Motivation Grundbegriffe

Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

Automatentheorie und ihre Anwendungen Teil 2: endliche Automaten auf endlichen Bäumen

> Wintersemester 2018/19 Thomas Schneider

> > AG Theorie der künstlichen Intelligenz (TdKI)

http://tinyurl.com/ws1819-autom

Teil 2: endliche Bäume

Automatentheorie und ihre Anwendungen http://tinyurl.com/ws1819-autom

8:30

20.11.18 S.215 (F.85)

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Überblick

Motivation

- Motivation: semistrukturierte Daten
- ② Grundbegriffe
- 3 Charakterisierungen erkennbarer Baumsprachen
- 4 Top-down-Baumautomaten
- 6 Abschlusseigenschaften
- 6 Entscheidungsprobleme
- Anwendung: XML-Schemasprachen

8:30

XML

2 Grundbegriffe

Charakt.

1 Motivation: semistrukturierte Daten

Abschlusseig.

XML

Und nun ...

Motivation

5 Abschlusseigenschaften



Motivation Grun

Grundbegriffe

Charakt.

Top-down-BAs Absch

Abschlusseig.

Entscheid.-probl.

Semistrukturierte Daten sind

- ein Datenmodell zur Beschreibung von Entitäten und Attributen,
 das weniger formale Struktur voraussetzt als z. B. relationale Datenbanken
- ein Vorläufer von XML
- gut geeignet, um
 - Dokumentansichten (z. B. Webseiten) und
 - strukturierte Daten (z.B. Datenbank-Tabellen)

zu repräsentieren und miteinander zu verbinden

 ein Datemodell zur Beschribung von Entstäten und Attrikuten,
das weniger formals Struktur vorzusstetzt
als z. B. relizionien Datembarien
w ein Vorläufer von XML
 get gesigner,
u. a. Obekrenstamielsten (z. B. Weberien) und
anterhalterins Dente (n. B. Destenbark-Tabelien)

zu renräsentieren und miteinander zu verhinder

Samistrukturierte Daten sind

8:31

Motivation Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Merkmale semistrukturierter Daten

Daten werden in Entitäten (Einheiten) zusammengefasst

Markierung von Entitäten durch Tags

Charakt.

- Bildung von Hierarchien
- Gruppieren ähnlicher Entitäten
- Entitäten derselben Gruppe können verschiedene (oder keine) Attribute haben
- Reihenfolge der Attribute kann eine Rolle spielen

Teil 2: endliche Bäume Motivation: semistrukturierte Daten -Merkmale semistrukturierter Daten Merkmale semistrukturierter Daten

Daten werden in Entitäten (Einheiten) zusammengefasst

w Markierung von Entitäten durch Tags Bildung von Hierarchien

a Gruppieren ähnlicher Entitäten

verschiedene (oder keine) Attribute haben · Reihenfolge der Attribute kann eine Rolle spielen

8:32

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl

Charakt.

Merkmale semistrukturierter Daten

Daten werden in Entitäten (Einheiten) zusammengefasst

- Markierung von Entitäten durch Tags
- Bildung von Hierarchien
- Gruppieren ähnlicher Entitäten
- Entitäten derselben Gruppe können verschiedene (oder keine) Attribute haben
- Reihenfolge der Attribute *kann* eine Rolle spielen (Mengen oder Listen z. B. von Telefonnummern?)

```
Person: {Name: {VN: "Thomas", NN: "Schneider"},
                   Telnr: 64432,
Beispiel:
                   Telnr: 43776243,
                   Email: "ts@informatik..."}
```

Teil 2: endliche Bäume Daten werden in Entitäten (Einheiten) zusammengefasst Motivation: semistrukturierte Daten w Markierung von Entitäten durch Tags Bildung von Hierarchien a Gruppieren ähnlicher Entitäten verschiedene (oder keine) Attribute haber Reihenfolge der Attribute kann eine Rolle spielen -Merkmale semistrukturierter Daten (Mengen oder Listen z. B. von Telefonnummern?) Person: {Name: {VN: "Thomas", NN: "Schneider" Telnr: 43776243. Enail: "te@informatik . . . "}

Merkmale semistrukturierter Dater

8:32

Grundbegriffe

Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Datenstruktur: Baum



Datenstruktur: Baum Teil 2: endliche Bäume - Motivation: semistrukturierte Daten └ Datenstruktur: Baum

Grundbegriffe

Top-down-BAs

Abschlusseig.

XML

Entscheid.-probl.

Datenstruktur: Baum

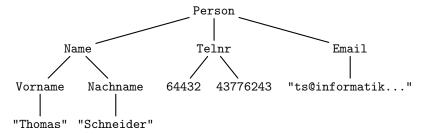
Person: {Name: {VN: "Thomas", NN: "Schneider"},

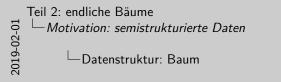
Telnr: 64432, Telnr: 43776243,

Email: "ts@informatik..."}

Repräsentation im Baum ist naheliegend:

Charakt.





Datenstruktur: Baum Person: {Name: {VH: "Thomas", NN: "Schneider"} Telmr: 64432, Telmr: 43776243, Email: "sedinformatik..."}

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

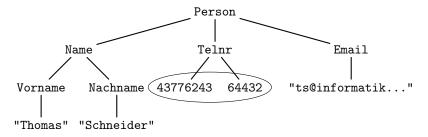
Datenstruktur: Baum

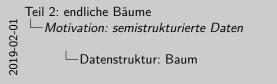
Person: {Name: {VN: "Thomas", NN: "Schneider"},

Telnr: 64432, Telnr: 43776243,

Email: "ts@informatik..."}

Ist das derselbe oder ein anderer Baum?







Motivation Grundbegriffe

Charakt.

Abschlusseig.

Entscheid.-probl.

-Automaten auf endlichen Bäumen

Automaten auf endlichen Bäumen

- ... sind wichtig für semistrukturierte Daten, weil sie ...
 - XML-Schemasprachen und -validierung zugrunde liegen
 - XML-Anfragesprachen auf ihnen aufgebaut sind

8:35

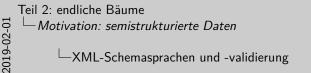
Automatentheorie u. i. A. WiSe 2018/19

Motivation Grundbegriffe

Top-down-BAs

Abschlusseig.

