http://tinyurl.com/ws1819-autom

Wintersemester 2018/19 Thomas Schneider

AG Theorie der künstlichen Intelligenz (TdKI)

http://tinyurl.com/ws1819-autom

8:30 11.1. \rightarrow Folie 23

TODO: siehe Verbesserungsvorschläge in TODO.txt

Model-Checking CTL Komplementierung Baumautomaten

Überblick

Computation Tree Logic (CTL)

- Grenzen von LTL: kann nicht über Pfade quantifizieren
- Berechnungsbäume und CTL
- Ausdrucksvermögen von LTL und CTL im Vergleich
- Model-Checking mit CTL

Büchi-Automaten auf unendlichen Bäumen

- Definitionen und Beispiele
- äquivalente Automatenmodelle: Muller-, Paritätsautomaten
- Abschlusseigenschaften; Komplementierung von Muller-Automaten Λ



2019-02-01

Teil 4: unendliche Bäume

 Ausdrucksvermögen von LTL und CTL im Vergleich · Model-Checking mit CTL Riichi, Automaten auf unendlichen Räumer –Überblick

 Definitionen und Beispiele w äquivalente Automatenmodelle: Muller-, Paritätsautomaten

Überblick

Computation Tree Logic (CTL)

Δhorblusseigenschaften

u Grenzen von LTL: kann nicht über Pfade quantifizieren Berechnungsbäume und CTL

Komplementierung von Muller-Automaten 🙏

Überblick

Model-Checking mit CTL

■ Komplementierung

Überblick

1 Model-Checking mit CTL

2 Automaten auf unendlichen Bäumen

3 Komplementierung

8:32

2019-02-01

Teil 4: unendliche Bäume

Komplementierung

2019-02-01

- 2 Automaten auf unendlichen Bäumen

Teil 4: unendliche Bäume - Model-Checking mit CTL Und nun ...

- System gegeben als Kripke-Struktur $S = (S, S_0, R, \ell)$
- LTL-Formel φ_F beschreibt Pfade, die Eigenschaft E erfüllen
- Beispiel: "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben." $G(e \rightarrow F \neg e)$ $(e \in AV \text{ steht für "Error"})$
- Umwandlung φ_F in GNBA A_F , der zulässige Pfade beschreibt
- lösen damit Model-Checking-Problem:
 - Gilt E für alle Pfade ab S_0 in S? (universelle Variante)
 - Gilt E für mindestens einen Pfad ab S_0 in S? (existenzielle Variante)

LTL 1977 eingeführt durch Amir Pnueli, 1941-2009. israelischer Informatiker (Haifa, Weizmann-Inst., Stanford, Tel Aviv, New York)

Erinnerung an LTL Teil 4: unendliche Bäume 2019-02-01 u LTL-Formel φ_E beschreibt Pfade, die Eigenschaft E erfüller Model-Checking mit CTL -Erinnerung an LTL (Linear Temporal Logic) ■ Gilt E für mindestens einen Pfad ab S. in S LTL 1977 eingeführt durch Amir Prueil, 1941-2009,

8:33

- System gegeben als Kripke-Struktur $S = (S, S_0, R, \ell)$

"LTL-Formel φ_F beschreibt Pfade, die Eigenschaft E erfüllen"

Nicht ausdrückbar: zu jedem Zeitpunkt ist es immer *möglich*, die Berechnung auf eine gewisse Weise fortzusetzen

Beispiel: "Wenn ein Fehler auftritt, ist es *möglich*, ihn nach endlicher Zeit zu beheben."

 $G(e \rightarrow F \neg e)$ oder $GF \neg e$ sind

- zu stark in Verbindung mit universellem Model-Checking T4.1
- zu schwach in Verbindung mit existenziellem MC T 4.1 Forts.

Teil 4: unendliche Bäume

Model-Checking mit CTL

Grenzen von LTL

"LTL-Formel φ_E beschreibt Pfade, die Eigenschaft E erfüllen"

Nicht ausdrückbar: zu jedem Zeitpunkt ist es immer möglich, die Berechnung auf eine gewisse Weise fortzusetzen Beispiel: "Wenn ein Fehler auftritt, st es möglich, ihn nach endlicher Zeit zu beheben."

Grenzen von LTL

G(e→F¬e) oder GF¬e sind
a zu stark in Verbindung mit universellem Model-Checking T4.1
u zu schwach in Verbindung mit existenziellem MC T4.1 Forts.

8:34 bis 8:46

Abhilfe: Betrachten Berechnungsbäume statt Pfaden

Sei also $S = (S, S_0, R, \ell)$ eine Kripke-Struktur

Berechnungsbaum für $s_0 \in S_0$

- entsteht durch "Auffalten" von S in s_0
- enthält alle unendlichen Pfade, die in s_0 starten d. h.: jeder Zustand $s \in S$ hat als Kinder alle seine Nachfolgerzustände aus S

 ${\mathcal S}$ ist eine endliche Repräsentation aller ∞ Berechnungsbäume

Teil 4: unendliche Bäume

Model-Checking mit CTL

Ein Fall für CTL

(Computation Tree Logic)

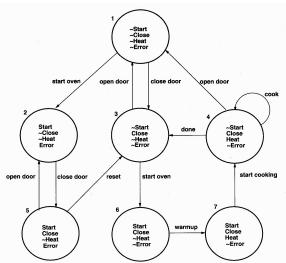
Ein Fall für CTL

(Computation Tree Logic)

S is des G = (5, 5, R, I) was Morphe-Structure Berechnegsphalme statt Platen
Sei and G = (5, 5, R, I) was Morphe-Structure Sein and G = (5, 5, R, I) was Morphe-Structure Sein and G = (5, 5, R, I) was Morphe-Structure Sein and G = (5, 5, R, I) was Morphe-Structure Sein and Sein a

8:46

Komplementierung



aus: E. M. Clarke et al., Model Checking, MIT Press 1999

T 4.2

and EM Chair at all Made Chaing, Bit Then 1909

8:47 bis 8:54

Baumautomaten Komplementierung

CTL intuitiv

Model-Checking CTL

CTL enthält Pfadquantoren A, E:

Operatoren, die über alle oder einige Berechnungen sprechen, die in einem bestimmten Zustand beginnen

Teil 4: unendliche Bäume

Model-Checking mit CTL

CTL intuitiv

CTL enthält Pfadquantoren A, E: Operatoren, die über alle oder einige Berechnungen sprechen, die in einem bestimmten Zustand beginnen

CTL intuitiv

8:54

CTL intuitiv

CTL enthält Pfadquantoren A, E:

Operatoren, die über alle oder einige Berechnungen sprechen, die in einem bestimmten Zustand beginnen

Beispiel: $AGEF \neg e$

Für alle Berechnungen, die hier starten (A), gibt es zu jedem Zeitpunkt in der Zukunft (G) eine Möglichkeit, die Berechnung fortzusetzen (E), so dass irgendwann in der Zukunft (F) kein Fehler auftritt $(\neg e)$

CTL enthält Pfadquantoren A, E:
Operatoren, die über alle oder einige Berechnungen sprechen,
die in einem bestimmten Zustand bezinnen

CTL intuitiv

Belopiol: $AGEF \neg e$ Für alle Berechnungen, die hier starten (A), gibt es zu jedem Zeitpunkt in der Zukunft (G)eine Möglichkeit, die Berechnung fortzusstrzen (E), so dass irgendwaren in der Zukunft (F)kein Feihler auftritt $(\neg e)$

8:54

CTL intuitiv

CTL enthält Pfadquantoren A, E:

Operatoren, die über alle oder einige Berechnungen sprechen, die in einem bestimmten Zustand beginnen

Beispiel: $AGEF \neg e$

Für alle Berechnungen, die hier starten (A), gibt es zu jedem Zeitpunkt in der Zukunft (*G*) eine Möglichkeit, die Berechnung fortzusetzen (E), so dass irgendwann in der Zukunft (F)kein Fehler auftritt $(\neg e)$

CTL 1981 eingeführt durch Edmund M. Clarke, *1945, Informatiker, Carnegie Mellon Univ. (Pittsburgh) E. Allen Emerson, *1954, Informatiker, Univ. of Texas, Austin, USA (beide Turing-Award-Träger 2007)

Teil 4: unendliche Bäume 2019-02-01 Model-Checking mit CTL -CTL intuitiv

CTL intuitiv

CTL enthält Pfadouantoren A. E. Operatoren, die über alle oder einige Berechnungen sprechen,

die in einem bestimmten Zustand bezinnen Beispiel: AGEF-e Für alle Berechnungen, die hier starten (A)

gibt es zu jedem Zeitpunkt in der Zukunft (G) eine Möelichkeit, die Berechnung fortzusetzen (E). so dass irgendwann in der Zukunft (F) kein Fehler auftritt (¬e)

Edmund M. Clarke, *1945. Informatiker, Carnegie Mellon Univ. (Pittsburgh) E. Allen Emerson, *1954. Informatiker, Univ. of Texas, Austin, USA (beide Turing-Award-Träger 2007)

8:54

Zustandsformeln drücken Eigenschaften eines Zustandes aus

$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(p: Aussagenvariable; ζ, ζ_1, ζ_2 : Zustandsformeln; ψ : Pfadformel)

Teil 4: unendliche Bäume Model-Checking mit CTL CTL exakt

CTL exakt Trennung von Zustands- und Pfadformein Zustandsformeln drücken Eigenschaften eines Zustandes aus $\zeta ::= \rho \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$ (p: Aussagenvariable; ζ,ζ,ζ; Zustandsformeln; ψ: Pfadformel)

8:56

Trennung von Zustands- und Pfadformeln:

Zustandsformeln drücken Eigenschaften eines Zustandes aus

$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(p: Aussagenvariable; ζ, ζ_1, ζ_2 : Zustandsformeln; ψ : Pfadformel)

Pfadformeln drücken Eigenschaften eines Pfades aus

$$\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$$

 $(\zeta, \zeta_1, \zeta_2$: Zustandsformeln)

Teil 4: unendliche Bäume Model-Checking mit CTL CTL exakt

Trennung von Zustands- und Pfadformeln Zustandsformeln drücken Eigenschaften eines Zustandes aus $\zeta ::= \rho \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$ Pfadformeln drücken Eigenschaften eines Pfades aus $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 U \zeta_2$

(C. G. Cr. Zuntandsformeln)

8:56

Trennung von Zustands- und Pfadformeln

Trennung von Zustands- und Pfadformeln:

Zustandsformeln drücken Eigenschaften eines Zustandes aus

$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(p: Aussagenvariable; ζ, ζ_1, ζ_2 : Zustandsformeln; ψ : Pfadformel)

Pfadformeln drücken Eigenschaften eines Pfades aus

$$\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$$

 $(\zeta, \zeta_1, \zeta_2$: Zustandsformeln)

- → in zulässigen CTL-Formeln muss
 - jeder Pfadquantor von einem temporalen Operator gefolgt werden
 - jeder temporale Operator direkt einem Pfadquantor folgen

Komplementierung

Teil 4: unendliche Bäume Model-Checking mit CTL

└─CTL exakt

 $\zeta := \rho \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$ Pfadformeln drücken Eigenschaften eines Pfades aus $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 U \zeta_2$

Zustandsformeln drücken Eigenschaften eines Zustandes aus

~ in zulässigen CTL-Formeln muss u jeder Pfadouantor von einem temporalen Operator gefolgt werden v ieder temporale Operator direkt einem Pfadquantor folger

8:56

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp
 $E(p \cup q)$
 $A((p \vee \neg p) \cup q)$
 $E(p \vee AXq)$
 $EX(p \vee AXq)$
 $EF(p \cup q)$
 $EFA(p \cup q)$

EFA(p U a)

9:00 Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$
 $A((p \vee \neg p) \cup q)$
 $E(p \vee AXq)$
 $EX(p \vee AXq)$
 $EF(p \cup q)$
 $EFA(p \cup q)$

Teil 4: unendliche Bäume

Total A: unendliche Bäume

Model-Checking mit CTL

Quiz: zulässige Formeln

Quiz: zulässige Formeln

Quiz: $\{E_i\}_{i \in I}$ Quiz: $\{E_i\}_{i \in I}$ $\{$

9:00

Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

EFA(p U a)

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$
 $E(p \vee AXq)$
 $EX(p \vee AXq)$
 $EF(p \cup q)$
 $EFA(p \cup q)$

9:00

Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$ \checkmark (äquivalent zu AFq)
 $E(p \vee AXq)$
 $EX(p \vee AXq)$
 $EF(p \cup q)$
 $EFA(p \cup q)$

9:00 Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04 Komplementierung

Zur Erinnerung:

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$ \checkmark (äquivalent zu AFq)
 $E(p \vee AXq)$ \checkmark (E nicht gefolgt von F, G, X, U)
 $EX(p \vee AXq)$
 $EF(p \cup q)$
 $EFA(p \cup q)$

Quiz: zulässige Formeln Teil 4: unendliche Bäume Model-Checking mit CTL (ZF) $\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$ (PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 U \zeta_2$ Duizfrage: Welche der folgenden Formeln sind zulässig? └─Quiz: zulässige Formeln ★ (E nicht gefolgt von F, G, X, U) $EX(o \lor AXo$ EF(pUa) EFA(p U a)

9:00

Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

(ZF)
$$\zeta ::= p \mid \zeta_1 \land \zeta_2 \mid \zeta_1 \lor \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$ \checkmark (äquivalent zu AFq)
 $E(p \vee AXq)$ \checkmark (E nicht gefolgt von F, G, X, U)
 $EX(p \vee AXq)$ \checkmark
 $EF(p \cup q)$
 $EFA(p \cup q)$

9:00 Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04 Komplementierung

Zur Erinnerung:

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$ \checkmark (äquivalent zu AFq)
 $E(p \vee AXq)$ \checkmark (E nicht gefolgt von F , G , X , U)
 $EX(p \vee AXq)$ \checkmark
 $EF(p \cup q)$ \checkmark (U folgt nicht E oder A)
 $EFA(p \cup q)$

9:00

Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

(ZF)
$$\zeta ::= p \mid \zeta_1 \wedge \zeta_2 \mid \zeta_1 \vee \zeta_2 \mid \neg \zeta \mid E\psi \mid A\psi$$

(PF) $\psi ::= F\zeta \mid G\zeta \mid X\zeta \mid \zeta_1 \cup \zeta_2$

Quizfrage: Welche der folgenden Formeln sind zulässig?

$$p \wedge q$$
 EFp AXp \checkmark
 $E(p \cup q)$ \checkmark
 $A((p \vee \neg p) \cup q)$ \checkmark (äquivalent zu AFq)
 $E(p \vee AXq)$ \checkmark (E nicht gefolgt von F, G, X, U)
 $EX(p \vee AXq)$ \checkmark
 $EF(p \cup q)$ \checkmark (U folgt nicht E oder A)
 $EFA(p \cup q)$ \checkmark

9:00

Als Aufgabe, 2 min Nachdenken, 2 min Auflösung → bis 9:04

CTL-Semantik

CTI -Formeln werden über Zuständen und Pfaden von Kripke-Strukturen $S = (S, S_0, R, \ell)$ interpretiert

Schreibweisen

- $s \models \zeta$ für Zustände $s \in S$ und Zustandsformeln ζ
- $\pi \models \psi$ für Pfade π und Pfadformeln ψ

Hilfsbegriffe

- Paths(s): Menge aller Pfade, die in Zustand s beginnen
- $\bullet \pi[i]$: i-ter Zustand auf dem Pfad π d. h. wenn $\pi = s_0 s_1 s_2 \dots$, dann $\pi[i] = s_i$

Teil 4: unendliche Bäume Model-Checking mit CTL

-CTL-Semantik

 s |= (für Zustände s ∈ S und Zustandsformeln (π l= ψ für Pfade π und Pfadformein ψ

CTL-Formein werden über Zuständen und Pfaden von

Kripke-Strukturen $S = (S, S_c, R, \ell)$ interpretiert

Menze aller Pfade, die in Zustand s beginner

9:04

2019-02-01

Sei
$$S = (S, S_0, R, \ell)$$
 eine Kripke-Struktur.

Definition 4.1

Erfülltheit von Zustandsformeln in Zuständen $s \in S$

$$s \models p$$
 falls $p \in \ell(s)$, für alle $p \in AV$

$$s \models \neg \zeta$$
 falls $s \not\models \zeta$

$$s \models \zeta_1 \land \zeta_2$$
 falls $s \models \zeta_1$ und $s \models \zeta_2$ (analog für $\zeta_1 \lor \zeta_2$)

$$s \models E\psi$$
 falls $\pi \models \psi$ für ein $\pi \in \mathsf{Paths}(s)$

$$s \models A\psi$$
 falls $\pi \models \psi$ für alle $\pi \in \mathsf{Paths}(s)$

Teil 4: unendliche Bäume - Model-Checking mit CTL └─CTL-Semantik

Sei $S = (S, S_0, R, \ell)$ eine Kripke-Struktu Erfülltheit von Zustandsformeln in Zuständen $s \in S$ falls $\pi \models \psi$ für alle $\pi \in Paths(s)$

9:06

Sei $S = (S, S_0, R, \ell)$ eine Kripke-Struktu

Erfülltheit von Zustandsformeln in Zuständen $s \in S$

Sei $S = (S, S_0, R, \ell)$ eine Kripke-Struktur.

Definition 4.1

Erfülltheit von Zustandsformeln in Zuständen $s \in S$

$$s \models p$$
 falls $p \in \ell(s)$, für alle $p \in AV$

$$s \models \neg \zeta$$
 falls $s \not\models \zeta$

$$s \models \zeta_1 \land \zeta_2$$
 falls $s \models \zeta_1$ und $s \models \zeta_2$ (analog für $\zeta_1 \lor \zeta_2$)

$$s \models E\psi$$
 falls $\pi \models \psi$ für ein $\pi \in \mathsf{Paths}(s)$

$$s \models A\psi$$
 falls $\pi \models \psi$ für alle $\pi \in \mathsf{Paths}(s)$

Erfülltheit von Pfadformeln in Pfaden π in S

$$\pi \models X \zeta \qquad \text{ falls } \quad \pi[1] \models \zeta \qquad \qquad \text{(analog für } F \zeta \text{ und } G \zeta \text{)}$$

$$\pi \models \zeta_1 U \zeta_2$$
 falls $\pi[j] \models \zeta_2$ für ein $j \geqslant 0$

und $\pi[k] \models \zeta_1$ für alle k mit $0 \leqslant k < j$

Teil 4: unendliche Bäume - Model-Checking mit CTL └─CTL-Semantik

falls $s \models \zeta_1 \text{ und } s \models \zeta_2$ (analog für $\zeta_1 \vee \zeta_2$ falls $\pi \models \psi$ für ein $\pi \in Paths(s)$ falls π l= ψ für alle π ∈ Paths(s) $\pi \models \zeta_1 U \zeta_2$ falls $\pi[j] \models \zeta_2$ für ein $j \ge 0$ und $\pi[k] \models \zeta_1$ für alle k mit $0 \leqslant k < j$

9:06

Sei $S = (S, S_0, R, \ell)$ eine Kripke-Struktur.

Definition 4.1

Erfülltheit von Zustandsformeln in Zuständen $s \in S$

$$s \models p$$
 falls $p \in \ell(s)$, für alle $p \in AV$

$$s \models \neg \zeta$$
 falls $s \not\models \zeta$

$$s \models \zeta_1 \land \zeta_2$$
 falls $s \models \zeta_1$ und $s \models \zeta_2$ (analog für $\zeta_1 \lor \zeta_2$)

$$s \models E\psi$$
 falls $\pi \models \psi$ für ein $\pi \in \mathsf{Paths}(s)$
 $s \models A\psi$ falls $\pi \models \psi$ für alle $\pi \in \mathsf{Paths}(s)$

Erfülltheit von Pfadformeln in Pfaden
$$\pi$$
 in \mathcal{S}

$$\pi \models X\zeta \qquad \text{falls} \quad \pi[1] \models \zeta \qquad \text{(analog für } F\zeta \text{ und } G\zeta)$$

$$\pi \models \zeta_1 \ U \ \zeta_2$$
 falls $\pi[j] \models \zeta_2$ für ein $j \geqslant 0$
und $\pi[k] \models \zeta_1$ für alle k mit $0 \leqslant k < j$

Schreiben $S \models \zeta$ falls $s_0 \models \zeta$ für alle $s_0 \in S_0$

Teil 4: unendliche Bäume - Model-Checking mit CTL └─CTL-Semantik

Sei $S = (S, S_0, R, \ell)$ eine Kripke-Struktu Erfülltheit von Zustandsformeln in Zuständen $s \in S$ falls s |= C1 und s |= C2 (analog für C1 V C2 falls $\pi \models \psi$ für ein $\pi \in Paths(s)$ falls π |= ψ für alle π ∈ Paths(s $\pi \models \zeta_1 U \zeta_2$ falls $\pi[j] \models \zeta_2$ für ein $j \ge 0$ und $\pi[k] \models \zeta_1$ für alle k mit $0 \le k < j$

Schreiben $S \models \zeta$ falls $s_0 \models \zeta$ für alle $s_0 \in S_0$

CTL-Semantik

9:06

Model-Checking CTL Baumautomaten

Zurück zu unseren Beispielen: Spezifikationen in CTL

Komplementierung

Beispiel Nebenläufigkeit

• Beide Teilprogramme sind nie zugleich im kritischen Bereich.

$$AG \neg (p_{12} \wedge p_{22})$$

 $(p_i \in AV: "Programmzähler in Zeile i")$

Zurück zu unseren Beispielen: Spezifikationen in CTL Teil 4: unendliche Bäume - Model-Checking mit CTL

Zurück zu unseren Beispielen: Spezifikationen in

9:10

CTL

Model-Checking CTL Baumautomaten Komplementierung

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Nebenläufigkeit

• Beide Teilprogramme sind nie zugleich im kritischen Bereich.

$$AG \neg (p_{12} \land p_{22})$$
 $(p_i \in AV: "Programmzähler in Zeile $i")$$

Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich.
 AGAFp₁₂ ∧ AGAFp₂₂

Zurück zu unseren Beispielen: Spezifikationen in CTL

9:10

2019-02-01

Automatentheorie u. i. A. WiSe 2018/19

Model-Checking CTL Komplementierung Baumautomaten

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Nebenläufigkeit

• Beide Teilprogramme sind nie zugleich im kritischen Bereich.

$$AG \neg (p_{12} \land p_{22})$$
 $(p_i \in AV: "Programmzähler in Zeile $i")$$

• Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich. $AGAFp_{12} \wedge AGAFp_{22}$

• Jedes Teilprog. kann beliebig oft in seinen kB kommen.

```
AGEFp_{12} \wedge AGEFp_{22}
```

Zurück zu unseren Beispielen: Spezifikationen in CTL Teil 4: unendliche Bäume Model-Checking mit CTL -Zurück zu unseren Beispielen: Spezifikationen in

Jedes Teilprog, kann beliebig oft in seinen kB kommen

9:10

CTL

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Nebenläufigkeit

- Beide Teilprogramme sind nie zugleich im kritischen Bereich.
 - $AG \neg (p_{12} \wedge p_{22})$ $(p_i \in AV: "Programmzähler in Zeile i")$
- Jedes Teilprog. kommt beliebig oft in seinen krit. Bereich. $AGAFp_{12} \wedge AGAFp_{22}$
- Jedes Teilprog. kann beliebig oft in seinen kB kommen. $AGEFp_{12} \wedge AGEFp_{22}$

Liveness properties:

 $AG\zeta$ besagt: " ζ ist in allen Berechnungen immer wahr"

 $AGAF\zeta$ besagt: " ζ ist in allen Berechnungen ∞ oft wahr"

AGEF ζ besagt: ", jede begonnene Berechnung kann so fortgesetzt werden, dass ζ irgendwann wahr wird."

