subset \Sigma^*$ und jede Büchi-erkennbare Sprache  $L \subseteq \Sigma^{\omega}$  gilt:

WL ist Büchi-erkennbar.

#### Beweis:

Wie Abgeschlossenheit der regulären Sprachen unter Konkatenation.

Von regulären zu Büchi-erkennbaren Sprachen (2) Teil 3: unendliche Wörter Charakterisierung Für iede reguläre Sprache  $W \subseteq \Sigma^*$ └Von regulären zu Büchi-erkennbaren Sprachen (2)

16:50

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Satz von Büchi

#### Satz 3.9

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

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

Teil 3: unendliche Wörter Charakterisierung
Satz von Büchi

16:52 bis 17:02

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Satz von Büchi

#### Satz 3.9

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

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

Beweisskizze: (Quiz: Welche der Richtungen  $\Rightarrow$ ,  $\Leftarrow$  ist leichter?)

Teil 3: unendliche Wörter
Charakterisierung
Satz von Büchi

Satz von Büchi 
Satz 30 
Satz von Büchi 
Satz 30 
Satz von Satz 30 
Satz 3

16:52 bis 17:02

Motiv. I

Büchi-Aut.

Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

T 3.4 □

### Satz von Büchi

#### Satz 3.9

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

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

#### Beweisskizze:

" $\Leftarrow$ ": folgt aus Lemmas 3.5, 3.7 und 3.8

" $\Rightarrow$ ": bilden  $V_i,\,W_i$  aus denjenigen Wörtern, die zum jeweils nächsten Vorkommen eines akzeptierenden Zustandes führen

Details siehe Tafel.

019-01-31 \_\_\_\_

Teil 3: unendliche Wörter
Charakterisierung

-Satz von Büchi

16:52 bis 17:02

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

## Satz von Büchi

#### Satz 3.9

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

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

#### Beweisskizze:

"←": folgt aus Lemmas 3.5, 3.7 und 3.8

" $\Rightarrow$ ": bilden  $V_i$ ,  $W_i$  aus denjenigen Wörtern, die zum jeweils nächsten Vorkommen eines akzeptierenden Zustandes führen

Details siehe Tafel. T 3.4

### Konsequenz:

Büchi-erkennbare Sprachen durch  $\omega$ -reguläre Ausdrücke darstellbar:

$$r_1 s_1^{\omega} + \cdots + r_n s_n^{\omega}$$
  $(r_i, s_i \text{ sind reguläre Ausdrücke})$ 

Teil 3: unendliche Wörter Charakterisierung
Satz von Büchi

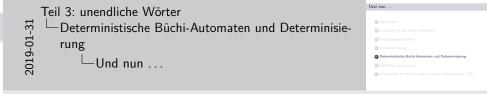


16:52 bis 17:02

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Und nun ...

- Motivatio
- 2 Grundbegriffe und Büchi-Automaten
- Abschlusseigenschafter
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierung
- 6 Entscheidungsprobleme
- Anwendung: Model-Checking in Linearer Temporallogik (LTL)



Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## Ziel dieses Abschnitts

### Wollen zeigen:

- det. und nichtdet. Büchi-Automaten sind **nicht** gleichmächtig d. h.: es gibt  $\omega$ -Sprachen, die von NBAs akzeptiert werden, aber nicht von DBAs
- Komplement-Abgeschlossenheit gilt trotzdem (der Beweis wird aber anspruchsvoll sein)

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Ziel dieses Abschnitts

Wolten zeigen:

• det: und nichtdet: Blichi-Automaten sind nicht gleichmücht;
d. h.: en gilt = Sprachen, die von NIBA akseptiert werden,
aber nicht von DIBAs

• Komplement-Angeschlossenheit gilt trotzdem
(der Breess wird aber ampruchvord sein)

Ziel dieses Abschnitts

17:02

"nicht gleichmächtig": Überraschung!:)

Beachte: hier wieder "genau ein" (Papierkörbe sind wieder einfach, wie bei DEAs)

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

#### Model-Checking

### Ziel dieses Abschnitts

### Wollen zeigen:

- det. und nichtdet. Büchi-Automaten sind nicht gleichmächtig d. h.: es gibt  $\omega$ -Sprachen, die von NBAs akzeptiert werden, aber nicht von DBAs
- Komplement-Abgeschlossenheit gilt trotzdem (der Beweis wird aber anspruchsvoll sein)

#### Definition 3.10

Ein deterministischer Büchi-Automat (DBA) ist ein NBA  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  mit

- |I| = 1
- $|\{q' \mid (q, a, q') \in \Delta\}| = 1$  für alle  $(q, a) \in Q \times \Sigma$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Ziel dieses Abschnitts

#### u det. und nichtdet. Büchi-Automaten sind nicht gleichmächtig d. h.: es gibt ω-Sprachen, die von NBAs akzeptiert werder w Komplement-Abeeschlossenheit eilt trotzdem Ein deterministischer Büchl-Automat (DBA) ist ein NBA $A = (Q, \Sigma, \Delta, I, F)$ mit |I| = 1• $|\{\sigma' \mid (\sigma, a, \sigma') \in \Delta\}| = 1$ für alle $(\sigma, a) \in Q \times \Sigma$

7ial diagae Abechnitte

#### 17:02

"nicht gleichmächtig": Überraschung!:)

Beachte: hier wieder "genau ein" (Papierkörbe sind wieder einfach, wie bei DEAs)

Determinismus

Entscheidungsprobl.

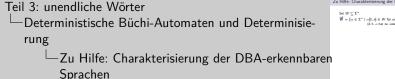
Model-Checking

Sei 
$$W \subseteq \Sigma^*$$
.

$$\overrightarrow{W} = \{ \alpha \in \Sigma^{\omega} \mid \alpha[0, n] \in W \text{ für unendlich viele } n \}$$

$$(d. h. \alpha \text{ hat } \infty \text{ viele Präfixe in } W)$$

T 3.5



17:04

2019-01

Dazu zunächst auch eine Charakt. der DBA-erkennbaren Sprachen, die uns erlauben wird, NBAs und DBAs bezüglich der Mächtigkeit zu trennen.

T3.5 bis 17:14

T3.6 bis 17:24

## Zu Hilfe: Charakterisierung der DBA-erkennbaren Sprachen

Sei 
$$W \subseteq \Sigma^*$$
.

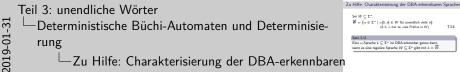
$$\overrightarrow{W} = \{ \alpha \in \Sigma^{\omega} \mid \alpha[0, n] \in W \text{ für unendlich viele } n \}$$

$$(d. h. \alpha \text{ hat } \infty \text{ viele Präfixe in } W)$$

T 3.5

#### Satz 3.11

Eine  $\omega$ -Sprache  $L\subseteq \Sigma^{\omega}$  ist DBA-erkennbar genau dan<u>n</u>, wenn es eine reguläre Sprache  $W \subset \Sigma^*$  gibt mit  $L = \overrightarrow{W}$ .



#### 17:04

Sprachen

Dazu zunächst auch eine Charakt. der DBA-erkennbaren Sprachen, die uns erlauben wird, NBAs und DBAs bezüglich der Mächtigkeit zu trennen.

T3.5 bis 17:14

T3.6 bis 17:24

Sei  $W \subset \Sigma^*$ .

$$\overrightarrow{W} = \{ \alpha \in \Sigma^{\omega} \mid \alpha[0, n] \in W \text{ für unendlich viele } n \}$$

$$(d. h. \alpha \text{ hat } \infty \text{ viele Präfixe in } W)$$

T 3.5

#### Satz 3.11

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

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

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

T 3.6 □

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Zu Hilfe: Charakterisierung der DBA-erkennbaren



17:04

2019-01

Dazu zunächst auch eine Charakt. der DBA-erkennbaren Sprachen, die uns erlauben wird, NBAs und DBAs bezüglich der Mächtigkeit zu trennen.

T3.5 bis 17:14

Sprachen

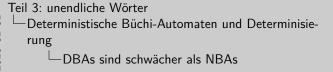
T3.6 bis 17:24

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.



DBAs sind schwächer als NBAs

Es gibt eine Büchi-erkennbare Sprache.

die nicht durch einen DBA erkannt wird

17:24 bis 17:29

Abschlusseig.

Charakt. Determinismus

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -DBAs sind schwächer als NBAs

DBAs sind schwächer als NBAs

Es gibt eine Büchi-erkennbare Sprache.

die nicht durch einen DBA erkannt wird

• Betrachte  $L = \{\alpha \in \{a,b\}^{\omega} \mid \#_{\alpha}(\alpha) \text{ ist endlich} \}$ 

17:24 bis 17:29

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -DBAs sind schwächer als NBAs

DBAs sind schwächer als NBAs Es gibt eine Büchi-erkennbare Sprache. die nicht durch einen DBA erkannt wird • Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_{\alpha}(\alpha) \text{ ist endlich} \}$ • L ist Büchi-erkennbar:  $L = \Sigma^*\{b\}^\omega$ , wende Satz 3.9 an

17:24 bis 17:29

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme, L sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -DBAs sind schwächer als NBAs

DBAs sind schwächer als NBAs

Es gibt eine Büchi-erkennbare Sprache.

die nicht durch einen DBA erkannt wird

• Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$ 

17:24 bis 17:29

2019-01

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme, L sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W
  - $\Rightarrow$  Wegen  $b^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1} \in W$

DBAs sind schwächer als NBAs Teil 3: unendliche Wörter Es gibt eine Büchi-erkennbare Sprache. Deterministische Büchi-Automaten und Determinisiedie nicht durch einen DBA erkannt wird rung • Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$ -DBAs sind schwächer als NBAs ⇒ Wegen b" ∈ L gibt as ain nichtlagen Wort b" ∈ W

17:24 bis 17:29

2019-01

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme. L sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W
  - $\Rightarrow$  Wegen  $b^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1} \in W$ Wegen  $b^{n_1}ab^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1}ab^{n_2} \in W$

DBAs sind schwächer als NBAs Teil 3: unendliche Wörter Es gibt eine Büchi-erkennbare Sprache. Deterministische Büchi-Automaten und Determinisiedie nicht durch einen DBA erkannt wird rung • Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_{\sigma}(\alpha) \text{ ist endlich} \}$ -DBAs sind schwächer als NBAs ⇒ Satz 3.11: L = W für eine reguläre Sprache W ⇒ Wegen b" ∈ L gibt as ain nichtlagen Wort b" ∈ W

Wegen  $b^n ab^n \in L$  gibt as an nichtleares Wort  $b^n ab^n \in M$ 

17:24 bis 17:29

2019-01-

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme, L sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W
  - $\Rightarrow$  Wegen  $b^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1} \in W$ Wegen  $b^{n_1}ab^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1}ab^{n_2} \in W$

DBAs sind schwächer als NBAs Teil 3: unendliche Wörter Es gibt eine Büchi-erkennbare Sprache. Deterministische Büchi-Automaten und Determinisiedie nicht durch einen DBA erkannt wird rung • Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$ -DBAs sind schwächer als NBAs

⇒ Wegen b" ∈ L gibt as an nichtlagen Wort b" ∈ W

17:24 bis 17:29

2019-01-

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

### DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme, L sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W
  - $\Rightarrow$  Wegen  $b^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1} \in W$ Wegen  $b^{n_1}ab^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1}ab^{n_2} \in W$

 $\Rightarrow \alpha := b^{n_1}ab^{n_2}ab^{n_3} \dots \in \overrightarrow{W}$ 

2019-01rung -DBAs sind schwächer als NBAs ⇒ a := b~ab~ab~... ∈ ₩

Deterministische Büchi-Automaten und Determinisie-

17:24 bis 17:29

Teil 3: unendliche Wörter

Am Anfang fragen: Ideen für so eine Sprache?

Es gibt eine Büchi-erkennbare Sprache. die nicht durch einen DBA erkannt wird

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_{\sigma}(\alpha) \text{ ist endlich} \}$
- ⇒ Satz 3.11: L = W für eine reguläre Sprache W ⇒ Wegen b" ∈ L gibt as ain nichtlagen Wort b" ∈ W
  - We sen  $b^{\alpha}ab^{\alpha} \in L$  with as an nichtleans Wort  $b^{\alpha}ab^{\alpha} \in M$

Aotiv.

Büchi-Aut.

Abschlusseig. Ch

Charakt. Determinismus

Entscheidungsprobl.

ngsprobl. Model-Checking

### DBAs sind schwächer als NBAs

#### Satz 3.12

Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

#### Beweis.

- Betrachte  $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich}\}$
- L ist Büchi-erkennbar:  $L = \Sigma^* \{b\}^{\omega}$ , wende Satz 3.9 an
- Annahme, *L* sei DBA-erkennbar.
  - $\Rightarrow$  Satz 3.11:  $L = \overrightarrow{W}$  für eine reguläre Sprache W
  - $\Rightarrow$  Wegen  $b^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1} \in W$ Wegen  $b^{n_1}ab^{\omega} \in L$  gibt es ein nichtleeres Wort  $b^{n_1}ab^{n_2} \in W$

 $\Rightarrow \alpha := b^{n_1}ab^{n_2}ab^{n_3} \dots \in \overrightarrow{W}$ 

Widerspruch:  $\alpha \notin L$ 

··· C // Widerspraem

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

DBAs sind schwächer als NBAs

Es gibt eine Bitchi-rekennbare Sprache, die nicht durch einem DBA erkannt wird.

Brook.

Bitzuche  $L = \{\alpha \in \{a,b\}^n \mid \theta_n(\alpha) \text{ ist endich} \}$ Annahme.  $L = \Sigma^n \{a,b\}^n \mid \theta_n(\alpha) \text{ ist endich} \}$ Annahme.  $L = \Sigma^n \{b\}^n \text{ ender Start 30 an Annahme. L. of DBA-rekennbe. }$ Size 311.1 – Wie eine regelüer Sprache W

Size 311.1 – Wie eine regelüer Sprache W

Womp  $h^2 \in L$  gibt ein nichtkens Wort  $h^2 \in W$ Womp  $h^2 \in L$  gibt ein nichtkens Wort  $h^2 \in W$ 

DBAs sind schwächer als NBAs

17:24 bis 17:29

2019-01

Am Anfang fragen: Ideen für so eine Sprache?

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

- $L = \{\alpha \in \{a, b\}^{\omega} \mid \#_a(\alpha) \text{ ist endlich} \}$ wird von keinem DBA erkannt
- aber  $\overline{L}$  wird von einem DBA erkannt (Ü)

Nebenprodukt des letzten Beweises Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Nebenprodukt des letzten Beweises

Die DRA-erkennharen Smarhen sind nicht unter Kommlemen

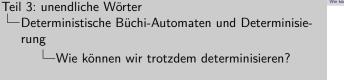
 $u L = \{\alpha \in \{a, b\}^{\omega} \mid \#_{a}(\alpha) \text{ ist endlich} \}$ aber L wird von einem DBA erkannt (0)

17:29 bis 17:30  $\rightarrow$  hoffentlich Punktlandung!

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

9

### Wie können wir trotzdem determinisieren?



#### 8:30

Erinnerung vom letzten Mal: haben DBAs eingeführt und gezeigt, dass sie weniger mächtig sind als NBAs (über versch. Charakterisierungen mittels regulärer Sprachen)

Heute: wollen geänderte Automatenmodelle einführen und zeigen, dass ihre deterministischen Varianten genauso mächtig sind wie NBAs.

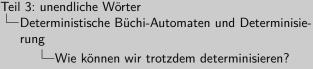
Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

2019-01

### Wie können wir trotzdem determinisieren?

Indem wir das Automatenmodell ändern!

Genauer: ändern die Akzeptanzbedingung



: Konnen wir trotzoem determinis Indem wir das Automatenmodell ändern! Genauer: ändern die Akzeptanzbedingung

#### 8:30

Erinnerung vom letzten Mal: haben DBAs eingeführt und gezeigt, dass sie weniger mächtig sind als NBAs (über versch. Charakterisierungen mittels regulärer Sprachen)

Heute: wollen geänderte Automatenmodelle einführen und zeigen, dass ihre deterministischen Varianten genauso mächtig sind wie NBAs.

v. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Wie können wir trotzdem determinisieren?

#### Indem wir das Automatenmodell ändern!

Genauer: ändern die Akzeptanzbedingung

### Zur Erinnerung

**NBA** ist 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  mit

- ...
- $F \subseteq Q$  (Menge der akz. Zustände)

Erfolgreicher Run:  $r = q_0 q_1 q_2 \dots$  mit  $q_0 \in I$  und  $Inf(r) \cap F \neq \emptyset$ 

Idee: r erfolgreich  $\Leftrightarrow$  ein Zustand aus F kommt  $\infty$  oft in r vor

(Julius Richard Büchi, 1924–1984, Logiker/Mathematiker; Zürich, Lafayette)

2019-01-31 ==

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Wie können wir trotzdem determinisieren?



8:30

Erinnerung vom letzten Mal: haben DBAs eingeführt und gezeigt, dass sie weniger mächtig sind als NBAs (über versch. Charakterisierungen mittels regulärer Sprachen)

Heute: wollen geänderte Automatenmodelle einführen und zeigen, dass ihre deterministischen Varianten genauso mächtig sind wie NBAs.

Model-Checking

(David E. Muller, 1924–2008, Math./Inf.; Illinois)

#### Definition 3.13

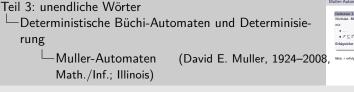
Nichtdet. Muller-Automat (NMA) ist 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ mit

- . . .
- (Kollektion von Endzustandsmengen)

**Erfolgreicher Run**  $r = q_0 q_1 q_2 \dots$  mit  $q_0 \in I$  und  $Inf(r) \in \mathcal{F}$ 

Idee: r erfolgreich  $\Leftrightarrow$  Inf(r) stimmt mit einer Menge aus  $\mathcal{F}$  überein

T 3.7



8:32 bis 8:44

2019-01

### Rabin-Automaten (Michael O. Rabin, \*1931, Inf.; Jerusalem, Princeton, Harvard)

#### Definition 3.14

Nichtdet. Rabin-Automat (NRA) ist 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

- . . . .
- $\mathcal{P} = \{(E_1, F_1), \ldots, (E_n, F_n)\}$  mit  $E_i, F_i \subset Q$ (Menge "akzeptierender Paare")

**Erfolgreicher Run**  $r = q_0 q_1 q_2 \dots$  mit  $q_0 \in I$  und

$$\exists i \in \{1, ..., n\}$$
 mit  $Inf(r) \cap E_i = \emptyset$  und  $Inf(r) \cap F_i \neq \emptyset$ 

**Idee**: r erfolgreich  $\Leftrightarrow$  es gibt Paar  $(E_i, F_i)$ , so dass

- mindestens ein Zustand aus F<sub>i</sub> unendlich oft in r vorkommt &
- alle Zustände aus *E<sub>i</sub>* nur endlich oft in *r* vorkommen T 3.8

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Rabin-Automaten (Michael O. Rabin, \*1931, Inf.; Jerusalem, Princeton,

8:44 bis 8:54

#### Definition 3.15

Nichtdet. Streett-Automat (NSA) ist 5-Tupel  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

Determinismus

- . . . .
- $\mathcal{P} = \{(E_1, F_1), \ldots, (E_n, F_n)\}$  mit  $E_i, F_i \subset Q$ (Menge "fairer Paare")

**Erfolgreicher Run**  $r = q_0 q_1 q_2 \dots$  mit  $q_0 \in I$  und

$$\forall i \in \{1, \dots, n\}$$
: wenn  $Inf(r) \cap F_i \neq \emptyset$ , dann  $Inf(r) \cap E_i \neq \emptyset$ 

Idee: r erfolgreich  $\Leftrightarrow$  für alle Paare  $(E_i, F_i)$  gilt:

- wenn ein Zustand aus  $F_i$  unendlich oft in r vorkommt,
- dann kommt ein Zustand aus  $E_i$  unendlich oft in r vor T 3.9

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisiev P = {(E<sub>1</sub>, F<sub>1</sub>), . . . , (E<sub>n</sub>, F<sub>n</sub>)} mit E<sub>i</sub>, F<sub>i</sub> ⊂ Q rung -Streett-Automaten (Robert S. Streett, ?; Boston, Oakland) dann kommt ein Zustand aus E: unendlich oft in r vor
 T3.5

(Robert S. Streett, 7: Boston, Oakland

8:54 bis 9:06

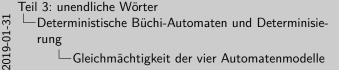
2019-01

Charakt.

## Gleichmächtigkeit der vier Automatenmodelle

Für  $X \in \{Muller, Rabin, Streett\}$  werden analog definiert:

- $L_{\omega}(A)$  für (nichtdeterministische) X-Automaten
- X-erkennbar



Charakt.

### Gleichmächtigkeit der vier Automatenmodelle

Für  $X \in \{Muller, Rabin, Streett\}$  werden analog definiert:

- $L_{\omega}(A)$  für (nichtdeterministische) X-Automaten
- X-erkennbar

#### Satz 3.16

Für jede Sprache  $L \subseteq \Sigma^{\omega}$  sind die folgenden Aussagen äquivalent.

- *L* ist Büchi-erkennbar. (R) L ist Rabin-erkennbar.
- L ist Muller-erkennbar. L ist Streett-erkennbar.

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Gleichmächtigkeit der vier Automatenmodelle

Für iede Sprache  $L \subset \Sigma^{\omega}$  sind die folgenden Aussagen äguivalent (M) L ist Muller-erkennbar. (S) L ist Streett-erkennbar

Gleichmächtigkeit der vier Automatenmodelle

9:06

9

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

### Gleichmächtigkeit der vier Automatenmodelle

Für  $X \in \{Muller, Rabin, Streett\}$  werden analog definiert:

- $L_{\omega}(A)$  für (nichtdeterministische) X-Automaten
- X-erkennbar

#### Satz 3.16

Für jede Sprache  $L \subseteq \Sigma^{\omega}$  sind die folgenden Aussagen äquivalent.

- L ist Büchi-erkennbar. (R) *L* ist Rabin-erkennbar.
- *L* ist Muller-erkennbar. L ist Streett-erkennbar.

Beweis: Konsequenz aus Lemmas 3.17–3.19. T 3.10

Gleichmächtigkeit der vier Automatenmodelle Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Gleichmächtigkeit der vier Automatenmodelle

Für iede Sprache  $L \subset \Sigma^{\omega}$  sind die folgenden Aussagen äguivalent

Beweis: Konsequenz aus Lemmas 3.17-3.19. 1 T3.10

9:06

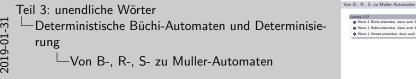
2019-01

Büchi-Aut. Charakt. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

### Von B-, R-, S- zu Muller-Automaten

#### Lemma 3.17

- Wenn L Büchi-erkennbar, dann auch Muller-erkennbar.
- Wenn L Rabin-erkennbar, dann auch Muller-erkennbar.
- Wenn L Streett-erkennbar, dann auch Muller-erkennbar.



A Wenn L Büchi-erkennbar, dann auch Muller-erkennba Wenn L Rabin-erkennbar, dann auch Muller-erkennbar Wenn L Streett-erkennbar, dann auch Muller-erkennbar

9:08

Idee: Kodiere F in  $\mathcal{F}$ .

Die Q' sind alle erlaubten Unendlichkeitsmengen Inf(r).

Abschlusseig.

Von B-, R-, S- zu Muller-Automaten

Charakt.

Determinismus

Model-Checking

#### Von B-, R-, S- zu Muller-Automaten Deterministische Büchi-Automaten und Determinisie-A Wenn L Büchi-erkennbar, dann auch Muller-erkennba Wenn L Rabin-erkennbar, dann auch Muller-erkennbar Wenn L Streett-erkennbar, dann auch Muller-erkennbar (1) Sei A = (Q, Σ, Δ, I, F) NBA. Konstruiare NMA $A' = (O \Sigma \land I F)$ mit └Von B-, R-, S- zu Muller-Automaten $F = \{O' \subset O \mid O' \cap F \neq \emptyset\}$ Leicht zu sehen: $L_{\omega}(A') = L_{\omega}(A)$ .

#### Lemma 3.17

- Wenn L Büchi-erkennbar, dann auch Muller-erkennbar.
- Wenn L Rabin-erkennbar, dann auch Muller-erkennbar.
- Wenn L Streett-erkennbar, dann auch Muller-erkennbar.

### Beweis.

(1) Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NMA  $A' = (Q, \Sigma, \Delta, I, F)$  mit

$$\mathcal{F} = \{ Q' \subseteq Q \mid Q' \cap F \neq \emptyset \}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

#### 9:08

2019-01

Idee: Kodiere F in  $\mathcal{F}$ .

Die Q' sind alle erlaubten Unendlichkeitsmengen Inf(r).

Abschlusseig.

Charakt. Determinismus

Entscheidungsprobl.

### Von B-, R-, S- zu Muller-Automaten

#### Lemma 3.17

- Wenn L Büchi-erkennbar, dann auch Muller-erkennbar.
- Wenn L Rabin-erkennbar, dann auch Muller-erkennbar.
- Wenn L Streett-erkennbar, dann auch Muller-erkennbar.

#### Beweis.

(2) Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P})$  NRA.

Konstruiere NMA  $A' = (Q, \Sigma, \Delta, I, F)$  mit

$$\mathcal{F} = \{ Q' \subseteq Q \mid \exists i \leq n : Q' \cap E_i = \emptyset \text{ und } Q' \cap F_i \neq \emptyset \}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung

└Von B-, R-, S- zu Muller-Automaten

A Wenn L Büchi-erkennbar, dann auch Muller-erkennba Wenn L Rabin-erkennbar, dann auch Muller-erkennbar ♦ Wenn L Streett-erkennbar, dann auch Muller-erkennbar (2) Sei  $A = (Q, \Sigma, \Delta, I, P)$  NRA. Konstruiere NMA  $A' = (Q, \Sigma, \Delta, I, F)$  mit  $F = \{O' \subseteq O \mid \exists i \leq n : O' \cap E = \emptyset \text{ und } O' \cap F \neq \emptyset\}$ 

Von B-, R-, S- zu Muller-Automaten

Leicht zu sehen:  $L_{\omega}(A') = L_{\omega}(A)$ .

#### 9:11

2019-01

Dieselbe Idee: Kodiere  $\mathcal{P}$  in  $\mathcal{F}$ .

... und natürlich auch bei Streett-Automaten ...

Notiv. Büchi-Aut.

Abschlusseig.

Charakt.

#### Lemma 3.17

- Wenn L Büchi-erkennbar, dann auch Muller-erkennbar.
- 2 Wenn L Rabin-erkennbar, dann auch Muller-erkennbar.
- Wenn L Streett-erkennbar, dann auch Muller-erkennbar.

#### Beweis.

(2) Sei  $A = (Q, \Sigma, \Delta, I, P)$  NRA.

Konstruiere NMA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{F})$  mit

$$\mathcal{F} = \{ Q' \subset Q \mid \exists i < n : Q' \cap E_i = \emptyset \text{ und } Q' \cap F_i \neq \emptyset \}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

(3) Analog.

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Von B-, R-, S- zu Muller-Automaten

Lemma 3.17

• When E Bick-résembar, dam aush Miller-sésembar.
• When E Balos-résembar, dam aush Miller-sésembar.
• When E Balos-résembar, dam aush Miller-sésembar.
• Damb.

(1) Set  $A = \{0, \Sigma, \Delta, L, F\}$  100A.

\*\*Fontationism MiA,  $A' = \{0, \Sigma, \Delta, L, F\}$  neit  $F = \{0^{r} \subseteq Q, F\} \le c: Q' \cap E = \emptyset \text{ and } Q' \cap F_i \neq \emptyset\}.$ Linkt zu saller.  $L_i(A') = L_i(A)$ .
(3) Avalug.

Von B-, R-, S- zu Muller-Automaten

#### 9:11

2019-01

Dieselbe Idee: Kodiere  $\mathcal{P}$  in  $\mathcal{F}$ .