- Schemasprachen zur Beschreibung gültiger Bäume
- Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten



Grundbegriffe

Charakt.

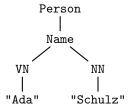
Top-down-BAs

Abschlusseig.

Entscheid.-probl.

- Schemasprachen zur Beschreibung gültiger Bäume
- Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben



Teil 2: endliche Bäume Motivation: semistrukturierte Daten XML-Schemasprachen und -validierung XML-Schemasprachen und -validierung

8:35

Teil 2: endliche Bäume

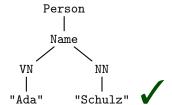
Motivation: semistrukturierte Daten

XML-Schemasprachen und -validierung

XML-Schemasprachen und -validierung

- Schemasprachen zur Beschreibung gültiger Bäume
- Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben



8:35

Grundbegriffe

Top-down-BAs

As Abschlusseig

σ Entsch

Entscheid.-probl.

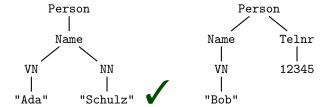
XML

XML-Schemasprachen und -validierung

Charakt.

- Schemasprachen zur Beschreibung gültiger Bäume
- Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben



Teil 2: endliche Bäume

Motivation: semistrukturierte Daten

Schemapschen und -validierung

Schemapschen zur Buchnehung glöbg

Välderung - Erkemen glöbge und en ger

Schemapschen zur Buchnehung glöbg

Välderung - Erkemen glöbge und en ger

Schemapschen und den ger

Schemapschen und validierung

Texture muss eines Vor. und Nar

T

8:35

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

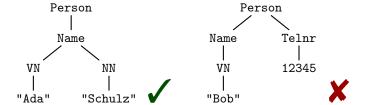
XML

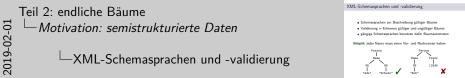
Entscheid.-probl.

XML-Schemasprachen und -validierung

- Schemasprachen zur Beschreibung gültiger Bäume
- Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben





Grundbegriffe Charakt. Top-down-BAs Abschlusseig.

XML-Anfragesprachen

Motivation

Entscheid.-probl.

Teil 2: endliche Bäume - Motivation: semistrukturierte Daten

u beantworten Anfragen mit Daten aus gegebenen Bäumen

XML-Anfragesprachen

XML-Anfragesprachen

• beantworten Anfragen mit Daten aus gegebenen Bäumen

Motivation Grundbegriffe Charakt. Top-down-BAs

XML-Anfragesprachen

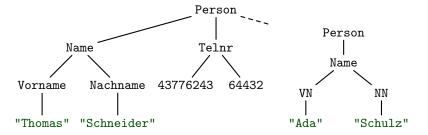
• beantworten Anfragen mit Daten aus gegebenen Bäumen

Abschlusseig.

Entscheid.-probl.

XML

Beispiel: gib alle Namen von Personen zurück





Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Und nun ...

Motivation

Motivation: semistrukturierte Daten

- 2 Grundbegriffe
- 3 Charakterisierungen erkennbarer Baumsprache
- 4 Top-down-Baumautomater
- 5 Abschlusseigenschaften
- 6 Entscheidungsprobleme
- Anwendung: XMI -Schemaspracher

XML

Grundbegriffe

Top-down-BAs

Abschlusseig.

XML

Entscheid.-probl.

Positionen im Baum

• positive natürliche Zahlen: \mathbb{N}_+

Charakt.

• Position: Wort $p \in \mathbb{N}_+^*$

Idee: Wurzel ist ε j-ter Nachfolger von p ist pj

Beispiel:



Teil 2: endliche Bäume -Grundbegriffe Positionen im Baum



8:37

Motivation Grundbegriffe

Alphabet mit Stelligkeit

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

	A1 1 1 . -	,	· -	

- (auf Englisch: ranked alphabet) hier: r-Alphabet Σ
- nichtleere endliche Menge von Symbolen; jedem Symbol ist eine Stelligkeit $\in \mathbb{N}$ zugeordnet
- Σ_m = Menge der Symbole mit Stelligkeit m
- Schreibweise: $\Sigma = \{a_1/r_1, \ldots, a_n/r_n\}$ heißt: Σ enthält die Symbole a_i mit Stelligkeit r_i , $i = 1, \ldots, n$

Teil 2: endliche Bäume -Grundbegriffe

-Alphabet mit Stelligkeit

8:38

2019-02-01

Alphabet mit Stelligkeit

a hier: r-Alphabet Σ (auf Englisch: ranked alphabet u nichtleere endliche Menze von Symbolen:

jedem Symbol ist eine Stelligkeit ∈ N zugeordnet $\mathbf{v} \; \Sigma_m = \mathsf{Menge} \; \mathsf{der} \; \mathsf{Symbole} \; \mathsf{mit} \; \mathsf{Stelligkeit} \; m$

• Schreibweise: $\Sigma = \{a_1/r_1, \dots, a_n/r_n\}$ heißt: Σ enthält die Symbole a_i mit Stelligkeit r_i , $i=1,\ldots,n$

Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl.