Zurück zu unseren Beispielen: Spezifikationen in CTL Teil 4: unendliche Bäume a Beide Teilgrogramme sind nie zugleich im kritischen Bereich Model-Checking mit CTL –Zurück zu unseren Beispielen: Spezifikationen in

w Jedes Teilprog, kann beliebig oft in seinen kB kommen

AGζ besagt: "ζ ist in allen Berechnungen immer wahr AGAF (besagt: ... ist in allen Berechnungen ∞ oft wahr"

9:10

CTL

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Mikrowelle

• "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben."

$$AG(e \rightarrow AF \neg e)$$

 $(e \in \mathsf{AV} \mathsf{\ steht\ f\"{u}r\ "Error"})$

Komplementierung



9:12

Beispiel Mikrowelle

• "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben."

$$AG(e \rightarrow AF \neg e)$$

$$(e \in AV \text{ steht für "Error"})$$

• "Wenn Fehler auftritt, *kann* er nach endl. Z. behoben werden" $AG(e \rightarrow EF \neg e)$

—Zurück zu unseren Beispielen: Spezifikationen in CTL

9:12

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Mikrowelle

• "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben."

$$AG(e \rightarrow AF \neg e)$$

$$(e \in AV \text{ steht für "Error"})$$

• "Wenn Fehler auftritt, kann er nach endl. Z. behoben werden" $AG(e \rightarrow EF \neg e)$

• "Wenn die Mikrowelle gestartet wird, beginnt sie nach endlicher Zeit zu heizen."

$$AG(s \rightarrow AFh)$$

 $(s, h \in AV \text{ stehen für "Start" bzw. "Heat"})$

Teil 4: unendliche Bäume Model-Checking mit CTL

Zurück zu unseren Beispielen: Spezifikationen in

. Wenn Fehler auftritt, kann er nach endl. Z. behoben werder . Wenn die Mikrowelle gestartet wird.

Zurück zu unseren Beispielen: Spezifikationen in CTL

9:12

CTL

Zurück zu unseren Beispielen: Spezifikationen in CTL

Beispiel Mikrowelle

• "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben."

$$AG(e \rightarrow AF \neg e)$$

$$(e \in AV \text{ steht für "Error"})$$

- "Wenn Fehler auftritt, *kann* er nach endl. Z. behoben werden" $AG(e \rightarrow EF \neg e)$
- "Wenn die Mikrowelle gestartet wird, beginnt sie nach endlicher Zeit zu heizen."

$$AG(s \to AFh)$$
 (s, $h \in AV$ stehen für "Start" bzw. "Heat")

• "Wenn die Mikrowelle gestartet wird, ist es möglich, dass sie nach endlicher Zeit zu heizen beginnt." $AG(s \rightarrow EFh)$

a "Wenn Fishler auftritt, kann er nach endl. Z. behoben werder $AG(e \rightarrow EF - e)$ u "Wenn die Mitzowelle gestartet wird, beginnt sie nach endlicher Zeit zu beizen." $AG(s \rightarrow AFh) \qquad (s, h \in R seinen für "Start" hzw. "Heat$

Zurück zu unseren Beispielen: Spezifikationen in CTL

 "Wenn die Mikrowelle gestartet wird, ist es möglich, dass sie nach endlicher Zeit zu heizen beginn AG(s → EFh)

9:12

CTL

Beispiel Mikrowelle

• "Wenn Fehler auftritt, ist er nach endlicher Zeit behoben."

$$AG(e \rightarrow AF \neg e)$$

 $(e \in AV \text{ steht für "Error"})$

Komplementierung

- "Wenn Fehler auftritt, kann er nach endl. Z. behoben werden" $AG(e \rightarrow EF \neg e)$
- "Wenn die Mikrowelle gestartet wird, beginnt sie nach endlicher Zeit zu heizen."

$$AG(s \rightarrow AFh)$$

 $(s, h \in AV \text{ stehen für "Start" bzw. "Heat"})$

• "Wenn die Mikrowelle gestartet wird, ist es möglich, dass sie nach endlicher Zeit zu heizen beginnt." $AG(s \rightarrow EFh)$

Progress properties: $AG(\zeta_1 \to AF\zeta_2)$, $AG(\zeta_1 \to EF\zeta_2)$ bedeuten:

Wann immer ζ_1 eintritt, ist nach endlicher Zeit ζ_2 "garantiert"

Teil 4: unendliche Bäume Model-Checking mit CTL

> Zurück zu unseren Beispielen: Spezifikationen in CTL

. Wenn Fehler auftritt, kann er nach endl. Z. behoben werder

. Wenn die Mikrowelle gestartet wird.

ist es mörlich, dass sie nach endlicher Zeit zu heizen beginn $AG(s \rightarrow EFh)$

Progress properties: $AG(\zeta_1 \rightarrow AF(\zeta_2), AG(\zeta_1 \rightarrow EF(\zeta_2))$ bedeute Wann immer () eintritt, ist nach endlicher Zeit () "garantiert

9:12

Komplementierung

Definition 4.2

Seien ζ eine CTL-Zustandsformel und φ eine LTL-Formel.

 ζ und φ sind äquivalent, geschrieben $\zeta \equiv \varphi$, wenn für alle Kripke-Strukturen $S = (S, S_0, R, \ell)$ gilt:

$$\mathcal{S} \models \zeta$$
 gdw. $\mathcal{S} \models \varphi$

Zur Erinnerung:

- $S \models \zeta$, wenn $s_0 \models \zeta$ für alle $s_0 \in S_0$
- $S \models \varphi$, wenn $\pi, 0 \models \varphi$ für alle $\pi \in Paths(s_0)$ und alle $s_0 \in S_0$

Teil 4: unendliche Bäume Model-Checking mit CTL Ausdrucksstärke von CTL versus LTL

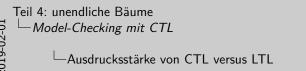
 ζ und φ sind liquivalent, geschrieben $\zeta \equiv \varphi$, wenn für alle Kripke-Strukturen $S = (S, S_0, R, \ell)$ gilt: $S \models \zeta \quad \text{gdw.} \quad S \models \varphi$ a S In C. wenn so In C für alle so ∈ So $\quad \textbf{a} \; \mathcal{S} \models \varphi, \text{ wenn } \pi, 0 \models \varphi \; \text{für alle } \pi \in \mathsf{Paths}(s_0) \; \text{und alle } s_0 \in \mathcal{S}_0$

9:15: 5 min Pause, dann 2 min für Folie

Lemma 4.3 AFAGp ≠ FGp

Lemma 4.3

 $AFAGp \not\equiv FGp$



Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur S: - s_0 s_1 s_2

• alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp

Teil 4: unendliche Bäume

-- Model-Checking mit CTL

-- Ausdrucksstärke von CTL versus LTL

Ausdrucksstärke von CTL versus LTL

[Limma 4.3]

ARGG ≠ FGp

Breach: Betrachte Kripke Struktur S: →

** silte Plade ≈ C Puble(s) wführe FGp

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur S: \longrightarrow (s_0) \longrightarrow (s_1) \longrightarrow (s_2)

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

 Ausdrucksstärke von CTL versus LTL

[Lemma 4.3]

ARAGE # FGP

Breek. Betrachte Krighe Strekter S:

a lite Pfade # © Pathol(a) william FGP

a late S | ARAGE

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur S: \longrightarrow $(s_0)^p$ (s_1) $(s_2)^p$

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

$$s_0 s_1 s_2^{\omega} \not\models Gp$$
 wegen $p \notin \ell(s_1)$

Teil 4: unendliche Bäume

-- Model-Checking mit CTL

-- Ausdrucksstärke von CTL versus LTL

Ausdrucksstärke von CTL versus LTL

Samma, 8.1

ATROG # EN

Bronch. Bitrischte Krighe Straktur S:

• All Pfalt r C Frankful) wilden FCp

• aller S # ATROG

• weep p # (n)

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur $S: \longrightarrow (s_0) \longrightarrow (s_1)$

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

$$s_0 s_1 s_2^{\omega} \not\models Gp$$
 wegen $p \notin \ell(s_1)$
 $\Rightarrow s_0 \not\models AGp$ weil $s_0 s_1 s_2^{\omega} \in \mathsf{Paths}(s_0)$

Teil 4: unendliche Bäume - Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL Ausdrucksstärke von CTL versus LTL Beweis. Betrachte Kripke-Struktur S: +65+65+65 a alle Pfade $\pi \in Paths(s_0)$ erfüllen FGp

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur
$$S$$
: \longrightarrow (s_0) \longrightarrow (s_1) \longrightarrow (s_2)

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

$$s_0 s_1 s_2^{\omega} \not\models Gp$$
 wegen $p \notin \ell(s_1)$
 $\Rightarrow s_0 \not\models AGp$ weil $s_0 s_1 s_2^{\omega} \in \mathsf{Paths}(s_0)$
 $\Rightarrow s_0^{\omega} \not\models FAGp$ weil s_0^{ω} nur aus s_0 besteht

Teil 4: unendliche Bäume - Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL

Beweis. Betrachte Kripke-Struktur S: +65+65+65 a alle Pfade $\pi \in Paths(s_0)$ erfüllen FGp

9:22

Komplementierung

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur S: \longrightarrow (s_0) \longrightarrow (s_1) \longrightarrow (s_2)

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

$$s_0s_1s_2^{\omega} \not\models Gp$$
 wegen $p \notin \ell(s_1)$
 $\Rightarrow s_0 \not\models AGp$ weil $s_0s_1s_2^{\omega} \in Paths(s_0)$
 $\Rightarrow s_0^{\omega} \not\models FAGp$ weil s_0^{ω} nur aus s_0 besteht
 $\Rightarrow s_0 \not\models AFAGp$ weil $s_0^{\omega} \in Paths(s_0)$

 $\begin{aligned} & \operatorname{Gamma}(A), \\ & \operatorname{AliGo}(p) \notin Fig. \\ & \text{ with Pinks} \in \operatorname{Crabbin}(a) \text{ within Fig.} \\ & \text{ with Pinks} \in \operatorname{Crabbin}(a) \text{ within Fig.} \\ & \text{ with } G \mid_{A} \operatorname{AliGo}(a) \\ & \text{ within } G \mid_{A} \operatorname{AliGo}$

Ausdrucksstärke von CTL versus LTL

Lemma 4.3

 $AFAGp \not\equiv FGp$

Beweis. Betrachte Kripke-Struktur
$$S$$
: $\rightarrow S_0$ $\rightarrow S_1$ $\rightarrow S_2$

- alle Pfade $\pi \in \mathsf{Paths}(s_0)$ erfüllen FGp
- aber $S \not\models AFAGp$:

$$s_0s_1s_2^{\omega} \not\models Gp$$
 wegen $p \notin \ell(s_1)$
 $\Rightarrow s_0 \not\models AGp$ weil $s_0s_1s_2^{\omega} \in Paths(s_0)$
 $\Rightarrow s_0^{\omega} \not\models FAGp$ weil s_0^{ω} nur aus s_0 besteht
 $\Rightarrow s_0 \not\models AFAGp$ weil $s_0^{\omega} \in Paths(s_0)$

Teil 4: unendliche Bäume

— Model-Checking mit CTL

— Ausdrucksstärke von CTL versus LTL

Lemma 4.4

Sei ζ eine CTL-Zustandsformel und ζ' die LTL-Formel, die man durch Entfernen aller Pfadquantoren aus ζ erhält. Dann gilt:

 $\zeta \equiv \zeta'$ oder es gibt keine zu ζ äquivalente LTL-Formel.

Ohne Beweis. (Clarke, Draghicescu 1988)

Teil 4: unendliche Bäume - Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL

9:26 bis 9:38

Ohne Beweis. (Clarke, Draghicescu 1988)

Sei (eine CTL-Zustandsformel und (' die LTL-Formel, die man durch Entfernen aller Pfadquantoren aus (erhält. Dann gilt: ⟨ ≡ ⟨ oder es gibt keine zu ⟨ äguivalente LTL-Formel

Lemma 4.4

Sei ζ eine CTL-Zustandsformel und ζ' die LTL-Formel, die man durch Entfernen aller Pfadquantoren aus ζ erhält. Dann gilt:

 $\zeta \equiv \zeta'$ oder es gibt keine zu ζ äquivalente LTL-Formel.

Ohne Beweis. (Clarke, Draghicescu 1988)

Lemma 4.5

- (1) Es gibt keine zu AFAGp äquivalente LTL-Formel.
- (2) Es gibt keine zu *FGp* äquivalente CTL-Zustandsformel.

Ohne Bowis. (Clarke, Draghicescu 1988)

Lemma 4.5

(1) Es gibt keine zu AFAGp äquivalente LTL-Formel.

(2) Es gibt keine zu FGp äquivalente CTL-Zustandsformel.

Sei C eine CTL-Zustandsformel und C' die LTL-Formel, die mar

durch Entfernen aller Pfadquantoren aus ζ erhält. Dann gilt: $\zeta \equiv \zeta'$ oder es gibt keine zu ζ äquivalente LTL-Forme

Ausdrucksstärke von CTL versus LTL

9:26 bis 9:38

(1) folet aus Lemmas 4.3 und 4.4

Ausdrucksstärke von CTL versus LTL

Lemma 4.4

Sei ζ eine CTL-Zustandsformel und ζ' die LTL-Formel, die man durch Entfernen aller Pfadquantoren aus ζ erhält. Dann gilt:

 $\zeta \equiv \zeta'$ oder es gibt keine zu ζ äquivalente LTL-Formel.

Ohne Beweis. (Clarke, Draghicescu 1988)

Lemma 4.5

- (1) Es gibt keine zu AFAGp äquivalente LTL-Formel.
- (2) Es gibt keine zu *FGp* äquivalente CTL-Zustandsformel.

Beweis.

(1) folgt aus Lemmas 4.3 und 4.4

(= q" oder es gibt heine zu (Squivalente LTL-Formel.

Ohne Bowels. (Clarke, Draghicsecu 1988)

Lemma 4-5.

(1) Es gibt heine zu AFAGo Squivalente LTL-Formel.

(2) Es gibt heine zu AFGo Squivalente LTL-Formel.

Bowels.

Sei C eine CTL-Zustandsformel und C' die LTL-Formel, die mar

durch Entfernen aller Pfadquantoren aus (erhält. Dann gilt:

9:26 bis 9:38

Lemma 4.4

Sei ζ eine CTL-Zustandsformel und ζ' die LTL-Formel, die man durch Entfernen aller Pfadquantoren aus ζ erhält. Dann gilt:

$$\zeta \equiv \zeta'$$
 oder es gibt keine zu ζ äquivalente LTL-Formel.

Ohne Beweis. (Clarke, Draghicescu 1988)

Lemma 4.5

- (1) Es gibt keine zu AFAGp äquivalente LTL-Formel.
- (2) Es gibt keine zu FGp äquivalente CTL-Zustandsformel.

Beweis.

- (1) folgt aus Lemmas 4.3 und 4.4
- (2) siehe Tafel

T 4.3

Teil 4: unendliche Bäume Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL

Sei C eine CTL-Zustandsformel und C' die LTL-Formel, die mar durch Entfernen aller Pfadquantoren aus (erhält. Dann gilt: ⟨ ≡ ⟨ oder es eibt keine zu ⟨ ăquivalente LTL-Forme (1) Es gibt keine zu AFAGp äquivalente LTL-Formel. (2) Es gibt keine zu FGp äquivalente CTL-Zustandsformel (1) folet aus Lemmas 4.3 und 4.4

T4.3 🗆

Ausdrucksstärke von CTL versus LTL

(2) siehe Tafel

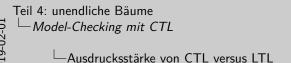
9:26 bis 9:38

Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Forme

Auch progress properties sind **nicht** in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.



Sei $\zeta = AG(\rho \to EF\rho')$. Es gibt keine zu ζ äquivalente LTL-Forme Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Forme

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Betrachte Kripke-Strukturen



Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Forme

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Betrachte Kripke-Strukturen



Dann gilt $S_1 \models \zeta$.

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Betrachte Kripke-Strukturen



Dann gilt $S_1 \models \zeta$.

Also auch $S_1 \models \varphi$.

Teil 4: unendliche Bäume Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL

Dann gilt $S_1 \models \zeta$. Also auch $S_1 \models \varphi$

Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Forme

9:38

Komplementierung

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Betrachte Kripke-Strukturen



Dann gilt $S_1 \models \zeta$.

Also auch $S_1 \models \varphi$.

Da Paths $(s_2) \subset \text{Paths}(s_0)$, gilt auch $S_2 \models \varphi$.

 S_1 \longrightarrow S_2 \longrightarrow S_2 \longrightarrow S_3 \longrightarrow S_4 \longrightarrow S_4 Dann gilt $S_1 \models \varphi$.

Also auch $S_1 \models \varphi$.

Da Pathol $(s_0) \subseteq P$ athol (s_0) , gilt auch $S_2 \models \varphi$.

Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Form

Auch progress properties sind nicht in LTL ausdrückbar:

Lemma 4.6

Sei $\zeta = AG(p \to EFp')$. Es gibt keine zu ζ äquivalente LTL-Formel.

Beweis. Angenommen, es gebe LTL-Formel $\varphi \equiv \zeta$.

Betrachte Kripke-Strukturen

$$S_1 \longrightarrow S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow S_2$$

Dann gilt $S_1 \models \zeta$.

Also auch $S_1 \models \varphi$.

Da Paths $(s_2) \subseteq \text{Paths}(s_0)$, gilt auch $S_2 \models \varphi$.

Aber offensichtlich $S_2 \not\models \zeta$. 4

Teil 4: unendliche Bäume Model-Checking mit CTL -Ausdrucksstärke von CTL versus LTL

Dann gilt $S_1 \models \zeta$. Also auch $S_1 \models \varphi$ Da Paths(s₂) \subset Paths(s₀), gilt auch $S_2 \models \varphi$. Aber offensichtlich St lé C. 5

Sei $\zeta = AG(p \rightarrow EFp')$. Es gibt keine zu ζ liquivalente LTL-Form

Erweiterung von LTL und CTL: CTL*
CTL*: 1986 von E. A. Emerson und J. Y. Halpern (*1953, Inform., Cornell)



Standard-Algorithmus ("bottom-up labelling", ohne Automaten):

Eingabe: Kripke-Str. S, Zust. s_0 , CTL-Zustandsformel ζ Frage: $s_0 \models \zeta$?

Vorgehen:

Teil 4: unendliche Bäume - Model-Checking mit CTL

☐ Model-Checking für CTL (Skizze)

Model-Checking für CTL (Skizze)

Model-Checking für CTL (Skizze)

Standard-Algorithmus ("bottom-up labelling", ohne Automaten):

Eingabe: Kripke-Str. S, Zust. s_0 , CTL-Zustandsformel ζ Frage: $s_0 \models \zeta$?

Vorgehen:

• Stelle ζ als Baum dar

(Bsp. siehe Tafel)

T 4.4

Teil 4: unendliche Bäume - Model-Checking mit CTL

☐ Model-Checking für CTL (Skizze)

Model-Checking für CTL (Skizze)

Model-Checking für CTL (Skizze)

Standard-Algorithmus ("bottom-up labelling", ohne Automaten):

Eingabe: Kripke-Str. S, Zust. s_0 , CTL-Zustandsformel ζ Frage: $s_0 \models \zeta$?

Vorgehen:

- Stelle ζ als Baum dar (Bsp. siehe Tafel) T 4.4
- Gehe Baum von unten nach oben durch und markiere Zustände s in S mit der jeweiligen Teilformel, wenn sie in s erfüllt ist

 T 4.4 Forts.

Teil 4: unendliche Bäume

Model-Checking mit CTL

Model-Checking für CTL (Skizze)

Implies (virginis-tr. S, Lint. N), C11-Lintentrovermin (
regisher:

Stelle (als Baum dar (Bop siehe Tafel) T4.4

Caleb Baum von unten nach oben durch

and markiera Zustände s in S mit der jeweiligen Teilformst,

wenn sie in a erfüllt ist.