... und natürlich auch bei Streett-Automaten ...

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

Von Büchi- zu R- und S-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung └─Von Büchi- zu R- und S-Automaten

Streett-erkennbar.

9:13 bis 9:15, 5 min Pause.

Determinismus

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

#### Beweis.

(1) Sei  $A = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NRA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

$$\mathcal{P} =$$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Büchi- zu R- und S-Automaten

Nenn L Büchi-erkennbar, dann auch Streett-erkennbar (1) Sei A = (Q, Σ, Δ, I, F) NBA Konstruiere NRA  $A' = (Q, \Sigma, \Delta, I, P)$  mit

Von Büchi- zu R- und S-Automaten

9:13 bis 9:15, 5 min Pause.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

ca Rabin,erkennbar und Streett-erkennbar

(1) Sei A = (Q, Σ, Δ, I, F) NBA.

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

#### Beweis.

(1) Sei  $A = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NRA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

$$\mathcal{P} = \{(\emptyset, F)\}.$$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Büchi- zu R- und S-Automaten

9:13 bis 9:15, 5 min Pause.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

2019-01

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

#### Beweis.

(1) Sei  $A = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NRA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

$$\mathcal{P} = \{(\emptyset, F)\}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Büchi- zu R- und S-Automaten

Von Riichi- zu R- und S-Automaten

Wenn L Büchi-erkennbar, dann auch ca Rabin,erkennbar und

(1) Sei A = (Q, Σ, Δ, I, F) NBA Konstruiere NRA  $A' = (Q, \Sigma, \Delta, I, P)$  mi

Leicht zu sehen:  $L_{\omega}(A') = L_{\omega}(A)$ .

A Streett-erkennhar

9:13 bis 9:15, 5 min Pause.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

Wenn L Büchi-erkennbar, dann auch

ca Rabin,erkennbar und A Streett-erkennhar

(2) Analog, aber mit P =

(1) Sei A = (Q, Σ, Δ, I, F) NBA Konstruiere NRA  $A' = (Q, \Sigma, \Delta, I, P)$  mit Leicht zu sehen:  $L_{\omega}(A') = L_{\omega}(A)$ .

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

#### Beweis.

(1) Sei  $A = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NRA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

$$\mathcal{P} = \{(\emptyset, F)\}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

(2) Analog, aber mit  $\mathcal{P} =$ 

9:13 bis 9:15, 5 min Pause.

2019-01

rung

Teil 3: unendliche Wörter

Jeweils vorm Aufdecken von  $\mathcal{P}$ : wer weiß es?

Deterministische Büchi-Automaten und Determinisie-

-Von Büchi- zu R- und S-Automaten

Abschlusseig.

Charakt.

2019-01

### Von Büchi- zu R- und S-Automaten

#### Lemma 3.18

Wenn L Büchi-erkennbar, dann auch

- Rabin-erkennbar und
- Streett-erkennbar.

#### Beweis.

(1) Sei  $A = (Q, \Sigma, \Delta, I, F)$  NBA.

Konstruiere NRA  $\mathcal{A}' = (Q, \Sigma, \Delta, I, \mathcal{P})$  mit

$$\mathcal{P} = \{(\emptyset, F)\}.$$

Leicht zu sehen:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

(2) Analog, aber mit  $\mathcal{P} = \{(F, Q)\}.$ 

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Büchi- zu R- und S-Automaten

Von Riichi- zu R- und S-Automaten Wenn L Büchi-erkennbar, dann auch ca Rabin,erkennbar und A Streett-erkennhar (1) Sei A = (Q, Σ, Δ, I, F) NBA Konstruiere NRA  $A' = (Q, \Sigma, \Delta, I, P)$  mit

Leicht zu sehen:  $L_{\omega}(A') = L_{\omega}(A)$ . (2) Analog, aber mit  $P = \{(F, Q)\}$ 

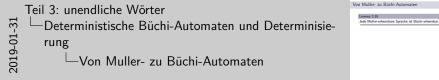
9:13 bis 9:15, 5 min Pause.

Büchi-Aut. Abschlusseig. Charakt. **Determinismus** Entscheidungsprobl. *Model-Checking* 

### Von Muller- zu Büchi-Automaten

#### Lemma 3.19

Jede Muller-erkennbare Sprache ist Büchi-erkennbar.



Charakt.

Determinismus

Entscheidungsprobl.

### Von Muller- zu Büchi-Automaten

#### Lemma 3.19

Jede Muller-erkennbare Sprache ist Büchi-erkennbar.

### Beweis.

• Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$  ein Muller-Automat

Von Muller- zu Büchi-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

Jede Muller-erkennbare Sprache ist Büchi-erkennbar

tiv. Büchi-Aut. Abschlusseig.

Charakt.

### Von Muller- zu Büchi-Automaten

#### Lemma 3.19

Jede Muller-erkennbare Sprache ist Büchi-erkennbar.

#### Beweis.

- Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$  ein Muller-Automat
- Dann ist  $L_{\omega}(A) = \bigcup_{F \in \mathcal{F}} L_{\omega}((Q, \Sigma, \Delta, I, \{F\}))$

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Von Muller- zu Büchi-Automaten

Von Muller- zu Büchi-Automaten

Jede Muller-erkennbare Sprache ist Büchi-erkennbar

Abschlusseig.

# Von Muller- zu Büchi-Automaten

#### Lemma 3.19

Jede Muller-erkennbare Sprache ist Büchi-erkennbar.

#### Beweis.

- Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$  ein Muller-Automat
- Dann ist  $L_{\omega}(A) = \bigcup_{F \in \mathcal{F}} L_{\omega}((Q, \Sigma, \Delta, I, \{F\}))$
- Wegen ∪-Abgeschlossenheit genügt es zu zeigen, dass  $L_{\omega}((Q, \Sigma, \Delta, I, \{F\}))$  Büchi-erkennbar ist

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

Jede Muller-erkennbare Sprache ist Büchi-erkennbar L.((Q.Σ.Δ.I.{F})) Büchi-erkennbar ist

Von Muller- zu Büchi-Automaten

9:20

2019-01

### Von Muller- zu Büchi-Automaten

#### Lemma 3.19

Jede Muller-erkennbare Sprache ist Büchi-erkennbar.

#### Beweis.

- Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$  ein Muller-Automat
- Dann ist  $L_{\omega}(A) = \bigcup_{F \in \mathcal{F}} L_{\omega}((Q, \Sigma, \Delta, I, \{F\}))$
- Wegen ∪-Abgeschlossenheit genügt es zu zeigen, dass  $L_{\omega}((Q, \Sigma, \Delta, I, \{F\}))$  Büchi-erkennbar ist
- Konstruiere Büchi-Automaten  $\mathcal{A}' = (Q', \Sigma, \Delta', I, F')$ , der
  - A simuliert
  - einen Zeitpunkt rät, ab dem nur noch Zustände aus F vorkommen
  - ab dort sicherstellt, dass *alle* diese unendlich oft vorkommen

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

Jede Muller-erkennbare Sprache ist Büchi-erkennbar L.((Q. E. A. I. (F3)) Büchi-erkennbar ist Konstruiere Büchi-Automaten A' = (Q', Σ, Δ', I, F'), der

· ab dort sicherstellt, dass alle diese unendlich oft vorkomme

• einen Zeitpunkt rät

Von Muller- zu Büchi-Automaten

9:20

2019-01

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Von Muller- zu Büchi-Automaten

Sei also 
$$\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$$
 (Muller-Automat)  
Konstruieren NBA  $\mathcal{A}' = (Q', \Sigma, \Delta', I', F')$  mit

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Von Muller- zu Büchi-Automaten

9:23

2019-01

**TODO** Vorschlag (Tryggve, WiSe 18/19):

Man kann auch die Zustände aus F ordnen  $(f_1,\ldots,f_n)$  und dann analog zur Produktkonstruktion n Modi verwenden, d. h. Modus i bedeutet "erwarte  $f_i$ ". Dann sind die Zustände der Phase 2 nur Paare aus EZ und Modus, und  $\Delta'$  hat vielleicht eine angenehmere Notation.  $\leadsto$  Ausprobieren!

Von Muller- zu Büchi-Automaten

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Von Muller- zu Büchi-Automaten

Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = \underbrace{Q}_{\text{Phase 1}} \cup \underbrace{\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}}_{\text{Phase 2}}$$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung

-Von Muller- zu Büchi-Automaten

Von Muller- zu Büchi-Automaten

9:23

**TODO** Vorschlag (Tryggve, WiSe 18/19):

Man kann auch die Zustände aus F ordnen  $(f_1, \ldots, f_n)$  und dann analog zur Produktkonstruktion *n* Modi verwenden, d. h. Modus *i* bedeutet "erwarte fi". Dann sind die Zustände der Phase 2 nur Paare aus EZ und Modus, und  $\Delta'$  hat vielleicht eine angenehmere Notation.  $\rightsquigarrow$  Ausprobieren!

Abschlusseig.

Charakt. Determinismus

Entscheidungsprobl.

Model-Checking

### Von Muller- zu Büchi-Automaten

Sei also  $\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = \underbrace{Q}_{\text{Phase 1}} \cup \underbrace{\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}}_{\text{Phase 2}}$$

Ph. 1:  $\mathcal{A}'$  simuliert  $\mathcal{A}$ , bis  $\mathcal{A}$  irgendwann in einem  $g_f \in \mathcal{F}$  ist

Ph. 2:  $\mathcal{A}'$  will nur noch Zustände  $\in F$  sehen und jeden  $\infty$  oft

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

Ph. 1: A' simuliert A, bis A invendwarn in einem  $a_i \in F$  ist Ph 2: 4' will not noch Zostände C F sehen und infen och di

Von Muller- zu Büchi-Automaten

9:23

2019-01

**TODO** Vorschlag (Tryggve, WiSe 18/19):

Man kann auch die Zustände aus F ordnen  $(f_1, \ldots, f_n)$  und dann analog zur Produktkonstruktion *n* Modi verwenden, d. h. Modus *i* bedeutet "erwarte fi". Dann sind die Zustände der Phase 2 nur Paare aus EZ und Modus, und  $\Delta'$  hat vielleicht eine angenehmere Notation.  $\rightsquigarrow$  Ausprobieren!

### Von Muller- zu Büchi-Automaten

Sei also  $\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = Q$$
  $\cup$   $\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}$  Phase 2

Ph. 1:  $\mathcal{A}'$  simuliert  $\mathcal{A}$ , bis  $\mathcal{A}$  irgendwann in einem  $g_f \in \mathcal{F}$  ist

Ph. 2:  $\mathcal{A}'$  will nur noch Zustände  $\in F$  sehen und jeden  $\infty$  oft

- $\mathcal{A}'$  we chselt in  $\langle q_f, S \rangle$  mit  $S = \{q_f\}$
- S enthält die seit dem letzten Zurücksetzen besuchten  $g \in F$
- Wenn S = F, wird S auf  $\emptyset$  "zurückgesetzt"
- akz. Zustände: ein  $\langle q_f, F \rangle$  muss  $\infty$  oft gesehen werden

Von Muller- zu Büchi-Automaten Teil 3: unendliche Wörter Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Deterministische Büchi-Automaten und Determinisie- $Q' = Q \cup \{(q_f, S) \mid q_f \in F, S \subseteq F\}$ Ph. 1: A' simuliert A, bis A invendwarn in einem  $a_i \in F$  ist rung Ph.2: A' will nur noch Zustände ∈ F sehen und jeden ∞ oft • A' wechselt in  $(q_V, S)$  mit  $S = \{q_V\}$ S enthält die seit dem letzten Zurücksetzen besuchten g ∈ Wenn S = F, wird S auf Ø "zunückensetzt" -Von Muller- zu Büchi-Automaten akz. Zustände: ein (av. F) muss oo oft gesehen werden

9:23

2019-01

**TODO** Vorschlag (Tryggve, WiSe 18/19):

Man kann auch die Zustände aus F ordnen  $(f_1, \ldots, f_n)$  und dann analog zur Produktkonstruktion *n* Modi verwenden, d. h. Modus *i* bedeutet "erwarte fi". Dann sind die Zustände der Phase 2 nur Paare aus EZ und Modus, und  $\Delta'$  hat vielleicht eine angenehmere Notation.  $\rightsquigarrow$  Ausprobieren!

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Von Muller- zu Büchi-Automaten

Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat)

Charakt.

Konstruieren NBA  $\mathcal{A}' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = \underbrace{Q}_{\text{Phase 1}} \cup \underbrace{\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}}_{\text{Phase 2}}$$

• 
$$\Delta' = \Delta$$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Von Muller- zu Büchi-Automaten

Von Muller- zu Büchi-Automaten

Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $\mathcal{A}' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = Q$$
  $\cup$   $\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}$ 

• 
$$\Delta' = \Delta$$
  
 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$ 

Von Muller- zu Büchi-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Von Muller- zu Büchi-Automaten

 $\cup \{(q, a, (qr, \{qr\})) \mid (q, a, qr) \in \Delta, qr \in F\}$ 

Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat)

Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = \underbrace{Q}_{\text{Phase 1}} \cup \underbrace{\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}}_{\text{Phase 2}}$$

• 
$$\Delta' = \Delta$$
  
 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$   
 $\cup \{(\langle q, S \rangle, a, \langle q', S \cup \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F, S \neq F\}$ 

Von Muller- zu Büchi-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

 $\cup \{((a.S), a, (a'.S \cup \{a'\})) \mid (a.a.a') \in \Delta, a.a' \in F, S \neq F\}$ 

 $\bullet \Delta' = \Delta$ 

Phase 1

Sei also  $A = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

 $\bullet \ Q' = Q \cup \{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}$ 

 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$ 

 $\cup \{(\langle q, S \rangle, a, \langle q', S \cup \{q'\} \rangle) \mid (q, a, q') \in \Delta, \ q, q' \in F, \ S \neq F\}$ 

 $\cup \{(\langle q, F \rangle, a, \langle q', \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F\}$ 

# Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Von Muller- zu Büchi-Automaten

-Von Muller- zu Büchi-Automaten

Sei also  $\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat)

Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = \underbrace{Q}_{\text{Phase 1}} \cup \underbrace{\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}}_{\text{Phase 2}}$$

• 
$$\Delta' = \Delta$$
  
 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$   
 $\cup \{(\langle q, S \rangle, a, \langle q', S \cup \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F, S \neq F\}$   
 $\cup \{(\langle q, F \rangle, a, \langle q', \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F\}$   
•  $I' = I$ 

Von Muller- zu Büchi-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Von Muller- zu Büchi-Automaten

 $\cup \{((q,S), a, (q',S \cup \{q'\})) \mid (q,a,q') \in \Delta, q,q' \in F, S \neq F\}$ 

# Von Muller- zu Büchi-Automaten

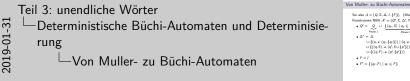
Sei also  $\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat) Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

• 
$$Q' = Q$$
  $\cup$   $\{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}$ 

• 
$$\Delta' = \Delta$$
  
 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$   
 $\cup \{(\langle q, S \rangle, a, \langle q', S \cup \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F, S \neq F\}$   
 $\cup \{(\langle q, F \rangle, a, \langle q', \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F\}$ 

• 
$$I' = I$$

• 
$$F' = \{\langle q_f, F \rangle \mid q_f \in F\}$$



9:26 bis spätestens 9:56

Phase 1

•  $F' = \{\langle q_f, F \rangle \mid q_f \in F\}$ Dann gilt:  $L_{\omega}(\mathcal{A}') = L_{\omega}(\mathcal{A})$ .

 $\bullet \wedge' = \wedge$ 

• I' = I

 $\cup \{(q, a, \langle q_f, \{q_f\} \rangle) \mid (q, a, q_f) \in \Delta, q_f \in F\}$ 

 $\cup \{(\langle q, S \rangle, a, \langle q', S \cup \{q'\} \rangle) \mid (q, a, q') \in \Delta, \ q, q' \in F, \ S \neq F\}$ 

T 3.11

 $\bigcup \{(\langle q, F \rangle, a, \langle q', \{q'\} \rangle) \mid (q, a, q') \in \Delta, q, q' \in F\}$ 

Sei also  $\mathcal{A} = (Q, \Sigma, \Delta, I, \{F\})$  (Muller-Automat)

 $\bullet \ Q' = Q \cup \{\langle q_f, S \rangle \mid q_f \in F, S \subseteq F\}$ 

Konstruieren NBA  $A' = (Q', \Sigma, \Delta', I', F')$  mit

–Von Muller- zu Büchi-Automaten

 $\cup \{((q,S), a, (q',S \cup \{q'\})) \mid (q,a,q') \in \Delta, q,q' \in F, S \neq F\}$  $\mathbf{u} F' = \{(ar, F) \mid ar \in F\}$ 

Von Muller- zu Rüchi-Automaten

9:26 bis spätestens 9:56

lotiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Abschlusseigenschaften

### Direkte Konsequenz aus

- Satz 3.4 (Abschlusseigenschaften der Büchi-erkennbaren Spr.)
- und Satz 3.16 (Gleichmächtigkeit der Automatenmodelle):

# Folgerung 3.20

Die Menge der

- Muller-erkennbaren Sprachen,
- Rabin-erkennbaren Sprachen,
- Streett-erkennbaren Sprachen

ist abgeschlossen unter den Operationen  $\cup$  und  $\cap$ .

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Abschlusseigenschaften

Die Menge der

Muller-erkennbaven Sprachen,

Rabin-erkennbaven Sprachen,

Streett-erkennbaven Sprachen
ist abgeschlossen unter den Operationen U und n.

und Satz 3.16 (Gleichmächtigkeit der Automatenmodelle)

Abschlusseigenschaften

#### 9:56

2019-01

Tief durchatmen; wir sind so gut wie fertig für heute. :)

ptiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Abschlusseigenschaften

#### Direkte Konsequenz aus

- Satz 3.4 (Abschlusseigenschaften der Büchi-erkennbaren Spr.)
- und Satz 3.16 (Gleichmächtigkeit der Automatenmodelle):

# Folgerung 3.20

Die Menge der

- Muller-erkennbaren Sprachen,
- Rabin-erkennbaren Sprachen,
- Streett-erkennbaren Sprachen

ist abgeschlossen unter den Operationen  $\cup$  und  $\cap$ .

### Zu Komplement-Abgeschlossenheit kommen wir jetzt.

Benötigen zunächst deterministische Varianten von Muller-, Rabin-, Streett-Automaten.

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Abschlusseigenschaften

sen San 14 (Abschlusseigenschaften Gesche Gesche des Büch erkenbaren

von San 15 (Abschlusseigenschaften Gesche Gesche

Benötigen zunächst deterministische Varianten von Muller-, Rabir

9:56

2019-01

Tief durchatmen; wir sind so gut wie fertig für heute. :)

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Deterministische Varianten

Deterministische Varianten sind analog zu NBA definiert:

Ein Muller-, Rabin- oder Streett-Automat  $\mathcal{A} = (Q, \Sigma, \Delta, I, Acc)$ ist deterministisch, wenn gilt:

- |I| = 1
- $|\{q' \mid (q, a, q') \in \Delta\}| = 1$  für alle  $(q, a) \in Q \times \Sigma$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Deterministische Varianten

Deterministische Varianten Fin Muller, Rabin, order Structt, Δυτοπατ. 4 = (Ο Σ Λ Ι Δος)  $\mathbf{u} \mid \{q' \mid (q, a, q') \in \Delta\} \mid = 1 \text{ für alle } (q, a) \in Q \times \Sigma$ 

#### 9:58 bis $10:00 \rightarrow$ Punktlandung?

Wenn Zeit, dann was zum Ablauf Prüfungen sagen.

Satz 3.21 folgt nicht unmittelbar aus den bisherigen Resultaten für NxAs. Er wird stückweise in Meghyns Skript bewiesen; dort sind Muller-, Rabin- und Streett-Automaten immer deterministisch.

0

### Deterministische Varianten

Deterministische Varianten sind analog zu NBA definiert:

Ein Muller-, Rabin- oder Streett-Automat  $\mathcal{A} = (Q, \Sigma, \Delta, I, Acc)$ ist deterministisch, wenn gilt:

- |I| = 1
- $|\{q' \mid (q, a, q') \in \Delta\}| = 1$  für alle  $(q, a) \in Q \times \Sigma$

Zu Satz 3.16 analoge Aussage:

#### Satz 3.21

Für jede Sprache  $L \subseteq \Sigma^{\omega}$  sind die folgenden Aussagen äquivalent.

- L ist von einem deterministischen Muller-Autom. erkennbar.
- L ist von einem deterministischen Rabin-Autom. erkennbar.
- L ist von einem deterministischen Streett-Autom. erkennbar.

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Deterministische Varianten

Ein Muller-, Rabin- oder Streett-Automat  $A = (Q, \Sigma, \Delta, I, Acc)$  $\mathbf{u} \mid \{q' \mid (q, a, q') \in \Delta\} \mid = 1 \text{ für alle } (q, a) \in Q \times \Sigma$ Zu Satz 3.16 analoge Aussage:

Für jede Sprache  $L\subseteq \Sigma^\omega$  sind die folgenden Aussagen liquivalen

(M) L ist von einem deterministischen Muller-Autom, erkennbar (R) L ist von einem deterministischen Rabin-Autom. erkennbar.

#### 9:58 bis $10:00 \rightarrow$ Punktlandung?

Wenn Zeit, dann was zum Ablauf Prüfungen sagen.

Satz 3.21 folgt nicht unmittelbar aus den bisherigen Resultaten für NxAs. Er wird stückweise in Meghyns Skript bewiesen; dort sind Muller-, Rabin- und Streett-Automaten immer deterministisch.

Abschlusseig.

Determinismus

Deterministische Varianten sind analog zu NBA definiert:

Ein Muller-, Rabin- oder Streett-Automat  $\mathcal{A} = (Q, \Sigma, \Delta, I, Acc)$ ist deterministisch, wenn gilt:

- |I| = 1
- $|\{q' \mid (q, a, q') \in \Delta\}| = 1$  für alle  $(q, a) \in Q \times \Sigma$

Zu Satz 3.16 analoge Aussage:

#### Satz 3.21

Für jede Sprache  $L \subseteq \Sigma^{\omega}$  sind die folgenden Aussagen äquivalent.

- L ist von einem deterministischen Muller-Autom. erkennbar.
- L ist von einem deterministischen Rabin-Autom. erkennbar.
- L ist von einem deterministischen Streett-Autom. erkennbar.

Ohne Beweis (ähnlich wie Lemmas 3.17–3.19).

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Deterministische Varianten

Ein Muller-, Rabin- oder Streett-Automat  $A = (Q, \Sigma, \Delta, I, Acc)$  $\mathbf{u} \mid \{q' \mid (q, a, q') \in \Delta\} \mid = 1 \text{ für alle } (q, a) \in Q \times \Sigma$ Zu Satz 3.16 analoge Aussage:

Für jede Sprache  $L\subseteq \Sigma^\omega$  sind die folgenden Aussagen äquivalen (M) L ist von einem deterministischen Muller-Autom, erkennbar

#### 9:58 bis $10:00 \rightarrow$ Punktlandung?

Wenn Zeit, dann was zum Ablauf Prüfungen sagen.

Satz 3.21 folgt nicht unmittelbar aus den bisherigen Resultaten für NxAs. Er wird stückweise in Meghyns Skript bewiesen; dort sind Muller-, Rabin- und Streett-Automaten immer deterministisch.

0

# Überblick der Automatenmodelle

### Büchi-Automat (NBA):

- $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  mit  $F \subset Q$
- Erfolgreicher Run r:  $Inf(r) \cap F \neq \emptyset$

#### Muller-Automat (NMA):

- $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F}) \text{ mit } \mathcal{F} \subset 2^Q$
- Erfolgreicher Run r: Inf $(r) \in \mathcal{F}$

### Rabin-Automat (NRA):

- $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P}) \text{ mit } \mathcal{P} \subset 2^Q \times 2^Q$
- Erfolg:  $\exists (E,F) \in \mathcal{P} : \mathsf{Inf}(r) \cap F \neq \emptyset \text{ und } \mathsf{Inf}(r) \cap E = \emptyset$

### Streett-Automat (NSA):

- $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{P}) \text{ mit } \mathcal{P} \subset 2^Q \times 2^Q$
- Erfolg:  $\forall (E, F) \in \mathcal{P} : \operatorname{Inf}(r) \cap F \neq \emptyset$  impliziert  $\operatorname{Inf}(r) \cap E \neq \emptyset$

Teil 3: unendliche Wörter Biichi, Automat (NRA):  $A = (Q, \Sigma, \Delta, I, F)$  mit F ⊆ QDeterministische Büchi-Automaten und Determinisie-■ Erfolgreicher Run r: Inf $(r) \cap F \neq \emptyset$ Muller-Automat (NMA):  $A = (Q, \Sigma, \Delta, I, F)$  mit  $F \subseteq 2^Q$ rung w Erfolgreicher Run r:  $Inf(r) \in \mathcal{F}$ Rabin-Automat (NRA):  $u A = (Q, \Sigma, \Delta, I, P) \text{ mit } P \subseteq 2^Q \times 2^Q$ Überblick der Automatenmodelle • Erfolg:  $\exists (E,F) \in \mathcal{P} : Inf(r) \cap F \neq \emptyset \text{ und } Inf(r) \cap E = \emptyset$ Streett-Automat (NSA) •  $A = (Q, \Sigma, \Delta, I, P)$  mit  $P \subseteq 2^Q \times 2^Q$ • Erfolg:  $\forall (E,F) \in P : Inf(r) \cap F \neq \emptyset$  impliziert  $Inf(r) \cap E \neq \emptyset$ 

Überblick der Automatenmodelle

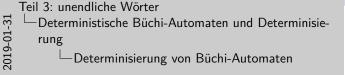
16:00

2019-01-

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.



Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache,

16:02

Determinismus

Entscheidungsprobl.

Model-Checking

# Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

# Ziel

Prozedur zur Umwandlung eines gegebenen NBA in einen äguivalenten deterministischen Rabin-Automaten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Determinisierung von Büchi-Automaten

Determinisierung von Büchi-Automaten Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache Prozedur zur Umwandlung eines gegebenen NBA

16:02

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

# Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

# Ziel

Prozedur zur Umwandlung eines gegebenen NBA in einen äguivalenten deterministischen Rabin-Automaten

→ wegen Satz 3.21 erhält man daraus auch äquivalente deterministische Muller-/Streett-Automaten

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Determinisierung von Büchi-Automaten

Determinisierung von Büchi-Automaten Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache Prozedur zur Umwandlung eines gegebenen NBA

16:02

# Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

# Ziel

Prozedur zur Umwandlung eines gegebenen NBA in einen äguivalenten deterministischen Rabin-Automaten

- → wegen Satz 3.21 erhält man daraus auch äquivalente deterministische Muller-/Streett-Automaten
  - Resultat geht auf McNaughton zurück (1965 von Robert McNaughton, Philosoph/Inform., Harvard, Rensselaer)

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Determinisierung von Büchi-Automaten

1965 von Robert McNaughton, Philosoph/Inform., Harvard, Renssel

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache

Determinisierung von Büchi-Automaten

16:02

# Determinisierung von Büchi-Automaten

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache, die nicht durch einen DBA erkannt wird.

# Ziel

Prozedur zur Umwandlung eines gegebenen NBA in einen äguivalenten deterministischen Rabin-Automaten

- → wegen Satz 3.21 erhält man daraus auch äquivalente deterministische Muller-/Streett-Automaten
  - Resultat geht auf McNaughton zurück (1965 von Robert McNaughton, Philosoph/Inform., Harvard, Rensselaer)
- Wir verwenden intuitiveren Beweis von Safra (1988 von Shmuel Safra, Informatiker, Tel Aviv)

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Determinisierung von Büchi-Automaten

Wir verwenden intuitiveren Beweis von Safra

Erinnerung an Satz 3.12: Es gibt eine Büchi-erkennbare Sprache

Determinisierung von Büchi-Automaten

Prozedur zur Umwandlung eines gegebenen NBA

16:02

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Potenzmengenkonstruktion versagt

### Zwei naheliegende Versuche:

■ NBA 

→ DBA mittels Potenzmengenkonstruktion (PMK) muss wegen Satz 3.12 fehlschlagen – Bsp. siehe Tafel

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Potenzmengenkonstruktion versagt

Potenzmengenkonstruktion versagt

16:03

T3.12 bis 16:10

T3.13 bis 16:17

insg. bis 16:19

Büchi-Aut. Charakt. Determinismus Model-Checking Abschlusseig. Entscheidungsprobl.