Alphabet mit Stelligkeit

- hier: r-Alphabet Σ (auf Englisch: ranked alphabet)
- nichtleere endliche Menge von Symbolen; jedem Symbol ist eine Stelligkeit $\in \mathbb{N}$ zugeordnet
- Σ_m = Menge der Symbole mit Stelligkeit m
- Schreibweise: $\Sigma = \{a_1/r_1, \ldots, a_n/r_n\}$ heißt: Σ enthält die Symbole a_i mit Stelligkeit r_i , $i=1,\ldots,n$

Beispiel: $\Sigma = \{a/2, b/3, c/0, d/0\}$



a "passt" in Position 2 b "passt" in Position ε c, d "passen" in Pos. 1, 21, 22, 3 Teil 2: endliche Bäume -Grundbegriffe

—Alphabet mit Stelligkeit

 Σ enthält die Symbole a_i mit Stelligkeit r_i , $i=1,\ldots,r$ Beispiel: $\Sigma = \{a/2, b/3, c/0, d/0\}$ c, d "passen" in Pos. 1, 21, 22, 3

a hier: r-Alphabet Σ (auf Enelisch: ranked alphabet

jedem Symbol ist eine Stelligkeit ∈ N zugeordnet ν Σ_m = Menge der Symbole mit Stelligkeit m • Schreibweise: $\Sigma = \{a_1/r_1, \dots, a_s/r_s\}$ heißt:

Alphabet mit Stelligkeit

8:38

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Alphabet mit Stelligkeit

- hier: r-Alphabet Σ (auf Englisch: ranked alphabet)
- nichtleere endliche Menge von Symbolen; jedem Symbol ist eine Stelligkeit $\in \mathbb{N}$ zugeordnet
- Σ_m = Menge der Symbole mit Stelligkeit m

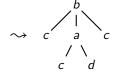
Charakt.

• Schreibweise: $\Sigma = \{a_1/r_1, \ldots, a_n/r_n\}$ heißt: Σ enthält die Symbole a_i mit Stelligkeit r_i , $i=1,\ldots,n$

Beispiel: $\Sigma = \{a/2, b/3, c/0, d/0\}$



a "passt" in Position 2 b "passt" in Position arepsilonc, d "passen" in Pos. 1, 21, 22, 3



Baum über Σ

Teil 2: endliche Bäume -Grundbegriffe

-Alphabet mit Stelligkeit

Beispiel: $\Sigma = \{a/2, b/3, c/0, d/0\}$

jedem Symbol ist eine Stelligkeit ∈ N zugeordne ν Σ_m = Menge der Symbole mit Stelligkeit m • Schreibweise: $\Sigma = \{a_1/r_1, \dots, a_s/r_s\}$ heißt:

Alphabet mit Stelligkeit

Motivation Grundbegriffe

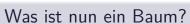
Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML





Teil 2: endliche Bäume -Grundbegriffe



└─Was ist nun ein Baum?

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Was ist ein Baum?

Definition 2.1

Ein endlicher geordneter Baum über dem r-Alphabet Σ ist ein Paar T = (P, t), wobei

- (1) $P \subseteq \mathbb{N}_{+}^{*}$ eine nichtleere endl. präfix-abgeschlossene Menge ist,
- (2) $t: P \to \Sigma$ eine Funktion ist mit den folgenden Eigenschaften.
 - (a) Wenn $t(p) \in \Sigma_0$, dann $\{j \mid pj \in P\} = \emptyset$.
 - (b) Wenn $t(p) \in \Sigma_m$, $m \ge 1$, dann $\{j \mid pj \in P\} = \{1, ..., m\}$.

Teil 2: endliche Bäume -Grundbegriffe

└─Was ist ein Baum?

Was ist ein Baum?

Ein endlicher geordneter Baum über dem r-Alphabet Σ ist ein Paa

(1) P ⊂ N* eine nichtleere endl. präfix-abzeschlossene Menge ist. (a) Wenn $t(p) \in \Sigma_0$, dann $\{j \mid pj \in P\} = \emptyset$. (b) Wenn $t(p) \in \Sigma_m$, $m \ge 1$, dann $\{j \mid pj \in P\} = \{1, ..., m\}$

Top-down-BAs

Abschlusseig.

Was ist ein Baum?

Definition 2.1

Ein endlicher geordneter Baum über dem r-Alphabet Σ ist ein Paar T = (P, t), wobei

- (1) $P \subseteq \mathbb{N}_{+}^{*}$ eine nichtleere endl. präfix-abgeschlossene Menge ist,
- (2) $t: P \to \Sigma$ eine Funktion ist mit den folgenden Eigenschaften.
 - (a) Wenn $t(p) \in \Sigma_0$, dann $\{j \mid pj \in P\} = \emptyset$.
 - (b) Wenn $t(p) \in \Sigma_m$, $m \ge 1$, dann $\{j \mid pj \in P\} = \{1, ..., m\}$.

Erklärungen:

- (1) P: Menge der vorhandenen Positionen Präfix-Abgeschlossenheit: Baum ist wohlgeformt (z. B.: wenn Position 31 existiert, dann auch Position 3 und ε)
- (2) (a) und (b) sagen: Stelligkeit des Zeichens an Position p muss mit der Anzahl der Kinder von p übereinstimmen.

Teil 2: endliche Bäume Grundbegriffe

└─Was ist ein Baum?

(1) P: Menge der vorhandenen Positionen

Präfix-Abgeschlossenheit: Baum ist wohlgeform (z. B.: wenn Position 31 existient, dann auch Position 3 und a (2) (a) und (b) sagen: Stelligkeit des Zeichens an Position p muss mit der Anzahl der Kinder von p übereinstimmen

Ein endlicher geordneter Baum über dem r-Alphabet Σ ist ein Paa

(1) P ⊂ N* eine nichtleere endl. präfix-abzeschlossene Menge ist. (2) t : P → Σ eine Funktion ist mit den folgenden Eigenschafte (a) Wenn $t(p) \in \Sigma_0$, dann $\{j \mid pj \in P\} = \emptyset$. (b) Wenn $t(p) \in \Sigma_m$, $m \ge 1$, dann $\{j \mid pj \in P\} = \{1, ..., m\}$

Grundbegriffe

Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Was ist ein Baum?