T4.4 Forts

Model-Checking für CTL (Skizze)

Model-Checking für CTL (Skizze)

Standard-Algorithmus ("bottom-up labelling", ohne Automaten):

Eingabe: Kripke-Str. S, Zust. s_0 , CTL-Zustandsformel ζ Frage: $s_0 \models \zeta$?

Vorgehen:

- Stelle ζ als Baum dar (Bsp. siehe Tafel) T 4.4
- Gehe Baum von unten nach oben durch und markiere Zustände s in S mit der jeweiligen Teilformel, T 4.4 Forts. wenn sie in s erfüllt ist
- Akzeptiere gdw. s_0 mit ζ markiert ist

Teil 4: unendliche Bäume Model-Checking mit CTL ☐ Model-Checking für CTL (Skizze)

a Stelle (als Baum dar g Gehe Baum von unten nach oben durch

Akzeptiere edw. so mit (markiert ist

Model-Checking für CTL (Skizze)

☐ Model-Checking für CTL (Skizze)

Standard-Algorithmus ("bottom-up labelling", ohne Automaten):

Eingabe: Kripke-Str. S, Zust. s_0 , CTL-Zustandsformel ζ Frage: $s_0 \models \zeta$?

Vorgehen:

- Stelle ζ als Baum dar (Bsp. siehe Tafel) T 4.4
- Gehe Baum von unten nach oben durch und markiere Zustände s in S mit der jeweiligen Teilformel, wenn sie in s erfüllt ist

 T 4.4 Forts.
- Akzeptiere gdw. s_0 mit ζ markiert ist

Komplexität: P-vollständig (LTL-MC: PSpace-vollständig)

Dafür ist CTL-SAT ExpTime-vollständig (LTL-SAT: PSpace-vollst.).

w Akzeptiere gdw. s₀ mit ζ markiert ist

Model-Checking für CTL (Skizze)

Model-Checking für CTL mit Baumautomaten

Automatenbasierte Entscheidungsprozedur für CTL

... basiert auf alternierenden Baumautomaten

(Erweiterung des Begriffs der nichtdeterminist. Baumautomaten; siehe Teil 5 der Vorlesung)



└─Model-Checking für CTL mit Baumautomaten

9:53 bis 9:55, 5 min Reserve.

CTL*-MC ist PSpace-vollst., CTL*-SAT 2ExpTime-vollst. Siehe Baier & Katoen S. 430.

Model-Checking für CTL mit Baumautomaten

Automatenbasierte Entscheidungsprozedur für CTL

basiert auf alternierenden Baumautomaten

(Erweiterung des Begriffs der nichtdeterminist. Baumautomaten; siehe Teil 5 der Vorlesung)

Verwandt:

Automatenbasierte Entscheidungsprozedur für CTL*-Erfüllbarkeit

- basiert auf nichtdeterministischen Rabin-Baumautomaten
- technisch aufwändige Konstruktion
- hier nicht behandelt

Teil 4: unendliche Bäume

Model-Checking mit CTL

└─Model-Checking für CTL mit Baumautomaten

Model-Checking für CTL mit Baumautomaten

basiert auf alternierenden Baumautematen
 (Erweiterung des Begriffs der nichtdeterminist. Baumautemates
 inte Tall 6 des Medicense)

Automatenbasierte Entscheidungsprozedur für CTL*-Erfüllbar

- technical aufmindian Kenstruktion

w hier nicht behandelt

9:53 bis 9:55, 5 min Reserve.

CTL*-MC ist PSpace-vollst., CTL*-SAT 2ExpTime-vollst. Siehe Baier & Katoen S. 430.

2019-02-01

Model-Checking für CTL mit Baumautomaten

Automatenbasierte Entscheidungsprozedur für CTL

... basiert auf alternierenden Baumautomaten

(Erweiterung des Begriffs der nichtdeterminist. Baumautomaten; siehe Teil 5 der Vorlesung)

Verwandt:

Automatenbasierte Entscheidungsprozedur für CTL*-Erfüllbarkeit

- basiert auf nichtdeterministischen Rabin-Baumautomaten
- technisch aufwändige Konstruktion
- hier nicht behandelt

Es folgt:

Überblick "klassische" nichdeterministische Baumautomaten

Teil 4: unendliche Bäume

Model-Checking mit CTL

└─Model-Checking für CTL mit Baumautomaten

Model-Checking für CTL mit Baumautomaten

Automatenbasierte Entscheidungsprozedur für CTL
... basiert auf alternierenden Baumautomaten
(Erweiterunz des Bezriffs der nichtdeterminist. Baumautomaten:

Oberblick Vassische" nichdeterministische Raumautomate

Verwand:
Automatenbasierte Entscheidungsprozedur für CTL*-Erfüllbarks

basiert auf nichtdeterministrichen Rab
 technisch aufwändige Konstruktion

w hier nicht behandelt

9:53 bis 9:55, 5 min Reserve.

CTL*-MC ist PSpace-vollst., CTL*-SAT 2ExpTime-vollst. Siehe Baier & Katoen S. 430.

2019-02-01

Und nun ...

2019-02-01

Und nun ...

Komplementierung

1 Model-Checking mit CTL

- 2 Automaten auf unendlichen Bäumen
- 2 Komplementierung

Baumautomaten: Grundbegriffe

Betrachten unendlichen vollständigen Binärbaum

- Positionen: alle Wörter aus $\{0,1\}^*$
- jeder Knoten p hat linkes und rechtes Kind: p0, p1
- Tiefe von Knoten p: |p|
- Ebene k: alle Knoten der Tiefe k
- p_2 ist Nachfolger von p_1 , geschrieben $p_1 \sqsubseteq p_2$, wenn $p_2 = p_1 p$ für ein $p \in \{0, 1\}^*$

T 4.5

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen -Baumautomaten: Grundbegriffe

Betrachten unendlichen vollständigen Binärbaur ■ Positionen: alle Wörter aus {0.1}*

- w ieder Knoten ø hat linkes und rechtes Kind: ø0. ø1

Baumautomaten: Grundbegriffe

wenn $\rho_1 = \rho_1 \rho$ für ein $\rho \in \{0, 1\}^*$

16:00

2019-02-01

Baumautomaten: Grundbegriffe

Betrachten unendlichen vollständigen Binärbaum

- Positionen: alle Wörter aus {0, 1}*
- jeder Knoten p hat linkes und rechtes Kind: p0, p1
- Tiefe von Knoten p: |p|
- Ebene k: alle Knoten der Tiefe k
- p_2 ist Nachfolger von p_1 , geschrieben $p_1 \sqsubseteq p_2$, wenn $p_2 = p_1 p$ für ein $p \in \{0, 1\}^*$

T 4.5

Pfad: Teilmenge $\pi \subset \{0,1\}^*$ mit $\varepsilon \in \pi$ und:

- wenn $p \in \pi$, dann genau eins der Kinder p0, p1 in π
- $\forall k$: von allen Knoten der Ebene k ist genau einer in π

T 4.5 Forts.

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen -Baumautomaten: Grundbegriffe

Betrachten unendlichen vollständigen Binärbaur ■ Positionen: alle Wörter aus {0.1}*

w ieder Knoten ø hat linkes und rechtes Kind: ø0. ø1

Baumautomaten: Grundbegriffe

a Ebene k: alle Knoten der Tiefe k

wenn $\rho_2 = \rho_1 \rho$ für ein $\rho \in \{0, 1\}^*$

Pfad: Teilmenge $\pi \subseteq \{0,1\}^n$ mit $\varepsilon \in \pi$ und: wenn $\rho \in \pi$, dann eenau eins der Kinder $\rho 0$, $\rho 1$ in π

▼ ∀k: von allen Knoten der Ebene k ist genau einer in π

16:00

2019-02-01

Betrachten unendlichen vollständigen Binärbaum

- Positionen: alle Wörter aus $\{0,1\}^*$
- jeder Knoten p hat linkes und rechtes Kind: p0, p1
- Tiefe von Knoten p: |p|
- Ebene k: alle Knoten der Tiefe k
- p_2 ist Nachfolger von p_1 , geschrieben $p_1 \sqsubseteq p_2$, wenn $p_2 = p_1 p$ für ein $p \in \{0, 1\}^*$

T 4.5

Pfad: Teilmenge $\pi \subset \{0,1\}^*$ mit $\varepsilon \in \pi$ und:

- wenn $p \in \pi$, dann genau eins der Kinder p0, p1 in π
- $\forall k$: von allen Knoten der Ebene k ist genau einer in π

T 4.5 Forts.

Σ-Baum t (Alphabet Σ ohne Stelligkeit):

Funktion $t: \{0,1\}^* \to \Sigma$

T 4.5 Forts.

2019-02-01

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

Baumautomaten: Grundbegriffe

Birtachten wandfahre willschipfen Birtachten Prelitioner. All Wilson and $(x, 1)^*$ is pink Kenter p hat links and release Kind (0, p) is pink Kenter p hat links and release Kind (0, p) is Table wan Kenter p $(x, 1)^*$ in $(x, 1)^*$ in

Baumautomaten: Grundbegriffe

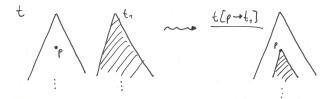
Funktion $t: \{0,1\}^* \rightarrow \Sigma$

Baumautomaten: etwas mehr Notation (1)

$$\hat{t} = t[p \rightarrow t_1]$$
:

der Baum, den man aus t erhält, wenn man den Teilbaum, der in p wurzelt, durch t_1 ersetzt

Skizze:



exakte Beschreibung:

$$\hat{t}(p') = \begin{cases} t_1(p'') & \text{wenn } p' = pp'' \\ t(p') & \text{wenn } p \not \sqsubseteq p' \end{cases}$$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Baumautomaten: etwas mehr Notation (1)

Baumautomaten: etwas mehr Notation (1)

Baumautomaten: etwas mehr Notation (2)

$$\hat{t}=a(t_0,t_1)$$

der Baum mit Wurzel a und Teilbäumen t_0, t_1 in den Wurzelkindern 0, 1:

Skizze:



exakte Beschreibung:

$$\hat{t}(p) = \begin{cases} a & \text{wenn } p = \varepsilon \\ t_0(p') & \text{wenn } p = 0p' \end{cases}$$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Baumautomaten: etwas mehr Notation (2)

Baumautomaten: etwas mehr Notation (2)

-Baumautomaten: Definition

Definition 4.7

Ein nichtdeterministischer Büchi-Baumautomat (NBBA) über Σ ist ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$, wobei

- Q eine endliche nichtleere Zustandsmenge ist,
- \bullet Σ ein Alphabet ist

Baumautomaten: Definition

- $\Delta \subseteq Q \times \Sigma \times Q \times Q$ die Überführungsrelation ist,
- $I \subseteq Q$ die Menge der Anfangszustände ist,
- $F \subset Q$ die Menge der akzeptierenden Zustände ist.

16:14

Automatentheorie u. i. A. WiSe 2018/19

Komplementierung

Definition 4.7

Ein nichtdeterministischer Büchi-Baumautomat (NBBA) über Σ ist ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$, wobei

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

(entsprechen offenbar Top-down-Automaten)

Raumautomaten: Definition Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen ist ein 5-Tupel $A = (Q, \Sigma, \Delta, I, F)$, wobe · Q eine endliche nichtleere Zustandsmense ist u Δ ⊂ Q × Σ × Q × Q die Überführungsrelation ist, -Baumautomaten: Definition v F ⊂ Q die Menge der akzeptierenden Zustände is (entsprechen offenbar Top-down-Automaten)

Muller- und Paritäts-Baumautomaten

Definition 4.8

Ein nichtdeterministischer Muller-Baumautomat (NMBA) über Σ ist ein 5-Tupel $\mathcal{A}=(Q,\Sigma,\Delta,I,\mathcal{F})$, wobei

- Q, Σ, Δ, I wie für NBBAs sind
- $\mathcal{F} \subset 2^Q$ die Akzeptanzkomponente ist

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Muller- und Paritäts-Baumautomaten

affaition 4.8 is the description of the state of the sta

Muller- und Paritäts-Baumautomaten

Q. Σ. Δ. / wie für NBBAs sind

└─Muller- und Paritäts-Baumautomaten

Muller- und Paritäts-Baumautomaten

Definition 4.8

Ein nichtdeterministischer Muller-Baumautomat (NMBA) über Σ ist ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$, wobei

- Q, Σ, Δ, I wie für NBBAs sind
- $\mathcal{F} \subset 2^Q$ die Akzeptanzkomponente ist

Ein nichtdeterministischer Paritäts-Baumautomat (NPBA) über Σ ist ein 5-Tupel $A = (Q, \Sigma, \Delta, I, c)$, wobei

- Q, Σ, Δ, I wie für NBBAs sind
- $c: Q \to \mathbb{N}$ die Akzeptanzkomponente ist

16:15

Automatentheorie u. i. A. WiSe 2018/19

Komplementierung

Definition 4.8

Ein nichtdeterministischer Muller-Baumautomat (NMBA) über Σ ist ein 5-Tupel $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$, wobei

- Q, Σ, Δ, I wie für NBBAs sind
- $\mathcal{F} \subset 2^Q$ die Akzeptanzkomponente ist

Ein nichtdeterministischer Paritäts-Baumautomat (NPBA) über Σ ist ein 5-Tupel $A = (Q, \Sigma, \Delta, I, c)$, wobei

- Q, Σ, Δ, I wie für NBBAs sind
- $c: Q \to \mathbb{N}$ die Akzeptanzkomponente ist

(Rabin- und Streett-Baumautomaten wie üblich definiert)

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen └─Muller- und Paritäts-Baumautomaten

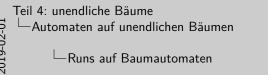
Muller- und Paritäts-Raumautomaten

ist ein 5-Tupel $A = (Q, \Sigma, \Delta, I, F)$, wobei # F ⊂ 2^Q die Akzeptanzkomponente is

 $\mathbf{u} \ c: Q \to \mathbb{N}$ die Akzeptanzkomponente ist

Run = Markierung der Positionen in {0,1}* mit Zuständen, verträelich mit Anfaneszuständen und Überführungsrelation

Run = Markierung der Positionen in $\{0,1\}^*$ mit Zuständen, verträglich mit Anfangszuständen und Überführungsrelation



Run = Markierung der Positionen in $\{0,1\}^*$ mit Zuständen verträglich mit Anfangszuständen und Überführungsrelation Definition 4.9 Ein Ban eines NBBA (NMBA, NPBA) A auf einem Σ -Baur sit eine Funktion $r:\{0,1\}^* \rightarrow O$, so dass

• für alle $\rho \in \{0,1\}^*$ gilt: $(r(\rho),t(\rho),r(\rho 0),r(\rho 1)) \in \Delta$

Runs auf Baumautomaten

Run = Markierung der Positionen in $\{0,1\}^*$ mit Zuständen, verträglich mit Anfangszuständen und Überführungsrelation

Definition 4.9

Ein Run eines NBBA (NMBA, NPBA) \mathcal{A} auf einem Σ -Baum t ist eine Funktion $r: \{0,1\}^* \to Q$, so dass

- $r(\varepsilon) \in I$;
- für alle $p \in \{0,1\}^*$ gilt: $(r(p), t(p), r(p0), r(p1)) \in \Delta$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Runs auf Baumautomaten

Run = Markierung der Positionen in $\{0,1\}^*$ mit Zuständen, verträglich mit Anfangszuständen und Überführungsrelation

Definition 4.9

Ein Run eines NBBA (NMBA, NPBA) \mathcal{A} auf einem Σ -Baum t ist eine Funktion $r: \{0,1\}^* \to Q$, so dass

- $r(\varepsilon) \in I$;
- für alle $p \in \{0,1\}^*$ gilt: $(r(p), t(p), r(p0), r(p1)) \in \Delta$

Erfolgreicher Run: verträglich mit Akzeptanzkomponente

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Runs auf Baumautomaten

• für alle $\rho \in \{0,1\}^*$ gilt: $(r(\rho),t(\rho),r(\rho 0),r(\rho 1)) \in \Delta$ Erfolgreicher Run: verträglich mit Akzeptanzkomponente

Run = Markierung der Positionen in $\{0,1\}^*$ mit Zuständen verträglich mit Anfangszuständen und Überführungsrelation Definition 4.9 Ein Ban eines NBBA (NMBA, NPBA) A auf einem Σ -Baur sit eine Funktion $r:\{0,1\}^* \rightarrow O$, so dass

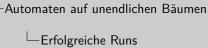
Runs auf Baumautomaten

 $Inf(r, \pi) = \{q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi\}$

Erfolgreiche Runs

Sei r Run eines NxBAs \mathcal{A} und π ein Pfad Betrachten wieder Unendlichkeitsmenge

$$Inf(r,\pi) = \{q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi\}$$



Teil 4: unendliche Bäume

16:18

Parity-Bedingung: in Aufgabe 4 auf Übungsblatt 4 war es max statt min - macht aber keinen Unterschied; ggf. Nummerierung der Zustände "umdrehen".

Komplementierung

Sei r Run eines NxBAs A und π ein Pfad Betrachten wieder Unendlichkeitsmenge

$$Inf(r, \pi) = \{ q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi \}$$

Definition 4.10

• Run r des NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls für alle Pfade π gilt: $Inf(r,\pi) \cap F \neq \emptyset$

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen -Erfolgreiche Runs



16:18

Parity-Bedingung: in Aufgabe 4 auf Übungsblatt 4 war es max statt min - macht aber keinen Unterschied; ggf. Nummerierung der Zustände "umdrehen".

Sei r Run eines NxBAs A und π ein Pfad

Run r des NMBA $A = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls

Betrachten wieder Uhrendlichkeitsmenge $\inf [r,\pi) = \{q \in Q \mid r(\rho) = q \text{ für unendlich wiele } \rho \in \pi \}$ [Definition 4.10]
• Run r des NBBA $A = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls

Erfolgreiche Runs

Sei r Run eines NxBAs $\mathcal A$ und π ein Pfad Betrachten wieder **Unendlichkeitsmenge**

$$Inf(r,\pi) = \{ q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi \}$$

Definition 4.10

- Run r des NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls für alle Pfade π gilt: $Inf(r, \pi) \cap F \neq \emptyset$
- Run r des NMBA $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ ist erfolgreich, falls für alle Pfade π gilt: $Inf(r, \pi) \in \mathcal{F}$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen —Erfolgreiche Runs

16:18

Parity-Bedingung: in Aufgabe 4 auf Übungsblatt 4 war es max statt min – macht aber keinen Unterschied; ggf. Nummerierung der Zustände "umdrehen".

Sei r Run eines NxBAs A und π ein Pfad Betrachten wieder Unendlichkeitsmenge

$$Inf(r,\pi) = \{ q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi \}$$

Definition 4.10

- Run r des NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls für alle Pfade π gilt: $Inf(r,\pi) \cap F \neq \emptyset$
- Run r des NMBA $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ ist erfolgreich, falls für alle Pfade π gilt: $Inf(r,\pi) \in \mathcal{F}$
- Run r des NPBA $\mathcal{A} = (Q, \Sigma, \Delta, I, c)$ ist erfolgreich, falls für alle Pfade π gilt: min $\{c(q) \mid q \in Inf(r, \pi)\}$ ist gerade

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen -Erfolgreiche Runs

Sei r Run eines NxBAs A und π ein Pfad Retrachten wieder Unenflichkeitsmenne $Inf(r, \pi) = \{q \in Q \mid r(p) = q \text{ für unendlich viele } p \in \pi\}$

• Run r des NBBA A = (Q, Σ, Δ, I, F) ist erfolgreich, falls

- für alle Pfade π gilt: $lnf(r, \pi) \cap F \neq \emptyset$ Run r des NMBA $A = (Q, \Sigma, \Delta, I, F)$ ist erfolgreich, falls
- für alle Pfade π gilt: min $\{c(g) \mid g \in Inf(r, \pi)\}$ ist gerade

16:18

Parity-Bedingung: in Aufgabe 4 auf Übungsblatt 4 war es max statt min - macht aber keinen Unterschied; ggf. Nummerierung der Zustände "umdrehen".

Komplementierung

Akzeptanz und erkannte Sprache

... sind wie üblich definiert:

Definition 4.11

Sei \mathcal{A} ein NBBA, NMBA oder NPBA, sei t ein Σ -Baum und L eine Menge von Σ -Bäumen.

- A akzeptiert t, wenn es einen erfolgreichen Run von A auf t gibt.
- $L_{\omega}(A) = \{t \mid A \text{ akzeptient } t\}$
- L heißt Büchi-erkennbar, wenn es einen NBBA A gibt mit $L_{\omega}(A) = L$.
- Analog: Muller-erkennbar und paritäts-erkennbar

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

State Ann BBA, Makade PEA.

State Ann BBA, Makade PEA.

State Ann BBB, Makade PEA.

A die in Externe and £ auto Mange von Externe.

A diergeber 1.

Akzeptanz und erkannte Sprache

Lock | 1 | (A diergeber 1.

Lock | 1 | (A diergeber 1.

Lock | 1 | (A diergeber 1.

Anage Make ektendar und public ektendar.

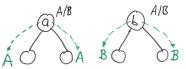
Anage Make ektendar und public ektendar.