# Potenzmengenkonstruktion versagt

### Zwei naheliegende Versuche:

- NBA 

  → DBA mittels Potenzmengenkonstruktion (PMK) muss wegen Satz 3.12 fehlschlagen – Bsp. siehe Tafel
- NBA → determ. Muller-(Rabin-/Streett-)Automat via PMK schlägt auch fehl – mit demselben Gegenbeispiel T 3.13

Potenzmengenkonstruktion versagt Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Potenzmengenkonstruktion versagt

Zwei naholionondo Vorsuche muss wegen Satz 3.12 fehlschlagen - Bsp. siehe Tafel T 3.12 NBA ~+ determ. Muller-(Rabin-/Streett-)Automat via PMK

16:03

T3.12 bis 16:10

T3.13 bis 16:17

insq. bis 16:19

Büchi-Aut. Abschlusseig. Determinismus Entscheidungsprobl.

# Potenzmengenkonstruktion versagt

### Zwei naheliegende Versuche:

- NBA 

  → DBA mittels Potenzmengenkonstruktion (PMK) muss wegen Satz 3.12 fehlschlagen – Bsp. siehe Tafel
- NBA 

  → determ. Muller-(Rabin-/Streett-)Automat via PMK schlägt auch fehl – mit demselben Gegenbeispiel T 3.13

#### Hauptproblem:

- Potenzautomat simuliert mehrere Runs gleichzeitig
- akzeptierende Zustände (akzZ) müssen dabei nicht synchron erreicht werden
- Bad runs:

Wenn DBA  $\mathcal{A}^d$  für  $\alpha$  eine  $\infty$  Folge von akzZ findet. dann können diese akzZ von verschiedenen Runs des NBA  $\mathcal{A}$ auf Präfixen von  $\alpha$  stammen.

Diese Runs müssen nicht zu einem Run auf  $\alpha$  fortsetzbar sein.

Potenzmengenkonstruktion versagt Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-9 rung Potenzmengenkonstruktion versagt

muss wegen Satz 3.12 fehlschlagen - Bsp. siehe Tafel T 3.12 NBA → determ. Muller-(Rabin-/Streett-)Automat via PMK schlägt auch fehl - mit demselben Gegenbeispiel

· Potenzautomat simuliert mehrere Runs gleichzeitig a akzeptierende Zustände (akzZ) müssen dabei nicht synchron erreicht werden

> Wenn DBA A<sup>d</sup> für α eine ∞ Folge von akzZ findet. dann können diese akzZ von verschiedenen Runs des NBA "A auf Präfixen von a stammer Diese Runs müssen nicht zu einem Run auf  $\alpha$  fortsetzbar sein

16:03

T3.12 bis 16:10

T3.13 bis 16:17

insq. bis 16:19

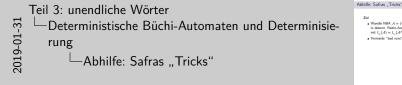
Model-Checking

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

# Abhilfe: Safras "Tricks"

#### Ziel

- Wandle NBA  $\mathcal{A}=(Q,\Sigma,\Delta,I,F)$  in determ. Rabin-Automaten  $\mathcal{A}^d=(Q^d,\Sigma,\Delta^d,I^d,\mathcal{P}^d)$  um mit  $L_\omega(\mathcal{A})=L_\omega(\mathcal{A}^d)$
- Vermeide "bad runs": Safras Tricks



$$\begin{split} & \text{Zid} \\ & \bullet \text{ Wandle NBA } \mathcal{A} = \left(Q, \Sigma, \Delta, l, F\right) \\ & \text{is detern. Rubin-Automaten } \mathcal{A}^d = \left(Q^d, \Sigma, \Delta^d, l^d, \mathcal{P}^d\right) \text{ urm} \\ & \text{mit } \mathbb{L}\left(\mathcal{A}\right) = \mathbb{L}\left(\mathcal{A}^d\right) \\ & \bullet \text{ Vermitide "bad runs": Salfast Tricks} \end{split}$$

16:19

Model-Checking

#### Ziel

- Wandle NBA  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ in determ. Rabin-Automaten  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P}^d)$  um  $mit L_{\omega}(\mathcal{A}) = L_{\omega}(\mathcal{A}^d)$
- Vermeide "bad runs": Safras Tricks

### Vorbetrachtungen

- Makrozustände: Zustände der alten PMK (Mengen  $M \subseteq Q$ )
- Zustände von  $\mathcal{A}^d$ : ≈ Bäume, deren Knoten mit Makrozuständen markiert sind
- Startzustand: Knoten I (Menge der Anfangszust., wie bei PMK)

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung -Abhilfe: Safras "Tricks"

Abhilfe: Safras "Tricks" Vermeide "had nuns": Safras Tricks

~ Diamo done Kenten mit Malacoustinden markinst si

16:19

otiv. Büchi-Aut. Abschlusseig. Charakt. **Determinismus** Entscheidungsprobl. *Model-Checking* 

# Safras Trick 1

#### Trick 1:

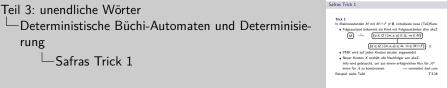
In Makrozuständen M mit  $M \cap F \neq \emptyset$ , initialisiere neue (Teil)Runs:

• Folgezustand bekommt ein Kind mit Folgezuständen aller akzZ

$$\begin{array}{ccc}
M & \xrightarrow{a} & \left[ \{ q \in Q \mid (m, a, q) \in \Delta, \ m \in M \} \right] \\
& & \downarrow \\
& \left[ \{ q \in Q \mid (m, a, q) \in \Delta, \ m \in M \cap F \} \right] X
\end{array}$$

- PMK wird auf jeden Knoten einzeln angewendet
- Neuer Knoten X enthält alle Nachfolger von akzZ; Info wird gebraucht, um aus einem erfolgreichen Run für  $\mathcal{A}^d$ einen für  $\mathcal{A}$  zu konstruieren  $\longrightarrow$  vermeidet  $\mathit{bad}$  runs

Beispiel: siehe Tafel T 3.14



#### 16:22

2019-01-

Safras Ideen bestehen aus drei Tricks, die ich jetzt halb formal, halb intuitiv vorstelle.

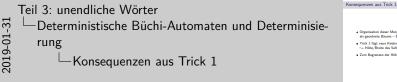
Anschließend präzise als Konstruktion mit 6 Schritten beschreiben.

Kinder im Baum sind immer unten; deshalb keine Pfeilspitzen!

Büchi-Aut. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

# Konsequenzen aus Trick 1

- Organisation dieser Mengen von Makrozuständen: als geordnete Bäume – Safra-Bäume
- Trick 1 fügt neue Kinder/Geschwister hinzu → Höhe/Breite des Safra-Baums wächst
- Zum Begrenzen der Höhe/Breite: Trick 2 und 3



 Organisation dieser Mengen von Makrozuständen als geordnete Bäume - Safra-Bäume

u Trick 1 fügt neue Kinder/Geschwister hinzu ~ Höhe/Breite des Safra-Baums wächst w Zum Begrenzen der Höhe/Breite: Trick 2 und 3

16:32

7. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Safras Trick 2

#### Trick 2:

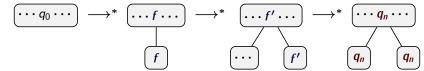
Erkenne zusammenlaufende Teilruns und lösche überflüssige Info

Bsp.: Betrachte Teilruns, die in demselben Zustand  $q_n$  enden:

$$r = q_0 q_1 q_2 \dots f \dots q_{n-1} \mathbf{q}_n$$
  

$$r' = q_0 q'_1 q'_2 \dots f' \dots q'_{n-1} \mathbf{q}_n \qquad (f, f' \in F)$$

Zugehörige *n* Schritte von  $A^d$  unter Anwendung von Trick 1:



Trick 2 vereinigt die beiden  $\{q_n\}$ -Kinder ("horizontal merge")

→ Weite von Safra-Bäumen wird beschränkt

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Safras Trick 2

Tick 2: Enteres parameteris/fueder Tailmens and tisches disarbitatings belower by the Distraction Tailmen, die in demandiere Zentand qu, ordens:  $r' = \log q_{1} \cdots \cdots q_{r-n} q_{r} \quad (r, r' \in \mathcal{F})$  Zuglichiger as Schötzte von  $\mathcal{A}'$  utter h Ammendung von Trok 1:  $r_{1} \cdots r_{r} \cdots r_$ 

Safras Trick 2

16:33

ptiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Safras Trick 3

#### Trick 3:

Gib überflüssige Makrozustände zur Löschung frei

Wenn alle Kinder eines MZ M bezeugen, dass jeder Zustand in M einen akz. Zustand als Vorgänger hat, dann können die Kinder gelöscht werden

Genauer: wenn M Kinder  $M_1, \ldots, M_n$  hat mit  $M_1 \cup \cdots \cup M_n = M$ , dann werden die  $M_i$  gelöscht und M mit (!) markiert

→ "vertical merge", beschränkt die **Tiefe** von Safra-Bäumen

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierrung

rung

Safras Trick 3

Teil 3: unendliche Wörter

Teil 4: Safras Trick 3

Teil 4: Matter und Michia Mitter und Mit

16:37 bis 16:39, dann 5min Pause

Sei Q Zustandsmenge des ursprünglichen NBA und *V* eine nichtleere Menge von Knotennamen.

Makrozustand (MZ) über Q: Teilmenge  $M \subseteq Q$ 

Safra-Baum über Q, V:

- ullet geordneter Baum mit Knoten aus V(der leere Baum ist erlaubt!)
- jeder Knoten mit einem **nichtleeren** MZ markiert und möglicherweise auch mit (!)
- Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$ markiert sind. dann:

  - $\bigcirc$   $M_i$  sind paarweise disjunkt

#### 16:44

2019-01-

Fragen: Wer ahnt, wozu die letzte Bedingung (1 und 2) wichtig ist? (Antw.: stellt sicher, dass es nicht zu viele mögliche SB gibt – zeigen wir jetzt!)

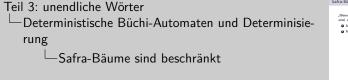
Charakt.

Determinismus

Entscheidungsprobl.

"Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$  markiert sind, dann:

- $M_1 \cup \cdots \cup M_n \subseteq M$
- M<sub>i</sub> sind paarweise disjunkt"



16:47

2019-01

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

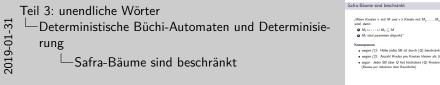
### Safra-Bäume sind beschränkt

"Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$  markiert sind, dann:

- M<sub>i</sub> sind paarweise disjunkt"

### Konseguenzen

- wegen (1): Höhe jedes SB ist durch |Q| beschränkt
- wegen (2): Anzahl Kinder pro Knoten kleiner als |Q|
- sogar: Jeder SB über Q hat höchstens |Q| Knoten (Beweis per Induktion über Baumhöhe)



16:47

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Safra-Bäume sind beschränkt

"Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$  markiert sind, dann:

- M<sub>i</sub> sind paarweise disjunkt"

### Konseguenzen

- wegen (1): Höhe jedes SB ist durch |Q| beschränkt
- wegen (2): Anzahl Kinder pro Knoten kleiner als |Q|
- sogar: Jeder SB über Q hat höchstens |Q| Knoten (Beweis per Induktion über Baumhöhe)
- → Anzahl der möglichen SB ist beschränkt durch ?

Safra-Bäume sind beschränkt Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie- $\bigcirc$  M<sub>1</sub>  $\cup \cdots \cup$  M<sub>n</sub>  $\subset$  M M; sind paarweise disjunkt rung weeren (1): H\(\tilde{t}\) he ierles SR ist durch |\(\O\)| heschr\(\tilde{t}\) he Safra-Bäume sind beschränkt - Δnzahl der möglichen SR ist heschränkt durch

16:47

2019-01-

Abschlusseig.

Determinismus

Entscheidungsprobl.

# Safra-Bäume sind beschränkt

"Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$  markiert sind, dann:

- M<sub>i</sub> sind paarweise disjunkt"

### Konseguenzen

- wegen (1): Höhe jedes SB ist durch |Q| beschränkt
- wegen (2): Anzahl Kinder pro Knoten kleiner als |Q|
- sogar: Jeder SB über Q hat höchstens |Q| Knoten (Beweis per Induktion über Baumhöhe)
- $\rightarrow$  Anzahl der möglichen SB ist beschränkt durch  $2^{O(|Q| \cdot \log |Q|)}$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung Safra-Bäume sind beschränkt

Safra-Bäume sind beschränkt

 $\bigcirc$  M<sub>1</sub>  $\cup \cdots \cup$  M<sub>n</sub>  $\subset$  M M; sind paarweise disjunkt

weeren (1): H\(\tilde{t}\) he ierles SR ist durch |\(\O\)| heschr\(\tilde{t}\) he

16:47

Abschlusseig.

Charakt.

Determinismus

2019-01-

16:50

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein NBA und  $V = \{1, \dots, 2|Q|\}$ . Konstruieren DRA  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P})$ :

- $Q^d$  = Menge aller Safra-Bäume über Q, V
- $I^d = Safra-Baum$  mit einzigem Knoten I
- $\Delta^d = \{(S, a, S') \mid S' \text{ wird aus } S \text{ wie folgt konstruiert}\}$



Knotennamen  $1, \ldots, 2|Q|$  reichen wegen der Beschränkungen, die wir für die Knotenzahl eines SB gerade aufgestellt haben.

Sei  $A = (Q, \Sigma, \Delta, I, F)$  ein NBA und  $V = \{1, ..., 2|Q|\}$ 

Δ<sup>d</sup> = {(S, a, S') | S' wird aus S wie folgt konstruiert}

Übergangsrelation folgt genau Safras Tricks (in jedem einzelnen Übergang!). Akzeptanzkomponente kommt am Ende.

Büchi-Aut. Charakt. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

# Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

• Beginne mit S; entferne alle Markierungen (!)

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung -Konstruktion von S' aus S in 6 Schritten

Konstruktion von S' aus S in 6 Schritten

Beginne mit S; entferne alle Markierungen ()

16:52

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1

Schritt 4 = Trick 2

Schritt 6 = Trick 3

Charakt.

Determinismus

Entscheidungsprobl.

# Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

- **1** Beginne mit S; entferne alle Markierungen (!)
- ② Für jeden Knoten v mit Makrozustand M und  $M \cap F \neq \emptyset$ , füge neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)

Konstruktion von S' aus S in 6 Schritten Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung

Beginne mit S; entferne alle Markierungen () Für jeden Knoten v mit Makrozustand M und M ∩ F ≠ Ø, Ture news Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu

-Konstruktion von S' aus S in 6 Schritten

#### 16:52

0

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1

Schritt 4 = Trick 2

Schritt 6 = Trick 3

Determinismus Entscheidungsprobl.

Model-Checking

### Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

- Beginne mit S; entferne alle Markierungen (!)
- ② Für jeden Knoten v mit Makrozustand M und  $M \cap F \neq \emptyset$ , füge neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
- **1** Wende Potenzmengenkonstruktion auf alle Knoten *v* an: ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen V' ⊂ V: sei a ∈ ∑

Beginne mit S; entferne alle Markierungen () Für jeden Knoten v mit Makrozustand M und M ∩ F ≠ Ø, füre neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu

Konstruktion von 5' aus 5 in 6 Schritten

(als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder) ersetze MZ M durch  $\{a \in Q \mid (m, a, a) \in \Delta \text{ für ein } m \in M\}$ 

16:52

0

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1

Schritt 4 = Trick 2

Schritt 6 = Trick 3

Determinismus

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

- Beginne mit S; entferne alle Markierungen (!)
- ② Für jeden Knoten v mit Makrozustand M und  $M \cap F \neq \emptyset$ , füge neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
- Wende Potenzmengenkonstruktion auf alle Knoten v an: ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$
- Horizontales Zusammenfassen: Für jeden Knoten v mit MZ M, lösche jeden Zustand q, der im MZ eines älteren Geschwisters vorkommt, aus M und aus den MZen der Kinder von v

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Konstruktion von S' aus S in 6 Schritten

Konstruktion von 5' aus 5 in 6 Schritten

Sei S Safra-Baum mit Knotennamen V' ⊂ V: sei a ∈ Σ Beginne mit S; entferne alle Markierungen ()

 Für jeden Knoten v mit Makrozustand M und M ∩ F ≠ Ø, füre neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)

ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$ Horizontales Zusammenfassen: Für ieden Knoten v mit MZ M. lösche jeden Zustand q, der im MZ eines älteren Geschwisters rorkommt, aus M und aus den MZen der Kinder von v

16:52

9

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1

Schritt 4 = Trick 2

Schritt 6 = Trick 3

Konstruktion von S' aus S in 6 Schritten

Determinismus

- (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
- ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$ Horizontales Zusammenfassen: Für ieden Knoten v mit MZ M.
- lösche jeden Zustand q, der im MZ eines älteren Geschwisters vorkommt, aus M und aus den MZen der Kinder von v
- A Entferne alle Knoten mit leeren MZen

# Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie--Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

- Beginne mit S; entferne alle Markierungen (!)
- ② Für jeden Knoten v mit Makrozustand M und  $M \cap F \neq \emptyset$ , füge neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
- **1** Wende Potenzmengenkonstruktion auf alle Knoten *v* an: ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$
- **4** Horizontales Zusammenfassen: Für jeden Knoten  $\nu$  mit MZ M, lösche jeden Zustand q, der im MZ eines älteren Geschwisters vorkommt, aus M und aus den MZen der Kinder von v
- Entferne alle Knoten mit leeren MZen

#### 16:52

rung

9

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1

Schritt 4 = Trick 2

Schritt 6 = Trick 3

lotiv. B

Büchi-Aut. Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

# Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subset V$ ; sei  $a \in \Sigma$ 

- **1** Beginne mit S; entferne alle Markierungen !
- ② Für jeden Knoten v mit Makrozustand M und  $M \cap F \neq \emptyset$ , füge neues Kind  $v' \in V \setminus V'$  mit Markierung  $M \cap F$  hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
- **③** Wende Potenzmengenkonstruktion auf alle Knoten v an: ersetze MZ M durch  $\{q \in Q \mid (m, a, q) \in \Delta \text{ für ein } m \in M\}$
- Horizontales Zusammenfassen: Für jeden Knoten v mit MZ M, lösche jeden Zustand q, der im MZ eines älteren Geschwisters vorkommt, aus M und aus den MZen der Kinder von v
- Entferne alle Knoten mit leeren MZen
- Vertikales Zusammenfassen: Für jeden Knoten v, dessen Markierung nur Zustände aus v's Kindern enthält, lösche alle Nachfolger von v und markiere v mit (!)

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Konstruktion von S' aus S in 6 Schritten

Konstruktion von S' aus S in 6 Schritten

Sei S Safra-Baum mit Knotennamen  $V' \subseteq V$ ; sei  $a \in \Sigma$  $\bullet$  Beginne mit S; entferne alle Markierungen  $\bigcirc$ 

Für jeden Knoten v mit Makrozustand M und M ∩ F ≠ ∅, füge nauss Kind v' ∈ V \ V' mit Markinerung M ∩ F hinzu (als jüngstes (rechtes) Geschwister aller evtl. vorhandenen Kinder)
 Wende Potenzmengenkonstruktion auf alle Knoten v an:

Wende Potenzinnigenkonstruktion auf alle Knoten v an: 
 ensetza MZ M durch { e Q | (m, a, q) ∈ \( \Limits\) für ein m ∈ M}
 Horizontakis Zusammenflassen: Für jeden Knoten v mit MZ M, 
 lösche jeden Zustand q, der im MZ eines älteren Geschwisters 
 verkommt. aus M und aus dem MZen der Kinder ven.

Entferne alle Knoten mit leeren MZen
 Vertikales Zusammenfassen: Für jeden Knoten v., dessen

Vertikales Zusammenfassen: Für jeden Knoten v., dessen Markierung nur Zustände aus v's Kindern enthält, lösche alle Nachfolger von v und markiere v mit ①

#### 16:52

2019-01

Und das sind die 6 Schritte, die auf den 3 Tricks von Safra beruhen.

Schritt 2 = Trick 1 Schritt 4 = Trick 2

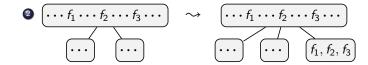
Schritt 6 = Trick 3

Abschlusseig.

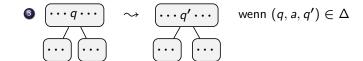
Determinismus

Entscheidungsprobl. Model-Checking

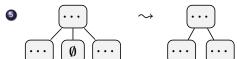
# Illustration der Schritte 2–5



Charakt.

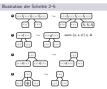






2019-01

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Illustration der Schritte 2-5



#### 16:58

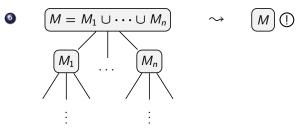
Hier wieder schematische Skizzen; als nächstes am konkreten Bsp. nur letztes Bild an Tafel (ist hier nicht eindeutig)

Charakt. Determinismus

Entscheidungsprobl.

Model-Checking

# Illustration von Schritt 6



- d. h. alle Zustände in M kommen im Makrozustand eines Kindes  $M_i$  vor
- d. h. jeder Zustand in M hat einen akzZ als Vorgänger!

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Illustration von Schritt 6

Illustration von Schritt 6

17:01

Abschlusseig.

Charakt. Determinismus Entscheidungsprobl.

Wenn Knoten v mit M und v's Kinder mit M ..... M.

# Erläuterungen zur Konstruktion

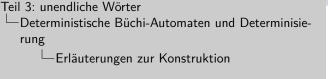
• S' ist wieder ein Safra-Baum:

Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$ markiert sind, dann:

"⊂": *Schritte 2, 3* "≠": Schritt 6

2 M<sub>i</sub> sind paarweise disjunkt

Schritt 4



17:04

2019-01

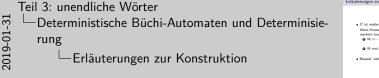
Def. der Akzeptanzkomponente kommt nach dem Beispiel.

Bsp. bis 17:28

• S' ist wieder ein Safra-Baum:

Wenn Knoten v mit M und v's Kinder mit  $M_1, \ldots, M_n$  markiert sind, dann:

Beispiel: siehe Tafel
 T 3.15



Filsuterungen zur Konstruktion

S ist wieder ein Salne-Baun:
Weine Knaten v mit M und v is Kinder mit M.....M.,
maleier sind, dans

G M, 'U'-U'-M.C. M. 'S' Salne

G M, will provinse dejankt

Biospeli siehe Tufel

T:

17:04

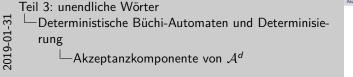
Def. der Akzeptanzkomponente kommt nach dem Beispiel.

Bsp. bis 17:28

F<sub>v</sub> = alle Safra-Bäume, in denen v mit 
 markiert ist

$$\mathcal{P} = \{(E_v, F_v) \mid v \in V\} \text{ mit }$$

- $E_v$  = alle Safra-Bäume ohne Knoten v
- $F_v = \text{alle Safra-B\"{a}ume}$ , in denen v mit ① markiert ist



#### 17:28 bis $17:30 \sim$ Punktlandung?

Jetzt müssen wir natürlich noch zeigen, dass die Konstruktion korrekt ist. Das tun wir nächste Woche! :)

2019-01-

# Akzeptanzkomponente von $\mathcal{A}^d$

$$\mathcal{P} = \{(E_v, F_v) \mid v \in V\} \text{ mit }$$

- $E_{V}$  = alle Safra-Bäume ohne Knoten V
- $F_v =$  alle Safra-Bäume, in denen v mit (!) markiert ist

 $\rightarrow$  d. h. Run  $r = S_0 S_1 S_2 \dots$  von  $\mathcal{A}^d$  ist erfolgreich, wenn es einen Knotennamen v gibt, so dass

- alle  $S_i$ , bis auf endlich viele, einen Knoten v haben und
- unendlich oft auf v Schritt 6 angewendet wurde, d. h. vorher kamen alle Zustände in v's MZ in v's Kindern vor T 3.15 Forts.

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung –Akzeptanzkomponente von  $\mathcal{A}^d$ 

 $P = \{(E_v, F_v) \mid v \in V\}$  mit 

Akzeptanzkomponente von Aa

a F. = alle Safra-Bäume, in denen v mit () markiert ist

 $\sim$  d. h. Run  $r = S_0S_1S_2...$  von  $\mathcal{A}^d$  ist erfolgreich venn es einen Knotennamen v gibt, so dass

unendlich oft auf v Schritt fi angewendet wurde

#### 17:28 bis 17:30 $\rightarrow$ Punktlandung?

Jetzt müssen wir natürlich noch zeigen, dass die Konstruktion korrekt ist. Das tun wir nächste Woche! :)

#### Lemma 3.22

Sei  $\mathcal{A}=(Q,\Sigma,\Delta,I,F)$  ein NBA und sei  $\mathcal{A}^d=(Q^d,\Sigma,\Delta^d,I^d,\mathcal{P})$  der DRA, den man nach Safras Konstruktion aus  $\mathcal{A}$  erhält.

Dann gilt  $L_{\omega}(\mathcal{A}^d) = L_{\omega}(\mathcal{A})$ .

Korrektheit:  $A^d$  akzeptiert nur Wörter, die A akzeptiert

$$L_{\omega}(\mathcal{A}^d) \subset L_{\omega}(\mathcal{A})$$

Vollständigkeit:

(Completeness)

(Soundness)

 $\mathcal{A}^d$  akzeptiert (mindestens) alle Wörter, die  $\mathcal{A}$  akzeptiert  $L_{\omega}(\mathcal{A}^d) \supset L_{\omega}(\mathcal{A})$ 

Beweis: Folgerung aus den nächsten beiden Lemmas

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Korrektheit und Vollständigkeit der Konstruktion

Korrektheit und Vollständigkeit der Konstruktion

Korektheit: (Soc  $A^{d}$  akzeptiert nur Wörter, die A akzeptiert  $L_{\omega}(A^{c}) \subseteq L_{\omega}(A)$ Vollständigkeit: (Compil  $A^{d}$  akzeptiert (mindestens) alle Wörter, die A akzeptiert

 $L_{\omega}(\mathcal{A}^d)\supseteq L_{\omega}(\mathcal{A})$ Beweis: Folgerung aus den nächsten beiden Lemmas

8:30

-01

Büchi-Aut. Korrektheit

Abschlusseig.

Charakt.

Determinismus

Model-Checking

Entscheidungsprobl.

#### Korrektheit

Dann gilt  $L_{\omega}(A^d) \subset L_{\omega}(A)$ 

Beweisidee. Sei  $I = \{q_l\}$  und  $I^d = \{S_l\}$ . Sei  $\alpha \in L_{\omega}(A^d)$ 

- Gesuchter Run von A ist ein ∞ Pfad in T

Deterministische Büchi-Automaten und Determinisierung -Korrektheit

8:32

2019-01-

Zuerst die Beweis**ide**e.

Teil 3: unendliche Wörter

#### Lemma 3.23

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein NBA und sei  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P})$ der DRA, den man nach Safras Konstruktion aus  $\mathcal{A}$  erhält.

Dann gilt  $L_{\omega}(\mathcal{A}^d) \subset L_{\omega}(\mathcal{A})$ .

Beweisidee. Sei  $I = \{g_I\}$  und  $I^d = \{S_I\}$ . Sei  $\alpha \in L_{\omega}(\mathcal{A}^d)$ .