Bezeichnungen

- Position p hat Kinder $p1, p2, \ldots$; p ist deren Elternteil
- jedes Präfix von p ist ein Vorgänger von p; p ist Nachfolger eines jeden Präfixes von p
- Blatt: Knoten ohne Kinder
- Höhe von p in T: Länge des längsten Pfades von p zu einem Blatt
- Höhe von T: Höhe von ε in T

Was ist ein Baum? Teil 2: endliche Bäume -Grundbegriffe Position p hat Kinder p1. p2.... p ist deren Elternteil p ist Nachfolger eines ieden Präfixes von p w Blatt: Knoten ohne Kinder └─Was ist ein Baum? Länge des längsten Pfades von p zu einem Blatt a Höhe von 7: Höhe von s in 7

8:45

Beispiel

$$\Sigma = \{a/2, b/3, c/0, d/0\}$$

$$P = \{\varepsilon, 1, 2, 3, 21, 22\}$$

$$t(\varepsilon) = b$$
, $t(1) = c$, $t(2) = a$, $t(3) = c$, $t(21) = c$, $t(22) = d$

Positionen P

Baum T = (P, t)

andere Schreibweise:

 $(\approx in-order-Tiefensuche)$

- 21 • Höhe: 2
- Blätter: 1, 21, 22, 3
- 21 ist Kind von 2 und hat Vorgänger 2, ε



2019-02-01

XML



8:46

Teil 2: endliche Bäume

└─Beispiel

Grundbegriffe

Abschlusseig.

Top-down-BAs

Bottom-up-Baumautomaten

Definition 2.2

Ein nichtdet. Bottom-up-Automat auf endl. geord. Bäumen (NEBA) ist ein Quadrupel $\mathcal{A} = (Q, \Sigma, \Delta, F)$, wobei

- Q eine endliche nichtleere Zustandsmenge ist.
- Σ ein r-Alphabet ist,
- Δ eine Menge von Überführungsregeln der Form

$$a(q_1,\ldots,q_m)\to q$$

ist mit $m \ge 0$, $a \in \Sigma_m$, $q, q_1, \ldots, q_m \in Q$, und

• $F \subset Q$ die Menge der akzeptierenden Zustände ist.

Teil 2: endliche Bäume -Grundbegriffe

-Bottom-up-Baumautomaten

ist ein Quadrupel $A = (Q, \Sigma, \Delta, F)$, wobei O eine endliche nichtleere Zustandsmense ist. ν Δ eine Menge von Überführungsregeln der Form ist mit $m \ge 0$, $a \in \Sigma_m$, $a, a_1, \dots, a_m \in Q$, and

8:50

2019-02-01

Keine Anfangszustände – wir sehen gleich, warum.

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Bottom-up-Baumautomaten: Intuitionen

Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1,\ldots,q_m) \to q$:

- Wenn A in Position p Zeichen a liest
- und in p's Kindern Zustände q_1, \ldots, q_m eingenommen hat, dann darf A in p Zustand q einnehmen. T 2.1

Teil 2: endliche Bäume Grundbegriffe

-Bottom-up-Baumautomaten: Intuitionen

8:52 bis 8:56

Fragen: Was sind dann die Anfangszustände?

Top-down-BAs

Entscheid.-probl.

Abschlusseig.

Charakt. Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1, \ldots, q_m) \rightarrow q$:

- Wenn A in Position p Zeichen a liest
- und in p's Kindern Zustände q_1, \ldots, q_m eingenommen hat, dann darf A in p Zustand q einnehmen. T 2.1

→ Andere Betrachtungsweise:

- A markiert Eingabebaum T bottom-up mit Zuständen
- \mathcal{A} akzeptiert T, wenn \mathcal{A} in der Wurzel einen akzeptierenden Zustand einnimmt

8:52 bis 8:56

2019-02-01

Fragen: Was sind dann die Anfangszustände?

-Bottom-up-Baumautomaten: Intuitionen

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1, \ldots, q_m) \rightarrow q$:

- Wenn A in Position p Zeichen a liest
- und in p's Kindern Zustände q_1, \ldots, q_m eingenommen hat, dann darf A in p Zustand q einnehmen. T 2.1

→ Andere Betrachtungsweise:

- A markiert Eingabebaum T bottom-up mit Zuständen
- \mathcal{A} akzeptiert T, wenn \mathcal{A} in der Wurzel einen akzeptierenden Zustand einnimmt

Was sind dann die Anfangszustände?

Teil 2: endliche Bäume Grundbegriffe

-Bottom-up-Baumautomaten: Intuitionen

Bottom-up-Baumautomaten: Intuitionen Wenn A in Position p Zeichen a liest a A markiert Eingabebaum T bottom-up mit Zuständen u A akzeptiert T, wenn A in der Wurzel einen akzeptierenden

Was sind dann die Anfangszustände?

8:52 bis 8:56

2019-02-01

Fragen: Was sind dann die Anfangszustände?

Top-down-BAs

Abschlusseig.

Entscheid.-probl

Charakt. Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1, \ldots, q_m) \rightarrow q$:

- Wenn A in Position p Zeichen a liest
- und in p's Kindern Zustände q_1, \ldots, q_m eingenommen hat, dann darf A in p Zustand q einnehmen. T 2.1

→ Andere Betrachtungsweise:

- A markiert Eingabebaum T bottom-up mit Zuständen
- \mathcal{A} akzeptiert T, wenn \mathcal{A} in der Wurzel einen akzeptierenden Zustand einnimmt

Was sind dann die Anfangszustände?

- Ü-Regeln $a() \rightarrow q$ deklarieren "zeichenspezifische" AZ: \mathcal{A} darf in mit a markierten Blättern in q starten
- Kurzschreibweise: $a \rightarrow q$

Teil 2: endliche Bäume Grundbegriffe

-Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1, \dots, q_m) \rightarrow q$ Wenn A in Position p Zeichen a liest und in p's Kindern Zustände gr..... gw eingenommen hat. a A markiert Eingabebaum T bottom-up mit Zuständen u A akzeptiert T, wenn A in der Wurzel einen akzeptierenden Was sind dann die Anfangszustände? ■ 0-Regeln a() → σ deklarieren "zeichenspezifische" AZ: A darf in mit a markierten Blättern in q starten

Bottom-up-Baumautomaten: Intuitionen

8:52 bis 8:56

2019-02-01

Fragen: Was sind dann die Anfangszustände?

Motivation Grundbegriffe Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

Berechnungen (analog zu NEAs)

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum Ein Run von A auf T ist eine Fkt. r : P → Q mit:

Berechnungen

(analog zu NEAs)

Definition 2.3

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.