16:20

• NBBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



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

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

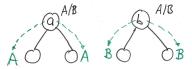
—Beispiele (Büchi)

Beispiele (Büchi)

* NBBA, $A = ((A,B), (A,b), \Delta, (A), (A)) = nt$ $\Delta = ((A,A,A,A), (B,A,AA), (A,B,B), (B,B,B))$ Skizze: AB = (A,A,A,A), (B,A,A,A), (A,B,B,B), (B,B,B)

• NBBA $\mathcal{A}=(\{A,B\},\ \{a,b\},\ \Delta,\ \{A\},\ \{A\})$ mit $\Delta=\{\ (A,a,A,A),\ (B,a,A,A),\ (A,b,B,B),\ (B,b,B,B)\ \}$

Skizze:



 $L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s}\}$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

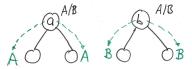
—Beispiele (Büchi)

Beispiele (Büchi)

16:21

• NBBA $\mathcal{A} = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\})$ mit $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$

Skizze:



$$L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s}\}$$

• derselbe NBBA, aber mit $F = \{B\}$ $L_{\omega}(A) = ?$ Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Beispiele (Büchi)

NBBA $A = \{\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\}\} \text{ mit}$ $\Delta = \{\{A, a, A, A\}, \{B, a, A, A\}, \{A, b, B, B\}, \{B, b, B, B\}\}$ Skizze: A^{R} A^{R} A^{R}

 $A = \{ t \mid \text{jeder Pfad hat } \infty \text{ viele a's} \}$

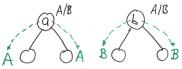
 $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele a} \}$ • derselbe NBBA, aber mit $F = \{B\}$ • $L_{\omega}(A) = 7$

Beispiele (Büchi)

16:21

• NBBA $\mathcal{A} = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\})$ mit $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$

Skizze:



$$L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s}\}$$

• derselbe NBBA, aber mit $F = \{B\}$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat } \infty \text{ viele } b\text{'s}\}$ Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Beispiele (Büchi)

Beispiele (Büchi)

u NBBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\})$ mit $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$ Skizze:

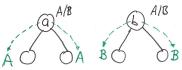
A) = {t | jeder Pfad hat ∞ viele a

derselbe NBBA, aber mit $F = \{B\}$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } b$'s

16:21

• NBBA $\mathcal{A} = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{A\})$ mit $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$

Skizze:



$$L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s}\}$$

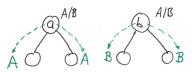
- derselbe NBBA, aber mit $F = \{B\}$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat } \infty \text{ viele } b'\text{s}\}$
- derselbe NBBA, aber mit $F = \{A, B\}$ $L_{\omega}(A) = ?$

-Beispiele (Büchi)

a derselbe NBBA, aber mit $F = \{A, B\}$

16:21

Skizze:



$$L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s}\}$$

- derselbe NBBA, aber mit $F = \{B\}$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat } \infty \text{ viele } b\text{'s}\}$
- derselbe NBBA, aber mit $F = \{A, B\}$ $L_{\omega}(A) = \{t \mid t \text{ ist ein } \Sigma\text{-Baum}\}$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

Beispiele (Büchi)

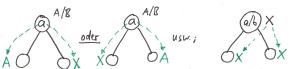
• NBBA $A = (A \cup B)$, $(A \cup B)$, $(A \cup A)$, $(A \cup A)$ and $A = (A \cup A \cup A)$, $(A \cup A \cup A)$, $(A \cup A \cup B)$, $(A \cup A \cup A \cup B)$, $(A \cup A \cup B)$, $(A \cup A \cup B)$, $(A \cup A \cup A \cup B)$, $(A \cup A \cup A \cup B)$, $(A \cup$

 $L_{-}(A) = \{t \mid t \text{ ist ein } \Sigma\text{-Baum}\}$

16:21

• NBBA $\mathcal{A} = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{ (A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) \}$



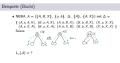


$$L_{\omega}(\mathcal{A}) = ?$$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Beispiele (Büchi)



16:24

• NBBA $A = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{(A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, X, X, X, X, X), (A, X, X, X, X, X, X), (A, X, X, X, X, X, X), (A, X, X, X, X, X, X, X, X), (A, X, X, X, X, X,$

$$(A, a, X, A, X), (B, a, X, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) }$$

Skizze:



 $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a's\}$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

Beispiele (Büch)

**NiBA $A = \{(A, B, X), (A, b), \Delta, (A), (A, X)\} \text{ ns. } \Delta = \{(A, A, X), (B, A, B, X), (B, B, B, X), (X, X, X), (A, A, X, B), (B, A, X, B$

16:24

• NBBA $\mathcal{A} = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{ (A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) \}$



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

• derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = ?$ Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

Beispiele (Büchi)

**NBB. $A = \{(A,B,X), \{a,b\}, \Delta, \{A\}, \{A,X\}\} \text{ on } \Delta = \{(A,A,X), (B,A,A,X), (B,A,B,X), (B,$

16:24

• NBBA $\mathcal{A} = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{ (A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) \}$



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

• derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } b's\}$ Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

Beispiele (Büchi)

16:24

• NBBA $\mathcal{A} = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{ (A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) \}$



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

- derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } b\text{'s}\}$
- derselbe NBBA, aber mit $F = \{X\}$ $L_{\omega}(A) = ?$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

* NIBLA \rightarrow $\{(A, B, X), \{A, B\}, \Delta, \{A\}, \{A, X\}\} \text{ in } \Delta = \{(A, A, X), \{B, A, X\}, \{C, A, X\}, \{C, A, X, A\}, \{C, A, X, A\}, \{C, A, X, A\}, \{C, A, X, B\}, \{C,$

Beispiele (Büchi)

 $L_{\omega}(A) = ?$

16:24

• NBBA $\mathcal{A} = (\{A, B, X\}, \{a, b\}, \Delta, \{A\}, \{A, X\}) \text{ mit } \Delta = \{ (A, a, A, X), (B, a, A, X), (A, b, B, X), (B, b, B, X), (X, a, X, X), (A, a, X, A), (B, a, X, A), (A, b, X, B), (B, b, X, B), (X, b, X, X) \}$



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

- derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } b\text{'s}\}$
- derselbe NBBA, aber mit $F = \{X\}$ $L_{\omega}(A) = \emptyset$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

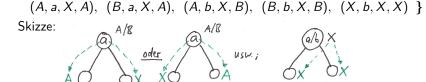
-Beispiele (Büchi)

* NIBEA \rightarrow * (AE EX), (a \rightarrow b), \rightarrow (a), (a \rightarrow C) are \rightarrow are (Ae \rightarrow AX), (a \rightarrow AX),

Beispiele (Büchi)

 $L_{\omega}(A) = \emptyset$

16:24



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

- derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } b \text{ s} \}$
- derselbe NBBA, aber mit $F = \{X\}$: $L_{\omega}(A) = \emptyset$
- derselbe NBBA, aber mit $F = \{A, B\}$: $L_{\omega}(A) = ?$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

Solve $\sum_{i \in \mathcal{I}} dx_i = \sum_{i \in \mathcal{I}} dx_i$

derselbe NBBA, aber mit $F = \{A, B\}$: $L_{-}(A) = 1$

Beispiele (Büchi)

16:24

Skizze:

$$\begin{split} \bullet \text{ NBBA } \mathcal{A} &= (\{A,B,X\},\ \{a,b\},\ \Delta,\ \{A\},\ \{A,X\}) \text{ mit } \Delta = \\ &\{\ (A,a,A,X),\ (B,a,A,X),\ (A,b,B,X),\ (B,b,B,X),\ (X,a,X,X),\\ &(A,a,X,A),\ (B,a,X,A),\ (A,b,X,B),\ (B,b,X,B),\ (X,b,X,X)\ \} \end{split}$$



$$L_{\omega}(\mathcal{A}) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } a\text{'s}\}$$

- derselbe NBBA, aber mit $F = \{B, X\}$ $L_{\omega}(A) = \{t \mid t \text{ hat mind. einen Pfad mit } \infty \text{ vielen } b \text{ s} \}$
- derselbe NBBA, aber mit $F = \{X\}$: $L_{\omega}(A) = \emptyset$
- derselbe NBBA, aber mit $F = \{A, B\}$: $L_{\omega}(A) = \emptyset$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

-Beispiele (Büchi)

 $\begin{cases} (A_n,A_n), (B_n,B_n), (B_n,B_n), (B_n,B_n), (B_n,A_n), \\ (A_n,A_n), (B_n,A_n), (A_n,A_n), (B_n,B_n), (A_n,A_n), \\ (A_n,A_n), (B_n,A_n), (A_n,A_n), (B_n,A_n), (A_n,A_n), \\ A_n,A_n) \end{cases}$ $\begin{cases} A_n \\ A_n \\ A_n \end{cases}$ $\downarrow_{L_n}(A_n) = \{1\} \text{ to tred. one PFad in ∞ within A_n}$ $\downarrow_{L_n}(A_n) = \{1\} \text{ to tred. one PFad in ∞ within A_n}$ $\downarrow_{L_n}(A_n) = \{1\} \text{ to tred. one PFad in ∞ within A_n}$ $\downarrow_{L_n}(A_n) = \{1\} \text{ to tred. one PFad in ∞ within A_n}$ $\downarrow_{L_n}(A_n) = \{1\} \text{ to tred. one PFad in ∞ within A_n}$

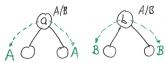
Beispiele (Büchi)

16:24

• NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



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

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

Beispiele (Muller)

Beispiele (Muller)

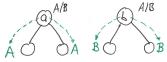
• $NMBA A = (\{A, B\}, \{A, b\}, \Delta, \{A\}, \{\{A\}\}\}) \text{ mit}$ $\Delta = \{\{A, A, A, A\}, \{B, A, A\}, \{A, b, B, B\}, \{B, b, B, B\}\}$ Skize: $L_{*}(A) = 7$

16:28

• NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



 $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

-Beispiele (Muller)

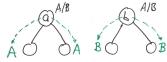
Beispiele (Muller)

• **MMA $A = (\{A, B\}, \{A, b\}, \Delta, \{A\}, \{A\})$ mit $\Delta = \{\{A, b, A\}, \{B, b, A, A\}, \{A, b, B, B\}, \{B, b, B, B\}\}$ Skizee. $L_i(A) = \{t \mid join \ Plad \ bit \ anticks visit \ b's) \ (1)$

 \bullet NMBA $\mathcal{A}=(\{A,B\},\ \{a,b\},\ \Delta,\ \{A\},\ \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\}$$
 (!)

• derselbe NMBA, aber mit $F = \{\{B\}\}\$ $L_{\omega}(A) = ?$ Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Beispiele (Muller)

BB. $A = \{(\hat{a}, \hat{B}), (a, b), \Delta, (A), \{\{A\}\}\} \text{ mit } \\ = \{(A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B)\} \\ \text{as:} \\ A) = \{t \mid \text{joder Plad hat endich viole } b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \{t\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \\ \text{d}) = \{t \mid \text{joder Plad hat endich viole} b^*b\} \\ \text{d}) = \{t \mid \text{joder Pl$

Beispiele (Muller)

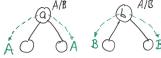
16:28

`

Beispiele (Muller)

• NMBA $\mathcal{A} = (\{A,B\}, \{a,b\}, \Delta, \{A\}, \{\{A\}\})$ mit $\Delta = \{ (A,a,A,A), (B,a,A,A), (A,b,B,B), (B,b,B,B) \}$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$$

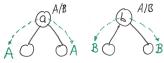
• derselbe NMBA, aber mit $F = \{\{B\}\}\$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat endlich viele } a's\}$ Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

-Beispiele (Muller)

16:28

• NMBA $\mathcal{A} = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\})$ mit $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$$

- derselbe NMBA, aber mit $F = \{\{B\}\}\$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } a's\}$
- derselbe NMBA, aber mit $F = \{\{A, B\}\}$ $L_{\omega}(A) = ?$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

-Beispiele (Muller)

16:28

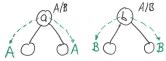
- expires: (wature!)

 NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\}\})$ mit $\Delta = \{(A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B)\}$ Skizze: AB = AB
 - $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b$'s} derselbe NMBA, aber mit $F = \{\{B\}\}$
- $L_{\omega}(A) = \{r \mid \text{jeder Pfad hat endlich viele a's}\}$ derselbe NMBA, aber mit $F = \{\{A, B\}\}$ $L_{\omega}(A) = 7$

• NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$$

- derselbe NMBA, aber mit $F = \{\{B\}\}\$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat endlich viele } a\text{'s}\}$
- derselbe NMBA, aber mit $F = \{\{A, B\}\}\$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s und } \infty \text{ viele } b\text{'s}\}$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Beispiele (Muller)

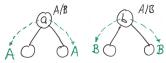
16:28

- NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\}\})$ mit $\Delta = \{(A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B)\}$ Skizze:
 - $A \bigcirc A \otimes \bigcirc C$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } I$
- derselbe NMBA, aber mit F = {{B}} L_v(A) = {t | jeder Pfad hat endlich viele a's}
- $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s und } \infty \text{ viele } b\text{'s}\}$

 \bullet NMBA $\mathcal{A}=(\{A,B\},\ \{a,b\},\ \Delta,\ \{A\},\ \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$$

- derselbe NMBA, aber mit $F = \{\{B\}\}\$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } a\text{'s}\}$
- derselbe NMBA, aber mit $F = \{\{A, B\}\}$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat } \infty \text{ viele } a\text{'s und } \infty \text{ viele } b\text{'s}\}$
- derselbe NMBA, aber mit $F = \{\{A\}, \{B\}\}$ $L_{\omega}(A) = ?$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

-Beispiele (Muller)

Beispiele (Muller)

**NOBEA $A = \{(A, B), \{A, b\}, \Delta, \{A\}, \{\{A\}\}\} \text{ mix}$ $\Delta = \{(A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B)\}$ Staze

**L_(A) = {r | jude Plad hat setlich vide b's} (!)

**densite NOBEA, sher mir F = {(B)}

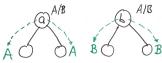
derselbe NMBA, aber mit F = {{A}, {B}}

16:28

• NMBA $\mathcal{A}=(\{A,B\},\ \{a,b\},\ \Delta,\ \{A\},\ \{\{A\}\})$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

Skizze:



$$L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b's\} (!)$$

- derselbe NMBA, aber mit $F = \{\{B\}\}$ $L_{\omega}(A) = \{t \mid \text{ jeder Pfad hat endlich viele } a's\}$
- derselbe NMBA, aber mit $F = \{\{A, B\}\}$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat } \infty \text{ viele } a\text{'s und } \infty \text{ viele } b\text{'s}\}$
- derselbe NMBA, aber mit $F = \{\{A\}, \{B\}\}\}$ $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endl. viele } b$'s oder endl. viele a's $\}$

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

-Beispiele (Muller)

derselbe NMBA, aber mit $F = \{\{A, B\}\}$ $L_i(A) = \{t \mid \text{jeder Plad hat } \infty \text{ viele } a's \text{ und } \infty \text{ viele } b's\}$ derselbe NMBA, aber mit $F = \{\{A\}, \{B\}\}$ $L_i(A) = \{t \mid \text{ider Plad hat endl. viele } b's \text{ oder endl. viele } a's\}$

u NMBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, \{\{A\}\})$ mit

 $L_{\omega}(A) = \{t \mid \text{jeder Pfad hat endlich viele } b$'s $\}$ (1)

 $\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$

Beispiele (Muller)

16:28

Zur Erinnerung:

Run r ist erfolgreich, wenn für alle Pfade $\pi \subseteq T$ gilt:

$$\min\{c(q) \mid q \in \inf(r,\pi)\}$$
 ist gerade

Teil 4: unendliche Bäume
—Automaten auf unendlichen Bäumen
—Beispiel (Parität)

Beispiel (Parität) Zor Einnenny: Run r ätt erfolgesich, seem für alle Pfade $\pi \subseteq T$ göt: $\min \{c(q) \mid q \in Inf(r, \pi)\} \text{ at gurade}$

Beispiel (Parität)

Zur Erinnerung:

Run r ist erfolgreich, wenn für alle Pfade $\pi \subset T$ gilt:

$$\min\{c(q) \mid q \in \inf(r,\pi)\}$$
 ist gerade

NPBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, c)$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

$$c(A) = 1$$

$$c(B) = 2$$



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

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Beispiel (Parität)

Beispiel (Parität)

 $L_{\omega}(A) = ?$

Run r ist erfolgreich, wenn für alle Pfade $\pi \subseteq T$ gilt $\min\{c(q) \mid q \in \inf\{r, \pi\}\} \text{ ist gerade}$ NPBA $\mathcal{A} = (\{A, B\}, \ \{a, b\}, \ \Delta, \ \{A\}, \ c)$ mit

Beispiel (Parität)

Zur Erinnerung:

Run r ist erfolgreich, wenn für alle Pfade $\pi \subset T$ gilt:

$$\min\{c(q) \mid q \in \inf(r,\pi)\}$$
 ist gerade

NPBA $A = (\{A, B\}, \{a, b\}, \Delta, \{A\}, c)$ mit

$$\Delta = \{ (A, a, A, A), (B, a, A, A), (A, b, B, B), (B, b, B, B) \}$$

$$c(A) = 1$$

$$c(B) = 2$$



$$L_{\omega}(\mathcal{A}) = \{t \mid \text{jeder Pfad hat endlich viele } a\text{'s}\}$$

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Beispiel (Parität)

Beispiel (Paritál)

Zu Estmong

Run r ist erfolgreich, were für alle Plade $\pi \subseteq T$ gilt: $\min(\mathcal{L}(q) \mid q \in \operatorname{Inf}(r, \pi))$ its gerade

NNBA $A = (A, B, B_1, A_2), \Delta_1(A_3, C)$ ent $\Delta = \{(A, B, A, A), (B, A, A, A), (A, B, B, B), (B, B, B, B)\}$ $\mathcal{L}(A) = 1$ $\mathcal{L}(A) = \{r \mid \operatorname{poler Plad hat endlich viole } x_1\}$

Satz 4.12

- Jede Büchi-erkennbare Sprache ist Muller-erkennbar.
- Nicht jede Muller-erkennbare Sprache ist Büchi-erkennbar.

Teil 4: unendliche Bäume -Automaten auf unendlichen Bäumen Büchi- versus Muller-Erkennbarkeit

Büchi- versus Muller-Erkennbarkeit

- Jede Büchi-erkennbare Sprache ist Muller-erkennbar.
- **Nicht jede** Muller-erkennbare Sprache ist Büchi-erkennbar.

Beweis.

• Wie im letzten Kapitel.

Teil 4: unendliche Bäume —Automaten auf unendlichen Bäumen

—Büchi- versus Muller-Erkennbarkeit

Sair 4.12

• Jede Büch-erkensbare Sprache ist Multer-erkensbar.

• Mott jede Multer-erkensbare Sprache ist Büch-erkensbar.

Steenik.

• With in letzten Kapitel.

Büchi- versus Muller-Erkennbarkeit

Büchi- versus Muller-Erkennbarkeit

Satz 4.12

- Jede Büchi-erkennbare Sprache ist Muller-erkennbar.
- 2 Nicht jede Muller-erkennbare Sprache ist Büchi-erkennbar.

Beweis.

- Wie im letzten Kapitel.
- Betrachten L = {t | jeder Pfad in t hat endlich viele a's}
 L ist Muller-erkennbar (siehe Bsp. auf Folie 34)
 Müssen zeigen: L nicht Büchi-erkennbar

Teil 4: unendliche Bäume

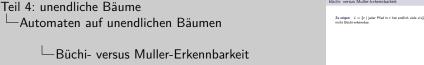
Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

O Jode Büch-erkembare Sprache ist Multe-erkembar.
Nicht jode Multe-erkembare Sprache ist Büchi-erkembar.
Breuek.
O Wie in letzten Kapitel.
O Betzschten L = {r | jeder Pfad in r hat endlich viele a's}.
List Multe-erkembar (siehe Big. auf Foisi 34)

Rüchi- versus Muller-Erkennbarkeit

Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Büchi-erkennbar.



Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Büchi-erkennbar.

Nehmen an, es gebe NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ mit $L_{\omega}(\mathcal{A}) = L$.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

Büchi- versus Muller-Erkennbarkeit

Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Büchi-erkennbar. Nehmen an, es gebe NBBA $A = (Q, \Sigma, \Delta, I, F)$ mit L, (A) = L

Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Büchi-erkennbar.

Nehmen an, es gebe NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ mit $L_{\omega}(\mathcal{A}) = L$.

O. B. d. A. sei
$$I = \{q_0\}$$
. Sei $n := |F|$.

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Büchi- versus Muller-Erkennbarkeit

Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Bächi-erkennbar. Nohmen zu so zehn NBRA $A = (O \mid X \mid A \mid F)$ mit L(A) = L

Büchi- versus Muller-Erkennbarkeit

Zu zeigen: $L = \{t \mid \text{jeder Pfad in } t \text{ hat endlich viele } a's\}$ nicht Büchi-erkennbar.

Nehmen an, es gebe NBBA $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ mit $L_{\omega}(\mathcal{A}) = L$.

O. B. d. A. sei
$$I = \{q_0\}$$
. Sei $n := |F|$.

Idee:

- Bestimme Baum $t \in L$ mit Run r und Pfad, auf dem zwischen 2 Besuchen desselben akzeptierenden Zustandes ein a auftritt
- "Pumpe" t, r so auf, dass dieser Teilpfad sich ∞ oft wiederholt
- 4 Neuer Baum wird akzeptiert, aber neuer Pfad hat ∞ viele a's

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen Büchi- versus Muller-Erkennbarkeit

 \mathbf{v} Bestimme Baum $t \in L$ mit Run r und Pfad, auf dem zwische "Pumpe" t, r so auf, dass dieser Teilpfad sich ∞ oft wiederhol § Neuer Baum wird akzeptiert, aber neuer Pfad hat ∞ viele a's

Nehmen an, es gebe NBBA $A = (Q, \Sigma, \Delta, I, F)$ mit $L_{-}(A) = L$

16:36

2019-02-01

Komplementierung

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i=1}^{n} (1^{+}0)^{i}$,

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten. Dadurch hat jeder Pfad nur max. n a's, also endlich viele.

Komplementierung

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i=1}^{n} (1^{+}0)^{i}$,

- d. h. t enthält ein a an allen Positionen, die man erreicht, wenn man bei der Wurzel startet und bis zu n-mal wie folgt läuft:
 - einmal oder mehrmals zum rechten Kind (beliebig oft)
 - einmal zum linken Kind ("Linksschritt")

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

schte Baum $t \in L$ mit $t(\rho) = a$ gdw. $\rho \in \bigcup_{i=k_1...n} (1^+0)^i$, t enthält ein a an allen Positionen, die man ersicht, man bei der Wurzel startet und bis zu n-mal wie folgt läuft: simmal oder mehrmals zum rechten Kind (beliebig oft)

Büchi- versus Muller-Erkennbarkeit

16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten. Dadurch hat jeder Pfad nur max. n a's, also endlich viele.