- Betrachte erfolgreichen Run s von  $A^d$  auf  $\alpha$ .
- "Konstruiere" daraus erfolgr. Run von  $\mathcal{A}$  auf  $\alpha$  stückweise:

$$s = S_1 \dots T_1 \dots T_2 \dots T_3 \dots$$
, (alle  $T_i$  laut  $\mathcal{P}$  gewählt)

- Jeder Teilrun  $T_i \dots T_{i+1}$  induziert Teilrun von  $\mathcal{A}$  auf Teilwort von  $\alpha$ , der einen akz. Zustand enthält
- Ordnen diese endl. Teilruns in einem  $\infty$  Baum  $\mathcal{T}$  an
- Gesuchter Run von  $\mathcal{A}$  ist ein  $\infty$  Pfad in  $\mathcal{T}$

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

### Korrektheit

Beweis. Sei also  $\alpha \in L_{\omega}(\mathcal{A}^d)$ .

Dann gibt es erfolgreichen Run  $s = S_0 S_1 S_2 \dots$  von  $\mathcal{A}^d$  auf  $\alpha$  und ein Knoten v, der (wegen  $\mathcal{P}^d$ )

- in allen Safra-Bäumen  $S_i, S_{i+1}, \ldots$  vorkommt, für ein  $j \ge 0$ , und
- in ∞ vielen Safra-Bäumen mit (!) markiert ist. Seien diese  $T_1, T_2, \ldots$  und sei  $T_0 = S_0$ :

$$s = T_0 \dots T_1 \dots T_2 \dots T_3 \dots$$

Teil 3: unendliche Wörter 2019-01-31 Deterministische Büchi-Automaten und Determinisierung -Korrektheit

Korrektheit Beweis. Sei also  $\alpha \in L_{\omega}(A^d)$ .

Dann gibt es erfolgreichen Run  $s = S_0S_1S_2...$  von  $\mathcal{A}^d$  auf  $\alpha$  und ein Knoten v. der (wegen Pd)  $\mathbf{v}$  in allen Safra-Bäumen  $S_j, S_{j+1}, \ldots$  vorkommt, für ein  $j \ge 0$ , und ø in ∞ vielen Safra-Bäumen mit () markiert ist

Seien diese  $T_1, T_2, ...$  und sei  $T_0 = S_0$ :  $s=T_0\dots T_1\dots T_2\dots T_3\dots$ 

8:36 Jetzt der eigentliche Beweis.

Bis 9:10

Abschlusseig.

Charakt.

Determinismus

### Korrektheit

Beweis. Sei also  $\alpha \in L_{\omega}(\mathcal{A}^d)$ .

Dann gibt es erfolgreichen Run  $s = S_0 S_1 S_2 \dots$  von  $\mathcal{A}^d$  auf  $\alpha$  und ein Knoten v, der (wegen  $\mathcal{P}^d$ )

- in allen Safra-Bäumen  $S_i, S_{i+1}, \ldots$  vorkommt, für ein  $j \ge 0$ , und
- in ∞ vielen Safra-Bäumen mit (!) markiert ist. Seien diese  $T_1, T_2, \ldots$  und sei  $T_0 = S_0$ :

$$s = T_0 \dots T_1 \dots T_2 \dots T_3 \dots$$

Zeigen Hilfsaussage [HA]:

Für alle  $T_i$  und alle Zustände p im MZ von v in  $T_{i+1}$ gibt es einen Zustand q im MZ von v in  $T_i$ und einen endlichen Run  $q \dots p$  von  $\mathcal{A}$  auf dem zugehörigen Teilwort von  $\alpha$ , der einen akzZ enthält.

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung Korrektheit

Korrektheit Beweis. Sei also  $\alpha \in L_{\omega}(A^d)$ Dann gibt es erfolgreichen Run  $s=S_0S_1S_2...$  von  $A^d$  auf  $\alpha$  und ein Knoten v. der (wegen Pd)

 $\mathbf{v}$  in allen Safra-Bäumen  $S_j, S_{j+1}, \dots$  vorkommt, für ein  $j \ge 0$ , und ø in ∞ vielen Safra-Bäumen mit () markiert ist Seien diese  $T_1, T_2, ...$  und sei  $T_0 = S_0$ :  $s=T_0\dots T_1\dots T_2\dots T_3\,.$ 

Für alle T: und alle Zustände p im MZ von v in T:... eilwort von a. der einen akzZ enthält.

8:36

Jetzt der eigentliche Beweis.

Bis 9:10

Abschlusseig.

Charakt. Determinismus Entscheidungsprobl.

Model-Checking

#### Korrektheit

Beweis. Sei also  $\alpha \in L_{\omega}(\mathcal{A}^d)$ .

Dann gibt es erfolgreichen Run  $s = S_0 S_1 S_2 \dots$  von  $\mathcal{A}^d$  auf  $\alpha$  und ein Knoten v, der (wegen  $\mathcal{P}^d$ )

- in allen Safra-Bäumen  $S_i, S_{i+1}, \ldots$  vorkommt, für ein  $j \ge 0$ , und
- in ∞ vielen Safra-Bäumen mit (!) markiert ist. Seien diese  $T_1, T_2, \ldots$  und sei  $T_0 = S_0$ :

$$s = T_0 \dots T_1 \dots T_2 \dots T_3 \dots$$

Zeigen Hilfsaussage [HA]:

Für alle  $T_i$  und alle Zustände p im MZ von v in  $T_{i+1}$ gibt es einen Zustand q im MZ von v in  $T_i$ und einen endlichen Run  $q \dots p$  von  $\mathcal{A}$  auf dem zugehörigen Teilwort von  $\alpha$ , der einen akzZ enthält.

Beweis der Hilfsaussage: s. Tafel

T 3.16

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung Korrektheit

Korrektheit Beweis. Sei also  $\alpha \in L_{\omega}(A^d)$ Dann gibt es erfolgreichen Run  $s=S_0S_1S_2...$  von  $A^d$  auf  $\alpha$  und ein Knoten v, der (wegen  $P^d$ )  $\mathbf{v}$  in allen Safra-Bäumen  $S_j, S_{j+1}, \dots$  vorkommt, für ein  $j \ge 0$ , und ø in ∞ vielen Safra-Bäumen mit () markiert ist Seien diese  $T_1, T_2, ...$  und sei  $T_0 = S_0$ :  $s=T_0\dots T_1\dots T_2\dots T_3\,.$ Für alle T: und alle Zustände p im MZ von v in T:... Beweis der Hilfsaussage: s. Tafel

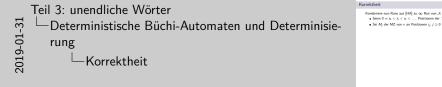
8:36 Jetzt der eigentliche Beweis.

Bis 9:10

### Korrektheit

Kombiniere nun Runs aus [HA] zu  $\infty$  Run von  $\mathcal A$ 

- Seien  $0 = i_0 < i_1 < i_2 < \dots$  Positionen der  $T_i$  in s
- ullet Sei  $M_j$  der MZ von v an Positionen  $i_j,\,j\geqslant 0$



Büchi-Aut. Model-Checking Motiv. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

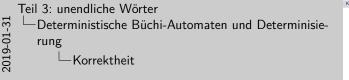
### Korrektheit

Kombiniere nun Runs aus [HA] zu  $\infty$  Run von  $\mathcal A$ 

- Seien  $0 = i_0 < i_1 < i_2 < \dots$  Positionen der  $T_i$  in s
- Sei  $M_i$  der MZ von v an Positionen  $i_i$ ,  $i \ge 0$

Konstruiere Baum  $\mathcal{T}$ :

- Knoten = Paare (q, j) mit  $q \in M_i$ ,  $j \ge 0$
- Jeder Knoten (p, j + 1) bekommt genau ein Elternteil: beliebiger (q, j) mit  $q \in M_i$  und  $\exists \text{Run } q \dots p \text{ wie in } [HA]$
- $\Rightarrow \infty$  viele Knoten, Verzweigungsgrad  $\leq |Q|$ , Wurzel  $(q_l, 0)$



Korrektheit

- Kombiniere nun Runs aus [HA] zu ∞ Run von A  $\mathbf{u}$  Sei M; der MZ von  $\mathbf{v}$  an Positionen  $i_i, j \ge 0$
- Konstruiere Baum T ■ Knoten = Paare (a, i) mit  $a \in M$ ,  $i \ge 0$
- Jeder Knoten (p. i + 1) bekommt genau ein Elternteil:
- beliebiger (q, j) mit  $q \in M_i$  und  $\exists Run \ q \dots p$  wie in [HA] ⇒ ∞ viele Knoten. Verzweigungsgrad ≤ |Q|. Wurzel (q<sub>1</sub>, 0)

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

#### Model-Checking

### Korrektheit

Kombiniere nun Runs aus [HA] zu  $\infty$  Run von  $\mathcal{A}$ 

- Seien  $0 = i_0 < i_1 < i_2 < \dots$  Positionen der  $T_i$  in s
- Sei  $M_i$  der MZ von v an Positionen  $i_i$ ,  $i \ge 0$

Konstruiere Baum  $\mathcal{T}$ :

- Knoten = Paare (q, j) mit  $q \in M_i$ ,  $j \ge 0$
- Jeder Knoten (p, j + 1) bekommt genau ein Elternteil: beliebiger (q, j) mit  $q \in M_i$  und  $\exists$  Run  $q \dots p$  wie in [HA]
- $\Rightarrow \infty$  viele Knoten, Verzweigungsgrad  $\leq |Q|$ , Wurzel  $(q_I, 0)$

Nach Lemma von König (nächste Folie) folgt:

- $\mathcal{T}$  hat einen  $\infty$  Pfad  $(q_1,0), (q_1,1), (q_2,2), \ldots$ ;
- Verkettung aller Teilruns entlang dieses Pfades ist ein Run von  $\mathcal{A}$  auf  $\alpha$ , der  $\infty$  oft einen akzZ besucht

$$\Rightarrow \alpha \in L_{\omega}(A)$$

Korrektheit Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01 rung Korrektheit

Kombiniere nun Runs aus [HA] zu ∞ Run von A

 $\mathbf{u}$  Sei M; der MZ von  $\mathbf{v}$  an Positionen  $i_i, j \ge 0$ Konstruiere Baum T

 $\mathbf{u}$  Knoten = Paare (q, j) mit  $q \in M_j, j \ge 0$ 

 Jeder Knoten (p. i + 1) bekommt genau ein Elternteil: beliebiger (q, j) mit  $q \in M_i$  und  $\exists Run \ q \dots p$  wie in [HA]

⇒ ∞ viele Knoten. Verzweigungsgrad ≤ |Q|. Wurzel (q<sub>1</sub>, 0)

• T hat einen  $\infty$  Pfad (q<sub>1</sub>, 0), (q<sub>1</sub>, 1), (q<sub>2</sub>, 2), u Verkettung aller Teilruns entlang dieses Pfades ist ein Run von A auf α, der ∞ oft einen akzZ besucht

 $\Rightarrow a \in L_1(A)$ 

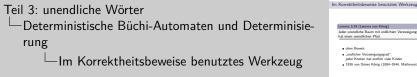
Büchi-Aut. Charakt. Determinismus Entscheidungsprobl. Abschlusseig.

### Im Korrektheitsbeweise benutztes Werkzeug

### Lemma 3.24 (Lemma von Kőnig)

Jeder unendliche Baum mit endlichem Verzweigungsgrad hat einen unendlichen Pfad.

- ohne Beweis
- "endlicher Verzweigungsgrad": jeder Knoten hat endlich viele Kinder
- 1936 von Dénes Kőnig (1884–1944, Mathematiker, Budapest)



a ohne Beweis ieder Knoten hat endlich viele Kinder

9:15

einschl. 4min Pause bis 9:20

Model-Checking

#### Lemma 3.25

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein NBA und sei  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P})$ der DRA, den man nach Safras Konstruktion aus  $\mathcal{A}$  erhält.

Dann gilt  $L_{\omega}(\mathcal{A}) \subseteq L_{\omega}(\mathcal{A}^d)$ .

#### Beweis.

- Sei  $\alpha \in L_{\omega}(\mathcal{A})$  und  $r = q_0 q_1 q_2 \dots$  erfolgr. Run von  $\mathcal{A}$  auf  $\alpha$
- $A^d$  hat eindeutigen Run  $s = S_0 S_1 S_2 \dots$  auf  $\alpha$
- Zu zeigen: s ist erfolgreich, d. h.:

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Vollständigkeit

Vollständigkeit

Dann gilt  $L_{\omega}(A) \subset L_{\omega}(A^d)$ 

. Zu zeigen: s ist erfolgreich, d.h.:

9:20 bis 9:55

### Vollständigkeit

#### Lemma 3.25

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein NBA und sei  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P})$ der DRA, den man nach Safras Konstruktion aus  $\mathcal{A}$  erhält.

Dann gilt  $L_{\omega}(A) \subseteq L_{\omega}(A^d)$ .

#### Beweis.

- Sei  $\alpha \in L_{\omega}(\mathcal{A})$  und  $r = q_0 q_1 q_2 \dots$  erfolgr. Run von  $\mathcal{A}$  auf  $\alpha$
- $A^d$  hat eindeutigen Run  $s = S_0 S_1 S_2 \dots$  auf  $\alpha$
- Zu zeigen: s ist erfolgreich, d. h.:

Es gibt einen Knotennamen v, für den gilt:

- (a)  $\exists m \ge 0$ :  $S_i$  enthält Knoten v für alle  $i \ge m$
- (b) v ist in  $\infty$  vielen  $S_i$  mit (!) markiert

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung -Vollständigkeit

Vollständigkeit

Dann gilt  $L_{\omega}(A) \subset L_{\omega}(A^d)$ 

(a) ∃m ≥ 0 : S enthält Knoten v für alle i ≥ m (b) v ist in ∞ vielen S; mit (1) markiert

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

# Vollständigkeit

#### Lemma 3.25

Sei  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  ein NBA und sei  $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, I^d, \mathcal{P})$ der DRA, den man nach Safras Konstruktion aus  $\mathcal{A}$  erhält.

Dann gilt  $L_{\omega}(A) \subseteq L_{\omega}(A^d)$ .

#### Beweis.

- Sei  $\alpha \in L_{\omega}(\mathcal{A})$  und  $r = q_0 q_1 q_2 \dots$  erfolgr. Run von  $\mathcal{A}$  auf  $\alpha$
- $A^d$  hat eindeutigen Run  $s = S_0 S_1 S_2 \dots$  auf  $\alpha$
- Zu zeigen: s ist erfolgreich, d. h.:

Es gibt einen Knotennamen v, für den gilt:

- (a)  $\exists m \ge 0$ :  $S_i$  enthält Knoten v für alle  $i \ge m$
- (b) v ist in  $\infty$  vielen  $S_i$  mit (!) markiert

Beweis dieser Aussage: s. Tafel

T 3.17

Model-Checking

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisie-2019-01rung -Vollständigkeit

Dann gilt  $L_{\omega}(A) \subset L_{\omega}(A^d)$ 

(a) ∃m ≥ 0 : S enthält Knoten v für alle i ≥ m

Beweis dieser Aussage: s. Tafel

Büchi-Aut. Abschlusseig.

Charakt.

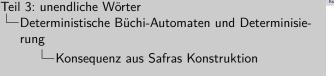
Determinismus

Entscheidungsprobl.

### Satz 3.26 (Satz von McNaughton)

Sei  $\mathcal{A}$  ein NBA. Dann gibt es einen DRA  $\mathcal{A}^d$  mit  $L_{\omega}(\mathcal{A}^d) = L_{\omega}(\mathcal{A})$ .

Beweis. Folgt aus Lemma 3.22.



Konsequenz aus Safras Konstruktion

9:55

2019-01

Büchi-Aut. Abschlusseig.

Determinismus

# Konsequenz aus Safras Konstruktion

### Satz 3.26 (Satz von McNaughton)

Sei  $\mathcal{A}$  ein NBA. Dann gibt es einen DRA  $\mathcal{A}^d$  mit  $L_{\omega}(\mathcal{A}^d) = L_{\omega}(\mathcal{A})$ .

Beweis. Folgt aus Lemma 3.22.

### Folgerung 3.27

Die Klasse der Büchi-erkennbaren Sprachen ist unter Komplement abgeschlossen.

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Die Klasse der Büchi-erkennbaren Sprachen ist unter Komplemen -Konsequenz aus Safras Konstruktion

9:55

2019-01

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Konsequenz aus Safras Konstruktion

### Satz 3.26 (Satz von McNaughton)

Sei  $\mathcal{A}$  ein NBA. Dann gibt es einen DRA  $\mathcal{A}^d$  mit  $L_{\omega}(\mathcal{A}^d) = L_{\omega}(\mathcal{A})$ .

Beweis. Folgt aus Lemma 3.22.

### Folgerung 3.27

Die Klasse der Büchi-erkennbaren Sprachen ist unter Komplement abgeschlossen.

Über folgende Transformationskette: Beweis.

NBA für 
$$L$$
  $\rightarrow$  DRA für  $L$  (gemäß Satz 3.26)  
 $\rightarrow$  DMA für  $L$  (gemäß Satz 3.21)  
 $\rightarrow$  DMA für  $\overline{L}$  (wie gehabt)  
 $\rightarrow$  NBA für  $\overline{L}$  (gemäß Satz 3.16)

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung Konsequenz aus Safras Konstruktion

Die Klasse der Büchi-erkennbaren Sprachen ist unter Komplemen

Sei A ein NBA. Dann gibt es einen DRA  $A^d$  mit  $L_-(A^d) = L_-(A$ 

Konsequenz aus Safras Konstruktion

9:55

Notiv. Büchi-Aut.

ut. Abschlusseig.

Charak

Determinismus Entsch

Entscheidungsprobl.

robl. Model-Checking

### Konsequenz aus Safras Konstruktion

### Satz 3.26 (Satz von McNaughton)

Sei  $\mathcal{A}$  ein NBA. Dann gibt es einen DRA  $\mathcal{A}^d$  mit  $L_{\omega}(\mathcal{A}^d) = L_{\omega}(\mathcal{A})$ .

Beweis. Folgt aus Lemma 3.22.

### Folgerung 3.27

Die Klasse der Büchi-erkennbaren Sprachen ist unter Komplement abgeschlossen.

Beweis. Über folgende Transformationskette:

NBA für  $L \rightarrow DRA$  für L (gemäß Satz 3.26)

 $\rightarrow$  DMA für *L* (gemäß Satz 3.21)

 $\rightarrow$  DMA für  $\overline{L}$  (wie gehabt)

NBA für  $\overline{L}$  (gemäß Satz 3.16)

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Konsequenz aus Safras Konstruktion

Sex 3.26 (Satz von McKlaughten)
Bross. Folgt aus Lemma 3.22.

Bross. Folgt aus Lemma 3.22.

Folgering 3.27

Die Kinnes der Bichli-erkensharen Sprachen ist unter Kumplement Algorichten.

Bross. Dier Gigende Transformationskrite.

NRA Fatz i = i DRA for i (genetid Stra 2.30)

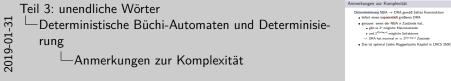
Konsequenz aus Safras Konstruktion

9:55

### Anmerkungen zur Komplexität

### Determinisierung NBA ightarrow DRA gemäß Safras Konstruktion

- liefert einen exponentiell größeren DRA
- genauer: wenn der NBA n Zustände hat,
  - gibt es 2<sup>n</sup> mögliche Makrozustände
  - und  $2^{O(n \log n)}$  mögliche Safrabäume
  - $\rightarrow$  DRA hat maximal  $m := 2^{O(n \log n)}$  Zustände
- Das ist optimal (siehe Roggenbachs Kapitel in LNCS 2500)



#### 9:57 bis Ende

**TODO:** Größe und Anzahl der Safra-Bäume sauber abschätzen & erklären! (siehe Folie  $\approx$  66)

Entscheidungsprobl. Model-Checking Büchi-Aut. Abschlusseig. Charakt. Determinismus

### Anmerkungen zur Komplexität

### **Determinisierung** NBA → DRA gemäß Safras Konstruktion

- liefert einen exponentiell größeren DRA
- genauer: wenn der NBA n Zustände hat,
  - gibt es 2<sup>n</sup> mögliche Makrozustände
  - und  $2^{O(n \log n)}$  mögliche Safrabäume
  - $\rightarrow$  DRA hat maximal  $m := 2^{O(n \log n)}$  Zustände
- Das ist optimal (siehe Roggenbachs Kapitel in LNCS 2500)

### Komplementierung beinhaltet auch den Schritt DMA $\rightarrow$ NBA

- liefert einen nochmal **exponentiell** größeren DBA: wenn der DMA *m* Zustände hat, hat der NBA  $O(m \cdot 2^m)$  Zustände
- $\rightarrow$  Resultierender NBA hat  $2^{2^{O(n^2)}}$  Zustände

Teil 3: unendliche Wörter Deterministische Büchi-Automaten und Determinisierung -Anmerkungen zur Komplexität

rminisierung NBA → DRA gemäß Safras Konstruktion

 eibt es 2º möeliche Makrozustände • und 2<sup>O(+log n)</sup> mögliche Safrabäume

Anmerkungen zur Komplexität

- On DRA hat maximal m := 20(nlegs) Zustände u Das ist optimal (siehe Roggenbachs Kapitel in LNCS 2500
  - Komplementierung beinhaltet auch den Schritt DMA → NBA
  - u liefert einen nochmal exponentiell größeren DBA wenn der DMA m Zustände hat, hat der NBA O(m · 2<sup>m</sup>) Zustände

9:57 bis Ende

TODO: Größe und Anzahl der Safra-Bäume sauber abschätzen & erklären! (siehe Folie  $\approx 66$ )

2019-01

### Anmerkungen zur Komplexität

### Determinisierung NBA ightarrow DRA gemäß Safras Konstruktion

- liefert einen exponentiell größeren DRA
- genauer: wenn der NBA n Zustände hat,
  - gibt es 2<sup>n</sup> mögliche Makrozustände
  - und  $2^{O(n \log n)}$  mögliche Safrabäume
  - $\rightarrow$  DRA hat maximal  $m := 2^{O(n \log n)}$  Zustände
- Das ist optimal (siehe Roggenbachs Kapitel in LNCS 2500)

#### Komplementierung beinhaltet auch den Schritt DMA $\rightarrow$ NBA

- liefert einen nochmal exponentiell größeren DBA: wenn der DMA m Zustände hat, hat der NBA  $O(m \cdot 2^m)$  Zustände
- $\rightarrow$  Resultierender NBA hat  $2^{2^{O(n^2)}}$  Zustände
  - Alternative Prozedur erfordert nur  $2^{O(n \log n)}$  Zustände

Teil 3: unendliche Wörter

Deterministische Büchi-Automaten und Determinisierung

Anmerkungen zur Komplexität

Determinishmen NRA — DIFA gendli Safras Komznáckou i John ci most opensomiel goldnen o serve opensomiel goldnen o serve opensomiel goldnen NRA zazladné lat.

\*\*A produce men de PRBA z zázladné lat.

\*\*\* and profession de produce de produce de la companya del companya del companya de la companya del companya del companya de la companya de la companya del compan

Alternative Prozedur erfordert nur 2<sup>O(n log n)</sup> Zustände

Anmerkungen zur Komplexität

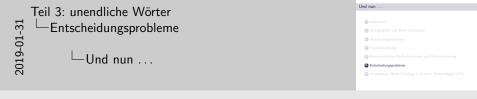
#### 9:57 bis Ende

2019-01

**TODO:** Größe und Anzahl der Safra-Bäume sauber abschätzen & erklären! (siehe Folie  $\approx$  66)

### Und nun ...

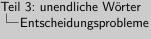
- Motivatio
- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschaften
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierun
- 6 Entscheidungsprobleme
- Anwendung: Model-Checking in Linearer Temporallogik (LTL



### Vorbetrachtungen

### Betrachten 4 Standardprobleme:

- Leerheitsproblem
- Wortproblem (Wort ist durch NBA gegeben)
- Äquivalenzproblem
- Universalitätsproblem



└─Vorbetrachtungen

trachtungen

Vorbetrachtungen

Betrachten 4 Standardprobleme:

u Leerheitsproblem
• Wortproblem (Wort ist durch NBA gegeben)
• Aquivalenzproblem
u Universalitätsproblem

16:00

### Vorbetrachtungen

### Betrachten 4 Standardprobleme:

- Leerheitsproblem
- Wortproblem (Wort ist durch NBA gegeben)
- Äquivalenzproblem
- Universalitätsproblem

### Beschränken uns auf das Leerheitsproblem – die anderen . . .

- lassen sich wie üblich darauf reduzieren
- aber teils mit (doppelt) exponentiellem "Blowup" (Determinisierung, Komplementierung, siehe Folie 80)

→ höhere Komplexität

Teil 3: unendliche Wörter

Entscheidungsprobleme

Entscheidungsprobleme

Vorpatien (für ist dech till geglen)

Apriliagestatien aus auf die Lenktrigestatien

Buschiebes aus auf die Lenktrigestatien - die anderen ...

Lusse ist der üblich dazuf radiatione

der unter in de (Special)

der unterscheide (Stemanisseriege, der Fein 80)

(Derminisseriege, furnisseriesege, sole Fein 80)

16:00

### Vorbetrachtungen

#### Betrachten 4 Standardprobleme:

- Leerheitsproblem
- Wortproblem (Wort ist durch NBA gegeben)
- Äquivalenzproblem
- Universalitätsproblem

Beschränken uns auf das Leerheitsproblem – die anderen . . .

- lassen sich wie üblich darauf reduzieren
- aber teils mit (doppelt) exponentiellem "Blowup" (Determinisierung, Komplementierung, siehe Folie 80)
   → höhere Komplexität

Beschränken uns auf NBA, aber Entscheidbarkeit überträgt sich auf die anderen Modelle

Teil 3: unendliche Wörter

Entscheidungsprobleme

Entscheidungsprobleme

Vorbetrachtungen

Vorbetrachtungen

Vorbetrachtungen

Vorbetrachtungen

Lundingsprobleme

Lundingsprobleme

Lundingsprobleme

Lundingsprobleme

Lundingsprobleme

Establishen uns all est Lundingsprobleme de andere ...

Lundingsproblemen de viole tille dieser de andere ...

Lundingsproblemen de state de landere ...

Lundingsprobl

16:00

2019-01

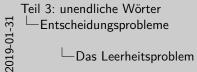
### Das Leerheitsproblem

**Zur Erinnerung:** 

Motiv.

Gegeben: NBA  ${\mathcal A}$ 

Frage: Gilt  $L_{\omega}(\mathcal{A}) = \emptyset$ ?



Zur Einzung: Gagben: Niñ A $\label{eq:continuous} \mbox{Gri $L_*(A) = \emptyset$} \mbox{7}$  Frage: Gri  $L_*(A) = \emptyset$  7

Das Leerheitsproblem

### Das Leerheitsproblem

**Zur Erinnerung:** 

Gegeben: NBA  ${\mathcal A}$ 

Frage: Gilt  $L_{\omega}(A) = \emptyset$ ?

Satz 3.28

Das Leerheitsproblem für NBAs ist entscheidbar.

Teil 3: unendliche Wörter
Entscheidungsprobleme

Das Leerheitsproblem

Das Leerheitsproblem

Zu Einnerung:
Gegeben: NRB. A
Frage: Git L.(A) = 0.7
Sarz 3.28

Das Leerheitsproblem für NRBA zit entscheidbar.

16:02

Motiv. Büchi-Aut.

Charakt.

t. Determinismus

Entscheidungsprobl.

Model-Checking

### Das Leerheitsproblem

**Zur Erinnerung:** 

Gegeben: NBA  ${\mathcal A}$ 

Frage: Gilt  $L_{\omega}(A) = \emptyset$ ?

Abschlusseig.

Satz 3.28

Das Leerheitsproblem für NBAs ist entscheidbar.