Teil 2: endliche Bäume Grundbegriffe (analog zu NEAs) Berechnungen

 $8:56 \rightarrow 5 \text{ min Puffer} \rightarrow 9:01$

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

2019-02-01

Berechnungen (analog zu NEAs)

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum Ein Run von A auf T ist eine Fkt. r : P → Q mit: • Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$

Berechnungen (analog zu NEAs)

Definition 2.3

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.
 - Wenn $t(p) = b \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$

Teil 2: endliche Bäume -Grundbegriffe (analog zu NEAs) Berechnungen

 $8:56 \rightarrow 5 \text{ min Puffer} \rightarrow 9:01$

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

2019-02-01

Berechnungen (analog zu NEAs)

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum Ein Run von A auf T ist eine Fkt. r : P → Q mit: • Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$ dann $b(q_1, \dots, q_m) \rightarrow q \in \Delta$.

Berechnungen (analog zu NEAs)

Charakt.

Definition 2.3

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.
 - Wenn $t(p) = b \in \Sigma_m$ $(m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann $b(q_1, \ldots, q_m) \to q \in \Delta$.

-Grundbegriffe (analog zu NEAs) Berechnungen

 $8:56 \rightarrow 5 \text{ min Puffer} \rightarrow 9:01$

Teil 2: endliche Bäume

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML Berechnungen (analog zu NEAs)

Berechnungen

(analog zu NEAs)

Charakt.

Definition 2.3

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.
 - Wenn $t(p) = b \in \Sigma_m$ $(m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann $b(q_1,\ldots,q_m)\to q\in\Delta$.

Also gilt:

• Blatt mit a kann q nur zugewiesen kriegen, wenn $a \to q \in \Delta$.

Teil 2: endliche Bäume -Grundbegriffe

2019-02-01

(analog zu NEAs)

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baun Fin Run von 4 auf T ist eine Flet x : P → Q mit-• Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$

Blatt mit a kann σ nur zugewiesen kriegen, wenn a → σ ∈ Δ

 $8:56 \rightarrow 5 \text{ min Puffer} \rightarrow 9:01$

Berechnungen

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

2019-02-01

Berechnungen

(analog zu NEAs)

Definition 2.3

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.
 - Wenn $t(p) = b \in \Sigma_m$ $(m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann $b(q_1,\ldots,q_m)\to q\in\Delta$.

Also gilt:

- Blatt mit a kann q nur zugewiesen kriegen, wenn $a \to q \in \Delta$.
- Nicht-Blatt mit b, dessen Kinder q_1, \ldots, q_m haben, kann q nur zugew. kriegen, wenn $b(q_1, \ldots, q_m) \to q \in \Delta$.

Teil 2: endliche Bäume -Grundbegriffe

> └─Berechnungen (analog zu NEAs)

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum Ein Run von A auf T ist eine Fkt. r : P → Q mit: • Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$ dann $b(q_1, \dots, q_m) \rightarrow q \in \Delta$.

Berechnungen (analog zu NEAs)

 \bullet Blatt mit a kann q nur zugewiesen kriegen, wenn $a \to q \in \Delta$

■ Nicht-Blatt mit b. dessen Kinder gr..... g_m haben. kann q nur zugew. kriegen, wenn $b(q_1, ..., q_m) \rightarrow q \in \Delta$.

 $8:56 \rightarrow 5 \text{ min Puffer} \rightarrow 9:01$

• Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$

Berechnungen, Akzeptanz und erkannte Sprache

Charakt.

Definition 2.3

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

- Ein Run von \mathcal{A} auf T ist eine Funktion $r: P \to Q$ mit:
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $a \to q \in \Delta$.
 - Wenn $t(p) = b \in \Sigma_m$ $(m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann $b(a_1,\ldots,a_m)\to a\in\Delta$.
- Ein Run r von \mathcal{A} auf T ist erfolgreich, wenn $r(\varepsilon) \in F$.
- A akzeptiert T, wenn es einen erfolgreichen Run von A auf T gibt.
- Die von \mathcal{A} erkannte Sprache ist $L(A) = \{ T \text{ "uber } \Sigma \mid A \text{ akzeptiert } T \}.$

Berechnungen, Akzeptanz und erkannte Sprache

 Ein Run r von A auf T ist erfolgreich, wenn r(e) ∈ F A akzeptiort T, wenn es einen erfolgreichen Run von A auf u Die von A erkannte Sprache ist $L(A) = \{T \text{ über } \Sigma \mid A \text{ akzentiert } T\}.$

Motivation Grundbegriffe

Beispiel 1

Charakt. Top-down-BAs

mit $\Delta = \{ b \rightarrow q_0, c \rightarrow q_0,$

 $a(q_0, q_0) \rightarrow q_1,$

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$

 $a(q_1, q_1) \to q_2, \quad a(q_1, q_1) \to q_3 \}.$

Abschlusseig.

Entscheid.-probl.

Teil 2: endliche Bäume -Grundbegriffe

Beispiel 1

Beispiel 1 mit $\Delta = \{b \rightarrow q_0, c \rightarrow q_0,$ $a(q_1, q_1) \rightarrow q_2, \quad a(q_1, q_1) \rightarrow q_1$ }.

9:05

Fragen: Welche Runs gibt es? Was ist die erkannte Sprache?

XML

• Sei
$$\Sigma = \{a/2, b/0, c/0\}$$
 und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$
mit $\Delta = \{b \to q_0, c \to q_0, a(q_0, q_0) \to q_1, a(q_1, q_1) \to q_2, a(q_1, q_1) \to q_3\}.$



Teil 2: endliche Bäume Grundbegriffe

Beispiel 1



9:05

2019-02-01

Sei
$$\Sigma = \{a/2, b\}$$

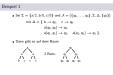
• Sei
$$\Sigma = \{a/2, b/0, c/0\}$$
 und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$
mit $\Delta = \{b \to q_0, c \to q_0, a(q_0, q_0) \to q_1, a(q_1, q_1) \to q_2, a(q_1, q_1) \to q_3\}.$





Teil 2: endliche Bäume -Grundbegriffe

Beispiel 1



9:05

• Sei
$$\Sigma = \{a/2, b/0, c/0\}$$
 und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$
mit $\Delta = \{b \to q_0, c \to q_0, a(q_0, q_0) \to q_1, a(q_1, q_1) \to q_2, a(q_1, q_1) \to q_3\}.$





Teil 2: endliche Bäume Grundbegriffe

Beispiel 1

mit $\Delta = \{b \rightarrow a_0, c \rightarrow a_0,$ $s(q_1, q_1) \rightarrow q_2, \quad s(q_1, q_1) \rightarrow q_1$ }.