Büchi- versus Muller-Erkennbarkeit

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in (1^+0)^i$,

d. h. t enthält ein a an allen Positionen, die man erreicht, wenn man bei der Wurzel startet und bis zu *n*-mal wie folgt läuft:

- einmal oder mehrmals zum rechten Kind (beliebig oft)
- einmal zum linken Kind ("Linksschritt")

An den übrigen Positionen enthält *t* ein *b*.

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

wenn man bei der Wurzel startet und bis zu n-mal wie folgt läuft einmal zum linken Kind ("Linksschritt") An den übrigen Positionen enthält t ein h

16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten.

Dadurch hat jeder Pfad nur max. n a's, also endlich viele.

Komplementierung

Büchi- versus Muller-Erkennbarkeit

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i=1}^{n} (1^{+}0)^{i}$,

d. h. t enthält ein a an allen Positionen, die man erreicht, wenn man bei der Wurzel startet und bis zu n-mal wie folgt läuft:

- einmal oder mehrmals zum rechten Kind (beliebig oft)
- einmal zum linken Kind ("Linksschritt")

An den übrigen Positionen enthält t ein b.

Skizze:

$$\begin{cases}
b \\
a \\
b
\end{cases}$$
b is zu
$$\begin{cases}
n-\text{max}
\end{cases}$$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Büchi- versus Muller-Erkennbarkeit

Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i \in L_{n-k}} (1, k)$ hill sin a an alian Positionen, die man orwicht, bei der Wurzel starest und bis zu n-mail wie folgt ill oder mehrmaks aum rechtnes Kindi (beliebig oft) il zum linken Kind ("Linksschritt") ill zum linken Kind ("Linksschritt") bis zu n-mail n-ma

16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten. Dadurch hat jeder Pfad nur max. n a's, also endlich viele.

Komplementierung

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i=1}^{n} (1^{+}0)^{i}$,

d. h. t enthält ein a an allen Positionen, die man erreicht, wenn man bei der Wurzel startet und bis zu n-mal wie folgt läuft:

- einmal oder mehrmals zum rechten Kind (beliebig oft)
- einmal zum linken Kind ("Linksschritt")

An den übrigen Positionen enthält t ein b.

Skizze:

$$\begin{cases}
b \\
a \\
b
\end{cases}$$
b bis zu
$$n-\text{mal}$$

Klar: $t \in L$. Sei r ein erfolgreicher Run.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit

Büchi- versus Muller-Erhennbarkeit

Bitrachte Bann $t \in I$ mit t(p) = a g d u, $p \in \bigcup_{i=1}^{n} \binom{n}{i} \binom{n}{i}$, d h, $i = m \log n$, $i = m \log$

16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten. Dadurch hat jeder Pfad nur max. n a's, also endlich viele.

Büchi- versus Muller-Erkennbarkeit

Betrachte Baum $t \in L$ mit t(p) = a gdw. $p \in \bigcup_{i=1}^{n} (1^{+}0)^{i}$,

d. h. t enthält ein a an allen Positionen, die man erreicht, wenn man bei der Wurzel startet und bis zu n-mal wie folgt läuft:

- einmal oder mehrmals zum rechten Kind (beliebig oft)
- einmal zum linken Kind ("Linksschritt")

An den übrigen Positionen enthält t ein b.

Skizze:

$$\begin{array}{c}
b \\
b \\
a \\
b
\\
b \\
a \\
b
\end{array}$$
 bis zu
$$\begin{array}{c}
n-\text{mal} \\
b \\
a \\
b \\
\end{array}$$

Klar: $t \in L$. Sei r ein erfolgreicher Run.

Details des Pumpens: s. Tafel

T 4.6 □

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Büchi- versus Muller-Erkennbarkeit



16:38 bis 16:50, 5 min Pause

Man beachte: a's tauchen in beliebiger Tiefe auf, aber auf jedem Pfad nur auf den ersten n Linksschritten.

Dadurch hat jeder Pfad nur max. *n* a's, also endlich viele.

Model-Checking CTL Baumautomaten Komplementierung

Folgerung aus dem Beweis von Satz 4.12

Folgerung 4.13

Die Klasse der Büchi-erkennbaren Baumsprachen ist nicht abgeschlossen unter ?

Folgerung aus dem Beweis von Satz 4.12 Teil 4: unendliche Bäume -Automaten auf unendlichen Bäumen

Folgerung aus dem Beweis von Satz 4.12

16:55

Vor Satz 4.14 sagen: Anfang des letzten Beweises erinnert nicht ohne Grund an Beweis der Ungleichmächtigkeit von DBAs und NBAs

Model-Checking CTL Baumautomaten Komplementierung

Folgerung aus dem Beweis von Satz 4.12

Folgerung 4.13

Die Klasse der Büchi-erkennbaren Baumsprachen ist **nicht** abgeschlossen unter Komplement.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Folgerung aus dem Beweis von Satz 4.12

Folgerung aus dem Beweis von Satz 4.12

Der Naue der Beheine der Beheine Bernegsahm ist nicht abgeschlossen unter Konglement.

Folgerung aus dem Beweis von Satz 4.12

16:55

Vor Satz 4.14 sagen: Anfang des letzten Beweises erinnert nicht ohne Grund an Beweis der Ungleichmächtigkeit von DBAs und NBAs

Komplementierung

Folgerung 4.13

Die Klasse der Büchi-erkennbaren Baumsprachen ist nicht abgeschlossen unter Komplement.

Man kann Satz 4.12 stärker formulieren (ohne Beweis):

Satz 4.14

Die Menge der Baumsprachen, die Muller-, aber nicht Büchi-erkennbar sind, ist

 $\{L_{\wedge} \mid L \text{ ist NBA-erkennbar, aber nicht DBA-erkennbar}\}.$

 $(L \subset \Sigma^{\omega} \text{ ist eine } \omega\text{-Sprache};$

 L_{\triangle} = Menge aller Σ -Bäume, deren Beschriftung entlang jedes Pfades in L liegt)

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen Folgerung aus dem Beweis von Satz 4.12



16:55

2019-02-01

Vor Satz 4.14 sagen: Anfang des letzten Beweises erinnert nicht ohne Grund an Beweis der Ungleichmächtigkeit von DBAs und NBAs

Model-Checking CTL Baumautomaten Komplementierung

Paritäts- versus Muller-Erkennbarkeit

Satz 4.15

- Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- ② Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Paritäts- versus Muller-Erkennbarkeit

16:57

Paritäts- versus Muller-Erkennbarkeit

Model-Checking CTL Baumautomaten Komplementierung

Paritäts- versus Muller-Erkennbarkeit

Satz 4.15

- **1** Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- 2 Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Beweis.

• Wie im letzten Kapitel.

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Paritäts- versus Muller-Erkennbarkeit

Paritäts- versus Muller-Erkennbarkeit

Satz 4.15

- Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- 2 Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Beweis.

- Wie im letzten Kapitel.
- Sei $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ NMBA, mit $I = \{q_0\}$ und $\mathcal{F} = \{F\}$ (o. B. d. A.) sowie n := |Q|.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Paritäts- versus Muller-Erkennbarkeit

$$\begin{split} & \pmb{\Phi} \text{ Wie im letzten Kapitel.} \\ & \pmb{\Phi} \text{ Sei } \mathcal{A} = \left(Q, \Sigma, \Delta, I, \mathcal{F}\right) \text{ NMBA, mit} \\ & I = \left\{q_{0}\right\} \text{ und } \mathcal{F} = \left\{F\right\} \left(\text{o.B.d.A.}\right) \text{ sowie } n := |Q|. \end{split}$$

Jede paritäts-erkennbare Sprache ist Muller-erkennbar

Jede Muller-erkennbare Sprache ist paritäts-erkennbare

Paritäts- versus Muller-Erkennbarkeit

Satz 4.15

- Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- 2 Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Beweis.

- Wie im letzten Kapitel.
- \bigcirc Sei $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ NMBA, mit $I = \{q_0\} \text{ und } \mathcal{F} = \{F\} \text{ (o. B. d. A.)} \text{ sowie } n := |Q|.$

Gesucht: äquivalenter NPBA \mathcal{A}'

Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen Paritäts- versus Muller-Erkennbarkeit

Paritäts- versus Muller-Erkennbarkeit Jede paritäts-erkennbare Sprache ist Muller-erkennbar Jede Muller-erkennbare Sprache ist paritäts-erkennbare

Wie im letzten Kapitel $I = \{a_1\} \text{ und } F = \{F\} \text{ (o. B. d. A.)} \text{ sowie } n := |Q|.$

Satz 4.15

- Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- 2 Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Beweis.

- Wie im letzten Kapitel.
- Sei $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ NMBA, mit $I = \{q_0\}$ und $\mathcal{F} = \{F\}$ (o. B. d. A.) sowie n := |Q|.

Gesucht: äquivalenter NPBA \mathcal{A}'

Idee: A' soll ...

• "sich merken", in welcher Reihenfolge die n Zustände zuletzt gesehen wurden (Permutation $q_1 \cdots q_n$ von Q)

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Paritäts- versus Muller-Erkennbarkeit

Jede paritäts-erkennbare Sprache ist Muller-erkennbar

Paritäte, vareus Muller-Erkennharkeit

16:57

2019-02-01

Wie im letzten Kapitel

A lede Muller, erkennhare Snranhe ist naritäts, erkennha

 $I = \{a_1\}$ and $F = \{F\}$ (o.B.d.A.) sowie a := |Q|

Satz 4.15

- Jede paritäts-erkennbare Sprache ist Muller-erkennbar.
- 2 Jede Muller-erkennbare Sprache ist paritäts-erkennbar.

Beweis.

- Wie im letzten Kapitel.
- Sei $\mathcal{A} = (Q, \Sigma, \Delta, I, \mathcal{F})$ NMBA, mit $I = \{q_0\}$ und $\mathcal{F} = \{F\}$ (o. B. d. A.) sowie n := |Q|.

Gesucht: äquivalenter NPBA \mathcal{A}'

Idee: A' soll ...

- "sich merken", in welcher Reihenfolge die n Zustände zuletzt gesehen wurden (Permutation $q_1 \cdots q_n$ von Q)
- sicherstellen, dass ab einem gewissen Zeitpunkt genau die Zustände aus F dauerhaft am Ende der Permutation stehen

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Paritäts- versus Muller-Erkennbarkeit

16:57

2019-02-01

Details der Konstruktion (1)

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_0\}, F)$ NMBA mit |Q| = n. Konstruieren NPBA $A' = (Q', \Sigma, \Delta', I', c)$ mit Zuständen

$$Q' = \{ \langle q_1 \cdots q_n, \ell \rangle \mid q_1 \cdots q_n \text{ ist Permutation von } Q, \\ \ell \in \{1, \dots, n\}$$

Idee:

- q_n ist der zuletzt besuchte Zustand auf dem aktuellen Pfad, q_{n-1} der zuletzt besuchte Zustand $\neq q_n$ usw.
- ℓ ist Position von q_n in der vorangehenden Permutation

Skizze: siehe Tafel T 4.7 Teil 4: unendliche Bäume Automaten auf unendlichen Bäumen $\ell \in \{1, \dots, n\}$ □ Details der Konstruktion (1) a ... 1 der zuletzt besuchte Zustand ≠ a usw. u f ist Position von a. in der vorangehenden Permutation

Details der Konstruktion (1)

Details der Konstruktion (2)

Zeigen zunächst folgende Hilfsaussage (HA) über Zustände von \mathcal{A}'

Sei $q_1q_2q_3\ldots$ eine Folge von Zuständen aus Q; sei $s_1s_2s_3\ldots$ die zugehörige Folge von Zuständen aus Q' mit $s_1=\langle t_1\cdots t_{n-1}q_1,\ 1\rangle$ und $s_i=\langle {\sf perm}_i,\ell_i\rangle$ für alle $i\geqslant 0$.

Dann gilt $Inf(q_1q_2q_3...) = S mit |S| = k gdw.$

- Für endlich viele i ist $\ell_i \leqslant n k$ und
- Für unendlich viele i gilt:
 - (a) $\ell_i = n k + 1$ und
 - (b) Die Menge der Zustände in den Positionen $\underbrace{n-k+1,\ldots,n}_{\text{letzte }k \text{ Positionen}}$

Beweis der Hilfsaussage: siehe Tafel

T 4.8

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

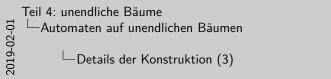
Sei (100): ... are Folge von Zutsiche Baumen (104) über Zutsich (10

17:06 bis 17:11 Tafelanschrieb nur, wenn viel mehr Zeit!

 $l' = \{(t_1 \cdots t_{n-1}q_1, 1) \mid t_1 \cdots t_{n-1} \text{ int Perm. von } Q \setminus \{q_0\}\}$

Können nun Konstruktion fortsetzen:

$$I' = \left\{ \langle t_1 \cdots t_{n-1} q_0, 1 \rangle \mid t_1 \cdots t_{n-1} \text{ ist Perm. von } Q \setminus \{q_0\} \right\}$$



- 17:11 bis 17:29; (wenn Zeit knapp, dann ohne Tafelanschrieb)
- (!) Wahl der Akzeptanzbedingung wird nur durch Beweis so richtig klar . . .

Details der Konstruktion (3)

Können nun Konstruktion fortsetzen:

$$I' = \left\{ \langle t_1 \cdots t_{n-1} q_0, 1 \rangle \mid t_1 \cdots t_{n-1} \text{ ist Perm. von } Q \setminus \{q_0\} \right\}$$

$$\Delta' = \left\{ \left(\langle i_1 \cdots i_{n-1} i, \ell \rangle, a, \langle i'_1 \cdots i'_{n-1} i', \ell' \rangle, \langle i''_1 \cdots i''_{n-1} i'', \ell'' \rangle \right) \mid d_1 \right\}$$

- $(i, a, i', i'') \in \Delta$
- $i'_1 \cdots i'_{n-1}$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i'
- $i_1'' \cdots i_{n-1}''$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i''
- ℓ' = Position von i' in $i_1 \cdots i_{n-1}i$
- ℓ'' = Position von i'' in $i_1 \cdots i_{n-1}i$

Teil 4: unendliche Bäume

—Automaten auf unendlichen Bäumen

—Details der Konstruktion (3)

```
Details der Konstruktion (3)  \begin{aligned} & \text{Rotens on an Konstruktion (Instructure)} \\ & & \text{Rotens on an Konstruktion Instructure} \\ & & \text{Pir} & \left\{ \left( (n_i, \dots, n_i, p_i) \mid p_i, \dots, p_i \right) \right\} \right. \\ & & \text{Air} & \left\{ \left( \left( (n_i, \dots, n_i, p_i), a_i \mid (n_i, \dots, n_i', p_i'), (p_i'' \dots, p_i'', p_i'') \right) \mid a_i \mid (n_i, p_i'') \mid a_i \mid (n_i, p_i'', p_i'') \mid a_i \mid (n_i, p_i'', p_i'') \mid a_i \mid (n_i, p_i'', p_i'', a_i) \right. \\ & & & & \text{Air} & \text{Air} & \text{and institute on } a_i \mid (n_i, p_i') \mid a_i \mid a_i \mid (n_i, p_i'', p_i'', a_i'', a_i) \right. \\ & & & & \text{Air} \right. \\ & & & & \text{Air} \right. \\ & & & & \text{Air} \right. \\ & & & & \text{Air} \right. \\ & & & & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text{Air} \\ & & & & \text{Air} & \text{Air} & \text
```

17:11 bis 17:29; (wenn Zeit knapp, dann ohne Tafelanschrieb)

(!) Wahl der Akzeptanzbedingung wird nur durch Beweis so richtig klar . . .

Komplementierung

Können nun Konstruktion fortsetzen:

$$I' = \left\{ \langle t_1 \cdots t_{n-1} q_0, 1 \rangle \mid t_1 \cdots t_{n-1} \text{ ist Perm. von } Q \setminus \{q_0\} \right\}$$

$$\Delta' = \left\{ \left(\langle i_1 \cdots i_{n-1} i, \ell \rangle, a, \langle i'_1 \cdots i'_{n-1} i', \ell' \rangle, \langle i''_1 \cdots i''_{n-1} i'', \ell'' \rangle \right) \mid A'' \right\}$$

- $(i, a, i', i'') \in \Delta$
- $i'_1 \cdots i'_{n-1}$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i'
- $i_1'' \cdots i_{n-1}''$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i''
- ℓ' = Position von i' in $i_1 \cdots i_{n-1}i$
- ℓ'' = Position von i'' in $i_1 \cdots i_{n-1}i$

$$c(s) = \begin{cases} 2\ell & \text{falls } s = \langle q_1 \cdots q_n, \ell \rangle \text{ und } \{q_\ell, \dots, q_n\} = F \\ 2\ell + 1 & \text{falls } s = \langle q_1 \cdots q_n, \ell \rangle \text{ und } \{q_\ell, \dots, q_n\} \neq F \end{cases}$$

Teil 4: unendliche Bäume -Automaten auf unendlichen Bäumen ☐ Details der Konstruktion (3)

Details der Konstruktion (3) $l' = \{(t_1 \cdots t_{n-1}q_1, 1) \mid t_1 \cdots t_{n-1} \text{ int Perm. von } Q \setminus \{q_0\}\}$ $\Delta' = \{(i_1 \cdots i_{s-1}i, \ell), x, (i'_1 \cdots i'_{s-1}i', \ell'), (i''_1 \cdots i''_{s-1}i'', \ell'')\}$

17:11 bis 17:29; (wenn Zeit knapp, dann ohne Tafelanschrieb)

(!) Wahl der Akzeptanzbedingung wird nur durch Beweis so richtig klar . . .

Können nun Konstruktion fortsetzen:

$$I' = \left\{ \langle t_1 \cdots t_{n-1} q_0, 1 \rangle \mid t_1 \cdots t_{n-1} \text{ ist Perm. von } Q \setminus \{q_0\} \right\}$$

$$\Delta' = \left\{ \left(\langle i_1 \cdots i_{n-1} i, \ell \rangle, \ a, \ \langle i_1' \cdots i_{n-1}' i', \ell' \rangle, \ \langle i_1'' \cdots i_{n-1}'' i'', \ell'' \rangle \right) \mid \right\}$$

- $(i, a, i', i'') \in \Delta$
- $i'_1 \cdots i'_{n-1}$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i'
- $i_1'' \cdots i_{n-1}''$ entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i''
- ℓ' = Position von i' in $i_1 \cdots i_{n-1}i$
- ℓ'' = Position von i'' in $i_1 \cdots i_{n-1}i$

$$c(s) = \begin{cases} 2\ell & \text{falls } s = \langle q_1 \cdots q_n, \ell \rangle \text{ und } \{q_\ell, \dots, q_n\} = F \\ 2\ell + 1 & \text{falls } s = \langle q_1 \cdots q_n, \ell \rangle \text{ und } \{q_\ell, \dots, q_n\} \neq F \end{cases}$$

Beweis der Korrektheit: siehe Tafel

T 4.9 □

Details der Konstruktion (3) Teil 4: unendliche Bäume $l' = \{(t_1 \cdots t_{n-1}q_1, 1) \mid t_1 \cdots t_{n-1} \text{ int Perm. von } Q \setminus \{q_0\}\}$ -Automaten auf unendlichen Bäumen $\Delta' = \{(i_1 \cdots i_{s-1}i, \ell), x, (i'_1 \cdots i'_{s-1}i', \ell'), (i''_1 \cdots i''_{s-1}i'', \ell'')\}$ ☐ Details der Konstruktion (3)

17:11 bis 17:29; (wenn Zeit knapp, dann ohne Tafelanschrieb)

(!) Wahl der Akzeptanzbedingung wird nur durch Beweis so richtig klar . . .

 $u \ i'_1 \cdots i'_{n-1}$, entsteht aus $i_1 \cdots i_{n-1}i$ durch Löschen von i

Die Klasse der . . .

- **1** Büchi-erkennbaren Sprachen ist abgeschlossen unter \cup und \cap , aber nicht unter $\overline{}$.
- 2 Muller-erkennbaren Sprachen ist abgeschlossen unter \cup , \cap , $\overline{}$.

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

Bürgerin in der Greiche der Greiche ist depenblosse unter Unit.

Abschlusseigenschaften

Abschlusseigenschaften

17:29 bis 17:30 → Ende Gelände

Satz 4.16

Die Klasse der ...

- **1** Büchi-erkennbaren Sprachen ist abgeschlossen unter \cup und \cap , aber nicht unter $\overline{}$.
- 2 Muller-erkennbaren Sprachen ist abgeschlossen unter \cup , \cap , $\overline{}$.

Beweis:

- U ∩ wie gehabt; siehe Folgerung 4.13.
- ② ∪∩ wie gehabt;
 - siehe nächsten Abschnitt

Auto

Teil 4: unendliche Bäume

Automaten auf unendlichen Bäumen

-Abschlusseigenschaften

Muller-erkennbaren Sorachen ist abeeschlossen unter ∪. ∩.

Abschlusseigenschaften

17:29 bis 17:30 → Ende Gelände

2019-02-01

- 2 Automaten auf unendlichen Bäumen
- 3 Komplementierung

Teil 4: unendliche Bäume -Komplementierung └─Und nun ...