Quiz: Welche Komplexität hat es? NL ... P ... höher?

Teil 3: unendliche Wörter Entscheidungsprobleme

 $\sqsubseteq \mathsf{Das}\ \mathsf{Leerheitsproblem}$ 

Zur Erimorung:  $Gageben NBA \ A$  Frage Gift  $L_{c}(A)=\emptyset$  7 Satz 320 Das Leerheitspoolsem für NBAs ät entscheidbar. Quiz: Welche Komplexolit hat es? NL... P... höher?

Das Leerheitsproblem

16:02

### Das Leerheitsproblem

#### **Zur Erinnerung:**

Gegeben: NBA  ${\cal A}$ 

Frage: Gilt  $L_{\omega}(A) = \emptyset$ ?

#### Satz 3.28

Das Leerheitsproblem für NBAs ist entscheidbar.

Quiz: Welche Komplexität hat es? NL ... P ... höher?

Beweis.  $L_{\omega}(\mathcal{A}) \neq \emptyset$  genau dann, wenn gilt:

Es gibt  $q_0 \in I$  und  $q_f \in F$ und einen Pfad von  $q_0$  zu  $q_f$  in  $\mathcal{A}$ und einen Pfad von  $q_f$  zu  $q_f$  in  $\mathcal{A}$ 

⇒ Reduktion zum Leerheitsproblem für NEAs:

Teil 3: unendliche Wörter

Entscheidungsprobleme

Entscheidungsprobleme

Das Leerheitsproblem

Das Leerheitsproblem

Das Leerheitsproblem

Og: Wideh Komplanki ku entscheidur.

Das Leerheitsproblem

Browk: L,(d) y 8 gena dem, sen gib.
Es gitt, q, E f m od ean Flat on q, x n q, n A
od ean Flat on q, x n q, n A
on flatekon zum Leekhingsplank fin MEA:

16:02

Bezeichne  $L(A_{q_1,q_2})$  die von A als **NEA** erkannte Sprache, wenn  $\{q_1\}$  Anfangs- und  $\{q_2\}$  Endzustandsmenge ist

Charakt.

Folgender Algorithmus entscheidet das Leerheitsproblem:

Rate nichtdeterministisch  $q_0 \in I$  und  $q_F \in F$ if  $L(A_{q_0,q_f}) \subseteq \{\varepsilon\}$  oder  $L(A_{q_f,q_f}) \subseteq \{\varepsilon\}$  then return "leer" return ..nicht leer"

Dabei ist 
$$L(A_{...}) \subseteq \{\varepsilon\}$$
 gdw.  $L(A_{...}) \cap \underbrace{(\Sigma \setminus \{\varepsilon\})}_{\text{konst. NEA}} = \emptyset$ 

$$\left( , \mathcal{L}(\mathcal{A}_{...}) = \emptyset ^{"} \text{ genügt nicht, denn } \mathcal{L}_{\omega}( \longrightarrow \bigcirc) = \emptyset . 
ight)$$

Teil 3: unendliche Wörter Entscheidungsprobleme Das Leerheitsproblem Bezeichne  $L(A_{\alpha_1,\alpha_2})$  die von A als NEA erkannte Sprach wenn {a;} Anfangs- und {a;} Endzustandsmenge ist Folgender Algorithmus entscheidet das Leerheitsproblem Rate nichtdeterministisch  $q_0 \in I$  und  $q_F \in F$  $(L(A_{-}) = \emptyset^{+} \text{ genügt nicht, denn } L_{-}(\longrightarrow) = \emptyset.$ 

☐ Das Leerheitsproblem

16:05 bis 16:07

TODO: besser formatieren! → die 2 Zeilen unter der Box als Tafelanschrieb?

Abschlusseig.

Determinismus

Entscheidungsprobl.

# Das Leerheitsproblem

Bezeichne  $L(A_{q_1,q_2})$  die von A als **NEA** erkannte Sprache, wenn  $\{q_1\}$  Anfangs- und  $\{q_2\}$  Endzustandsmenge ist

Folgender Algorithmus entscheidet das Leerheitsproblem:

Rate nichtdeterministisch  $q_0 \in I$  und  $q_F \in F$ if  $L(A_{q_0,q_f}) \subseteq \{\varepsilon\}$  oder  $L(A_{q_f,q_f}) \subseteq \{\varepsilon\}$  then return "leer" return ..nicht leer"

Dabei ist 
$$L(\mathcal{A}_{...}) \subseteq \{\varepsilon\}$$
 gdw.  $L(\mathcal{A}_{...}) \cap \underbrace{(\Sigma \setminus \{\varepsilon\})}_{\text{konst. NEA}} = \emptyset$ 

$$(L(\mathcal{A}_{...}) = \emptyset$$
" genügt nicht, denn  $L_{\omega}(-) = \emptyset$ .

Das ist ein NL-Algorithmus (eigentlich coNL, aber NL = coNL ist bekannt, Immerman–Szelepcsényi 1987)

Leerheit für NBAs ist NL-vollständig

Teil 3: unendliche Wörter Entscheidungsprobleme

-Das Leerheitsproblem

Bezeichne  $L(A_{\alpha_1,\alpha_2})$  die von A als NEA erkannte Sprach wenn {a;} Anfangs- und {a;} Endzustandsmenge ist Folgender Algorithmus entscheidet das Leerheitsproblem Rate nichtdeterministisch  $q_0 \in I$  und  $q_F \in F$ if  $L(A_{\alpha_1,\alpha_2}) \subset \{\varepsilon\}$  oder  $L(A_{\alpha_2,\alpha_2}) \subset \{\varepsilon\}$  then return "leer  $(L(A_{-}) = \emptyset^{+} \text{ genügt nicht, denn } L_{-}(\longrightarrow) = \emptyset.$ 

Das Leerheitsproblem

Leerheit für NBAs ist NL-vollständig

16:05 bis 16:07

TODO: besser formatieren! → die 2 Zeilen unter der Box als Tafelanschrieb?

# Überblick Entscheidungsprobleme für NBAs

Problem	entscheidbar?	Komplexität	effizient lösbar?
LP	✓	<b>NL</b> -vollständig	✓
WP	— macht keinen Sinn, da Eingabewort $\infty$ —		
ÄP	✓	PSpace-vollst.	<b>×</b> *
UP	✓	PSpace-vollst.	<b>X</b> *

<sup>\*</sup> unter den üblichen komplexitätstheoretischen Annahmen (z. B. PSpace ≠ P)

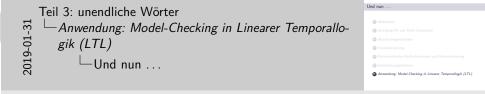
# Teil 3: unendliche Wörter Entscheidungsprobleme Diberblick Entscheidungsprobleme für NBAs

#### 16:07

Hier nochmal für Euch als Überblick. Situation dieselbe wie für NEAs; nur WP macht hier keinen Sinn.

### Und nun ...

- Motivation
- 2 Grundbegriffe und Büchi-Automaten
- 3 Abschlusseigenschafter
- 4 Charakterisierung
- 5 Deterministische Büchi-Automaten und Determinisierung
- 6 Entscheidungsprobleme
- Anwendung: Model-Checking in Linearer Temporallogik (LTL)



Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

### Reaktive Systeme und Verifikation

#### **Reaktive Systeme**

- interagieren mit ihrer Umwelt
- terminieren oft nicht
- Beispiele:
  - Betriebssysteme, Bankautomaten, Flugsicherungssysteme, ...
  - s. a. Philosophenproblem, Konsument-Produzent-Problem



16:08

2019-01

Model-Checking

Abschlusseig.

Charakt. Determinismus Entscheidungsprobl.

### Reaktive Systeme und Verifikation

#### **Reaktive Systeme**

- interagieren mit ihrer Umwelt
- terminieren oft nicht
- Beispiele:
  - Betriebssysteme, Bankautomaten, Flugsicherungssysteme, ...
  - s. a. Philosophenproblem, Konsument-Produzent-Problem

#### **Verifikation** = Prüfen von Eigenschaften eines Systems

- Eingabe-Ausgabe-Verhalten hat hier keine Bedeutung
- Andere Eigenschaften sind wichtig,
  - z. B.: keine Verklemmung (deadlock) bei Nebenläufigkeit

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Reaktive Systeme und Verifikation

16:08

2019-01

Reaktive Systeme

- a interagieren mit ihrer Umwelt
- . . . Ohilassahannahlam Wassamant Dankamat Danklas
- Verifikation = Prüfen von Eigenschaften eines Systems
- w Eingabe-Ausgabe-Verhalten hat hier keine Bedeutun
  - z. B.: keine Verklemmung (deadlock) bei Nebenläufigkei

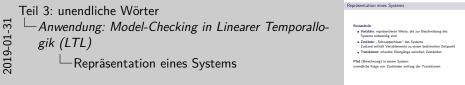
otiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

### Repräsentation eines Systems

#### Bestandteile

- Variablen: repräsentieren Werte, die zur Beschreibung des Systems notwendig sind
- Zustände: "Schnappschüsse" des Systems
   Zustand enthält Variablenwerte zu einem bestimmten Zeitpunkt
- Transitionen: erlaubte Übergänge zwischen Zuständen

Pfad (Berechnung) in einem System: unendliche Folge von Zuständen entlang der Transitionen



Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

#### Transitionsgraph als Kripke-Struktur\*

#### Definition 3.29

Sei AV eine Menge von Aussagenvariablen. Eine Kripke-Struktur  $\mathcal{S}$  über AV ist ein Quadrupel  $\mathcal{S}=(S,S_0,R,\ell)$ , wobei

- S eine endliche nichtleere Menge von Zuständen ist,
- $S_0 \subseteq S$  die Menge der Anfangszustände ist,

Model-Checking

Teil 3: unendliche Wörter

- Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Transitionsgraph als Kripke-Struktur\*

<sup>\*</sup> Saul Kripke, geb. 1940, Philosoph und Logiker, Princeton und New York, USA

Büchi-Aut. Abschlusseig. Determinismus

#### Transitionsgraph als Kripke-Struktur\*

#### Definition 3.29

Sei AV eine Menge von Aussagenvariablen. Eine Kripke-Struktur  ${\mathcal S}$ über AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobei

- S eine endliche nichtleere Menge von Zuständen ist,
- $S_0 \subset S$  die Menge der Anfangszustände ist,
- $R \subset S \times S$  eine Übergangsrelation ist, die total ist:  $\forall s \in S \ \exists s' \in S : sRs'$

Fransitionsgraph als Kripke-Struktur\* Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallo-

gik (LTL)

-Transitionsgraph als Kripke-Struktur\*

ii AV eine Menze von Aussagenvariablen. Eine Kripke-Struktur J iber AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobe

<sup>\*</sup> Saul Kripke, geb. 1940, Philosoph und Logiker, Princeton und New York, USA

Büchi-Aut. Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Transitionsgraph als Kripke-Struktur\*

#### Definition 3.29

Sei AV eine Menge von Aussagenvariablen. Eine Kripke-Struktur  ${\mathcal S}$ über AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobei

- S eine endliche nichtleere Menge von Zuständen ist,
- $S_0 \subset S$  die Menge der Anfangszustände ist,
- $R \subset S \times S$  eine Übergangsrelation ist, die total ist:  $\forall s \in S \ \exists s' \in S : sRs'$
- $\ell: S \to 2^{AV}$  eine Funktion ist, die Markierungsfunktion.  $\ell(s) = \{p_1, \dots, p_m\}$  bedeutet: in s sind genau  $p_1, \dots, p_m$  wahr

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Teil 3: unendliche Wörter

-Transitionsgraph als Kripke-Struktur\*

Transitionsgraph als Kripke-Struktur\* ii AV eine Menze von Aussagenvariablen. Eine Kripke-Struktur J

iber AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobe

 ℓ: S → 2<sup>RV</sup> eine Funktion ist, die Markierungsfunktion  $f(x) = \{p_1, \dots, p_m\}$  bedeutet: in x sind genau  $p_1, \dots, p_m$  wahr

<sup>\*</sup> Saul Kripke, geb. 1940, Philosoph und Logiker, Princeton und New York, USA

#### Definition 3.29

Sei AV eine Menge von Aussagenvariablen. Eine Kripke-Struktur  ${\mathcal S}$ über AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobei

- S eine endliche nichtleere Menge von Zuständen ist,
- $S_0 \subset S$  die Menge der Anfangszustände ist,
- $R \subseteq S \times S$  eine Übergangsrelation ist. die total ist:  $\forall s \in S \ \exists s' \in S : sRs'$
- $\ell: S \to 2^{AV}$  eine Funktion ist, die Markierungsfunktion.  $\ell(s) = \{p_1, \dots, p_m\}$  bedeutet: in s sind genau  $p_1, \dots, p_m$  wahr

Ein **Pfad** in S ist eine unendliche Folge  $\pi = s_0 s_1 s_2 \dots$  von Zuständen mit  $s_0 \in S_0$  und  $s_i R s_{i+1}$  für alle  $i \ge 0$ .

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Transitionsgraph als Kripke-Struktur\*

ii AV eine Menze von Aussagenvariablen. Eine Kripke-Struktur J

 $a : S \rightarrow 2^{RV}$  eine Funktion ist, die Markierungsfunktion  $\ell(s) = \{p_1, \dots, p_m\}$  bedeutet: in s sind genau  $p_1, \dots, p_m$  wahr

Transitionsgraph als Kripke-Struktur\*

über AV ist ein Quadrupel  $S = (S, S_0, R, \ell)$ , wobe a S eine endliche nichtleere Menze von Zuständen ist. u S₁ ⊂ S die Menze der Anfangszustände ist.

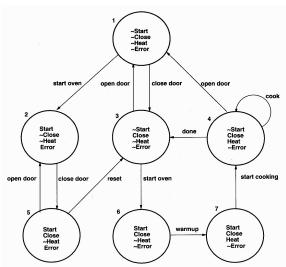
Ein Pfad in S ist eine unendliche Folge  $\pi = s_0 s_1 s_2$  .

Zuständen mit  $s_0 \in S_n$  und  $s_i R s_{i+1}$  für alle  $i \ge 0$ .

<sup>\*</sup> Saul Kripke, geb. 1940, Philosoph und Logiker, Princeton und New York, USA

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Beispiel 1: Mikrowelle



aus: E. M. Clarke et al., Model Checking, MIT Press 1999

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Beispiel 1: Mikrowelle



16:14

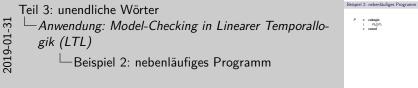
2019-01

**TODO:** Mikrowellenbeispielbild selber tikzen; vernünftige Bezeichnungen ( $\neg$  statt  $\sim$ , "offen" statt "Close", keine Pfeilbeschriftungen usw.)

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

# Beispiel 2: nebenläufiges Programm

0 cobegin 1  $P_0||P_1$ 2 coend



Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Beispiel 2: nebenläufiges Programm

```
\begin{array}{cccc} P & 0 & \text{cobegin} \\ & 1 & P_0 \| P_1 \\ & 2 & \text{coend} \end{array} \begin{array}{cccc} P_0 & 10 & \text{while(true) do} \\ & 11 & \text{wait(turn} = 0) \\ & 12 & \text{turn} \leftarrow 1 \\ & 13 & \text{end while} \end{array}
```



16:17

2019-01

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Beispiel 2: nebenläufiges Programm

```
cobegin
             P_0 || P_1
          coend
          while(true) do
P_0
             wait(turn = 0)
                                 kritischer Bereich
             turn \leftarrow 1
          end while
```



P<sub>0</sub> 10 while(true) do 11 wait(turn = 0)
12 turn ← 1
13 end while

16:17

2019-01

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

### Beispiel 2: nebenläufiges Programm

```
cobegin
             P_0 || P_1
           coend
          while(true) do
P_0
             wait(turn = 0)
                                 kritischer Bereich
             turn \leftarrow 1
          end while
P_1
           while(true) do
             wait(turn = 1)
             turn \leftarrow 0
           end while
```



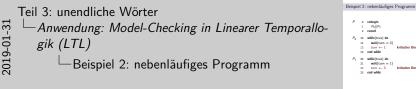
16:17

Model-Checking

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

## Beispiel 2: nebenläufiges Programm

```
cobegin
             P_0 || P_1
          coend
          while(true) do
P_0
             wait(turn = 0)
                                kritischer Bereich
             turn \leftarrow 1
          end while
          while(true) do
             wait(turn = 1)
             turn \leftarrow 0
                                kritischer Bereich
          end while
```



16:17

Model-Checking

Abschlusseig.

Charakt.

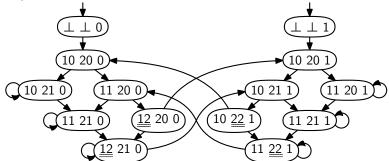
Determinismus

## Beispiel 2: nebenläufiges Programm

Variablen in der zugehörigen Kripke-Struktur:  $v_1$ ,  $v_2$ ,  $v_3$  mit

- $v_1, v_2$ : Werte der Programmzähler für  $P_0, P_1$ (einschl. ⊥: Teilprogramm ist nicht aktiv)
- v<sub>3</sub>: Werte der gemeinsamen Variable turn

Kripke-Struktur:



Beispiel 2: nebenläufiges Programm Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Beispiel 2: nebenläufiges Programm

#### 16:20

Hier benutzen wir für jede Programmzeile eine getrennte Aussagenvariable (auch wenn wir die wenigen möglichen Zustände mit weniger AVs kodieren könnten).

**TODO:** Variablen sind falsch beschrieben. Es gibt nicht nur  $v_1, v_2, v_3$ , sondern 10, 11, 12, 13, 20, 21, 22, 23,  $\perp_0$ ,  $\perp_1$ ,  $t_0$ ,  $t_1$  (letztere für "turn"). → Beschreibung und Bild anpassen

Büchi-Aut. Abschlusseig. Charakt. Entscheidungsprobl. Model-Checking Determinismus

### Spezifikationen

... sind Zusicherungen über die Eigenschaften eines Systems, z. B.:

- "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."
- "Wenn die Mikrowelle gestartet wird, fängt sie immer nach endlicher Zeit an zu heizen."
- "Wenn die Mikrowelle gestartet wird, ist es möglich, danach zu heizen."

Teil 3: unendliche Wörter 2019-01-31 Anwendung: Model-Checking in Linearer Temporallogik (LTL) Spezifikationen

- sind Zusicherungen über die Eigenschaften eines Systems. z. B. · Wonn ein Fehler auftritt ist er nach endlicher Zeit hehnher w "Wenn die Mikrowelle gestartet wird,
- fänet sie immer nach endlicher Zeit an zu heizen.
- .Wenn die Mikrowelle gestartet wird,

Abschlusseig. Model-Checking Büchi-Aut. Charakt. Determinismus Entscheidungsprobl.

### Spezifikationen

... sind Zusicherungen über die Eigenschaften eines Systems, z. B.:

- "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."
- "Wenn die Mikrowelle gestartet wird, fängt sie immer nach endlicher Zeit an zu heizen."
- "Wenn die Mikrowelle gestartet wird, ist es möglich, danach zu heizen."
- "Es kommt nie vor, dass beide Teilprogramme zugleich im kritischen Bereich sind."
- "Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich."
- "Jedes Teilprogramm kann beliebig oft in seinen kritischen Bereich gelangen."

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallogik (LTL) Spezifikationen

Spezifikationen

- sind Zusicherungen über die Eigenschaften eines Systems. z. B. · Wonn ein Fehler auftritt ist er nach endlicher Zeit hehnher
- w "Wenn die Mikrowelle gestartet wird, fänet sie immer nach endlicher Zeit an zu heizen.
- .Wenn die Mikrowelle gestartet wird,
- ist es möglich, danach zu heizen a .Es kommt nie vor
- dass beide Teilprogramme zugleich im kritischen Bereich sind u "Jedes Teilprog, kommt beliebig oft in seinen krit. Bereich: "Jedes Teilprogramm kann beliebig oft in seinen kritischer

16:24

2019-01-31

Model-Checking Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

### Spezifikationen

... sind Zusicherungen über die Eigenschaften eines Systems, z. B.:

- "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."
- "Wenn die Mikrowelle gestartet wird, fängt sie immer nach endlicher Zeit an zu heizen."
- "Wenn die Mikrowelle gestartet wird, ist es möglich, danach zu heizen."
- "Es kommt nie vor, dass beide Teilprogramme zugleich im kritischen Bereich sind."
- "Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich."
- "Jedes Teilprogramm kann beliebig oft in seinen kritischen Bereich gelangen."

• . . .

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) Spezifikationen

Spezifikationen

- sind Zusicherungen über die Eigenschaften eines Systems. z. B. · Wonn ein Fehler auftritt ist er nach endlicher Zeit hehnher
- w "Wenn die Mikrowelle gestartet wird,
- fänet sie immer nach endlicher Zeit an zu heizen. .Wenn die Mikrowelle gestartet wird,
- ist es mórfich danach zu heizen
- a .Es kommt nie vor
- dass beide Teilprogramme zugleich im kritischen Bereich sind u "Jedes Teilprog, kommt beliebig oft in seinen krit. Bereich: "Jedes Teilprogramm kann beliebig oft in seinen kritischer

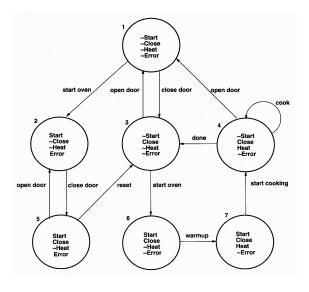
Abschlusseig.

Determinismus

Entscheidungsprobl.

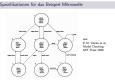
Model-Checking

### Spezifikationen für das Beispiel Mikrowelle



aus: E. M. Clarke et al., Model Checking, MIT Press 1999

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) Spezifikationen für das Beispiel Mikrowelle



16:26

2019-01

Jeweils fragen: Ist die Zusicherung erfüllt?

Abschlusseig.

start over

close door

Start

~Close ~Heat

Error

open door

close door

~Start Close

~Error

Start Close

start cooking

open door

~Start Close ~Heat ~Error

Start Close

~Error

aus: E. M. Clarke et al., Model Checking,

MIT Press 1999

"Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

Teil 3: unendliche Wörter

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Spezifikationen für das Beispiel Mikrowelle

Spezifikationen für das Beispiel Mikrowelle

16:26

2019-01

Jeweils fragen: Ist die Zusicherung erfüllt?

Vorsicht: Es kommt darauf an, ob gemeint ist "in jedem Lauf" oder "es gibt einen Lauf". (univ. vs. exist. Model-Checking) – Diskutieren!

Start Close

Error

Büchi-Aut. Abschlusseig.

Spezifikationen für das Beispiel Mikrowelle

Determinismus

E. M. Clarke et al.,

Model Checking,

MIT Press 1999

start over

close door

Start

~Close ~Heat

Error

open door

close door

aus:

open door

~Start Close ~Heat ~Error

Start Close

~Error

~Start Close

~Error

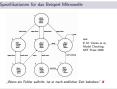
Start Close

start cooking

gik (LTL)

Spezifikationen für das Beispiel Mikrowelle

-Anwendung: Model-Checking in Linearer Temporallo-



16:26

Jeweils fragen: Ist die Zusicherung erfüllt?

Vorsicht: Es kommt darauf an, ob gemeint ist "in jedem Lauf" oder "es gibt einen Lauf". (univ. vs. exist. Model-Checking) – Diskutieren!



Start Close

Error

start over

close door

Start

~Close ~Heat

Error

open door

close door

~Start Close

~Error

Start Close

start cooking

open door

~Start

Close ~Heat ~Error

Start Close

~Error

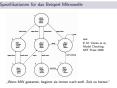
aus: E. M. Clarke et al.,

Model Checking, MIT Press 1999

"Wenn MW gestartet, beginnt sie immer nach endl. Zeit zu heizen."

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-

gik (LTL) Spezifikationen für das Beispiel Mikrowelle



16:26

2019-01

Jeweils fragen: Ist die Zusicherung erfüllt?

Vorsicht: Es kommt darauf an, ob gemeint ist "in jedem Lauf" oder "es gibt einen Lauf". (univ. vs. exist. Model-Checking) – Diskutieren!

Start Close

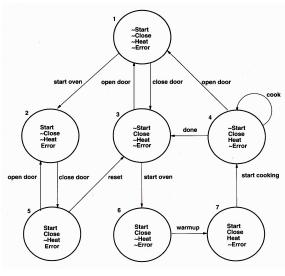
Error

Büchi-Aut. Abschlusseig.

Determinismus

Entscheidungsprobl.

2019-01

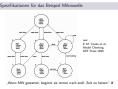


aus: E. M. Clarke et al., Model Checking, MIT Press 1999

"Wenn MW gestartet, beginnt sie immer nach endl. Zeit zu heizen." 🗶

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Spezifikationen für das Beispiel Mikrowelle

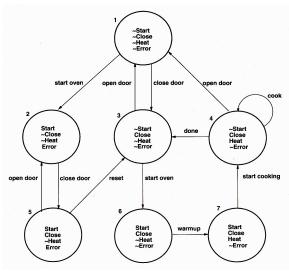


16:26

Jeweils fragen: Ist die Zusicherung erfüllt?

Determinismus

Model-Checking



aus: E. M. Clarke et al., Model Checking, MIT Press 1999

"Wenn MW gestartet, ist es möglich, danach zu heizen."

Teil 3: unendliche Wörter

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Spezifikationen für das Beispiel Mikrowelle

Spezifikationen für das Beispiel Mikrowelle

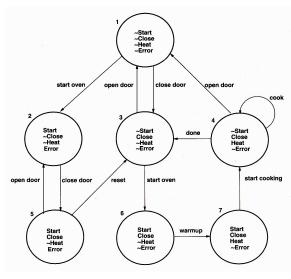
16:26

2019-01

Jeweils fragen: Ist die Zusicherung erfüllt?

Determinismus

# Spezifikationen für das Beispiel Mikrowelle



aus: E. M. Clarke et al., Model Checking, MIT Press 1999

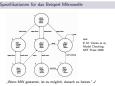
"Wenn MW gestartet, ist es *möglich*, danach zu heizen." 🗸

2019-01

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Spezifikationen für das Beispiel Mikrowelle

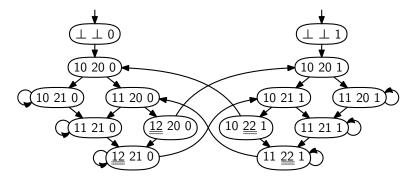


16:26

Jeweils fragen: Ist die Zusicherung erfüllt?

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Spezifikationen für das Beispiel Nebenläufigkeit



Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Spezifikationen für das Beispiel Nebenläufigkeit

16:30

Jeweils wieder fragen ...

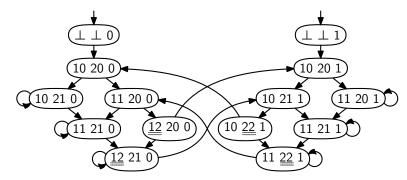
Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl

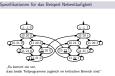
# Spezifikationen für das Beispiel Nebenläufigkeit



"Es kommt nie vor, dass beide Teilprogramme zugleich im kritischen Bereich sind." Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



16:30

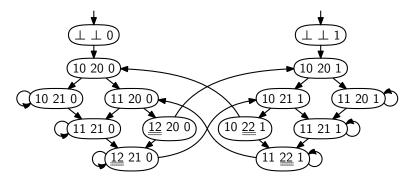
Jeweils wieder fragen . . .

Abschlusseig.

Charakt.

Determinismus

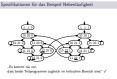
# Spezifikationen für das Beispiel Nebenläufigkeit



"Es kommt nie vor, dass beide Teilprogramme zugleich im kritischen Bereich sind." 🗸 Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



16:30

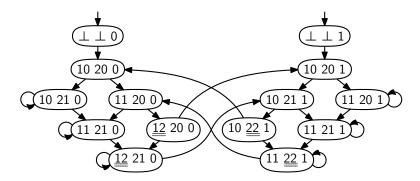
Jeweils wieder fragen . . .