Beispiel 1

9:05

2019-02-01

Beispiel 1

• Sei
$$\Sigma = \{a/2, b/0, c/0\}$$
 und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$
mit $\Delta = \{b \to q_0, c \to q_0, a(q_0, q_0) \to q_1, a(q_1, q_1) \to q_2, a(q_1, q_1) \to q_3\}.$

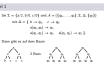
• Dann gibt es auf dem Baum







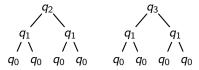
Teil 2: endliche Bäume -Grundbegriffe Beispiel 1



9:05

• Sei
$$\Sigma = \{a/2, b/0, c/0\}$$
 und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$
mit $\Delta = \{b \to q_0, c \to q_0, a(q_0, q_0) \to q_1, a(q_1, q_1) \to q_2, a(q_1, q_1) \to q_3\}.$

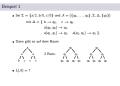




•
$$L(A) = ?$$

Teil 2: endliche Bäume -Grundbegriffe

Beispiel 1

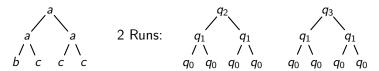


9:05

Top-down-BAs

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$ mit $\Delta = \{ b \rightarrow q_0, c \rightarrow q_0,$ $a(q_0, q_0) \rightarrow q_1,$ $a(q_1, q_1) \to q_2, \quad a(q_1, q_1) \to q_3 \}.$

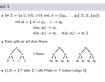
• Dann gibt es auf dem Baum



• $L(A) = \{ T \text{ "uber } \Sigma \mid \text{ alle Pfade in } T \text{ haben L"ange 2} \}$

Teil 2: endliche Bäume -Grundbegriffe

Beispiel 1



9:05

Beispiel 1

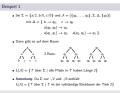
mit $\Delta = \{ b \rightarrow q_0, c \rightarrow q_0, c \rightarrow q_0, a \rightarrow q_0, a \rightarrow q_0, c \rightarrow q_0, a \rightarrow q_0, c \rightarrow q_0,$

Dann gibt es auf dem Baum

 $a(q_1, q_1) \to q_2, \quad a(q_1, q_1) \to q_3 \}.$

Teil 2: endliche Bäume -Grundbegriffe

Beispiel 1



9:05

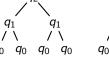
Fragen: Welche Runs gibt es? Was ist die erkannte Sprache?

Automatentheorie u. i. A. WiSe 2018/19



 $a(q_0,q_0) \rightarrow q_1,$

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und $\mathcal{A} = (\{q_0, \dots, q_3\}, \Sigma, \Delta, \{q_2\})$





- $L(A) = \{ T \text{ "uber } \Sigma \mid \text{ alle Pfade in } T \text{ haben L"ange 2} \}$
- Anmerkung: Da Σ nur ./2 und ./0 enthält: $L(A) = \{ T \text{ "uber } \Sigma \mid T \text{ ist der vollst" and ige Bin" arbaum der Tiefe 2} \}$

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Beispiel 2

Motivation

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NEBA erkennt $\{T \text{ "uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister}\}$?

Teil 2: endliche Bäume
Grundbegriffe
Beispiel 2

Sei $\Sigma = \{a/2,\ b/1,\ c/0,\ d/0\}$. Welcher NEBA erkennt $\{T \ \text{liber} \ \Sigma \mid \ \text{jedes} \ c\text{-Blatt hat ein rechtes} \ d\text{-Geschwister}\}$?

Beispiel 2

9:08 bis 9:15

Gemeinsam machen, Zeit nehmen!

XML

Motivation

Grundbegriffe Charakt.

 $\mathcal{A} = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\})$ mit

Top-down-BAs

Abschlusseig.

XML

Entscheid.-probl.

Beispiel 2

Sei
$$\Sigma = \{a/2, b/1, c/0, d/0\}$$
. Welcher NEBA erkennt $\{T \text{ "uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister}\}$?

$$\Delta = \{ c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f, \\ a(q_c, q_d) \rightarrow q_f, \\ a(q_f, q_f) \rightarrow q_f, \\ b(q_f) \rightarrow q_f \}$$

Teil 2: endliche Bäume Grundbegriffe Beispiel 2

```
Sei \Sigma = \{a/2, b/1, c/0, d/0\}. Welcher NEBA erkennt
{ T über Σ | jedes c-Blatt hat ein rechtes d-Geschwister}?
A = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\}) mit
            \Delta = \{c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f,
                      a(q_c, q_d) \rightarrow q_f
                       a(q_f, q_f) \rightarrow q_f
                          b(q_f) \rightarrow q_f }
```

Beispiel 2

9:08 bis 9:15

Gemeinsam machen, Zeit nehmen!

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NEBA erkennt { T über Σ | jedes c-Blatt hat ein rechtes d-Geschwister}?

$$\mathcal{A} = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\}) \text{ mit}$$

$$\Delta = \{ c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f,$$

$$a(q_c, q_d) \rightarrow q_f,$$

$$a(q_f, q_f) \rightarrow q_f,$$

$$b(q_f) \rightarrow q_f \}$$

Übergang $a(q_d, q_d) \rightarrow q_f$ ist überflüssig: $d \rightarrow q_f$ und $a(q_f, q_f) \rightarrow q_f$.