Komplementierung

Dann folet $L_{-}(A') = \overline{L_{-}(A)}$

Überblick

Ziel dieses Abschnitts:

Lösen Komplementierung mit Hilfe eines bekannten Resultates über Gewinnstrategien in einer bestimmten Art (abstrakter) Spiele

Vorgehen:

- Ordnen jedem NPBA \mathcal{A} und Baum t ein 2-Personen-Spiel $G_{\mathcal{A},t}$ zu (Beschränkung auf NPBAs ist unerheblich, siehe Satz 4.15)
- Dann wird leicht zu sehen sein:

 \mathcal{A} akzeptiert $t \iff \mathsf{Spielerin} \ 1$ hat Gewinnstrategie in $\mathcal{G}_{\mathcal{A},t}$

- Ein Resultat aus der Spieltheorie impliziert:
 In G_{A,t} hat immer genau eine Spielerin eine Gewinnstrategie,
 die nicht vom bisherigen Spielverlauf abhängt
- Konstruieren \mathcal{A}' , so dass gilt: \mathcal{A}' akzeptiert $t\Leftrightarrow$ Spielerin 2 hat Gewinnstrategie in $G_{\mathcal{A},t}$ Dann folgt $L_{\omega}(\mathcal{A}') = \overline{L_{\omega}(\mathcal{A})}$

Teil 4: unendliche Bäume

Komplementierung

Überblick

16:00

2019-02-01

Model-Checking CTL Baumautomaten Komplementierung

Intuitive Beschreibung des Spiels $G_{A,t}$

Zwei Spielerinnen Aut (Automat), PF (Pfadfinderin)

- sind abwechselnd an der Reihe
- bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_I), q_I \in I$

Teil 4: unendliche Bäume

Komplementierung

 \sqsubseteq Intuitive Beschreibung des Spiels $G_{A,t}$

16:03 bis 16:12

 q_I bezeichne ab hier einen beliebigen Anfangszustand; brauchen " q_0 " später noch anderweitig.

Es wird also ein Baumautomat mit beliebiger Akzeptanzbedingung (Büchi, Muller, whatever) zugrunde gelegt.

Intuitive Beschreibung des Spiels $G_{A,t}$

wei Spielerinnen Aut (Automat), PF (Pfadfinderin

a bewegen sich pro Runde 1 Schritt im Baum durch Markieren

Model-Checking CTL Komplementierung

Intuitive Beschreibung des Spiels $G_{A,t}$

Zwei Spielerinnen Aut (Automat), PF (Pfadfinderin)

- sind abwechselnd an der Reihe.
- bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_I), q_I \in I$

In jeder Runde wählt

- Aut eine Transition, die auf die markierte Position anwendbar ist
- PF einen Kindknoten und verschiebt Markierung dorthin

Teil 4: unendliche Bäume Komplementierung Intuitive Beschreibung des Spiels $G_{A,t}$

Intuitive Beschreibung des Spiels G.,

a bewegen sich pro Runde 1 Schritt im Baum durch Markierer von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_l), q_l \in I$

B PF einen Kindknoten und verschiebt Markierung dorthin

16:03 bis 16:12

q₁ bezeichne ab hier einen beliebigen Anfangszustand; brauchen " q_0 " später noch anderweitig.

Es wird also ein Baumautomat mit beliebiger Akzeptanzbedingung (Büchi, Muller, whatever) zugrunde gelegt.

Zwei Spielerinnen Aut (Automat), PF (Pfadfinderin)

- sind abwechselnd an der Reihe.
- bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_I), q_I \in I$

In jeder Runde wählt

- Aut eine Transition, die auf die markierte Position anwendbar ist
- PF einen Kindknoten und verschiebt Markierung dorthin

Spiel läuft ∞ lange, erzeugt ∞ Folge $r = q_0 q_1 q_2$ von Zuständen (bestimmt durch die gewählten Transitionen)

Teil 4: unendliche Bäume Komplementierung Intuitive Beschreibung des Spiels $G_{A,t}$

wei Spielerinnen Aut (Automat), PF (Pfadfinderin

Intuitive Beschreibung des Spiels G.,

- a bewegen sich pro Runde 1 Schritt im Baum durch Markierer von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_l), q_l \in I$
- · Aut eine Transition, die auf die markierte Position anwendbar ist
- # PF einen Kindknoten und verschiebt Markierung dorthin

Spiel läuft ∞ lange, erzeugt ∞ Folge $r = q_1q_1q_2$ von Zuständer

16:03 bis 16:12

q₁ bezeichne ab hier einen beliebigen Anfangszustand; brauchen " q_0 " später noch anderweitig.

Es wird also ein Baumautomat mit beliebiger Akzeptanzbedingung (Büchi, Muller, whatever) zugrunde gelegt.

Zwei Spielerinnen Aut (Automat), PF (Pfadfinderin)

- sind abwechselnd an der Reihe
- bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_I), q_I \in I$

In jeder Runde wählt

- Aut eine Transition, die auf die markierte Position anwendbar ist
- PF einen Kindknoten und verschiebt Markierung dorthin

Spiel läuft ∞ lange, erzeugt ∞ Folge $r=q_0q_1q_2$ von Zuständen (bestimmt durch die gewählten Transitionen)

Aut gewinnt, wenn r der Akzeptanzbedingung von \mathcal{A} entspricht; sonst gewinnt PF (d. h. Aut versucht, \mathcal{A} zum Akzeptieren zu bringen; PF versucht das zu verhindern)

Teil 4: unendliche Bäume

Komplementierung

Intuitive Beschreibung des Spiels $G_{\mathcal{A},t}$

Intuitive Beschreibung des Spiels $G_{A,t}$

- wei Spielerinnen Aut (Automat), PF (Pfadfinderin)
- bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: (ε, q), q; ∈ I
- In jeder Runde wählt

 Aut eine Transition, die auf die markierte Position anwendbar is
- Aut eine Transition, die auf die markierte Position anwendb
 PF einen Kindknoten und verschiebt Markierung dorthin

Spiel läuft ∞ lange, erzeugt ∞ Folge $r = q_0q_1q_2$ von Zuständer (bestimmt durch die gewählten Transitionen)

at gewinnt, wenn r der Akzeptanzbedingung von $\mathcal A$ entspric onst gewinnt PF 1.b. Aut verucht, $\mathcal A$ zum Akzeptieren zu brieger; PF versucht das zu v

16:03 bis 16:12

 q_I bezeichne ab hier einen beliebigen Anfangszustand; brauchen " q_0 " später noch anderweitig.

Es wird also ein Baumautomat mit beliebiger Akzeptanzbedingung (Büchi, Muller, whatever) zugrunde gelegt.

wei Spielerinnen Aut (Automat), PF (Pfadfinderin Komplementierung a bewegen sich pro Runde 1 Schritt im Baum durch Markierer

- Zwei Spielerinnen Aut (Automat), PF (Pfadfinderin)
 - sind abwechselnd an der Reihe.

Intuitive Beschreibung des Spiels $G_{A,t}$

• bewegen sich pro Runde 1 Schritt im Baum durch Markieren von Positionen mit Zuständen; zu Beginn: $(\varepsilon, q_I), q_I \in I$

In jeder Runde wählt

- Aut eine Transition, die auf die markierte Position anwendbar ist
- PF einen Kindknoten und verschiebt Markierung dorthin

Spiel läuft ∞ lange, erzeugt ∞ Folge $r = q_0 q_1 q_2$ von Zuständen (bestimmt durch die gewählten Transitionen)

Aut gewinnt, wenn r der Akzeptanzbedingung von A entspricht; sonst gewinnt PF (d. h. Aut versucht, A zum Akzeptieren zu bringen; PF versucht das zu verhindern)

T 4.10

Komplementierung

16:03 bis 16:12

q₁ bezeichne ab hier einen beliebigen Anfangszustand; brauchen " q_0 " später noch anderweitig.

Intuitive Beschreibung des Spiels $G_{A,t}$

Es wird also ein Baumautomat mit beliebiger Akzeptanzbedingung (Büchi, Muller, whatever) zugrunde gelegt.

Komplementierung

Genaue Beschreibung des Spiels $G_{A,t}$

Spiel ist ein unendlicher Graph

- Knoten sind die Spielpositionen:
 - für Aut: $\{(p,q) \mid p \in \{0,1\}^*, q \in Q\}$ (Positionen im Baum)
 - für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen)

Teil 4: unendliche Bäume

Komplementierung

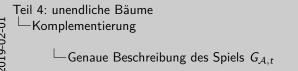
Genaue Beschreibung des Spiels $G_{A,t}$

16:12

Achtung: Ab jetzt bezeichnet p Positionen im Baum, nicht mehr Zustände!

Spiel ist ein unendlicher Graph

- Knoten sind die Spielpositionen:
 - für Aut: $\{(p,q) \mid p \in \{0,1\}^*, q \in Q\}$ (Positionen im Baum)
 - für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen)
- Kanten sind die möglichen **Spielzüge**:
 - $(p,q) \to (q,t(p),q_0,q_1)$
 - $(q, t(p), q_0, q_1) \rightarrow (p_0, q_0)$ \rightarrow $(p1, q_1)$



piel ist ein unendlicher Graph • für Aut: $\{(\rho,q)\mid \rho\in\{0,1\}^*,\ q\in Q\}$ (Positionen im Baum) • $(q, t(p), q_1, q_1) \rightarrow (p0, q_2)$

16:12

Achtung: Ab jetzt bezeichnet p Positionen im Baum, nicht mehr Zustände!

Komplementierung

Spiel ist ein unendlicher Graph

- Knoten sind die Spielpositionen:
 - für Aut: $\{(p,q) \mid p \in \{0,1\}^*, q \in Q\}$ (Positionen im Baum)
 - für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen)
- Kanten sind die möglichen **Spielzüge**:
 - $(p,q) \to (q,t(p),q_0,q_1)$
 - $(q, t(p), q_0, q_1) \rightarrow (p_0, q_0)$ \rightarrow $(p1, q_1)$
- Startknoten: (ε, q_I) für $q_I \in I$ (o. B. d. A. $I = \{q_I\}$)

Teil 4: unendliche Bäume Komplementierung Genaue Beschreibung des Spiels $G_{A,t}$

• für Aut: $\{(\rho,q)\mid \rho\in\{0,1\}^*,\ q\in Q\}$ (Positionen im Baum) • $(q, t(p), q_1, q_1) \rightarrow (p0, q_1)$ a Startknoten: (ε, q_l) für $q_l \in I$ (o.B.d.A. $I = \{q_l\}$)

16:12

2019-02-01

Achtung: Ab jetzt bezeichnet p Positionen im Baum, nicht mehr Zustände!

Komplementierung

Genaue Beschreibung des Spiels $G_{A,t}$

Spiel ist ein unendlicher Graph

- Knoten sind die Spielpositionen:
 - für Aut: $\{(p,q) \mid p \in \{0,1\}^*, q \in Q\}$ (Positionen im Baum)
 - für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen)
- Kanten sind die möglichen **Spielzüge**:
 - $(p,q) \to (q,t(p),q_0,q_1)$
 - $(q, t(p), q_0, q_1) \rightarrow (p_0, q_0)$ \rightarrow $(p1, q_1)$
- Startknoten: (ε, q_I) für $q_I \in I$ (o. B. d. A. $I = \{q_I\}$)

Jede mögliche ∞ Folge von Spielzügen entspricht einem ∞ Pfad im Spielbaum $G_{A,t}$

Genaue Beschreibung des Spiels G., Teil 4: unendliche Bäume piel ist ein unendlicher Graph Komplementierung • für Aut: $\{(\rho,q)\mid \rho\in\{0,1\}^*,\ q\in Q\}$ (Positionen im Baum) • für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen) • $(q, t(p), q_1, q_1) \rightarrow (p0, q_1)$ Genaue Beschreibung des Spiels G_{A+} a Startknoten: (ε, q_l) für $q_l \in I$ (o.B.d.A. $I = \{q_l\}$) Jede mögliche ∞ Folge von Spielzügen entspricht

16:12

2019-02-01

Achtung: Ab jetzt bezeichnet p Positionen im Baum, nicht mehr Zustände!

- Knoten sind die Spielpositionen:
 - für Aut: $\{(p,q) \mid p \in \{0,1\}^*, q \in Q\}$ (Positionen im Baum)
 - für PF: $\{(q, t(p), q_0, q_1) \in \Delta \mid p \in \{0, 1\}^*\}$ (Transitionen)
- Kanten sind die möglichen **Spielzüge**:
 - $(p,q) \to (q,t(p),q_0,q_1)$
 - $(q, t(p), q_0, q_1) \rightarrow (p_0, q_0)$ \rightarrow $(p1, q_1)$
- Startknoten: (ε, q_I) für $q_I \in I$ (o. B. d. A. $I = \{q_I\}$)

Jede mögliche ∞ Folge von Spielzügen entspricht einem ∞ Pfad im Spielbaum $G_{A,t}$

Knoten v' erreichbar von Knoten v: es gibt endliche Folge von Spielzügen von ν nach ν'

Genaue Beschreibung des Spiels G., Teil 4: unendliche Bäume Komplementierung Genaue Beschreibung des Spiels $G_{A,t}$ Jede mögliche ∞ Folge von Spielzügen entspricht einem oo Pfad im Spielbaum Ga -

16:12

2019-02-01

Achtung: Ab jetzt bezeichnet p Positionen im Baum, nicht mehr Zustände!

• für Aut: $\xi(\rho, \sigma) \mid \rho \in \{0, 1\}^*, \ \sigma \in Q\}$ (Positionen im Baum)

• $(q, t(p), q_1, q_1) \rightarrow (p0, q_1)$

Knoten v² erreichbar von Knoten v es gibt endliche Folge von Spielzügen von v nach v

a Startknoten: (ε, q_l) für $q_l \in I$ (o.B.d.A. $I = \{q_l\}$)

Strategie ab Spielposition ν für Spielerin $X \in \{Aut, PF\}$:

einen in v' möglichen Zug zuweist

Funktion, die jeder Zugfolge $v \dots v'$ mit v' Spielposition für X

(legt fest, welchen Zug X in jeder von v aus erreichbaren Spielposition macht)

└─Spielstrategien

Teil 4: unendliche Bäume

16:17 bis 16:26

Automatentheorie u. i. A. WiSe 2018/19

Spielstrategien

Strategie ab Spielposition ν für Spielerin $X \in \{Aut, PF\}$:

Funktion, die jeder Zugfolge $v \dots v'$ mit v' Spielposition für Xeinen in v' möglichen Zug zuweist (legt fest, welchen Zug X in jeder von v aus erreichbaren Spielposition macht)

Gewinnstrategie für Spielerin $X \in \{Aut, PF\}$:

Strategie, die sicherstellt, dass X gewinnt, unabhängig von den Zügen der Gegenspielerin T 4.11 Teil 4: unendliche Bäume Komplementierung -Spielstrategien

Strategie ab Spielposition v für Spielerin X ∈ {Aut, PF}: Funktion, die ieder Zurfolge v...v' mit v' Spielposition für X

Gewinnstrateoie für Spielerin X ∈ {Aut. PF} Strategie, die sicherstellt, dass X gewinnt,

Spielstrategien

unabhängig von den Zügen der Gegenspielerin

16:17 bis 16:26

Funktion, die jeder Zugfolge $v \dots v'$ mit v' Spielposition für X einen in v' möglichen Zug zuweist (legt fest, welchen Zug X in jeder von v aus erreichbaren Spielposition macht)

Gewinnstrategie für Spielerin $X \in \{Aut, PF\}$:

Strategie, die sicherstellt, dass X gewinnt, unabhängig von den Zügen der Gegenspielerin T 4.11

gedächtnislose Strategie:

Strategie, die nur von v' abhängt, nicht von den vorigen Positionen

Teil 4: unendliche Bäume
Co-6106

Spielstrategien

Strategie ab Spielposition v für Spielerin $X \in \{\text{Aut}, \text{PF}\}$: Funktion, die jeder Zugfelge $v \dots v^r$ mit v^r Spielposition für X einen in v^r möglichen Zug zuweist

Spielstrategien

Gewinnstrategie für Spielerin X ∈ {Aut, PF}:

Strategie, die sicherstellt, dass X gewinnt,
unabhänus von den Züsen der Gesenscielerin

dSrbteidose Stratenie

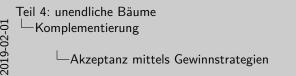
Strategie, die nur von √ abhängt, nicht von den vorigen Positionen

16:17 bis 16:26

Lemma 4.17

Seien $\mathcal{A}=(Q,\Sigma,\Delta,\{q_I\},c)$ ein NPBA und t ein Σ -Baum. Dann gilt:

 $t \in L_{\omega}(\mathcal{A}) \iff$ Aut hat Gewinnstrategie in $G_{\mathcal{A},t}$ ab Position (ε, q_I)



Lemms 4.17 $\label{eq:continuous} \begin{array}{ll} \text{Seion } \mathcal{A} = \{O, \Sigma, \Delta, \{\phi\}, c\} \text{ sin NPBA and } f \text{ sin } \Sigma\text{-Baum.} \\ \text{Dann git:} \\ \varepsilon \in \mathbb{L}(\mathcal{A}) \Leftrightarrow \text{ Aut hat Geninnstrategie in } G_{AI} \text{ ab Position } (\varepsilon, \phi) \end{array}$

Akzeptanz mittels Gewinnstrategien

16:26

Lemma 4.17

Seien $A = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum. Dann gilt:

 $t \in L_{\omega}(A) \Leftrightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon, q_I)$

Beweis:

Konstruiere Gewinnstrategie direkt aus einem erfolgreichen Run und umgekehrt



Seien $A = (Q, \Sigma, \Delta, \{q_t\}, c)$ ein NPBA und t ein Σ -Baum $t \in L_{\omega}(A)$ on Aut hat Gewinnstrategie in $G_{A,t}$ ab Position (ε, q)

16:26

" $t \in L_{\omega}(\mathcal{A}) \Rightarrow ext{Aut hat Gewinnstrategie in } G_{\mathcal{A},t} ext{ ab Position } (arepsilon,q_I)$ "

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für **Aut** wie folgt aus r. Teil 4: unendliche Bäume

Komplementierung

Akzeptanz mittels Gewinnstrategien

Akzeptanz mittels Gewinnstrategien

 $_{a}t\in L_{\omega}(A) \Rightarrow \text{ Aut hat Gowinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon, q_{t})^{n}$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf tKonstruiere Gewinnstrategie für Aut wie folgt aus r.

16:28 bis 16:38

 $,t \in L_{\omega}(A) \Rightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_l)^{"}$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für Aut wie folgt aus r.

• in Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$

Teil 4: unendliche Bäume Komplementierung

-Akzeptanz mittels Gewinnstrategien

 $_{a}t\in L_{\omega}(A) \implies \text{Aut hat Gowinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_{t})^{n}$ Gelte $t \in L$ (A) und sei r erfolgreicher Run von A auf • in Startposition (ε, q_f) withle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$

Akzeptanz mittels Gewinnstrategien

16:28 bis 16:38

Komplementierung

Konstruiere Gewinnstrateaie für Aut wie folgt aus r • in Startoosition (ε, q_f) withle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$ a in allen anderen Spielpos. (p. a) wähle (a. t(p), r(p0), r(p1))

Akzeptanz mittels Gewinnstrategien

 $u,t \in L_{\omega}(A) \Rightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_t)^{*}$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für **Aut** wie folgt aus r.

- in Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$
- in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))

Teil 4: unendliche Bäume Komplementierung

-Akzeptanz mittels Gewinnstrategien

16:28 bis 16:38

Akzeptanz mittels Gewinnstrategien

 $,t \in L_{\omega}(A) \Rightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_l)^*$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für **Aut** wie folgt aus r.

- in Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$
- in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))

Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r.

-Akzeptanz mittels Gewinnstrategien

 $t \in L_{\omega}(A) \implies \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon, q_t)^* \|$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für Aut wie folgt aus r.

Akzeptanz mittels Gewinnstrategien

Konstruiere Gewinnstrategie für Aut wie folgt aus r. • in Startposition (r, a_i) wähle (r(a), (a)), (0), r(0), r(1)• in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeuste Zustandomense einem Pfad in r.

16:28 bis 16:38

2019-02-01

 $,t \in L_{\omega}(A) \Rightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_l)^*$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für **Aut** wie folgt aus r.

- in Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$
- in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))

Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r.

Da r erfolgreich, gewinnt **Aut** nach Definition von $G_{A,t}$.

Teil 4: unendliche Bäume

Komplementierung

—Akzeptanz mittels Gewinnstrategien

Konstruiere Gewinnstrategie für Aut wie folgt aus r. \mathbf{v} in Startposition (ℓ, a) wähle (r(e), t(e), r(0), r(1)) a in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r.

Da r erfolgreich, gewinnt Aut nach Definition von $G_{A,t}$.

 $_{a}t\in L_{\omega}(A) \implies \text{Aut hat Gowinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_{t})^{\alpha}$

Gelte $t \in L$ (A) und sei r erfolgreicher Run von A auf t

16:28 bis 16:38

 $t \in L_{\omega}(A) \Rightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon, q_t)^*$

Gelte $t \in L_{\omega}(A)$ und sei r erfolgreicher Run von A auf t. Konstruiere Gewinnstrategie für **Aut** wie folgt aus r.

• in Startposition (ε, q_I) wähle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$

- in allen anderen Spielpos. (p, q) wähle (q, t(p), r(p0), r(p1))

Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r.

Da r erfolgreich, gewinnt **Aut** nach Definition von $G_{A,t}$.

"Aut hat Gewinnstrategie in $G_{A,t}$ ab Position $(\varepsilon, q_l) \Rightarrow t \in L_{\omega}(A)$ "

T 4.12 □

Akzeptanz mittels Gewinnstrategien Teil 4: unendliche Bäume

Komplementierung

-Akzeptanz mittels Gewinnstrategien

Wenn Aut diese Strategie befolgt, dann entspricht die im Spiel erzeugte Zustandsmenge einem Pfad in r. Da r erfolgreich, gewinnt Aut nach Definition von $G_{A,r}$

a in allen anderen Spielpos. (p. g) wähle (g. t(p), r(p0), r(p1)

 $_{a}t\in L_{\omega}(A) \implies \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon,q_{l})^{n}$

Gelte $t \in L$ (A) und sei r erfolgreicher Run von A auf Konstruiere Gewinnstrategie für Aut wie folgt aus r. • in Startposition (ε, q_f) withle $(r(\varepsilon), t(\varepsilon), r(0), r(1))$

"Aut hat Gewinnstrategie in $G_{A,t}$ ab Position $(\varepsilon, q_t) \implies t \in L_{\omega}(A)^{-1}$

16:28 bis 16:38

Determiniertheit von Paritätsspielen

Klassisches Resultat aus der Spieltheorie, hier nicht bewiesen:

Satz 4.18 (Emerson & Jutla 1991, Mostowski 1991)

Alle Paritätsspiele sind **gedächtnislos determiniert**: genau eine der Spielerinnen hat eine gedächtnislose Gewinnstrategie.