Abschlusseig.

Charakt. Determinismus Entscheidungsprobl

Model-Checking

# Spezifikationen für das Beispiel Nebenläufigkeit



"Jedes P<sub>i</sub> kommt beliebig oft in seinen kritischen Bereich."

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



16:30

Jeweils wieder fragen . . .

Abschlusseig.

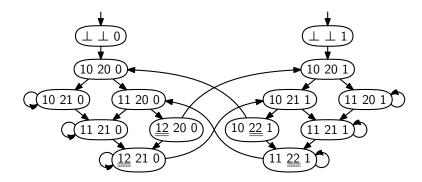
Charakt.

Determinismus

Entscheidungsprobl

Model-Checking

# Spezifikationen für das Beispiel Nebenläufigkeit

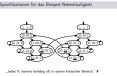


"Jedes  $P_i$  kommt beliebig oft in seinen kritischen Bereich." X

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



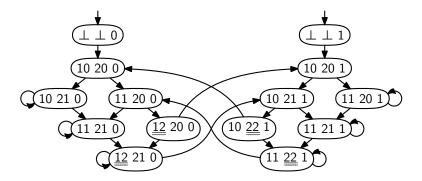
16:30

Jeweils wieder fragen . . .

Büchi-Aut. Abschlusseig.

Determinismus

# Spezifikationen für das Beispiel Nebenläufigkeit

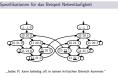


"Jedes P<sub>i</sub> kann beliebig oft in seinen kritischen Bereich kommen."

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



16:30

Jeweils wieder fragen . . .

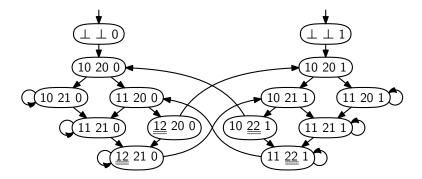
Büchi-Aut. Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl

# Spezifikationen für das Beispiel Nebenläufigkeit

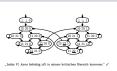


"Jedes  $P_i$  kann beliebig oft in seinen kritischen Bereich kommen."  $\checkmark$ 

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Spezifikationen für das Beispiel Nebenläufigkeit



Spezifikationen für das Beispiel Nebenläufigkeit

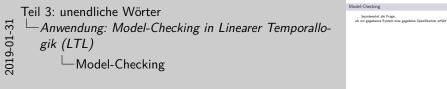
16:30

Jeweils wieder fragen . . .

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

## Model-Checking

... beantwortet die Frage, ob ein gegebene System eine gegebene Spezifikation erfüllt



Büchi-Aut. Charakt. Determinismus Model-Checking Abschlusseig. Entscheidungsprobl.

### Model-Checking

beantwortet die Frage, ob ein gegebenes System eine gegebene Spezifikation erfüllt

#### Definition 3.30 (Model-Checking-Problem MCP)

Gegeben ein System S und eine Spezifikation E,

- gilt E für jeden Pfad in S? (universelle Variante)
- qibt es einen Pfad in S, der E erfüllt? (existenzielle Variante)

Model-Checking Teil 3: unendliche Wörter ob ein gegebenes System eine gegebene Spezifikation erfüllt -Anwendung: Model-Checking in Linearer Temporallogik (LTL) a silt E für ieden Pfad in S? (universelle Variante) u gibt es einen Pfad in S, der E erfüllt? (existenzielle Variante) -Model-Checking

2019-01

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

## Model-Checking

... beantwortet die Frage, ob ein gegebene System eine gegebene Spezifikation erfüllt

#### Definition 3.30 (Model-Checking-Problem MCP)

Gegeben ein System  ${\mathcal S}$  und eine Spezifikation  ${\mathcal E}$ ,

- gilt E für jeden Pfad in S? (universelle Variante)
- gibt es einen Pfad in S, der E erfüllt? (existenzielle Variante)

Frage: Wie kann man Model-Checking

- exakt beschreiben und
- algorithmisch lösen?

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogisk (LTL)

Ballow (LTL)

Model-Checking

Ballow (LTL)

Model-Checking

Fig. We kan man Model Checking

\*\*expective to the graph of the first of the control of the control

16:33

2019-01-

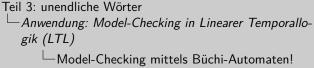
Model-Checking

iv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking Tail 2: uppendliche Wärter

# Model-Checking mittels Büchi-Automaten!

#### Schritt 1

- $\bullet$  Stellen System  ${\cal S}$  als NBA  ${\cal A}_{\cal S}$  dar
  - $\leadsto$  Pfade in  ${\mathcal S}$  sind erfolgreiche Runs von  ${\mathcal A}_{\mathcal S}$



Schritt 1

• Stellen System S als NBA  $A_S$  dar  $\rightarrow$  Pfade in S sind erfolgreiche Runs von  $A_S$ 

16:35

2019-01

Nun zunächst Schritt 1.

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# Model-Checking mittels Büchi-Automaten!

#### Schritt 1

- Stellen System S als NBA  $A_S$  dar  $\sim$  Pfade in S sind erfolgreiche Runs von  $A_S$
- Stellen Spezifikation E als NBA  $A_E$  dar  $\rightarrow$   $A_E$  beschreibt die Pfade, die E erfüllen

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

Model-Checking mittels Büchi-Automaten!

Model-Checking mittels Büchi-Automaten!

 $_{ullet}$  Stellen System  ${\cal S}$  als NBA  ${\cal A}_{<}$  dar

→ Pfade in S sind erfolgreiche Runs von A.
■ Stellen Sozzifikation E als NBA A.r. dar

16:35

2019-01

Nun zunächst Schritt 1.

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## Model-Checking mittels Büchi-Automaten!

#### Schritt 1

- Stellen System S als NBA  $A_S$  dar  $\rightarrow$  Pfade in S sind erfolgreiche Runs von  $A_S$
- Stellen Spezifikation E als NBA  $A_F$  dar  $\rightarrow$   $A_E$  beschreibt die Pfade, die E erfüllen
- $\rightarrow$  Universelles MCP =  $_{n}L(A_{S}) \subset L(A_{E})$ ?" Existenzielles MCP =  $L(A_S) \cap L(A_E) \neq \emptyset$ ?" (beide reduzierbar zum Leerheitsproblem, benutzt Abschlusseigenschaften)

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -Model-Checking mittels Büchi-Automaten!

16:35

Nun zunächst Schritt 1.

Model-Checking mittels Büchi-Automaten!

- $_{ullet}$  Stellen System  ${\cal S}$  als NBA  ${\cal A}_{<}$  dar
- → Pfade in S sind erfolgreiche Runs von A. Stellen Spezifikation E als NBA Ar dar
- ~ Ag beschreibt die Pfade, die E erfüllen

Motiv. Büchi-Aut. Abschlusseig.

ig. Charakt.

## Schritt 1

- Stellen System S als NBA  $A_S$  dar  $\sim$  Pfade in S sind erfolgreiche Runs von  $A_S$
- Stellen Spezifikation E als NBA  $A_E$  dar  $\rightarrow$   $A_E$  beschreibt die Pfade, die E erfüllen
- Universelles MCP = " $L(A_S) \subseteq L(A_E)$ ?"

  Existenzielles MCP = " $L(A_S) \cap L(A_E) \neq \emptyset$ ?"

  (beide reduzierbar zum Leerheitsproblem, benutzt Abschlusseigenschaften)

# Schritt 2

- intuitivere Beschreibung von E mittels Temporallogik
- Umwandlung von Temporallogik-Formel  $\varphi_F$  in Automaten  $\mathcal{A}_F$

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallo-

gik (LTL)

-- Model-Checking mittels Büchi-Automaten!

■ Stellen Spezifikation E als NBA  $A_E$  dar  $\sim A_E$  beschreibt die Pfade, die E erfüllen  $\sim$  Universelles MCP =  ${}_{-}L(A_E) \subseteq L(A_E)$ ?\*

Existenzielles MCP =  ${}_{-}L(A_E) \cap L(A_E) \neq \emptyset$ ?\*

itt 2

Model-Checking mittels Büchi-Automaten!

~ Pfade in S sind erfolgreiche Runs von A.

 $_{ullet}$  Stellen System  ${\cal S}$  als NBA  ${\cal A}_{<}$  dar

a intuitivere Beschreibung von E mittels Temporallogik u Umwandlung von Temporallogik-Formel  $\varphi_E$  in Automaten  $A_\delta$ 

16:35

2019-01

Nun zunächst Schritt 1.

tiv. Büchi-Aut. Abschl

sseig. Cl

Determinismus

Entscheidungsprobl.

. Model-Checking

# Konstruktion des NBA $\mathcal{A}_{\mathcal{S}}$ für das System $\mathcal{S}$

Erinnerung: S gegeben als Kripke-Struktur  $S = (S, S_0, R, \ell)$  (Zustände, Anfangszustände, Transitionen, Markierungen)

Teil 3: unendliche Wörter

Etnommer S. grephen Ale Kriph Straktur S = (5. St.

-Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Konstruktion des NBA  $\mathcal{A}_{\mathcal{S}}$  für das System  $\mathcal{S}$ 

#### 16:37

2019-01

Kripke-Struktur in Automaten umwandeln ist ganz einfach: im Prinzip ist die KS bereits der Automat.

Konstruktion des NBA  $\mathcal{A}_{\mathcal{S}}$  für das System  $\mathcal{S}$ 

# Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Konstruktion des NBA  $\mathcal{A}_{\mathcal{S}}$  für das System  $\mathcal{S}$ 

 $Q = S \boxtimes \{q_1\}$  $a = \{a_0\}$  $\mathbf{u} F = Q$  $\cup \{(s, t(s'), s') \mid (s, s') \in R\}$ 

**Erinnerung:** S gegeben als Kripke-Struktur  $S = (S, S_0, R, \ell)$ (Zustände, Anfangszustände, Transitionen, Markierungen)

**Zugehöriger Automat**  $A_S = (Q, \Sigma, \Delta, I, F)$ :

- $\Sigma = 2^{AV}$
- $Q = S \oplus \{q_0\}$
- $I = \{q_0\}$
- $\bullet$  F = Q
- $\Delta = \{ (q_0, \ell(s), s) \mid s \in S_0 \}$  $\cup \{ (s, \ell(s'), s') \mid (s, s') \in R \}$

16:37

2019-01

Kripke-Struktur in Automaten umwandeln ist ganz einfach: im Prinzip ist die KS bereits der Automat.

# Konstruktion des NBA $\mathcal{A}_{\mathcal{S}}$ für das System $\mathcal{S}$

**Erinnerung:** S gegeben als Kripke-Struktur  $S = (S, S_0, R, \ell)$ (Zustände, Anfangszustände, Transitionen, Markierungen)

**Zugehöriger Automat**  $A_S = (Q, \Sigma, \Delta, I, F)$ :

• 
$$\Sigma = 2^{AV}$$

$$Q = S \uplus \{q_0\}$$

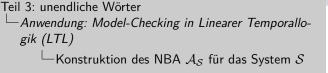
• 
$$I = \{q_0\}$$

$$\bullet$$
  $F = Q$ 

• 
$$\Delta = \{ (q_0, \ell(s), s) \mid s \in S_0 \}$$
  
 $\cup \{ (s, \ell(s'), s') \mid (s, s') \in R \}$ 

Beispiel: siehe Tafel.

T 3.18



Konstruktion des NBA  $A_c$  für das System S $Q = S \boxtimes \{q_1\}$  $a = \{q_0\}$  $\mathbf{u} F = Q$  $\bullet \Delta = \{(q_0, \ell(s), s) | s \in S_0\}$  $\cup \{ (s, \ell(s'), s') \mid (s, s') \in R \}$ 

16:37

2019-01

Kripke-Struktur in Automaten umwandeln ist ganz einfach: im Prinzip ist die KS bereits der Automat.

# Beschreibung von E durch NBA $A_E$

# Beispiel Mikrowelle (siehe Bild auf Folie 90)

"Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

Determinismus

Entscheidungsprobl.

- (b) "Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen."
- "Wenn die Mikrowelle gestartet wird, ist es möglich, danach zu heizen."

## Beispiel Nebenläufigkeit (siehe Bild auf Folie 92)

- (d) "Es kommt nie vor, dass beide Teilprog. zugleich im kritischen Bereich sind."
- (e) "Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich."
- "Jedes Teilprogramm kann beliebig oft in seinen kritischen Bereich gelangen."

T 3.19

Model-Checking

2019-01

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Beschreibung von E durch NBA  $\mathcal{A}_E$ 

16:43 bis 16:55 und 5min Pause

- a) "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behober
  - b) "Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen
- (c) .Wenn die Mikrowelle gestartet wird ist es möglich, danach zu heizen."
- Beispiel Nebenläufigkeit (siehe Bild auf Folie 92
- dass beide Teilprog, zugleich im kritischen Bereich sind.
- (e) \_Jedes Teilprog, kommt beliebig oft in seinen krit. Bereich.

Motiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus

# Verifikation mittels der konstruierten NBAs

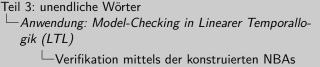
Gegeben sind wieder System  $\mathcal S$  und Spezifikation E.

Entscheidungsprobl.

Model-Checking

#### Universelles MCP

• Gilt E für jeden Pfad in S?



Verifikation mittels der konstruierten NBAs
Gegeben sind wieder System S und Spezifikation E.
Universaltes MCP

Gilt E für jeden Pfad in S?

#### 17:00

9

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen **doppelt** exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren, während man den Algo. für das Leerheitsproblem laufen lässt.

Charakt.

Determinismus

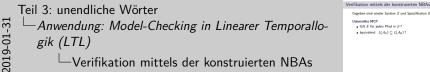
Entscheidungsprobl.

# Verifikation mittels der konstruierten NBAs

Gegeben sind wieder System S und Spezifikation E.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subseteq L(A_E)$ ?



#### 17:00

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Gilt E für jeden Pfad in S? • äquivalent:  $L(A_S) \subset L(A_E)$ 

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren, während man den Algo. für das Leerheitsproblem laufen lässt. Büchi-Aut.

Abschlusseig.

Determinismus

Entscheidungsprobl.

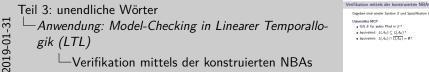
# Verifikation mittels der konstruierten NBAs

Gegeben sind wieder System S und Spezifikation E.

Charakt.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subseteq L(A_E)$ ?
- äquivalent:  $L(A_S) \cap \overline{L(A_E)} = \emptyset$ ?



#### 17:00

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Gilt E für jeden Pfad in S?

u ăquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren, während man den Algo. für das Leerheitsproblem laufen lässt. Büchi-Aut

Abschlusseig.

Charakt. Determinismus

Entscheidungsprobl.

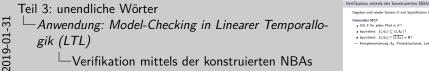
Model-Checking

# Verifikation mittels der konstruierten NBAs

Gegeben sind wieder System S und Spezifikation E.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subseteq L(A_E)$ ?
- äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?
- $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem



#### 17:00

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo. für das Leerheitsproblem laufen lässt.

Charakt.

Determinismus

Gegeben sind wieder System S und Spezifikation E.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subset L(A_E)$ ?
- äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?
- $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem
  - Komplexität: PSpace (exponentielle Explosion bei Komplementierung)

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Verifikation mittels der konstruierten NBAs

Gilt E für jeden Pfad in S?

- Verifikation mittels der konstruierten NBAs
- āquivalent: L(A<sub>S</sub>) ⊂ L(A<sub>E</sub>)?

#### 17:00

2019-01

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo. für das Leerheitsproblem laufen lässt. Büchi-Aut

Universelles MCP

Charakt.

Determinismus

Entscheidungsprobl.

Gegeben sind wieder System S und Spezifikation E.

• Gilt E für jeden Pfad in S?

• äquivalent:  $L(A_S) \subset L(A_E)$ ?

• äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?

• Gibt es einen Pfad in S. der E erfüllt?

 $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem

• Komplexität: PSpace (exponentielle Explosion bei Komplementierung)

Model-Checking Teil 3: unendliche Wörter

2019-01

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Verifikation mittels der konstruierten NBAs

u Gibt es einen Pfad in S, der E erfüllt?

Verifikation mittels der konstruierten NBAs

Gilt E für jeden Pfad in S? āquivalent: L(A<sub>S</sub>) ⊂ L(A<sub>E</sub>)?

#### 17:00

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo, für das Leerheitsproblem laufen lässt.

Existenzielles MCP

# Verifikation mittels der konstruierten NBAs

Gegeben sind wieder System S und Spezifikation E.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subset L(A_E)$ ?
- äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?
- $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem
  - Komplexität: PSpace (exponentielle Explosion bei Komplementierung)

#### Existenzielles MCP

- Gibt es einen Pfad in S. der E erfüllt?
- äquivalent:  $L(A_S) \cap L(A_E) \neq \emptyset$ ?

Verifikation mittels der konstruierten NBAs Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallo-Gilt E für jeden Pfad in S? āquivalent: L(A<sub>S</sub>) ⊂ L(A<sub>E</sub>)? gik (LTL) a aquivalent:  $L(A_S) \cap \overline{L(A_E)} = \emptyset$ ? Komplementianung 4r Produktautomat Leerheitsproblem -Verifikation mittels der konstruierten NBAs u Gibt es einen Pfad in S, der E erfüllt? a äquivalent:  $L(A_c) \cap L(A_c) \neq \emptyset$ ?

#### 17:00

2019-01

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo, für das Leerheitsproblem laufen lässt.

Büchi-Aut

Verifikation mittels der konstruierten NBAs

Entscheidungsprobl.

Model-Checking

Gegeben sind wieder System S und Spezifikation E.

• äquivalent:  $L(A_S) \subset L(A_E)$ ?

Universelles MCP

• äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?

• Gilt E für jeden Pfad in S?

 $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem

• Komplexität: PSpace (exponentielle Explosion bei Komplementierung)

# Existenzielles MCP

- Gibt es einen Pfad in S. der E erfüllt?
- äquivalent:  $L(A_S) \cap L(A_E) \neq \emptyset$ ?
- → Produktautomat, Leerheitsproblem

Teil 3: unendliche Wörter iegeben sind wieder System S und Spezifikation EAnwendung: Model-Checking in Linearer Temporallo-Gilt E für jeden Pfad in S? āquivalent: L(A<sub>S</sub>) ⊂ L(A<sub>E</sub>)? gik (LTL) - Komplementierung 4r Produktautomat Leerheitsprobler -Verifikation mittels der konstruierten NBAs a äquivalent:  $L(A_c) \cap L(A_c) \neq \emptyset$ ?

Verifikation mittels der konstruierten NBAs

-- Produktautomat, Leerheitsproblem

#### 17:00

2019-01

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo, für das Leerheitsproblem laufen lässt. Büchi-Aut

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

# Verifikation mittels der konstruierten NBAs

Gegeben sind wieder System S und Spezifikation E.

Charakt.

#### Universelles MCP

- Gilt E für jeden Pfad in S?
- äquivalent:  $L(A_S) \subset L(A_E)$ ?
- äquivalent:  $L(A_S) \cap \overline{L(A_F)} = \emptyset$ ?
- $\rightarrow$  Komplementierung  $A_F$ , Produktautomat, Leerheitsproblem
  - Komplexität: PSpace (exponentielle Explosion bei Komplementierung)

### Existenzielles MCP

- Gibt es einen Pfad in S. der E erfüllt?
- äquivalent:  $L(A_S) \cap L(A_E) \neq \emptyset$ ?
- → Produktautomat, Leerheitsproblem
  - Komplexität: NL (keine exponentielle Explosion)

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-Gilt E für jeden Pfad in S? āquivalent: L(A<sub>S</sub>) ⊂ L(A<sub>E</sub>)? gik (LTL) -Verifikation mittels der konstruierten NBAs a äquivalent:  $L(A_c) \cap L(A_c) \neq \emptyset$ ? -- Produktautomat, Leerheitsproblem

Verifikation mittels der konstruierten NBAs

#### 17:00

2019-01

"Exponentielle Explosion": wie schon gesagt, liefert der Weg über die Safra-Konstruktion einen doppelt exp. Blowup.

Es gibt aber direkte Verfahren zur Komplementierung mit einfach exp. Blowup.

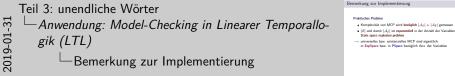
Außerdem kann man natürlich nicht den exp. großen Komplement-Automaten im Ganzen erzeugen, wenn man nur Polyplatz zur Verfügung hat.

Man muss ihn also "on the fly" stückchenweise generieren. während man den Algo, für das Leerheitsproblem laufen lässt. otiv. Büchi-Aut. Abschlusseig, Charakt. Determinismus Entscheidungsprobl.

# Bemerkung zur Implementierung

#### **Praktisches Problem**

- Komplexität von MCP wird bezüglich  $|A_S| + |A_E|$  gemessen
- |S| und damit  $|A_S|$  ist exponentiell in der Anzahl der Variablen: State space explosion problem
- → universelles bzw. existenzielles MCP sind eigentlich in ExpSpace bzw. in PSpace bezüglich Anz. der Variablen



### 17:06

"Abhilfe": macht natürlich nicht die Komplexität kleiner, vermeidet aber, den ganzen Automaten in den Speicher schreiben zu müssen.

Model-Checking

Praktisches Problem

Bemerkung zur Implementierung

State space explosion problem

"On-the-fly model checking"

nur bei Bedarf erzeugt

→ universelles bzw. existenzielles MCP sind eigentlich

ullet Zustände von  $\mathcal{A}_{\mathcal{S}}$  werden während des Leerheitstests

• Komplexität von MCP wird bezüglich  $|A_S| + |A_F|$  gemessen

in ExpSpace bzw. in PSpace bezüglich Anz. der Variablen

• |S| und damit  $|A_S|$  ist exponentiell in der Anzahl der Variablen:

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Bemerkung zur Implementierung

Bemerkung zur Implementierung

 Komplexität von MCP wird bezüglich |Ac| + |Ac| gemesser u ISI und damit I.Acl ist exponentiell in der Anzahl der Variablen

- universelles bzw. existenzielles MCP sind eigentlich in ExpSpace bzw. in PSpace bezüglich Anz. der Variabler

# nur bei Bedarf erzeugt

u \_On-the-fly model checking Zustände von Ac werden während des Leerheitstests

#### 17:06

2019-01

"Abhilfe": macht natürlich nicht die Komplexität kleiner, vermeidet aber, den ganzen Automaten in den Speicher schreiben zu müssen.

Abhilfe:

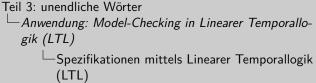
a intuitivere Beschreibung der Spezifikation E durch Formel der

a Prozedur zur Umwandlung use in As-(!) allerdings ist  $|A_E|$  exponentiell in  $|\varphi_E|$ 

# Spezifikationen mittels Linearer Temporallogik (LTL)

#### Nun zu Schritt 2. Ziele:

- intuitivere Beschreibung der Spezifikation E durch Formel  $\varphi_E$
- Prozedur zur Umwandlung  $\varphi_F$  in  $\mathcal{A}_F$ (!) allerdings ist  $|A_E|$  exponentiell in  $|\varphi_E|$
- dafür Explosion bei Komplementierung vermeiden: wandle  $\neg \varphi_F$  in Automaten um
- → beide MCP f
  ür LTL sind PSpace-vollst
  ändig



17:08

2019-01-

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

# LTL im Überblick

LTL = Aussagenlogik + Operatoren, die über Pfade sprechen:

F (Future)

 $F\varphi$  bedeutet " $\varphi$  ist irgendwann in der Zukunft wahr"

G (Global)

 $G\varphi$  bedeutet " $\varphi$  ist ab jetzt immer wahr"

X (neXt)

 $X\varphi$  bedeutet " $\varphi$  ist im nächsten Zeitpunkt wahr"

U: (Until)

 $\varphi U \psi$  bedeutet " $\psi$  ist irgendwann in der Zukunft wahr und bis dahin ist immer  $\varphi$  wahr"

Teil 3: unendliche Wörter

The Anwendung: Model-Checking in Linearer Temporallogisk (LTL)

gik (LTL)

LTL im Überblick

17:09

Model-Checking

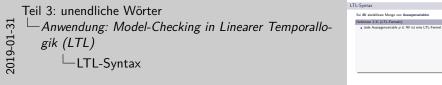
Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

# LTL-Syntax

Sei AV abzählbare Menge von Aussagenvariablen.

# Definition 3.31 (LTL-Formeln)

• Jede Aussagenvariable  $p \in AV$  ist eine LTL-Formel.



17:11

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

# LTL-Syntax

Sei AV abzählbare Menge von Aussagenvariablen.

### Definition 3.31 (LTL-Formeln)

- Jede Aussagenvariable  $p \in AV$  ist eine LTL-Formel.
- Wenn  $\varphi$  und  $\psi$  LTL-Formeln sind, dann sind die folgenden auch LTL-Formeln.

"in Zukunft irgendwann  $\psi$ ; bis dahin immer  $\varphi$ "

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogik (LTL)

LTL-Syntax

Definition 3.32 (ET-Formal)

A such extraoration by  $\sigma$  (R at size LT-Formal.

\*Ween  $\varphi$  und  $\psi$  (T-Formats ind,
densi side of the followings and LT-Formals.

\*Very

\*Property of the following and LT-Formals.

\*Property of the following and

17:11

2019-01

φ U ψ

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

# LTL-Syntax

Sei AV abzählbare Menge von Aussagenvariablen.

# Definition 3.31 (LTL-Formeln)

- Jede Aussagenvariable  $p \in AV$  ist eine LTL-Formel.
- Wenn  $\varphi$  und  $\psi$  LTL-Formeln sind, dann sind die folgenden auch LTL-Formeln.

 $\bullet \neg \varphi$  $\bullet \varphi \wedge \psi$ 

"nicht  $\varphi$ " " $\varphi$  und  $\psi$ "

Fφ

"in Zukunft irgendwann  $\varphi$ "

 $\bullet$   $G\varphi$ 

"in Zukunft immer  $\varphi$ "

Xφ

φ U ψ

"im nächsten Zeitpunkt  $\varphi$ " "in Zukunft irgendwann  $\psi$ ; bis dahin immer  $\varphi$ "

Verwenden die üblichen Abkürzungen  $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi),$  $\varphi \to \psi = \neg \varphi \lor \psi, \qquad \varphi \leftrightarrow \psi = (\varphi \to \psi) \land (\psi \to \varphi)$ 

Teil 3: unendliche Wörter

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -LTL-Syntax

LTL-Syntax Jede Aussagenvariable ρ ∈ AV ist eine LTL-Formel Wenn φ und ψ LTL-Formeln sind, "in Zukunft irgendwann ; "in Zukunft immer

17:11

104

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

# LTL-Semantik

**Pfad:** Abbildung  $\pi: \mathbb{N} \to 2^{\text{AV}}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 



17:13 bis Ende 17:40

8:30 - Kurzwdhlg.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

# LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ . Das **Erfülltsein** von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert. Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogie gik (LTL)

LTL-Semantik

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogie gik (LTL)

LTL-Semantik

17:13 bis Ende 17:40

8:30 - Kurzwdhlg.: Syntax+Semantik LTL; Bsp. a U b an Tafel

**Hinweisen:** "nicht strikte Semantik", also "≥" statt ">"; "für ein" (= "es gibt") vs. "für alle"

Model-Checking

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

## LTL-Semantik

**Pfad:** Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

•  $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$ 

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL)

LTL-Semantik ei  $\omega$  eine LTL-Formel.  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

17:13 bis Ende 17:40

-LTL-Semantik

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Charakt.

Determinismus

Entscheidungsprobl.

## LTL-Semantik

**Pfad:** Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$

Teil 3: unendliche Wörter - Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -LTL-Semantik

LTL-Semantik ei  $\omega$  eine LTL-Formel.  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

17:13 bis Ende 17:40

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Charakt.

Determinismus

Entscheidungsprobl.

## LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -LTL-Semantik

```
LTL-Semantik
        ei \omega eine LTL-Formel. \pi ein Pfad und i \in \mathbb{N}.
```

17:13 bis Ende 17:40

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Büchi-Aut.

Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

### LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$
- $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$

LTL-Semantik Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -LTL-Semantik

```
ei \omega eine LTL-Formel. \pi ein Pfad und i \in \mathbb{N}.
```

•  $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$ 

17:13 bis Ende 17:40

2019-01

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi$ ,  $i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$
- $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$
- $\pi, i \models G\varphi$ , falls  $\pi, i \models \varphi$  für alle  $i \geqslant i$

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -LTL-Semantik

LTL-Semantik ei  $\omega$  eine LTL-Formel.  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .  $\mathbf{v} = i \vdash \mathbf{p}$ , falls  $\mathbf{p} \in \pi$ ; für alle  $\mathbf{p} \in AV$  $a = i \vdash G \omega$ , falls  $\pi, i \vdash \omega$  für alle  $i \ge i$ 

17:13 bis Ende 17:40

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

2019-01

## LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi$ ,  $i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$
- $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$
- $\pi, i \models G\varphi$ , falls  $\pi, j \models \varphi$  für alle  $j \geqslant i$
- $\pi, i \models X\varphi$ , falls  $\pi, i+1 \models \varphi$

LTL-Semantik Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporalloei  $\omega$  eine LTL-Formel.  $\pi$  ein Pfad und  $i \in \mathbb{N}$ . Das Erfülltsein von  $\varphi$  in  $\pi, i$   $(\pi, i \models \varphi)$  ist wie folgt definiert  $\mathbf{v} = i \vdash \mathbf{p}$ , falls  $\mathbf{p} \in \pi$ ; für alle  $\mathbf{p} \in AV$ gik (LTL) -LTL-Semantik a  $\pi$ , i  $\models$   $G_{i}$ , falls  $\pi$ , i  $\models$   $\varphi$  für alle i ≥ i

17:13 bis Ende 17:40

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

ei  $\omega$  eine LTL-Formel.  $\pi$  ein Pfad und  $i \in \mathbb{N}$ Das Erfülltsein von  $\varphi$  in  $\pi, i$   $(\pi, i \models \varphi)$  ist wie folgt definiert

 $\mathbf{v} = i \vdash \mathbf{p}$ , falls  $\mathbf{p} \in \pi$ ; für alle  $\mathbf{p} \in AV$ 

a  $\pi$ , i  $\models$   $G_{i}$ , falls  $\pi$ , i  $\models$   $\varphi$  für alle i ≥ i

und  $\pi, k \models \varphi$  für alle k mit  $i \leqslant k < j$ 

# LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi$ , i  $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$
- $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$
- $\pi, i \models G\varphi$ , falls  $\pi, j \models \varphi$  für alle  $j \geqslant i$
- $\pi, i \models X\varphi$ , falls  $\pi, i+1 \models \varphi$
- $\pi, i \models \varphi \ U \ \psi$ , falls  $\pi, j \models \psi$  für ein  $j \geqslant i$ und  $\pi, k \models \varphi$  für alle k mit  $i \leq k < j$

Teil 3: unendliche Wörter - Anwendung: Model-Checking in Linearer Temporallogik (LTL) LTL-Semantik

17:13 bis Ende 17:40

2019-01

8:30 - Kurzwdhlq.: Syntax+Semantik LTL; Bsp. a U b an Tafel

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

# LTL-Semantik

Pfad: Abbildung  $\pi: \mathbb{N} \to 2^{AV}$  Schreiben  $\pi_0 \pi_1 \dots$  statt  $\pi(0)\pi(1)\dots$ 

#### Definition 3.32

Sei  $\varphi$  eine LTL-Formel,  $\pi$  ein Pfad und  $i \in \mathbb{N}$ .

Das Erfülltsein von  $\varphi$  in  $\pi, i$   $(\pi, i \models \varphi)$  ist wie folgt definiert.

- $\pi, i \models p$ , falls  $p \in \pi_i$ , für alle  $p \in AV$
- $\pi, i \models \neg \psi$ , falls  $\pi, i \not\models \psi$
- $\pi, i \models \varphi \land \psi$ , falls  $\pi, i \models \varphi$  und  $\pi, i \models \psi$
- $\pi, i \models F\varphi$ , falls  $\pi, j \models \varphi$  für ein  $j \geqslant i$
- $\pi, i \models G\varphi$ , falls  $\pi, j \models \varphi$  für alle  $j \geqslant i$
- $\pi, i \models X\varphi$ , falls  $\pi, i+1 \models \varphi$
- $\pi, i \models \varphi \ U \ \psi$ , falls  $\pi, j \models \psi$  für ein  $j \geqslant i$ und  $\pi, k \models \varphi$  für alle k mit  $i \leqslant k < j$

*j* T 3.20

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogisk (LTL)

LTL-Semantik

LTL-Semantik

LTL-Semantik

17:13 bis Ende 17:40

8:30 – Kurzwdhlg.: Syntax+Semantik LTL; Bsp. a U b an Tafel

**Hinweisen**: "nicht strikte Semantik", also "≥" statt ">"; "für ein" (= "es gibt") vs. "für alle"

2019-01

Notiv. Büchi-Aut. Abschlusseig.

Charal

Determinismu

Entscheidungsprobl.

# Beispiel-Spezifikationen als LTL-Formeln

Beispiel Mikrowelle (siehe Bild auf Folie 90)

• "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

$$G(e \rightarrow F \neg e)$$

 $(e \in AV \text{ steht für "Error"})$ 

8:32

2019-01

Wir brauchen das Bild nicht zu sehen; hier geht es nur um die Eigenschaften und die entsprechenden LTL-Formeln.

Beispiel-Spezifikationen als LTL-Formeln

Determinismus

Beispiel Mikrowelle (siehe Bild auf Folie 90)

• "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

$$G(e \rightarrow F \neg e)$$

 $(e \in AV \text{ steht für "Error"})$ 

• "Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen."

$$G(s \rightarrow Fh)$$

 $(s, h \in AV \text{ stehen für "Start" bzw. "Heat"})$ 



Beispiel-Spezifikationen als LTL-Formeln

8:32

2019-01

Wir brauchen das Bild nicht zu sehen; hier geht es nur um die Eigenschaften und die entsprechenden LTL-Formeln.

# Beispiel-Spezifikationen als LTL-Formeln

### Beispiel Mikrowelle (siehe Bild auf Folie 90)

• "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

$$G(e \to F \neg e)$$

$$(e \in AV \text{ steht für "Error"})$$

• "Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen."

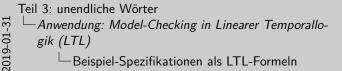
$$G(s \rightarrow Fh)$$

$$(s, h \in AV \text{ stehen für "Start" bzw. "Heat"})$$

• "Irgendwann ist für genau einen Zeitpunkt die Tür geöffnet."

$$F(c \wedge X(\neg c \wedge Xc))$$

 $(c \in AV \text{ steht für "Close"})$ 



8:32

Wir brauchen das Bild nicht zu sehen; hier geht es nur um die Eigenschaften und die entsprechenden LTL-Formeln.

Beispiel-Spezifikationen als LTL-Formeln

### Beispiel Mikrowelle (siehe Bild auf Folie 90)

• "Wenn ein Fehler auftritt, ist er nach endlicher Zeit behoben."

$$G(e \to F \neg e)$$

 $(e \in AV \text{ steht für "Error"})$ 

• "Wenn die Mikrowelle gestartet wird, fängt sie nach endlicher Zeit an zu heizen."

$$G(s \rightarrow Fh)$$

 $(s, h \in AV \text{ stehen für "Start" bzw. "Heat"})$ 

"Irgendwann ist für genau einen Zeitpunkt die Tür geöffnet."

$$F(c \wedge X(\neg c \wedge Xc))$$

 $(c \in AV \text{ steht für "Close"})$ 

• "Irgendwann ist für genau einen Zeitpunkt die Tür geöffnet, und bis dahin ist sie geschlossen."

$$c U (\neg c \wedge Xc)$$



Beispiel-Spezifikationen als LTL-Formeln

- $G(e \rightarrow F \neg e)$ .Wenn die Mikrowelle gestartet wird,

- w "Irgendwann ist für genau einen Zeitpunkt die Tür geöffnet und bis dahin ist sie geschlossen. cU(¬c∧Xc)

#### 8:32

2019-01

Wir brauchen das Bild nicht zu sehen; hier geht es nur um die Eigenschaften und die entsprechenden LTL-Formeln.

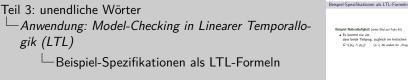
Büchi-Aut. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

# Beispiel-Spezifikationen als LTL-Formeln

## Beispiel Nebenläufigkeit (siehe Bild auf Folie 92)

 Es kommt nie vor, dass beide Teilprog. zugleich im kritischen Bereich sind.

$$G \neg (p_{12} \land p_{22})$$
  $(p_i \in AV \text{ stehen für "Programmzähler in Zeile } i")$ 



Beispiel Nebenläufigkeit (siehe Bild auf Folie 92) - Er howest nic yes dass beide Teilprog, zugleich im kritischen Bereich sind.

8:35

Notiv. Büchi-Aut. Abschlusseig.

g. Cha

Determinismus

Entscheidungsprobl.

Model-Checking

### Beispiel-Spezifikationen als LTL-Formeln

#### Beispiel Nebenläufigkeit (siehe Bild auf Folie 92)

- Es kommt nie vor, dass beide Teilprog. zugleich im kritischen Bereich sind.  $G \neg (p_{12} \land p_{22})$   $(p_i \in AV \text{ stehen für "Programmzähler in Zeile } i")$
- Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich.  $GFp_{12} \wedge GFp_{22}$

8:35

2019-01

Motiv. Büchi-Aut. Abschlusseig, Charakt. Determinismus Entscheidungsprobl.

### Model-Checking mit LTL-Formeln

### Zur Erinnerung:

#### Definition 3.30: Model-Checking-Problem MCP

Gegeben ein System  ${\mathcal S}$  und eine Spezifikation  ${\mathcal E}$ ,

- gilt E für jeden Pfad in S? (universelle Variante)
- gibt es einen Pfad in S, der E erfüllt? (existenzielle Variante)



8:37

Model-Checking

Büchi-Aut. Charakt. Determinismus Entscheidungsprobl. Model-Checking Abschlusseig.

### Model-Checking mit LTL-Formeln

#### Für LTL:

(jedem Pfad  $s_0 s_1 s_2 \dots$  in einer Kripke-Struktur  $S = (S, S_0, R, \ell)$ entspricht ein LTL-Pfad  $\pi_0\pi_1\pi_2\dots$  mit  $\pi_i=\ell(s_i)$ 

Model-Checking mit LTL-Formeln Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-

gik (LTL) -Model-Checking mit LTL-Formeln

8:38

Charakt. Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Model-Checking mit LTL-Formeln

#### Für LTL:

(jedem Pfad  $s_0 s_1 s_2 \dots$  in einer Kripke-Struktur  $\mathcal{S} = (S, S_0, R, \ell)$ entspricht ein LTL-Pfad  $\pi_0\pi_1\pi_2\dots$  mit  $\pi_i=\ell(s_i)$ 

#### Definition 3.33 (Model-Checking-Problem)

Gegeben Kripke-Struktur  $S = (S, S_0, R, \ell)$  und LTL-Formel  $\varphi$ ,

• gilt  $\pi$ ,  $0 \models \varphi$  für alle Pfade  $\pi$ , die in einem  $s_0 \in S_0$  starten? (universelle Variante)

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

 $\mathbf{v}$  gilt  $\pi$ .  $0 \models \omega$  für alle Pfade  $\pi$ . die in einem  $\mathbf{s}_1 \in S_1$  starten

(iedem Pfad  $s_0s_1s_2...$  in einer Kripke-Struktur  $S = (S, S_0, R, \ell)$ 

Model-Checking mit LTL-Formeln

-Model-Checking mit LTL-Formeln

8:38

Abschlusseig.

Determinismus

Entscheidungsprobl.

Model-Checking

### Model-Checking mit LTL-Formeln

#### Für LTL:

(jedem Pfad  $s_0 s_1 s_2 \dots$  in einer Kripke-Struktur  $\mathcal{S} = (S, S_0, R, \ell)$ entspricht ein LTL-Pfad  $\pi_0\pi_1\pi_2\dots$  mit  $\pi_i=\ell(s_i)$ 

#### Definition 3.33 (Model-Checking-Problem)

Gegeben Kripke-Struktur  $S = (S, S_0, R, \ell)$  und LTL-Formel  $\varphi$ ,

- gilt  $\pi$ ,  $0 \models \varphi$  für alle Pfade  $\pi$ , die in einem  $s_0 \in S_0$  starten? (universelle Variante)
- gibt es Pfad  $\pi$ , der in einem  $\pi_0 \in S_0$  startet, mit  $\pi, 0 \models \varphi$ ? (existenzielle Variante)

Teil 3: unendliche Wörter (iedem Pfad  $s_0s_1s_2...$  in einer Kripke-Struktur  $S = (S, S_0, R, \ell)$ -Anwendung: Model-Checking in Linearer Temporallogik (LTL)  $\mathbf{v}$  gilt  $\pi$ .  $0 \models \omega$  für alle Pfade  $\pi$ . die in einem  $\mathbf{s}_1 \in S_1$  starten a gibt as Pfad π, der in einem π₁ ∈ S₁ startet, mit π.0 lm φ i -Model-Checking mit LTL-Formeln

Model-Checking mit LTL-Formeln

8:38

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

### Model-Checking mit LTL-Formeln

#### Für LTL:

(jedem Pfad  $s_0 s_1 s_2 \dots$  in einer Kripke-Struktur  $\mathcal{S} = (S, S_0, R, \ell)$ entspricht ein LTL-Pfad  $\pi_0\pi_1\pi_2\dots$  mit  $\pi_i=\ell(s_i)$ 

#### Definition 3.33 (Model-Checking-Problem)

Gegeben Kripke-Struktur  $S = (S, S_0, R, \ell)$  und LTL-Formel  $\varphi$ ,

- gilt  $\pi$ ,  $0 \models \varphi$  für alle Pfade  $\pi$ , die in einem  $s_0 \in S_0$  starten? (universelle Variante)
- gibt es Pfad  $\pi$ , der in einem  $\pi_0 \in S_0$  startet, mit  $\pi, 0 \models \varphi$ ? (existenzielle Variante)
- ✓ Exakte Beschreibung des Model-Checking-Problems
- ▶ Algorithmische Lösung?

Teil 3: unendliche Wörter

-Anwendung: Model-Checking in Linearer Temporallogik (LTL)

-Model-Checking mit LTL-Formeln

Model-Checking mit LTL-Formeln (iedem Pfad  $s_0s_1s_2...$  in einer Kripke-Struktur  $S = (S, S_0, R, \ell)$  $\mathbf{v}$  gilt  $\pi$ .  $0 \models \omega$  für alle Pfade  $\pi$ . die in einem  $\mathbf{s}_1 \in S_1$  starten a gibt as Pfad π, der in einem π₁ ∈ S₁ startet, mit π.0 lm φ i

✓ Exakte Beschreibung des Model-Checking-Problems ► Algorithmische Lösung?

8:38

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

Teil 3: unendliche Wörter

### MCP weiterhin mittels Büchi-Automaten lösen!

#### Vorgehen wie gehabt:

- $\bullet$  Wandle Kripke-Struktur  ${\cal S}$  in NBA  ${\cal A}_{\cal S}$  um
  - $\sim$  Pfade in  ${\mathcal S}$  sind erfolgreiche Runs von  ${\mathcal A}_{\mathcal S}$

— Anwendung: Model-Checking in Linearer Temporallogisk (LTL)

—MCP weiterhin mittels Büchi-Automaten lösen!

MCP weiterhin mittels Büchi-Automaten lösen!

8:41

2019-01

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

### MCP weiterhin mittels Büchi-Automaten lösen!

#### Vorgehen wie gehabt:

- ullet Wandle Kripke-Struktur  ${\cal S}$  in NBA  ${\cal A}_{\cal S}$  um  $\rightarrow$  Pfade in S sind erfolgreiche Runs von  $A_S$
- Wandeln LTL-Formel  $\varphi_E$  in NBA  $\mathcal{A}_E$  um  $\rightarrow$   $A_F$  beschreibt Pfade, die E erfüllen

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -MCP weiterhin mittels Büchi-Automaten lösen! MCP weiterhin mittels Büchi-Automaten lösen!

 $_{f e}$  Wandle Kripke-Struktur  ${\cal S}$  in NBA  ${\cal A}_{\cal S}$  um

a Wandeln LTL-Formel use in NBA As um

8:41

2019-01

- $_{f e}$  Wandle Kripke-Struktur  ${\cal S}$  in NBA  ${\cal A}_{\cal S}$  um → Pfade in S sind erfolereiche Runs von A.
- a Wandeln LTL-Formel use in NBA As um

### MCP weiterhin mittels Büchi-Automaten lösen!

#### Vorgehen wie gehabt:

- Wandle Kripke-Struktur S in NBA  $A_S$  um  $\rightarrow$  Pfade in S sind erfolgreiche Runs von  $A_S$
- Wandeln LTL-Formel  $\varphi_E$  in NBA  $\mathcal{A}_E$  um  $\rightarrow$   $A_F$  beschreibt Pfade, die E erfüllen
- $\rightarrow$  Universelles MCP = " $L(A_S) \subset L(A_F)$ ?" Existenzielles MCP =  $_{n}L(A_{S}) \cap L(A_{E}) \neq \emptyset$ ?"

8:41

2019-01

Teil 3: unendliche Wörter

gik (LTL)

-Anwendung: Model-Checking in Linearer Temporallo-

-MCP weiterhin mittels Büchi-Automaten lösen!

#### Vorgehen wie gehabt:

- Wandle Kripke-Struktur S in NBA  $A_S$  um  $\rightarrow$  Pfade in S sind erfolgreiche Runs von  $A_S$
- ullet Wandeln LTL-Formel  $arphi_E$  in NBA  $\mathcal{A}_E$  um  $\rightarrow$   $A_F$  beschreibt Pfade, die E erfüllen
- $\rightarrow$  Universelles MCP = " $L(A_S) \subset L(A_F)$ ?" Existenzielles MCP =  $_{"}L(A_S) \cap L(A_F) \neq \emptyset$ ?"

**Noch zu klären:** Wie wandeln wir  $\varphi_F$  in  $\mathcal{A}_F$  um?

MCP weiterhin mittels Büchi-Automaten lösen! Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo- $_{f e}$  Wandle Kripke-Struktur  ${\cal S}$  in NBA  ${\cal A}_{\cal S}$  um → Pfade in S sind erfolereiche Runs von A. gik (LTL) a Wandeln LTL-Formel use in NBA As um -MCP weiterhin mittels Büchi-Automaten lösen!

Noch zu klären: Wie wandeln wir  $\varphi_E$  in  $A_E$  um?

8:41

2019-01

Wandeln  $\varphi_E$  in generalisierten Büchi-Automaten (GNBA) um:

- $\mathcal{A}_{\omega_{\mathcal{F}}} = (Q, \Sigma, \Delta, I, \mathcal{F}) \text{ mit } \mathcal{F} \subset 2^Q$
- $r = q_0 q_1 q_2 \dots$  ist erfolgreich:  $Inf(r) \cap F \neq \emptyset$  für alle  $F \in \mathcal{F}$
- GNBAs und NBAs sind äquivalent (nur quadratische Vergrößerung)

Umwandlung von LTL-Formeln in Automaten (Überblick) Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallo-Nandele or in generalisierten Rüchi, Automaten (GNRA) um gik (LTL)  $a.A... = (Q, \Sigma, \Delta, I, F) \text{ mit } F \subset 2^Q$ -Umwandlung von LTL-Formeln in Automaten

8:43

2019-01

"GNBAs und NBAs sind äquivalent":

Richtung NBA  $\rightarrow$  GNBA trivial.

(Überblick)

Richtung GNBA  $\rightarrow$  NBA:

Wenn  $\mathcal{F} = \{F_1, \dots, F_n\}$ , erzeuge *n* Kopien des GNBA.

Von jedem akzZ. der *i*-ten Kopie wechsle in  $((i + 1) \mod n)$ -te Kopie.

Neue akzZ: die bisherigen akzZ. einer beliebigen Kopie.

Büchi-Aut. Abschlusseig.

Charakt. Determinismus Entscheidungsprobl.

Model-Checking

### Vorbetrachtungen

Sei  $\varphi_F$  eine LTL-Formel, in der o. B. d. A.

• nur die Operatoren  $\neg$ ,  $\wedge$ , X, U vorkommen

Die anderen kann man mit diesen ausdrücken:

$$F\varphi \equiv (\neg(p \land \neg p)) \ U \varphi \qquad G\varphi \equiv \neg F \neg \varphi$$

keine doppelte Negation vorkommt

natürlich gilt 
$$\neg\neg\psi\equiv\psi$$
 für alle Teilformeln  $\psi$ 

(Hier steht 
$$\alpha \equiv \beta$$
 für  $\forall \pi \forall i : \pi, i \models \alpha$  gdw.  $\pi, i \models \beta$ )

Teil 3: unendliche Wörter Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Vorbetrachtungen

 $F_{\varphi} \equiv (\neg(\varphi \land \neg \varphi)) U_{\varphi}$   $G_{\varphi} \equiv \neg F \neg \varphi$ 

Vorbetrachtungen

8:45

2019-01-31

Die Einschränkung der vorkommenden Operatoren

- ist o. B. d. A., wie wir an den Äquivalenzen sehen;
- macht die folgenden Definitionen deutlich übersichtlicher;
- führt aber dazu, dass Automaten schon für kleine F-/G-Formeln riesig werden.

Deshalb nur kurze Beispiele hier und in ÜS.

## Vorbetrachtungen

Sei  $\varphi_F$  eine LTL-Formel, in der o. B. d. A.

- nur die Operatoren  $\neg$ ,  $\wedge$ , X, U vorkommen
  - Die anderen kann man mit diesen ausdrücken:

$$F\varphi \equiv (\neg(p \land \neg p)) \ U \varphi \qquad G\varphi \equiv \neg F \neg \varphi$$

- keine doppelte Negation vorkommt
  - natürlich gilt  $\neg \neg \psi \equiv \psi$  für alle Teilformeln  $\psi$

(Hier steht 
$$\alpha \equiv \beta$$
 für  $\forall \pi \forall i : \pi, i \models \alpha$  gdw.  $\pi, i \models \beta$ )

#### **Etwas Notation**

- $\operatorname{cl}(\varphi_F) = \{\psi, \sim \psi \mid \psi \text{ ist Teilformel von } \varphi_F\}$
- $\Sigma = 2^{AV}$

2019-01-31

## -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Vorbetrachtungen

### $F_{\varphi} \equiv (\neg(\varphi \land \neg \varphi)) U_{\varphi}$ $G_{\varphi} \equiv \neg F \neg \varphi$

Vorbetrachtungen

•  $cl(\varphi_E) = \{\psi, \sim \psi \mid \psi \text{ ist Teilformel von } \varphi_E\}$  $\sigma \Sigma = 2^{AV}$ 

#### 8:45

Die Einschränkung der vorkommenden Operatoren

- ist o. B. d. A., wie wir an den Äquivalenzen sehen;
- macht die folgenden Definitionen deutlich übersichtlicher;
- führt aber dazu, dass Automaten schon für kleine F-/G-Formeln riesig werden.

Deshalb nur kurze Beispiele hier und in ÜS.

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl.

#### Intuitionen

Motiv.

#### **Erweiterung von Pfaden**

- Betrachten Pfade  $\pi = s_0 s_1 s_2 \dots$  mit  $s_i \subset AV$
- Erweitern jedes  $s_i$  mit den  $\psi \in cl(\varphi_E)$ , für die  $\pi, i \models \psi$  gilt
- Resultat: Folge  $\overline{\pi} = t_0 t_1 t_2 \dots$  mit  $t_i \subseteq cl(\varphi_E)$

Skizze: s. Tafel T 3.21

Teil 3: unendliche Wörter

Anwendung: Model-Checking in Linearer Temporallogisk (LTL)

Intuitionen

8:48 bis 8:56

2019-01-31

Model-Checking

Büchi-Aut. Abschlusseig. Motiv. Charakt. Determinismus

Entscheidungsprobl.