Beispiel 2 Teil 2: endliche Bäume Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NEBA erkennt -Grundbegriffe { T über Σ | jedes c-Blatt hat ein rechtes d-Geschwister}? $A = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\})$ mit $\Delta = \{c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f,$ $a(q_c, q_d) \rightarrow q_f$ $a(q_f, q_f) \rightarrow q_f$ Beispiel 2 $b(q_f) \rightarrow q_f$ } Obergang $a(\alpha_i, \alpha_i) \rightarrow \alpha_i$ ist überflüssig: $d \rightarrow \alpha_i$ and $a(\alpha_i, \alpha_i) \rightarrow \alpha_i$.

9:08 bis 9:15

Gemeinsam machen, Zeit nehmen!

Motivation

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

2019-02-01

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NEBA erkennt { T über Σ | jedes c-Blatt hat ein rechtes d-Geschwister}?

$$\mathcal{A} = (\{\textit{q}_\textit{c}, \textit{q}_\textit{d}, \textit{q}_\textit{f}\}, \Sigma, \Delta, \{\textit{q}_\textit{f}\})$$
 mit

$$\Delta = \{ c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f, \\ a(q_c, q_d) \rightarrow q_f, \\ a(q_f, q_f) \rightarrow q_f, \\ b(q_f) \rightarrow q_f \}$$

Übergang $a(q_d, q_d) \rightarrow q_f$ ist überflüssig: $d \rightarrow q_f$ und $a(q_f, q_f) \rightarrow q_f$.

T 2.2 Beispielbaum und -run: siehe Tafel

Teil 2: endliche Bäume -Grundbegriffe Beispiel 2

Beispiel 2 Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NEBA erkennt { T über Σ | jedes c-Blatt hat ein rechtes d-Geschwister}? $A = (\{a_r, a_t, a_t\}, \Sigma, \Delta, \{a_t\})$ mit $\Delta = \{c \rightarrow q_c, d \rightarrow q_d, d \rightarrow q_f,$ $a(q_c, q_d) \rightarrow q_f$ $a(q_f, q_f) \rightarrow q_f$ $b(q_f) \rightarrow q_f$ } Obergang $a(a_i, a_i) \rightarrow a_i$ ist überflüssig: $d \rightarrow a_i$ und $a(a_i, a_i) \rightarrow a_i$. Beispielbaum und -run: siehe Tafel

9:08 bis 9:15

Gemeinsam machen, Zeit nehmen!

Motivation Grundbegriffe

Charakt.

Erkennbare Baumsprache

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Teil 2: endliche Bäume Grundbegriffe

Erkennbare Baumsprache

Eine Menge L von (endlichen geordneten) Bäumen über Σ wenn as einen NEBA A gibt mit L(A) = L.

Definition 2.4

Eine Menge L von (endlichen geordneten) Bäumen über Σ ist eine erkennbare Baumsprache,

wenn es einen NEBA \mathcal{A} gibt mit $L(\mathcal{A}) = L$.

9:15, 5 min Pause bis 9:20

Motivation Grundbegriffe Determinismus

Definition 2.5

Charakt.

Abschlusseig.

Entscheid.-probl.

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA.

höchstens eine Regel $a(q_1, \ldots, q_m) \rightarrow q$

dann ist \mathcal{A} ein deterministischer endlicher Baumautomat (DEBA).

Teil 2: endliche Bäume

Grundbegriffe

-Determinismus

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA Enthält Δ für jedes jedes $a \in \Sigma_m$ und alle $(q_1, \dots, q_m) \in Q^m$ öchstens eine Regel $a(q_1, ..., q_m) \rightarrow q$ dann ist A ein deterministischer endlicher Baumautomat (DEBA

hier "höchstens eine" statt "genau eine": vermeidet Papierkorbzusta

Determinismus

9:20

Determinismus, "höchstens eine Regel ... ":

Papierkorbzustand q^- ist hier umständlicher als bei NEAs:

- mehr Kombis $a(q_1, \ldots, q_m) \to q^-$
- man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) \rightarrow$ q^{-}

Beispiele: **beide** Aut. sind NEBAs.

Frage stellen!

¹hier "höchstens eine" statt "genau eine": vermeidet Papierkorbzustand

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl

Determinismus

Definition 2.5

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA.

Enthält Δ für jedes jedes $a \in \Sigma_m$ und alle $(q_1, \ldots, q_m) \in Q^m$ höchstens eine¹ Regel $a(q_1, \ldots, q_m) \rightarrow q$ dann ist \mathcal{A} ein deterministischer endlicher Baumautomat (DEBA).

- \rightarrow Nachfolgezustand für jedes (m+1)-Tupel $a(q_1,\ldots,q_m)$ ist eindeutig bestimmt (wenn er existiert)
- Jeder DEBA ist ein NEBA, aber nicht umgekehrt (z. B. die vergangenen 2 Beispiele).

¹hier "höchstens eine" statt "genau eine": vermeidet Papierkorbzustand

Teil 2: endliche Bäume Grundbegriffe

-Determinismus

Determinismus Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA Enthält Δ für jedes jedes $a \in \Sigma_m$ und alle $(q_1, \dots, q_m) \in Q^m$ nöchstens eine Regel $a(q_1, ..., q_m) \rightarrow q$ dann ist A ein deterministischer endlicher Baumautomat (DEBA) ist eindeutig bestimmt (wenn er existiert) aber nicht umgekehrt (z.B. die vergangenen 2 Beispiele

hier "höchstens eine" statt "genau eine": vermeidet Papierkorbzusta

9:20

2019-02-01

Determinismus, "höchstens eine Regel ... ": Papierkorbzustand q^- ist hier umständlicher als bei NEAs:

- mehr Kombis $a(q_1, \ldots, q_m) \to q^-$
- man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) \rightarrow$ q^{-}

Beispiele: **beide** Aut. sind NEBAs.

Frage stellen!

-Determinismus

Enthält Δ für jedes jedes $a \in \Sigma_m$ und alle $(q_1,...$ nöchstens eine Regel $a(q_1, \dots, q_m) \rightarrow q$

ind DEBAs und NEBAs gleichmächtig?

Definition 2.5

Determinismus

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA.

Enthält Δ für jedes jedes $a \in \Sigma_m$ und alle $(q_1, \ldots, q_m) \in Q^m$ höchstens eine Regel $a(q_1, \ldots, q_m) \rightarrow q$ dann ist \mathcal{A} ein deterministischer endlicher Baumautomat (DEBA).