"Paritätsspiel" bezeichnet dabei 2-Personen-Spiele, die

- auf Graphen gespielt werden, deren Knoten mit natürlichen Zahlen markiert sind:
- als Gewinnbedingung für unendliche Spielverläufe die Paritätsbedingung verwenden.

Für alle A und t ist $G_{A,t}$ ein Paritätsspiel.

Teil 4: unendliche Bäume Komplementierung -Determiniertheit von Paritätsspielen

Klassisches Resultat aus der Spieltheorie, hier nicht bewiese Satz 4.18 (Emerson & Jutla 1991, Mostowski 199 Alle Paritätsspiele sind gedächtnislos determiniert: genau eine der Spielerinnen hat eine gedächtnislose Gewinnstrateg deren Knoten mit natürlichen Zahlen markiert sind; u als Gewinnbedingung für unendliche Spielverläufe

die Paritätsbedingung verwender

16:38

Komplementierung

für (c, q) - eine gedächtnislose Gewinnstrategie für Aut oder PF

Folgerung aus Satz 4.18:

Folgerung 4.19

Seien $\mathcal{A} = (Q, \Sigma, \Delta, \{q_l\}, c)$ ein NPBA und t ein Σ -Baum.

Dann gibt es für jede Spielposition v in $G_{A,t}$ — und insbesondere für (ε, q_I) — eine gedächtnislose Gewinnstrategie für Aut oder PF.

Teil 4: unendliche Bäume

—Komplementierung

Determiniertheit von Paritätsspielen

16:39 bis 16:41, 5min Pause

Folgerung aus Satz 4.18:

Folgerung 4.19

Seien $\mathcal{A} = (Q, \Sigma, \Delta, \{q_l\}, c)$ ein NPBA und t ein Σ -Baum.

Dann gibt es für jede Spielposition v in $G_{A,t}$ — und insbesondere für (ε, q_I) — eine gedächtnislose Gewinnstrategie für Aut oder PF.

Folgerung 4.20 (aus Lemma 4.17 und Folgerung 4.19)

 $t \in \overline{L_{\omega}(A)} \Leftrightarrow \mathsf{PF}$ hat gedächtnislose GS ab (ε, q_I) in $G_{A,t}$

Ziel: konstruieren NPBA, um deren Existenz zu testen

Teil 4: unendliche Bäume

Komplementierung

Determiniertheit von Paritätsspielen

Following 4.19: Since $A = \{0, T, \Delta_i(\phi_i), c\}$ is NPBA and x on T. Blaim Since $A = \{0, T, \Delta_i(\phi_i), c\}$ is NPBA and x on T. Blaim Dans glat is the jade Spolyosition v in $G_{A,v}$ and independent for (x, ϕ) — one get-ficthen-loss Genientz-stage for Aut oder PF. Following 4.20 (aux Lamma 4.17 and Folgosumg 4.19) $x \in \mathbb{Z}[A] = 0$ or PF has get-fichen-loss GS is (x, ϕ) in $G_{A,t}$.

Determiniertheit von Paritätsspielen

16:39 bis 16:41, 5min Pause

Gewinnbäume

Betrachten gedächtnislose Gewinnstrategien für **PF** als Menge von Funktionen

$$f_p: \Delta \to \{0,1\}$$
 für jede Baumposition $p \in \{0,1\}^*$

Idee: f_p weist jeder Transition, die Aut in Baumposition p wählt, einen Spielzug (Richtung 0/1) zu

Teil 4: unendliche Bäume

Komplementierung

Gewinnbäume

16:46 bis 16:49

Gewinnbäume

Betrachten gedächtnislose Gewinnstrategien für PF als Menge von

Funktionen $f_p:\Delta \to \{0,1\}$ für jede Baumposition $p\in \{0,1\}^*$ floet: f_p weist jeder Transition, die Aut in Baumposition p wäheinen Spielzug (Richtung (0,1) zu

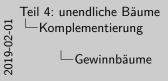
Gewinnbäume

Betrachten gedächtnislose Gewinnstrategien für PF als Menge von Funktionen

$$f_p: \Delta \to \{0,1\}$$
 für jede Baumposition $p \in \{0,1\}^*$

Idee: f_p weist jeder Transition, die Aut in Baumposition p wählt, einen Spielzug (Richtung 0/1) zu

- Sei *F* die Menge dieser Funktionen
- Ordnen die f_p in einem **F**-Baum s an (Strategiebaum)



Betrachten gedächtnislose Gewinnstrategien für PF als Menge vo $f_p : \Delta \to \{0,1\}$ für jede Baumposition $p \in \{0,1\}$

Gewinnbäume

· Ordnen die f., in einem F-Baum s an (Strategiebaum)

16:46 bis 16:49

Komplementierung

Betrachten gedächtnislose Gewinnstrategien für **PF** als Menge von Funktionen

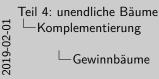
$$f_p: \Delta \to \{0,1\}$$
 für jede Baumposition $p \in \{0,1\}^*$

Idee: f_p weist jeder Transition, die Aut in Baumposition p wählt, einen Spielzug (Richtung 0/1) zu

- Sei *F* die Menge dieser Funktionen
- Ordnen die f_p in einem **F**-Baum s an (Strategiebaum)

PF-Gewinnbaum für t:

ein F-Baum, der eine Gewinnstrategie für **PF** in $G_{A,t}$ kodiert



 $f_p : \Delta \to \{0,1\}$ für jede Baumposition $p \in \{0,1\}$ · Ordnen die f., in einem F-Baum s an (Strategiebaum)

Betrachten gedächtnislose Gewinnstrategien für PF als Menge vo

ein F-Baum, der eine Gewinnstrategie für PF in GA, kodiert

16:46 bis 16:49

Betrachten gedächtnislose Gewinnstrategien für **PF** als Menge von Funktionen

$$f_p: \Delta \to \{0,1\}$$
 für jede Baumposition $p \in \{0,1\}^*$

Idee: f_p weist jeder Transition, die Aut in Baumposition p wählt, einen Spielzug (Richtung 0/1) zu

- Sei *F* die Menge dieser Funktionen
- Ordnen die f_p in einem **F**-Baum s an (Strategiebaum)

PF-Gewinnbaum für t:

ein F-Baum, der eine Gewinnstrategie für **PF** in $G_{A,t}$ kodiert

Folgerung 4.21 (aus Folgerung 4.20)

 $t \in L_{\omega}(\mathcal{A}) \iff$ es gibt einen **PF**-Gewinnbaum für t

Neues Ziel: bauen NPBA, um Existenz PF-Gewinnbaum zu testen

Teil 4: unendliche Bäume Komplementierung

-Gewinnbäume

16:46 bis 16:49

- $f_p : \Delta \rightarrow \{0,1\}$ für jede Baumposition $p \in \{0,1\}$
- · Ordnen die f., in einem F-Baum s an (Strategiebaum)
- ein F-Baum, der eine Gewinnstrategie für PF in Ga, kodier

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum **Zwischenziel**: Prüfen, ob gegebener F-Baum s **kein PF**-GB ist

Teil 4: unendliche Bäume

Komplementierung

Existenz von PF-Gewinnbäumen (**PF**-GB)

Existenz von PF-Gewinnbäumen (PF-GB)

16:49 bis 16:58

Model-Checking CTL Baumautomaten Komplementierung

Existenz von PF-Gewinnbäumen (PF-GB)

Sei $\mathcal{A}=(Q,\Sigma,\Delta,\{q_I\},c)$ ein NPBA und t ein Σ -Baum **Zwischenziel:** Prüfen, ob gegebener F-Baum s **kein PF**-GB ist

Idee:

- Benutzen NPA A' (ω -Wortautomat)
- \mathcal{A}' prüft für jeden Pfad π in t und jeden möglichen Spielzug von **Au**t separat, ob Akzeptanzbedingung von \mathcal{A} erfüllt ist
- \rightarrow \mathcal{A}' akzeptiert $\geqslant 1$ Pfad \Leftrightarrow s ist kein **PF**-Gewinnbaum für t

Teil 4: unendliche Bäume

Komplementierung

Existenz von PF-Gewinnbäumen (**PF**-GB)

Existenz von PF-Gewinnbäumen (PF-GB)

Sei $A = (Q, \Sigma, \Delta, \{q_l\}, c)$ ein NPBA und t ein Σ -Baum

 Benutzen NPA A' (ω-Wortautomat)
 A' prüft für jeden Pfad π in t und jeden möglichen Spielzug von Auf separat, ob Akzeptanzbedingung von A erfüllt ist
 Azspatiert ≥ 1 Pfad en s ist kein PF-Gewinnbaum für

16:49 bis 16:58

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum Zwischenziel: Prüfen, ob gegebener F-Baum s kein PF-GB ist

Idee:

- Benutzen NPA A' (ω -Wortautomat)
- \mathcal{A}' prüft für jeden Pfad π in t und jeden möglichen Spielzug von Aut separat, ob Akzeptanzbedingung von \mathcal{A} erfüllt ist
- \rightarrow \mathcal{A}' akzeptiert $\geqslant 1$ Pfad \Leftrightarrow s ist kein **PF**-Gewinnbaum für t

Sei $\pi \in \{0,1\}^{\omega}$ ein Pfad mit $\pi = \pi_1 \pi_2 \pi_3 \dots$

 \mathcal{A}' arbeitet auf Wörtern der folgenden Form:

$$\langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3 \rangle \dots$$

Sei L_{s,t} die Sprache aller dieser Wörter

Teil 4: unendliche Bäume Komplementierung

Existenz von PF-Gewinnbäumen (**PF**-GB)

Existenz von PF-Gewinnbäumen (PF-GB)

Sei $A = (Q, \Sigma, \Delta, \{q_f\}, c)$ ein NPBA und t ein Σ -Baun Zwischenziel: Prüfen, ob gegebener F-Baum s kein PF-GB ist

- Donotton MDA M (c. Mostautoma
- $\mathbf{u} \ \mathcal{A}'$ prüft für ieden Pfad π in t und ieden möglichen Spielzu von Aut separat, ob Akzeptanzbedingung von A erfüllt ist → A' akzeptiert ≥ 1 Pfad eo s ist kein PF-Gewinnbaum für.
- Sei $\pi \in \{0,1\}^{\omega}$ ein Pfad mit $\pi = \pi_1 \pi_2 \pi_3$. A' arbeitet auf Wörtern der folgenden Form:
- $\langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle$ $\langle s(\pi_1), t(\pi_1), \pi_2 \rangle$ $\langle s(\pi_1\pi_2), t(\pi_1\pi_2), \pi_3 \rangle$ Sei L., die Sprache aller dieser Wörter

16:49 bis 16:58

Zwischenziel: Prüfen, ob gegebener F-Baum s kein PF-GB ist

Idee:

- Benutzen NPA A' (ω -Wortautomat)
- \mathcal{A}' prüft für jeden Pfad π in t und jeden möglichen Spielzug von **Aut** separat, ob Akzeptanzbedingung von \mathcal{A} erfüllt ist
- \rightarrow \mathcal{A}' akzeptiert $\geqslant 1$ Pfad $\Leftrightarrow s$ ist kein **PF**-Gewinnbaum für t

Sei $\pi \in \{0,1\}^{\omega}$ ein Pfad mit $\pi = \pi_1 \pi_2 \pi_3 \dots$

 \mathcal{A}' arbeitet auf Wörtern der folgenden Form:

$$\langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \langle s(\pi_1 \pi_2), t(\pi_1 \pi_2), \pi_3 \rangle \dots$$

Sei $L_{s,t}$ die Sprache aller dieser Wörter

Beispiel: s. Tafel

T 4.13

Komplementierung

Teil 4: unendliche Bäume Komplementierung

Existenz von PF-Gewinnbäumen (PF-GB)

a. A poir für jeden Pfad π in trud jeden möglichen Spidzu von Aut speart, och Anzeptarchefüngen von A effilität vor. Af abzeptarchefüngen von A effilität vor. Af abzeptarch pfach mit $\pi = \pi_1\pi_2\pi_2\dots$ Sizi π ein $(0,1)^n$ ein Pfad mit $\pi = \pi_1\pi_2\pi_2\dots$ A arbeiteta ad Winnen der folgenden Form: $\left\langle s(c), t(c), \pi_2\right\rangle \left\langle s(\pi_1), t(\pi_1, \pi_2)\right\rangle \left\langle s(\pi_2, \pi_2), t(\pi_1\pi_2), \pi_2\right\rangle \dots$ Sizi L_{c_i} die Syzache aller dieser Wörter Bessele λ . Tall

16:49 bis 16:58

Sei $A = (Q, \Sigma, \Delta, \{a_i\}, c)$ ein NPBA und t ein Σ -Baun

Konstruieren NPA $A' = (Q, \Sigma', \Delta', \{q_i\}, c)$ wie folgt

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}' = (Q, \Sigma', \Delta', \{q_I\}, c)$ wie folgt:

Teil 4: unendliche Bäume Komplementierung

-Konstruktion des Wortautomaten für Gewinnbäume

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

 $\mathbf{v} \Sigma' = \{ \langle f, a, i \rangle \mid f \in F, a \in \Sigma, i \in \{0, 1\} \}$

Sei $\mathcal{A}=(Q,\Sigma,\Delta,\{q_I\},c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}'=(Q,\Sigma',\Delta',\{q_I\},c)$ wie folgt:

•
$$\Sigma' = \{ \langle f, a, i \rangle \mid f \in F, a \in \Sigma, i \in \{0, 1\} \}$$

Teil 4: unendliche Bäume Komplementierung

Konstruktion des Wortautomaten für Gewinnbäume

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

2019-02-01

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}' = (Q, \Sigma', \Delta', \{q_I\}, c)$ wie folgt:

$$\bullet \ \Sigma' = \left\{ \langle f, a, i \rangle \ \middle| \ f \in F, \ a \in \Sigma, \ i \in \{0, 1\} \right\}$$

• Q, c wie in A (wollen Akzeptanz von A prüfen)

Teil 4: unendliche Bäume

Komplementierung

Konstruktion des Wortautomaten für Gewinnbäume Sei $A = (Q, \Sigma, \Delta, \{q_i\}, c)$ ein NPBA und ϵ ein Σ -Baum Konstruieren NPA $A' = (Q, \Sigma', \Delta', \{q_i\}, c)$ wie folgt: • $\Sigma' = \{d', a, i\} \mid \ell \in \mathcal{F}, a \in \Sigma, i \in \{0, 1\}\}$ • Q, C wie in A (wollen Azegusav on A polisie)

—Konstruktion des Wortautomaten für Gewinnbäume

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_l\}, c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}' = (Q, \Sigma', \Delta', \{q_l\}, c)$ wie folgt:

- $\Sigma' = \{ \langle f, a, i \rangle \mid f \in F, a \in \Sigma, i \in \{0, 1\} \}$
- Q, c wie in A (wollen Akzeptanz von A prüfen)
- $\Delta' = \{ (q, \langle f, a, i \rangle, q'_i) \mid \langle f, a, i \rangle \in \Sigma', i \in \{0, 1\},$ es gibt $\delta = (q, a, q'_0, q'_1) \in \Delta \text{ mit } f(\delta) = i$

Teil 4: unendliche Bäume Komplementierung

> -Konstruktion des Wortautomaten für Gewinnbäume

Sei $A = (Q, \Sigma, \Delta, \{\alpha\}, c)$ ein NPBA und t ein Σ -Baun Konstruieren NPA $A' = (Q, \Sigma', \Delta', \{q_i\}, c)$ wie folgt $\mathbf{u} \; \Sigma' = \{ \langle f, a, i \rangle \mid f \in F, \; a \in \Sigma, \; i \in \{0, 1\} \}$ u Q, c wie in A (wollen Akzeptanz von A prüfen)

• $\Delta' = \{(q, \langle f, a, i \rangle, q'_i) \mid \langle f, a, i \rangle \in \Sigma', i \in \{0, 1\}.$ es gibt $\delta = (a, a, a', a') \in \Delta \text{ mit } f(\delta) = i$

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

Achtung: $f \in F$ bedeutet "Funktion aus der Menge aller Funktionen (Gewinnstrategien)", nicht "akzeptierender Zustand"!

Komplementierung

Komplementierung

2019-02-01

Sei $\mathcal{A}=(Q,\Sigma,\Delta,\{q_I\},c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}'=(Q,\Sigma',\Delta',\{q_I\},c)$ wie folgt:

- $\Sigma' = \{ \langle f, a, i \rangle \mid f \in F, a \in \Sigma, i \in \{0, 1\} \}$
- Q, c wie in A (wollen Akzeptanz von A prüfen)
- $\Delta' = \left\{ \left(q, \ \langle f, a, i \rangle, \ q'_i \right) \ \middle| \ \langle f, a, i \rangle \in \Sigma', \ i \in \{0, 1\}, \right.$ es gibt $\delta = (q, a, q'_0, q'_1) \in \Delta \ \text{mit} \ f(\delta) = i \right\}$

 \mathcal{A}' prüft für jeden möglichen Zug von Aut, ob Aut gewinnen kann

Teil 4: unendliche Bäume

Komplementierung

-Konstruktion des Wortautomaten für Gewinnbäume $16:58 \rightarrow \text{mit Beweis bis } 17:30$

s ist ein PF-Gewinnbaum für t eo L_{s} , $\cap L_{s}(A^{s}) = \emptyset$

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}' = (Q, \Sigma', \Delta', \{q_I\}, c)$ wie folgt:

•
$$\Sigma' = \{ \langle f, a, i \rangle \mid f \in F, a \in \Sigma, i \in \{0, 1\} \}$$

• Q, c wie in A (wollen Akzeptanz von A prüfen)

 \mathcal{A}' prüft für jeden möglichen Zug von Aut, ob Aut gewinnen kann

Lemma 4.22

s ist ein PF-Gewinnbaum für $t \Leftrightarrow L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$

Teil 4: unendliche Bäume

Komplementierung

Konstruktion des Wortautomaten für Gewinnbäume

Sis $A = (O, T, \Delta, \{a_i\}, c)$ is in PERA use t on Σ Elson Kontrainen IPA $A = (O, T, \Delta', \{a_i\}, c)$, which follows $V = \{t', s, a'\} \mid t' \in F, s \in \Sigma, i \in \{0, 1\}\}$ = (O, c) with A = (notes A boungars on a, points) $= \Delta' = \{(c, (r, s, i), c) \mid (r, s, i) \in V, i \in \{0, 1\}, c)$ $= \Delta' = \{(c, (r, s, i), c) \mid (r, s, i) \in V, i \in \{0, 1\}, c)$ $= \Delta' = \{(c, (r, s, i), c) \mid (r, s, i) \in V, i \in \{0, 1\}, c)$ $= \Delta' \text{ prift for judin miglician Zig von Aut, ob Aut govinness kann (Lemma 4.22).}$

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

Achtung: $f \in F$ bedeutet "Funktion aus der Menge aller Funktionen (Gewinnstrategien)", nicht "akzeptierender Zustand"!