Model-Checking

#### Intuitionen

#### **Erweiterung von Pfaden**

- Betrachten Pfade  $\pi = s_0 s_1 s_2 \dots$  mit  $s_i \subset AV$
- Erweitern jedes  $s_i$  mit den  $\psi \in cl(\varphi_E)$ , für die  $\pi, i \models \psi$  gilt
- Resultat: Folge  $\overline{\pi} = t_0 t_1 t_2 \dots$  mit  $t_i \subset cl(\varphi_E)$

# Bestandteile des GNBA $\mathcal{A}_{\varphi_{\mathcal{E}}}$

Skizze: s. Tafel T 3.21

- Zustände:  $\approx$  alle  $t_i$
- $\overline{\pi} = t_0 t_1 t_2 \dots$  wird ein Run von  $\mathcal{A}_{\varphi_F}$  für  $s_0 s_1 s_2 \dots$  sein
- Run  $\overline{\pi}$  wird erfolgreich sein gdw.  $\pi$ ,  $0 \models \varphi_E$
- Kodieren Bedeutung der logischen Operatoren in
  - Zustände (¬, ∧, teilweise U)
  - Überführungsrelation (X, teilweise U)
  - Akzeptanzbedingung (teilweise *U*)

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Intuitionen

Intuitionen Erweiterung von Pfaden a Betrachten Pfade π = sisisi . . . mit si ⊂ AV ■ Erweitern iedes  $s_i$  mit den  $\psi \in cl(w_C)$ , für die  $\pi, i \models \psi$  eilt Resultat: Folice ∓ = totato ... mit t: C d(ωc) Skizze: s. Tafel T 3.21 w Zustände: ≈ alle t:  $\bullet$   $\pi = t_1t_1t_2\dots$  wird ein Run von  $\mathcal{A}_{i,x}$  für  $s_1s_1s_2\dots$  sein Zustände (¬. ∧. teilweise U) Überführungsrelation (X, teilweise U) · Akzeptanzbedingung (teilweise U)

8:48 bis 8:56

2019-01-31

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

# Zustandsmenge des GNBA $\mathcal{A}_{\omega_{F}}$

Q = Menge aller elementaren Formelmengen, wobei  $t \subset cl(\varphi_E)$  elementar ist, wenn gilt:

- ① t ist konsistent bzgl. Aussagenlogik, d. h. für alle  $\psi_1 \wedge \psi_2 \in cl(\varphi_E)$  und  $\psi \in cl(\varphi_E)$ :
  - $\psi_1 \wedge \psi_2 \in t$  gdw.  $\psi_1 \in t$  und  $\psi_2 \in t$
  - wenn  $\psi \in t$ , dann  $\sim \psi \notin t$



Q = Menge aller elementaren Formelmenger

a r ist konsistent bzel. Aussagenlogik, d.h.  $\psi_1 \wedge \psi_2 \in t \text{ gdw. } \psi_1 \in t \text{ und } \psi_2 \in t$ 

wobei  $t \subseteq cl(\varphi_E)$  elementar ist, wenn gilt:

wenn v ∈ t, dann ~v ∉ t

# Zustandsmenge des GNBA $\mathcal{A}_{\varphi_{E}}$

Q= Menge aller elementaren Formelmengen, wobei  $t\subseteq \operatorname{cl}(\varphi_E)$  elementar ist, wenn gilt:

- t ist konsistent bzgl. Aussagenlogik, d. h. für alle  $\psi_1 \wedge \psi_2 \in \operatorname{cl}(\varphi_E)$  und  $\psi \in \operatorname{cl}(\varphi_E)$ :
    $\psi_1 \wedge \psi_2 \in t$  gdw.  $\psi_1 \in t$  und  $\psi_2 \in t$ 

  - wenn  $\psi \in t$ , dann  $\sim \psi \notin t$
- ② t ist lokal konsistent bzgl. des U-Operators, d. h. für alle  $\psi_1$  U  $\psi_2 \in \operatorname{cl}(\varphi_E)$ :
  - wenn  $\psi_2 \in t$ , dann  $\psi_1 \ U \ \psi_2 \in t$
  - wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t$ , dann  $\psi_1 \in t$

Zustandsmenge des GNBA  $A_{eg}$  Q · Meng aller demotation Formalismagne,
what is  $C \in G(x_0^2)$  demotate int, wenn git:  $\Phi$  it is busident back, Ansaspanlogis, d. h.
for all  $w \in V, v \in G(x_0^2)$  and  $v \in G(x_0^2)$   $v \in V, v \in G(x_0^2)$  and  $v \in G(x_0^2)$   $v \in V, v \in G(x_0^2)$  and  $v \in G(x_0^2)$   $v \in V, v \in G(x_0^2)$   $v \in G(x_0^2)$ 

• wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t, \ \text{dann} \ \psi_1 \in t$ 

# Zustandsmenge des GNBA $\mathcal{A}_{\omega_{E}}$

Q = Menge aller elementaren Formelmengen, wobei  $t \subset cl(\varphi_F)$  elementar ist, wenn gilt:

- ① t ist konsistent bzgl. Aussagenlogik, d. h. für alle  $\psi_1 \wedge \psi_2 \in \operatorname{cl}(\varphi_E)$  und  $\psi \in \operatorname{cl}(\varphi_E)$ :
  - $\psi_1 \wedge \psi_2 \in t$  gdw.  $\psi_1 \in t$  und  $\psi_2 \in t$
  - wenn  $\psi \in t$ , dann  $\sim \psi \notin t$
- 2 t ist lokal konsistent bzgl. des *U*-Operators, d. h. für alle  $\psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_F)$ :
  - wenn  $\psi_2 \in t$ , dann  $\psi_1 U \psi_2 \in t$
  - wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t$ , dann  $\psi_1 \in t$
- **3** t ist maximal, d. h. für alle  $\psi \in cl(\varphi_F)$ : wenn  $\psi \notin t$ , dann  $\sim \psi \in t$

Teil 3: unendliche Wörter 2019-01-31 Anwendung: Model-Checking in Linearer Temporallogik (LTL) –Zustandsmenge des GNBA  ${\cal A}_{\omega_F}$ 

Zustandsmenge des GNBA A., Q = Menge aller elementaren Formelmengen wobei  $t \subseteq cl(\varphi_E)$  elementar ist, wenn gilt: a r ist konsistent bzel. Aussagenlogik, d.h. für alle  $\psi_1 \wedge \psi_2 \in cl(\varphi_E)$  und  $\psi \in cl(\varphi_E)$ :  $\psi_1 \wedge \psi_2 \in t \text{ gdw. } \psi_1 \in t \text{ und } \psi_2 \in t$  wenn v ∈ t, dann ~v ∉ t a t ist lokal konsistent bzgl. des U-Operators, d.h für alle  $\psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)$ : wenn sis ∈ t, dann sis U sis ∈ t • wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t, \ \text{dann} \ \psi_1 \in t$  $\phi$  t ist maximal, d. h. für alle  $\psi \in cl(\varphi_E)$ 

wenn  $\psi \notin t$ , dann  $\sim \psi \in t$ 

- ① t ist konsistent bzgl. Aussagenlogik, d. h. für alle  $\psi_1 \wedge \psi_2 \in \operatorname{cl}(\varphi_E)$  und  $\psi \in \operatorname{cl}(\varphi_E)$ :
  - $\psi_1 \wedge \psi_2 \in t$  gdw.  $\psi_1 \in t$  und  $\psi_2 \in t$

  - wenn  $\psi \in t$ , dann  $\sim \psi \notin t$
- 2 t ist lokal konsistent bzgl. des *U*-Operators, d. h. für alle  $\psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_F)$ :
  - wenn  $\psi_2 \in t$ , dann  $\psi_1 U \psi_2 \in t$
  - wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t$ , dann  $\psi_1 \in t$
- **3** t ist maximal, d. h. für alle  $\psi \in cl(\varphi_F)$ : wenn  $\psi \notin t$ , dann  $\sim \psi \in t$

Beispiel:  $a U (\neg a \land b)$ , siehe Tafel

T 3.22

Zustandsmenge des GNBA A., Teil 3: unendliche Wörter Q = Menge aller elementaren Formelmenger wobei  $t \subseteq cl(\varphi_E)$  elementar ist, wenn gilt: 2019-01-31 Anwendung: Model-Checking in Linearer Temporalloa t ist konsistent bzel. Aussagenlogik, d.h. für alle  $\psi_1 \wedge \psi_2 \in cl(\varphi_E)$  und  $\psi \in cl(\varphi_E)$ :  $\psi_1 \wedge \psi_2 \in t \text{ gdw. } \psi_1 \in t \text{ und } \psi_2 \in t$ gik (LTL) wenn v ∈ t, dann ~v ∉ t a t ist lokal konsistent bzel, des U-Operators, d.h. für alle  $\psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)$ : wenn sis ∈ t, dann sis U sis ∈ t -Zustandsmenge des GNBA  $\mathcal{A}_{\scriptscriptstyle \mathcal{Q}_{F}}$ • wenn  $\psi_1 \ U \ \psi_2 \in t \ \text{und} \ \psi_2 \notin t, \ \text{dann} \ \psi_1 \in t$  $\phi$  t ist maximal, d. h. für alle  $\psi \in cl(\varphi_E)$ wenn  $\psi \notin t$ , dann  $\sim \psi \in t$ Beispiel:  $aU(\neg a \land b)$ , siehe Tafel

Büchi-Aut. Abschlusseig. Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

–Überführungsrelation des GNBA  $\mathcal{A}_{\omega_{F}}$ 

Überführungsrelation des GNBA  $\mathcal{A}_{\omega_{F}}$ 

Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{AV}$ )

 $\Delta$  besteht aus allen Tripeln (t, s, t') mit

(d. h. s besteht aus allen Aussagevariablen in t)

9:15 bis 9:26

Abschlusseig.

Charakt.

Determinismus

Entscheidungsprobl.

Model-Checking

# Überführungsrelation des GNBA $\mathcal{A}_{\omega_{\mathcal{F}}}$

Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{AV}$ )

 $\Delta$  besteht aus allen Tripeln (t, s, t') mit

- (d. h. s besteht aus allen Aussagevariablen in t)
- ② für alle  $X\psi \in cl(\varphi_E)$ :  $X\psi \in t$  gdw.  $\psi \in t'$

Überführungsrelation des GNBA .4.. Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{NV}$ ) gik (LTL) –Überführungsrelation des GNBA  $\mathcal{A}_{\omega_{F}}$ 

9:15 bis 9:26

Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{AV}$ )

 $\Delta$  besteht aus allen Tripeln (t, s, t') mit

- (d. h. s besteht aus allen Aussagevariablen in t)
- für alle  $\psi_1$  U  $\psi_2 \in \operatorname{cl}(\varphi_E)$ :  $\psi_1$  U  $\psi_2 \in t$  gdw.  $\psi_2 \in t$  oder  $(\psi_1 \in t \text{ und } \psi_1 \text{ } U \text{ } \psi_2 \in t')$ ("Aufschieben" von  $\psi_1$  U  $\psi_2$ )

9:15 bis 9:26

2019-01

Abschlusseig. Charakt. Determinismus

Entscheidungsprobl.

Model-Checking

2019-01

### Überführungsrelation des GNBA $\mathcal{A}_{\omega_{E}}$

Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{AV}$ )

 $\Delta$  besteht aus allen Tripeln (t, s, t') mit

 $\mathbf{0}$   $s = t \cap AV$ 

(d. h. s besteht aus allen Aussagevariablen in t)

② für alle  $X\psi \in cl(\varphi_E)$ :  $X\psi \in t$  gdw.  $\psi \in t'$ 

 $\bullet$  für alle  $\psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)$ :

 $\psi_1 \ U \ \psi_2 \in t \ \text{gdw.} \ \psi_2 \in t \ \text{oder} \ (\psi_1 \in t \ \text{und} \ \psi_1 \ U \ \psi_2 \in t')$ 

("Aufschieben" von  $\psi_1 U \psi_2$ )  $\wedge$ 

Skizzen: siehe Tafel T 3.23

Überführungsrelation des GNBA .4.. Teil 3: unendliche Wörter Seien  $t, t' \in Q$  (elementare Formelmengen) und  $s \in \Sigma$  ( $\Sigma = 2^{NV}$ ) -Anwendung: Model-Checking in Linearer Temporallogik (LTL) -Überführungsrelation des GNBA  $\mathcal{A}_{\omega_{F}}$ 

9:15 bis 9:26

Büchi-Aut. Abschlusseig.

Determinismus

Entscheidungsprobl.

## Anfangszustände und Akzeptanzkomponente von $\mathcal{A}_{\varphi_F}$

Menge der Anfangszustände alle elementaren Formelmengen, die  $\varphi_F$  enthalten

$$I = \{ t \in Q \mid \varphi_E \in t \}$$



9:26 bis 9:58

Abschlusseig.

Determinismus

#### Charakt. Anfangszustände und Akzeptanzkomponente von $\mathcal{A}_{\omega_E}$

Menge der Anfangszustände alle elementaren Formelmengen, die  $\varphi_F$  enthalten

$$I = \{ t \in Q \mid \varphi_E \in t \}$$

Menge der akzeptierenden Zustände

stellen sicher, dass kein  $\psi_1 U \psi_2$  für immer "aufgeschoben" wird

$$\mathcal{F} = \{ M_{\psi_1 U \psi_2} \mid \psi_1 \ U \ \psi_2 \in \mathsf{cl}(\varphi_E) \} \ \mathsf{mit}$$

$$M_{\psi_1 U \psi_2} = \{ t \in Q \mid \psi_1 U \psi_2 \notin t \text{ oder } \psi_2 \in t \}$$

Anfangszustände und Akzeptanzkomponente von A. Teil 3: unendliche Wörter alle elementaren Formelmenzen, die cor enthalten -Anwendung: Model-Checking in Linearer Temporallo- $I = \{t \in Q \mid \varphi_E \in t\}$ Menge der akzeptierenden Zustände gik (LTL) tellen sicher, dass kein ψ<sub>1</sub> U ψ<sub>2</sub> für immer "aufgeschoben" wird  $\mathcal{F} = \{M_{\varphi_1U\varphi_2} \mid \psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)\} \ \operatorname{mit}$  $M_{\psi_1U\psi_2} = \{t \in Q \mid \psi_1 \ U \ \psi_2 \notin t \ oder \ \psi_2 \in t\}$ -Anfangszustände und Akzeptanzkomponente von  $\mathcal{A}_{arphi_{E}}$ 

9:26 bis 9:58

2019-01

### Anfangszustände und Akzeptanzkomponente von $\mathcal{A}_{\omega_E}$

Menge der Anfangszustände alle elementaren Formelmengen, die  $\varphi_F$  enthalten

$$I = \{ t \in Q \mid \varphi_E \in t \}$$

Menge der akzeptierenden Zustände

stellen sicher, dass kein  $\psi_1 \ U \ \psi_2$  für immer "aufgeschoben" wird

$$\mathcal{F} = \{ \mathit{M}_{\psi_1 \mathit{U} \psi_2} \mid \psi_1 \; \mathit{U} \; \psi_2 \in \mathsf{cl}(\varphi_\mathit{E}) \} \; \mathsf{mit}$$

$$M_{\psi_1 U \psi_2} = \{ t \in Q \mid \psi_1 \ U \ \psi_2 \notin t \ \text{oder} \ \psi_2 \in t \}$$

Intuition: Ein  $t \in M_{\psi_1 U \psi_2}$  kommt unendlich oft vor gdw.  $\psi_1 U \psi_2$  immer nur höchstens endlich lange "aufgeschoben" wird

Anfangszustände und Akzeptanzkomponente von A. Teil 3: unendliche Wörter alle elementaren Formelmenzen, die cor enthalten -Anwendung: Model-Checking in Linearer Temporallo- $I = \{t \in Q \mid \varphi_E \in t\}$ Menge der akzeptierenden Zustände gik (LTL) itellen sicher, dass kein  $\psi_1 U \psi_2$  für immer "aufgeschoben" wird  $\mathcal{F} = \{M_{\varphi_1U\varphi_2} \mid \psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)\} \ \operatorname{mit}$  $M_{\psi_1U\psi_2} = \{t \in Q \mid \psi_1 \ U \ \psi_2 \notin t \ oder \ \psi_2 \in t\}$ -Anfangszustände und Akzeptanzkomponente von Intuition: Ein  $t \in M_{\psi_1 U \psi_2}$  kommt unendlich oft vor gdw  $\mathcal{A}_{arphi \mathsf{E}}$ 

9:26 bis 9:58

2019-01

Abschlusseig.

Charakt.

Entscheidungsprobl.

T 3.24

116

### Anfangszustände und Akzeptanzkomponente von $\mathcal{A}_{\omega_{E}}$

Menge der Anfangszustände alle elementaren Formelmengen, die  $\varphi_F$  enthalten

$$I = \{ t \in Q \mid \varphi_E \in t \}$$

Menge der akzeptierenden Zustände

stellen sicher, dass kein  $\psi_1 U \psi_2$  für immer "aufgeschoben" wird

$$\mathcal{F} = \{ M_{\psi_1 U \psi_2} \mid \psi_1 \ U \ \psi_2 \in \mathsf{cl}(\varphi_E) \} \ \mathsf{mit}$$

$$M_{\psi_1 U \psi_2} = \{ t \in Q \mid \psi_1 \ U \ \psi_2 \notin t \ \text{oder} \ \psi_2 \in t \}$$

Intuition: Ein  $t \in M_{\psi_1 U \psi_2}$  kommt unendlich oft vor gdw.  $\psi_1 U \psi_2$  immer nur höchstens endlich lange "aufgeschoben" wird

Beispiel: Xa, siehe Tafel

Automatentheorie u. i. A. WiSe 2018/19

Beispiel:  $(\neg a) U b$ , siehe Tafel T 3.25

Teil 3: unendliche Wörter

Teil 3: unendliche Wörter -Anwendung: Model-Checking in Linearer Temporallo-2019-01 gik (LTL) -Anfangszustände und Akzeptanzkomponente von  $\mathcal{A}_{arphi_{E}}$ 

9:26 bis 9:58

Anfangszustände und Akzeptanzkomponente von A. alle elementaren Formelmenzen, die cor enthalten Menge der akzeptierenden Zustände tellen sicher, dass kein ψ<sub>1</sub> U ψ<sub>2</sub> für immer "aufgeschoben" wird  $\mathcal{F} = \{M_{\varphi_1U\varphi_2} \mid \psi_1 \ U \ \psi_2 \in \operatorname{cl}(\varphi_E)\} \ \operatorname{mit}$  $M_{\psi_1U\psi_2} = \{t \in Q \mid \psi_1 \ U \ \psi_2 \notin t \ \text{oder} \ \psi_2 \in t\}$ 

Intuition: Ein  $t \in M_{n,m_0}$  kommt unendlich oft vor edw

Beispiel: (¬a) U b. siehe Tafel

Büchi-Aut. Abschlusseig. Charakt. Determinismus

### Abschließende Betrachtungen

- |Q| ist exponentiell in  $|\varphi_E|$
- Dafür kann man jetzt beim universellen MCP auf Komplementierung  $\mathcal{A}_{\varphi_E}$  verzichten: Wandle  $\neg \varphi_E$  in Automaten um
- → beide MCP-Varianten in PSpace
  - beide MCP-Varianten sind PSpace-vollständig (aber für bestimmte LTL-Fragmente NP- oder NL-vollständig)

A. Prasad Sistla, Edmund M. Clarke: *The Complexity of Propositional Linear Temporal Logics*. Journal of the ACM 32(3): 733-749 (1985)

Michael Bauland, Martin Mundhenk, Thomas Schneider, Henning Schnoor, Ilka Schnoor, Heribert Vollmer: *The Tractability of Model Checking for LTL: the Good, the Bad, and the Ugly Fragments.* ACM Trans. Comput. Log. 12(2): 13 (2011)

Teil 3: unendliche Wörter

Her Lander Model-Checking in Linearer Temporallogik (LTL)

Abschließende Betrachtungen

Abschließende Betrachtungen

 $\bullet$  |Q| ist exponentiell in  $|\varphi_E|$ 

 beide MCP-Varianten in PSpace
 beide MCP-Varianten sind PSpace-vollständig (aber für bestimmte LTL-Fragmente NP- oder NL-vollständig A. Prassd Sitals. Edmand M. Cather. The Combinator of Procontinual

 Dafür kann man jetzt beim universellen MCP auf Komplementierung A<sub>rix</sub> verzichten: Wandle ¬uzr in Automaten um

Michael Bauland, Martin Mundhenk, Thomas Schneider, Henning Schnolika Schnoor, Heribert Vollmer: The Tractability of Model Checking for

LTL: the Good, the Bad, and the Ugly Fragments. ACM Trans. Comput

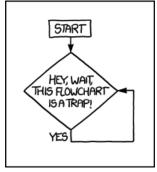
9:58 bis Ende – umblättern und evtl. Prüfungshinweise

Model-Checking

Entscheidungsprobl.

tiv. Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. Model-Checking

### Damit sind wir am Ende dieses Kapitels.



http://xkcd.com/1195 (CC BY-NC 2.5)

# Vielen Dank.

Teil 3: unendliche Wörter

Damit sind wir am Ende dieses Kapitels.



Damit sind wir am Ende dieses Kapitels.

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

### Literatur für diesen Teil (1)



Wolfgang Thomas.

Automata on Infinite Objects.

In J. van Leeuwen (Hrsg.):

Handbook of Theoretical Computer Science.

Volume B: Formal Models and Sematics.

Elsevier, 1990, S. 133-192.

SUB, Zentrale: a inf 400 ad/465-2



Wolfgang Thomas.

Languages, automata, and logic.

In G. Rozenberg and A. Salomaa (Hrsg.:)

Handbook of Formal Languages. Volume 3: Beyond Words.

Springer, 1997, S. 389-455.

SUB, Zentrale: a inf 330/168-3

Teil 3: unendliche Wörter

Literatur für diesen Teil (1)

Welfgarg Thomas.

Automatic on Infinite Objects.
In J. van Leussen (Files).

Hardrodox of Theoretical Computer Science.

Valums III. Fromt Middels and Senetics.

Enwier, 1900, S. 133-142.

SUB, Zentrale, and 400 ad/465-2.

Welfgarg Thomas.

Lenguage, nationatal, and logic.

SUB. Zentrale: a inf 330/168-3

Literatur für diesen Teil (1)

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

### Literatur für diesen Teil (2)



Markus Roggenbach.

Determinization of Büchi Automata.

In E. Grädel, W. Thomas, T. Wilke (Hrsg.):

Automata, Logics, and Infinite Games. LNCS 2500, Springer, 2002, S. 43–60.

Erklärt anschaulich Safras Konstruktion

http://www.cs.tau.ac.il/~rabinoa/Lncs2500.zip

Auch erhältlich auf Anfrage in der BB Mathematik im MZH:

19h inf 001 k/100-2500



Meghyn Bienvenu.

Automata on Infinite Words and Trees.

Vorlesungsskript, Uni Bremen, WS 2009/10. Kapitel 2.

http://www.informatik.uni-bremen.de/tdki/lehre/ws09/automata/automata-notes.pdf

Teil 3: unendliche Wörter

Literatur für diesen Teil (2)

Determination of Biolsh Automata.
In E. Graids W. Thomasa, T. Wilse (Heng.)
Automata, Lugics, and Indiret Garms.
LNCS 2000, Springer, 2002, S. 43–60.
Exidite methanisch Sufran Konstruktion,
http://www.cs.us.ac.il/Trainson/laca2000.stp
Auto-writiktich and Aufrage in der Bill Mathematik im MZH.
10h. saif 001 Julion-2000

Literatur für diesen Teil (2)

Markus Roggenbach.

Meghyn Bienvenss.

Automata on Infinite Words and Trees.

Vorlesungsskript, Unil Bremen, WS 2000/10. Kapitel 2
http://www.informatik.uni-bremen.de/tdki/lehre/set09/

Büchi-Aut. Abschlusseig. Charakt. Determinismus Entscheidungsprobl. *Model-Checking* 

### Literatur für diesen Teil (3)



Christel Baier, Joost-Pieter Katoen.

Principles of Model Checking.

MIT Press 2008.

Abschnitt 4.3 "Automata on Infinite Words" Abschnitt 5.2 "Automata-Based LTL Model Checking" SUB, Zentrale: a inf 440 ver/782, a inf 440 ver/782a



Edmund M. Clarke, Orna Grumberg, Doron A. Peled.

Model Checking.

MIT Press 1999.

Abschnitt 2 "Modeling Systems" bis Mitte S. 14, Abschnitt 2.2.3 +2.3 "Concurrent Programs" und "Example . . . ",

Abschnitt 3 "Temporal Logics",

Abschnitt 9.1 "Automata on Finite and Infinite Words".

SUB, Zentrale: a inf 440 ver/780(6), a inf 440 ver/780(6)a

Teil 3: unendliche Wörter

Literatur für diesen Teil (3)

Christel Baier, Josos-Pister Katoen,
Principles of Model Checking,
MIT Press 2008.
Alaschnitt 4.3 "Autornats on Infinite Words"
Alaschnitt 5.2 "Autornats-Baned LTL Model Checking"

Literatur für diesen Teil (3)

Edmund M. Clarka, Orna Grumberg, Doron A. Peled.

Model Checking.
MIT Press 1990.
MIT Press 1990.
Almobrist 2, Modeling Systems\* his Matte S. 14.
Almobrist 2, 24-23, Concurrent Peograms\* and Example . . . .
Almobrist 3, Temporal Lagger.
Almobrist 3, Temporal Lagger.
SUB Zentrolle a 1844 690 vser7006(5). a 1844 690 vser7006(5).

### Anhang: Beispiel Konsument-Produzent-Problem

- P erzeugt Produkte und legt sie einzeln in einem Lager ab
- K entnimmt Produkte einzeln dem Lager
- Lager fasst maximal 3 Stück

Teil 3: unendliche Wörter

 P erzeugt Produkte und legt sie einzeln in einem Lager ab K entnimmt Produkte einzeln dem Lager Lager fasst maximal 3 Stück —Anhang: Beispiel Konsument-Produzent-Problem

Anhang: Beispiel Konsument-Produzent-Problem

### Anhang: Beispiel Konsument-Produzent-Problem

- P erzeugt Produkte und legt sie einzeln in einem Lager ab
- K entnimmt Produkte einzeln dem Lager
- Lager fasst maximal 3 Stück

#### Modellierung durch endliches Transitionssystem

- Zustände 0, 1, 2, 3, Ü, U
  - 0,1,2,3: im Lager liegen 0,1,2,3 Stück
  - Überschuss: P will ein Stück im vollen Lager ablegen
  - Unterversorgung: K will ein Stück aus leerem Lager nehmen
- Aktionen P, K (P legt ab oder K entnimmt)

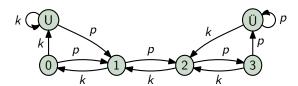
Teil 3: unendliche Wörter

2019-01

Anhang: Beispiel Konsument-Produzent-Problem

Anhang: Beispiel Konsument-Produzent-Problem

- P erzeuet Produkte und legt sie einzeln in einem Lager ab K entnimmt Produkte einzeln dem Lager
- Lager fasst maximal 3 Stück
- Zustände 0, 1, 2, 3, 0, U
- 0,1,2,3: im Lager liegen 0,1,2,3 Stück
- Überschuss: P will ein Stück im vollen Lager ablegen · Unterversorgung: K will ein Stück aus leerem Lazer nehmen Aktionen P. K (P leat ab oder K entnimmt

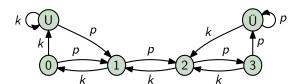


Teil 3: unendliche Wörter

Das Transitionssystem

2019-01-31

└─Das Transitionssystem



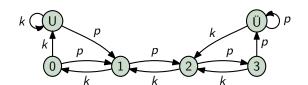
Eingaben in das System: unendliche Zeichenketten über  $\Sigma = \{p, k\}$  (Läufe)

Teil 3: unendliche Wörter

019-01-31

—Das Transitionssystem

Das Transitionssystem  $\begin{picture}(100,0) \put(0,0){\line(1,0){100}} \put(0,0){\line(1,0){100}$ 



**Eingaben in das System:** unendliche Zeichenketten über  $\Sigma = \{p, k\}$ (Läufe)

**Zufriedenheit**: *P* (*K*) möchte . . .

- beliebig oft Produkte produzieren (konsumieren)
- nur endlich oft Überschuss (Unterversorgung) erleiden

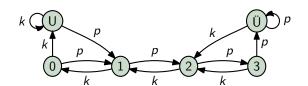
Teil 3: unendliche Wörter

2019-01-31

-Das Transitionssystem

Das Transitionssystem

Zufriedenheit: P (K) möchte .



**Eingaben in das System:** unendliche Zeichenketten über  $\Sigma = \{p, k\}$ (Läufe)

**Zufriedenheit**: *P* (*K*) möchte . . .

- beliebig oft Produkte produzieren (konsumieren)
- nur endlich oft Überschuss (Unterversorgung) erleiden

Lauf, der *P* und *K* zufrieden stellt:

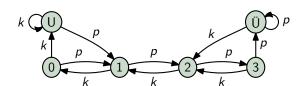
Teil 3: unendliche Wörter

2019-01-31

-Das Transitionssystem

- Zufriedenheit: P (K) möchte

Das Transitionssystem



**Eingaben in das System:** unendliche Zeichenketten über  $\Sigma = \{p, k\}$ (Läufe)

**Zufriedenheit:** P(K) möchte . . .

- beliebig oft Produkte produzieren (konsumieren)
- nur endlich oft Überschuss (Unterversorgung) erleiden

Lauf, der *P* und *K* zufrieden stellt:

$$p^3k^3p^3k^3...$$
 oder  $ppkpkpk...$  oder ...

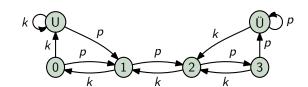
Teil 3: unendliche Wörter

2019-01-31

-Das Transitionssystem

Das Transitionssystem

Zufriedenheit: P (K) möchte



Eingaben in das System: unendliche Zeichenketten über  $\Sigma = \{p, k\}$  (Läufe)

Zufriedenheit: P (K) möchte ...

- beliebig oft Produkte produzieren (konsumieren)
- nur endlich oft Überschuss (Unterversorgung) erleiden

Lauf, der *P* und *K* zufrieden stellt:

 $p^3k^3p^3k^3...$  oder ppkpkpk... oder ...

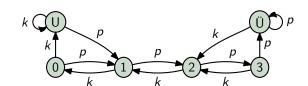
Lauf, der weder P noch K zufrieden stellt:

Teil 3: unendliche Wörter

2019-01-31

-Das Transitionssystem

auf, der P und K zufrieden stellt:  ${}^3k^3p^3k^3\dots$  oder  $ppkpkpk\dots$  oder  $\dots$ 



Eingaben in das System: unendliche Zeichenketten über  $\Sigma = \{p, k\}$  (Läufe)

Zufriedenheit: P (K) möchte ...

- beliebig oft Produkte produzieren (konsumieren)
- nur endlich oft Überschuss (Unterversorgung) erleiden

Lauf, der *P* und *K* zufrieden stellt:

$$p^3k^3p^3k^3...$$
 oder  $ppkpkpk...$  oder ...

Lauf, der weder P noch K zufrieden stellt:  $p^4k^4p^4k^4...$ 

Teil 3: unendliche Wörter

└─Das Transitionssystem

Das Transitionssystem  $\int\limits_{0}^{\infty} \int\limits_{0}^{\infty} \int\limits_{0}^{\infty$ 

Teil 3: unendliche Wörter