Komplementierung

Konstruktion des Wortautomaten für Gewinnbäume

Sei $\mathcal{A} = (Q, \Sigma, \Delta, \{q_l\}, c)$ ein NPBA und t ein Σ -Baum Konstruieren NPA $\mathcal{A}' = (Q, \Sigma', \Delta', \{q_l\}, c)$ wie folgt:

$$\bullet \ \Sigma' = \left\{ \langle f, a, i \rangle \ \middle| \ f \in F, \ a \in \Sigma, \ i \in \{0, 1\} \right\}$$

• Q, c wie in A (wollen Akzeptanz von A prüfen)

•
$$\Delta' = \left\{ \left(q, \ \langle f, a, i \rangle, \ q'_i \right) \ \middle| \ \langle f, a, i \rangle \in \Sigma', \ i \in \{0, 1\}, \right.$$

es gibt $\delta = (q, a, q'_0, q'_1) \in \Delta \ \text{mit} \ f(\delta) = i \right\}$

 \mathcal{A}' prüft für jeden möglichen Zug von Aut, ob Aut gewinnen kann

Lemma 4.22

s ist ein PF-Gewinnbaum für $t \Leftrightarrow L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$

Beweis: s. Tafel

T 4.14

Komplementierung

Teil 4: unendliche Bäume Sei $A = (Q, \Sigma, \Delta, \{a_i\}, c)$ ein NPBA und t ein Σ -Baun Komplementierung $\Sigma' = \{(f, a, i) | f \in F, a \in \Sigma, i \in \{0, 1\}\}$ u Q, c wie in A (wollen Akzeptanz von A prüfen) $\bullet \Delta' = \{(q, (f, a, i), q_i') | (f, a, i) \in \Sigma', i \in \{0, 1\}.$ Konstruktion des Wortautomaten für Gewinnbäume

 $16:58 \rightarrow \text{mit Beweis bis } 17:30$

Komplementierung: Was bisher geschah



- Gegeben: NPBA $\mathcal{A} = (Q, \Sigma, \Delta, \{q_I\}, c)$
- Ordnen A und jedem **Eingabebaum** t ein 2-Pers.-Spiel $G_{A,t}$ zu
- Spielerin Aut wählt Transition für aktuelle Position in t; **PF** wählt Kindsposition (→ gibt schrittweise Pfad vor)
- Aut gewinnt, wenn gespielter Pfad c entspricht

Lemma 4.17

 $t \in L_{\omega}(\mathcal{A}) \Leftrightarrow \text{Aut hat Gewinnstrategie in } G_{A,t} \text{ ab Position } (\varepsilon, q_I)$

Mittels Resultat aus der Spieltheorie folgt:

Folgerung 4.20

 $t \in L_{\omega}(A) \Leftrightarrow \mathsf{PF}$ hat gedächtnislose GS ab (ε, q_I) in $G_{A,t}$

Komplementierung: Was bisher geschah Teil 4: unendliche Bäume Komplementierung -Komplementierung: Was bisher geschah

a Ordnen A und iedem Eingabebaum t ein 2-Pers - Spiel Ga - zu Snielerin Aut wählt Transition für aktuelle Position in t: u Aut gewinnt, wenn gespielter Pfad c entsprich $t \in \overline{L_1(A)}$ exp PF hat gedächteislose GS ab (s, a_1) in G_A ,

8:30

Komplementierung: Was bisher geschah



 Betrachten gedächtnislose Gewinnstrategie für PF als Menge F von Funktionen

$$f_p:\Delta \to \{0,1\}$$
 für jede Baumposition $p\in \{0,1\}^*$

- Ordnen die f_p in F-Baum s an
- PF-Gewinnbaum: F-Baum für eine Gewinnstrategie von PF

Dann folgt sofort:

Folgerung 4.21

 $t \in \overline{L_{\omega}(\mathcal{A})} \iff$ es gibt einen **PF**-Gewinnbaum für t

Teil 4: unendliche Bäume

Komplementierung

Bushase gediktsinder Geninnstatige für FF als blein eine Ff Geninnstatig für FF als blein gestellt der Geninnstatig für FF als blein einer Ff Geninnstatig für FF als blein einer Ff Geninnstatig von FF Geninnstatig von FF Geninnstatig von FF Geninnstatig von FF Geninnstatig für Geninnstatig von FF Geninnstatig für gestellt für Geninnstatig für gestellt für Geninnstatig für gestellt für geninnstatig von FF Geninnstatig für geninnstatig für geninnstatig von FF Geninnstatig für geninnstat

8:32

F-Baum auch "Strategiebaum"; $F \triangleq \text{alle } f_p$

Komplementierung: Was bisher geschah

Konstruieren NPA \mathcal{A}' (ω -Wortautomat!), um **PF**-Gewinnbäume zu erkennen:

• Eingabewörter haben die Form

$$\langle s(\varepsilon), t(\varepsilon), \pi_1 \rangle \langle s(\pi_1), t(\pi_1), \pi_2 \rangle \langle s(\pi_1 \pi_2), t(\pi_1 \pi_2), \pi_3 \rangle \dots$$

• Sei $L_{s,t}$ die Menge aller solcher Wörter

Konstruktion von A' stellt sicher:

Lemma 4.22

s ist ein **PF**-Gewinnbaum für $t \Leftrightarrow L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$

Teil 4: unendliche Bäume

└─Komplementierung

└─Komplementierung: Was bisher geschah

8:34

2019-02-01

Konstruktion des Komplementautomaten für \mathcal{A}

Gesucht: (siehe Folgerung 4.21)

NPBA \mathcal{B} , der t akzeptiert gdw. es einen **PF**-Gewinnbaum für t gibt

Teil 4: unendliche Bäume Komplementierung

Konstruktion des Komplementautomaten für A

 \sqsubseteq Konstruktion des Komplementautomaten für $\mathcal A$

8:36 bis 8:40

$$L_{s,t} \subseteq \overline{L_{\omega}(\mathcal{A}')}$$
": ist äquivalent zu $L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$ " aus L. 4.22

Zu Schritt 1:

 $NPA \rightarrow NMA \rightarrow^* NBA - Safra \rightarrow^* DRA \rightarrow DMA \rightarrow^* DPA$ *Resultate aus Vorlesung, exp. Blowup Es gibt effizientere direkte Konstruktion NPA \rightarrow DPA (sage am Ende was dazu).

Zu Schritt 2:

 \mathcal{A}'' braucht ja als Eingabe Tripel, 1. Komponente $s(\cdot)$ aus Strategie. Diese Info wird schrittweise geraten.

Gesucht: (siehe Folgerung 4.21)

NPBA \mathcal{B} , der t akzeptiert gdw. es einen **PF**-Gewinnbaum für t gibt

Wegen Lemma 4.22 muss \mathcal{B} akzeptieren gdw. $L_{s,t} \subset \overline{L_{\omega}(\mathcal{A}')}$

Teil 4: unendliche Bäume Komplementierung

 \sqsubseteq Konstruktion des Komplementautomaten für \mathcal{A}

8:36 bis 8:40

$$L_{s,t} \subseteq \overline{L_{\omega}(\mathcal{A}')}$$
": ist äquivalent zu $L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$ " aus L. 4.22

Zu Schritt 1:

 $NPA \rightarrow NMA \rightarrow^* NBA - Safra \rightarrow^* DRA \rightarrow DMA \rightarrow^* DPA$ *Resultate aus Vorlesung, exp. Blowup Es gibt effizientere direkte Konstruktion NPA \rightarrow DPA (sage am Ende was dazu).

Zu Schritt 2:

 \mathcal{A}'' braucht ja als Eingabe Tripel, 1. Komponente $s(\cdot)$ aus Strategie. Diese Info wird schrittweise geraten.

Konstruktion des Komplementautomaten für \mathcal{A}

Gesucht: (siehe Folgerung 4.21)

NPBA \mathcal{B} , der t akzeptiert gdw. es einen **PF**-Gewinnbaum für t gibt

Wegen Lemma 4.22 muss \mathcal{B} akzeptieren gdw. $L_{s,t} \subset L_{\omega}(\mathcal{A}')$

Konstruktion von \mathcal{B} in 2 Schritten:

Schritt 1

- Sei $\mathcal{A}'' = (Q'', \Sigma', \Delta'', q'', c'')$ der DPA mit $L_{\omega}(\mathcal{A}'') = \overline{L_{\omega}(\mathcal{A}')}$
- A" ist deterministisch: Safra-Konstruktion (+ Umwandlung zwischen den Automatentypen)

Teil 4: unendliche Bäume Komplementierung

 \sqsubseteq Konstruktion des Komplementautomaten für ${\cal A}$

Konstruktion des Komplementautomaten für A

8:36 bis 8:40

$$L_{s,t} \subset \overline{L_{\omega}(\mathcal{A}')}$$
": ist äquivalent zu $L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$ " aus L. 4.22

Zu Schritt 1:

 $NPA \rightarrow NMA \rightarrow^* NBA - Safra \rightarrow^* DRA \rightarrow DMA \rightarrow^* DPA$ *Resultate aus Vorlesung, exp. Blowup Es gibt effizientere direkte Konstruktion NPA \rightarrow DPA (sage am Ende was dazu).

Zu Schritt 2:

 \mathcal{A}'' braucht ja als Eingabe Tripel, 1. Komponente $s(\cdot)$ aus Strategie. Diese Info wird schrittweise geraten.

Konstruktion des Komplementautomaten für ${\cal A}$

Gesucht: (siehe Folgerung 4.21)

NPBA \mathcal{B} , der t akzeptiert gdw. es einen **PF**-Gewinnbaum für t gibt

Wegen Lemma 4.22 muss \mathcal{B} akzeptieren gdw. $L_{s,t} \subseteq \overline{L_{\omega}(\mathcal{A}')}$

Konstruktion von \mathcal{B} in 2 Schritten:

Schritt 1

- Sei $\mathcal{A}'' = (Q'', \Sigma', \Delta'', q''_l, c'')$ der **D**PA mit $L_{\omega}(\mathcal{A}'') = \overline{L_{\omega}(\mathcal{A}')}$
- A" ist deterministisch: Safra-Konstruktion
 (+ Umwandlung zwischen den Automatentypen)

Schritt 2

 \mathcal{B} soll auf jedem Pfad von t

- A" laufen lassen
- und "parallel" dazu eine Strategie für **PF** raten

8:36 bis 8:40

$$L_{s,t} \subset \overline{L_{\omega}(\mathcal{A}')}$$
": ist äquivalent zu $L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$ " aus L. 4.22

Zu Schritt 1:

NPA \to NMA \to * NBA —Safra \to * DRA \to DMA \to * DPA *Resultate aus Vorlesung, exp. Blowup Es gibt effizientere direkte Konstruktion NPA \to DPA (sage am Ende was dazu).

Zu Schritt 2:

 \mathcal{A}'' braucht ja als Eingabe Tripel, 1. Komponente $s(\cdot)$ aus Strategie. Diese Info wird schrittweise geraten.

Konstruktion des Komplementautomaten für ${\cal A}$

Idee: NPBA $\mathcal B$ soll $\mathcal A''$ auf jedem Pfad simulieren, indem $\mathcal B$

- s rät (also pro Position $p \in f_p$)
- sich ansonsten wie \mathcal{A}'' verhält, (also pro Position die Folgezustände q_0, q_1 gemäß Δ'' setzt)

 \mathcal{A}'' deterministisch \Rightarrow Zustand pro Position p eindeutig bestimmt

Teil 4: unendliche Bäume

Komplementierung

Idea: NPBA \mathcal{B} soil \mathcal{A}'' auf jedem Pfad simulieren, indem \mathcal{B} \bullet s rikt (also pro Position p ein f_p)

• sich ansonsten wie \mathcal{A}'' verhält,

(also pro Position die Folgozzattinde q_0, q_1 gemäß Δ'' setzt) \mathcal{A}'' deterministisch \Rightarrow Zustand pro Position p eindeutig bestimmt

Konstruktion des Komplementautomaten für A

 $lue{}$ Konstruktion des Komplementautomaten für ${\mathcal A}$

 $8:40 \rightarrow bis 9:10$

DPA \mathcal{A}'' "bezeugt", dass s kein **PF**-Gewinnbaum für t ist

" \mathcal{A}'' deterministisch ... ": obwohl ∞ viele Pfade durch p gehen!

Komplementierung

Idee: NPBA \mathcal{B} soll \mathcal{A}'' auf jedem Pfad simulieren, indem \mathcal{B}

- s rät (also pro Position p ein f_p)
- sich ansonsten wie \mathcal{A}'' verhält. (also pro Position die Folgezustände q_0, q_1 gemäß Δ'' setzt)

 \mathcal{A}'' deterministisch \Rightarrow Zustand pro Position p eindeutig bestimmt

Konstruktion von $\mathcal{B} = (Q'', \Sigma, \Delta^{\text{neu}}, q''_{l}, c'')$:

- Q'', q'', c'' werden von A'' übernommen
- $\Delta^{\text{neu}} = \{(q, a, q_0, q_1) \mid \text{es gibt } f \in F \text{ mit } \}$

$$(q, \langle f, a, i \rangle, q_i) \in \Delta''$$
 für $i = 0, 1$

Teil 4: unendliche Bäume Komplementierung \sqsubseteq Konstruktion des Komplementautomaten für ${\cal A}$

also pro Position die Folgezustände q₀, q₁ gemäß Δ" setzt) " deterministisch => Zustand pro Position p eindeutig bestimm $a \Delta^{nou} = \{(q, a, q_0, q_1) \mid \text{es gibt } f \in F \text{ mit }$ $(q, (f, a, i), q_i) \in \Delta''$ für i = 0, 1

 $8:40 \to \text{bis } 9:10$

DPA A'' "bezeugt", dass s kein **PF**-Gewinnbaum für t ist

", \mathcal{A}'' deterministisch ...": obwohl ∞ viele Pfade durch p gehen!

 $f \in F$: Für jede mögliche Funktion $f : \Delta \to \{0,1\}$ ("... s rät"!) und zugehörigen Übergang in Δ'' , ein neuer Übergang

Komplementierung

Idee: NPBA \mathcal{B} soll \mathcal{A}'' auf jedem Pfad simulieren, indem \mathcal{B}

- s rät (also pro Position p ein f_p)
- sich ansonsten wie \mathcal{A}'' verhält. (also pro Position die Folgezustände q_0, q_1 gemäß Δ'' setzt)

 \mathcal{A}'' deterministisch \Rightarrow Zustand pro Position p eindeutig bestimmt

Konstruktion von $\mathcal{B} = (Q'', \Sigma, \Delta^{\text{neu}}, q''_{l}, c'')$:

- Q'', q'', c'' werden von A'' übernommen
- $\Delta^{\mathsf{neu}} = \big\{ (q, a, q_0, q_1) \mid \mathsf{es} \; \mathsf{gibt} \; f \in F \; \mathsf{mit} \big\}$

$$(q, \langle f, a, i \rangle, q_i) \in \Delta'' \text{ für } i = 0, 1$$

Lemma 4.23

 $t \in L_{\omega}(\mathcal{B})$ gdw. es gibt F-Baum s mit $L_{s,t} \subset L_{\omega}(\mathcal{A}'')$

Teil 4: unendliche Bäume Komplementierung

 \sqsubseteq Konstruktion des Komplementautomaten für ${\cal A}$

Konstruktion des Komplementautomaten für A also oro Position die Folgezustände q₀, q₁ gemäß Δ" setzt) " deterministisch - Zustand pro Position p eindeutig bestimm $a \Delta^{nou} = \{(q, a, q_0, q_1) \mid \text{es gibt } f \in F \text{ mit }$ $(q, (f, a, i), q) \in \Delta'' \text{ für } i = 0, 1$

 $t \in L_{\omega}(B)$ gdw. es gibt F-Baum s mit $L_{s,t} \subset L_{\omega}(A'')$

 $8:40 \to \text{bis } 9:10$

DPA A'' "bezeugt", dass s kein **PF**-Gewinnbaum für t ist

", \mathcal{A}'' deterministisch ...": obwohl ∞ viele Pfade durch p gehen!

 $f \in F$: Für jede mögliche Funktion $f : \Delta \to \{0,1\}$ ("... s rät"!) und zugehörigen Übergang in Δ'' , ein neuer Übergang

Lemma ankündigen mit: "es bleibt zu zeigen '

Idee: NPBA \mathcal{B} soll \mathcal{A}'' auf jedem Pfad simulieren, indem \mathcal{B}

- s rät (also pro Position p ein f_p)
- sich ansonsten wie \mathcal{A}'' verhält. (also pro Position die Folgezustände q_0, q_1 gemäß Δ'' setzt)

 \mathcal{A}'' deterministisch \Rightarrow Zustand pro Position p eindeutig bestimmt

Konstruktion von $\mathcal{B} = (Q'', \Sigma, \Delta^{\text{neu}}, q''_{l}, c'')$:

- Q'', q'', c'' werden von A'' übernommen
- $\Delta^{\mathsf{neu}} = \big\{ (q, a, q_0, q_1) \mid \mathsf{es} \; \mathsf{gibt} \; f \in F \; \mathsf{mit} \big\}$

$$(q,\langle f,a,i\rangle,q_i)\in\Delta''$$
 für $i=0,1$

Lemma 4.23

 $t \in L_{\omega}(\mathcal{B})$ gdw. es gibt F-Baum s mit $L_{s,t} \subset L_{\omega}(\mathcal{A}'')$

Beweis: siehe Tafel

T 4.15 \square

Komplementierung

Teil 4: unendliche Bäume Komplementierung \sqsubseteq Konstruktion des Komplementautomaten für ${\cal A}$

also oro Position die Folgezustände q₀, q₁ gemäß Δ" setzt) deterministisch - Zustand pro Position p eindeutig bestimm $a \Delta^{nou} = \{(q, a, q_0, q_1) \mid \text{ as gibt } f \in F \text{ mit }$ $(q, (f, a, i), q) \in \Delta'' \text{ für } i = 0, 1$

 $8:40 \to \text{bis } 9:10$

DPA A'' "bezeugt", dass s kein **PF**-Gewinnbaum für t ist

", \mathcal{A}'' deterministisch ...": obwohl ∞ viele Pfade durch p gehen!

 $f \in F$: Für jede mögliche Funktion $f : \Delta \to \{0,1\}$ ("... s rät"!) und zugehörigen Übergang in Δ'' , ein neuer Übergang

Lemma ankündigen mit: "es bleibt zu zeigen '

... Es darf aufgeatmet werden ...

9:10

-Komplementierung

Jetzt haben wir alles Technische geschafft und können das Hauptresultat sehr einfach beweisen.

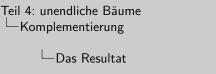
Danach haben wir uns eine Pause verdient. :)

Für jeden NPBA A gibt es einen NPBA B mit $L_{\omega}(B) = \overline{L_{\omega}(A)}$.

Das Resultat

Satz 4.24 (Rabin 1969)

Für jeden NPBA \mathcal{A} gibt es einen NPBA \mathcal{B} mit $L_{\omega}(\mathcal{B}) = \overline{L_{\omega}(\mathcal{A})}$.



9:10 bis 9:13

Jetzt müssen wir nur noch alle Lemmas zusammentun und erhalten das Hauptresultat.

"Konstr. \mathcal{A}'' ": ist Komplementärautomat zu \mathcal{A}'

TODO: Wenn Zeit ist, Beispiel für die Konstruktion vorführen (siehe Notizen in Sammlung nach T4.15, zweites=einfacheres Bsp.)

Satz 4.24 (Rabin 1969)

Für jeden NPBA \mathcal{A} gibt es einen NPBA \mathcal{B} mit $L_{\omega}(\mathcal{B}) = L_{\omega}(\mathcal{A})$.

Beweis:

Für den bisher konstruierten NPBA \mathcal{B} gilt:

$$t \in \mathcal{L}_{\omega}(\mathcal{B})$$
 gdw. $\exists s . L_{s,t} \subseteq \mathcal{L}_{\omega}(\mathcal{A}'')$

(Lemma 4.23)

gdw.
$$\exists s . L_{s,t} \subseteq L_{\omega}(\mathcal{A}')$$

(Konstr. \mathcal{A}'')

gdw.
$$\exists s . L_{s,t} \cap L_{\omega}(\mathcal{A}') = \emptyset$$

(Mengenlehre)

gdw.
$$\exists PF$$
-Gewinnbaum s für t

(Lemma 4.22)

gdw.
$$t \in \overline{L_{\omega}(\mathcal{A})}$$

(Folg. 4.21)

Teil 4: unendliche Bäume

Komplementierung

└─Das Resultat

Komplementierung

Für jeden NPBA A gibt es einen NPBA B mit $L_{\omega}(B) = \overline{L_{\omega}(A)}$.

9:10 bis 9:13

Jetzt müssen wir nur noch alle Lemmas zusammentun und erhalten das Hauptresultat.

"Konstr. \mathcal{A}'' ": ist Komplementärautomat zu \mathcal{A}'

TODO: Wenn Zeit ist, Beispiel für die Konstruktion vorführen (siehe Notizen in Sammlung nach T4.15, zweites=einfacheres Bsp.) Sei n = |Q| (Anzahl der Zustände des NBPA \mathcal{A}). Dann hat der NPA \mathcal{A}' dieselben n Zustände. DPA \mathcal{A}'' kann so konstruiert werden, dass $|Q''| \in O(2^{n \log n})$. \rightsquigarrow NBPA \mathcal{B} hat $O(2^{n \log n})$ Zustände. Teil 4: unendliche Bäume

Komplementierung

Dann hat der NPA A' disselben n Zustände.

DPA A'' kann so konstruiert werden, dass |Q''| ∈ O(2^{n log n}).

→ NBPA B hat O(2^{n log n}) Zustände.

Bemerkungen zur Komplexität der Konstruktion

Bemerkungen zur Komplexität der Konstruktion

9:13; auf Literatur verweisen, dann fertig? Oder noch Alternierung beginnen?

Punkt 3 folgt nicht aus den Resultaten dieser Vorlesung.

Naives Vorgehen:

NPA \rightarrow NMA \rightarrow * NBA —Safra \rightarrow * DRA \rightarrow DMA \rightarrow * DPA *Resultate aus Vorlesung, exp. Blowup

Es gibt offenbar eine effizientere direkte Konstruktion NPA ightarrow DPA

Literatur für diesen Teil (Basis, 1)

Teil 4: unendliche Bäume

Literatur für diesen Teil (Basis, 1)



E. Grädel, W. Thomas, T. Wilke (Hrsg.).

Automata, Logics, and Infinite Games.

LNCS 2500, Springer, 2002, S. 43-60.

Kapitel 6-9 über Paritätsspiele und Baumautomaten.

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 4.

http://www.informatik.uni-bremen.de/tdki/lehre/ws09/ automata/automata-notes.pdf

Literatur für diesen Teil (Basis, 2)



Christel Baier, Joost-Pieter Katoen.

Principles of Model Checking.

MIT Press 2008.

Abschnitt 6 "Computation Tree Logic".

SUB, Zentrale: a inf 440 ver/782, a inf 440 ver/782a

Baumautomaten



Edmund M. Clarke, Orna Grumberg, Doron A. Peled.

Model Checking.

MIT Press 1999.

Abschnitt 3 "Temporal Logics",

Abschnitt 4 "Model Checking".

SUB, Zentrale: a inf 440 ver/780(6), a inf 440 ver/780(6)a

Teil 4: unendliche Bäume

Literatur für diesen Teil (Basis, 2)

Literatur für diesen Teil (Basis, 2) Christel Baier, Joost-Pieter Katoen Principles of Model Checking. Edmund M. Clarke, Orna Grumberg, Doron A. Peled Model Checking

SUB. Zentrale: a inf 440 ver/780(6). a inf 440 ver/780(6)a

MIT Press 1999

Literatur für diesen Teil (weiterführend)



Edmund M. Clarke, I. A. Draghicescu

Expressibility Results for Linear-Time and Branching-Time Logics

REX Workshop 1988, S. 428-437.

https://doi.org/10.1007/BFb0013029

Teil 4: unendliche Bäume

Literatur für diesen Teil (weiterführend)

Edmund M. Clarke, I. A. Draghioscu Expressibility Results for Linear-Time and Branching-Time Logics

Literatur für diesen Teil (weiterführend)