- \rightarrow Nachfolgezustand für jedes (m+1)-Tupel $a(q_1,\ldots,q_m)$ ist eindeutig bestimmt (wenn er existiert)
 - Jeder DEBA ist ein NEBA. aber nicht umgekehrt (z. B. die vergangenen 2 Beispiele).

Frage

Sind DEBAs und NEBAs gleichmächtig?

9:20

Determinismus, "höchstens eine Regel ... ": Papierkorbzustand q^- ist hier umständlicher als bei NEAs:

- mehr Kombis $a(q_1, \ldots, q_m) \to q^-$
- man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) \rightarrow$ q^{-}

Beispiele: **beide** Aut. sind NEBAs.

Frage stellen!

¹hier "höchstens eine" statt "genau eine": vermeidet Papierkorbzustand

Motivation Grundbegriffe

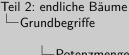
Charakt.

Potenzmengenkonstruktion

Antwort: Ja!

Satz 2.6

Für jeden NEBA \mathcal{A} gibt es einen DEBA \mathcal{A}^d mit $L(\mathcal{A}^d) = L(\mathcal{A})$.



☐ Potenzmengenkonstruktion

9:23 bis 9:50? \rightarrow 10 min Reserve

"ist DEBA": sogar genau 1 Folgezustand statt "höchstens 1"

Grundbegriffe

Potenzmengenkonstruktion

Charakt.

Abschlusseig.

Entscheid.-probl

Potenzmengenkonstruktion

Für ieden NEBA A eibt es einen DEBA A^d mit $L(A^d) = L(A)$

Sei $A = (Q, \Sigma, \Delta, F)$. Konstruieren $A^d = (Q^d, \Sigma, \Delta^d, F^d)$. $\mathbf{u} Q^d = 2^Q$ (Potenzmenze der Zustandsmenze)

 \bullet $a(S_1, \dots, S_m) \rightarrow S \in \Delta^d$ gdw. $S = \{q \mid \exists q_1 \in S_1, \dots, \exists q_m \in S_m : a(q_1, \dots, q_m) \rightarrow q \in \Delta\}$

└─Potenzmengenkonstruktion

Teil 2: endliche Bäume

-Grundbegriffe

9:23 bis 9:50? \rightarrow 10 min Reserve

"ist DEBA": sogar genau 1 Folgezustand statt "höchstens 1"

Antwort: Ja!

Satz 2.6

Für jeden NEBA \mathcal{A} gibt es einen DEBA \mathcal{A}^d mit $L(\mathcal{A}^d) = L(\mathcal{A})$.

Top-down-BAs

Beweis: (analog zur Potenzmengenkonstr. für NEAs) Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$. Konstruieren $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, F^d)$:

- $Q^d = 2^Q$ (Potenzmenge der Zustandsmenge)
- $F^d = \{S \subset Q \mid S \cap F \neq \emptyset\}$
- $a(S_1, \ldots, S_m) \to S \in \Delta^d$ gdw. $S = \{q \mid \exists q_1 \in S_1, \dots, \exists q_m \in S_m : a(q_1, \dots, q_m) \rightarrow q \in \Delta\}$

Antwort: Ja!

Satz 2.6

Für jeden NEBA \mathcal{A} gibt es einen DEBA \mathcal{A}^d mit $L(\mathcal{A}^d) = L(\mathcal{A})$.

Beweis: (analog zur Potenzmengenkonstr. für NEAs)

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$. Konstruieren $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, F^d)$:

- $Q^d = 2^Q$ (Potenzmenge der Zustandsmenge)
- $F^d = \{S \subset Q \mid S \cap F \neq \emptyset\}$
- $a(S_1, \ldots, S_m) \to S \in \Delta^d$ gdw. $S = \{q \mid \exists q_1 \in S_1, \dots, \exists q_m \in S_m : a(q_1, \dots, q_m) \rightarrow q \in \Delta\}$

 \mathcal{A}^d ist DEBA (klar) mit $L(\mathcal{A}^d) = L(\mathcal{A})$. T 2.3

Auch für NEBAs kann die Potenzmengenkonstruktion im schlimmsten Fall zu exponentiell vielen Zuständen führen. └─Potenzmengenkonstruktion

Für ieden NEBA A eibt es einen DEBA A^d mit $L(A^d) = L(A)$ Sei $A = (Q, \Sigma, \Delta, F)$. Konstruieren $A^d = (Q^d, \Sigma, \Delta^d, F^d)$. ■ Q^d = 2^Q (Potenzmenze der Zustandsmenze) im schlimmsten Fall zu exponentiell vielen Zuständen führen

9:23 bis 9:50? \rightarrow 10 min Reserve

"ist DEBA": sogar genau 1 Folgezustand statt "höchstens 1"

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Teil 2: endliche Bäume 2019-02-01

2 Grundbegriffe

Charakterisierungen erkennbarer Baumsprachen

6 Abschlusseigenschaften

Charakterisierungen erkennbarer Baumsprachen

Und nun ...

28

Motivation

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Nichterkennbare Baumsprachen: Intuitionen

Beispiel:

• r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$

Charakt.

• Baumautomat $\mathcal{A} = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit

$$\Delta = \{ c \to q_0, b(q_0) \to q_1, a(q_0, q_0) \to q_1, b(q_1) \to q_0, a(q_1, q_1) \to q_0 \}.$$

$$\rightarrow L(A) =$$

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

Automatentheorie u. i. A. WiSe 2018/19

Grundbegriffe

Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Beispiel:

- r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$
- Baumautomat $\mathcal{A} = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit

$$\Delta = \{ c \to q_0, b(q_0) \to q_1, a(q_0, q_0) \to q_1, b(q_1) \to q_0, a(q_1, q_1) \to q_0 \}.$$

$$\rightarrow$$
 $L(A) = \{ T \mid \text{alle Wurzel-Blatt-Pfade in } T \text{ haben gerade Länge} \}.$

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

Motivation

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

Beispiel:

• r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$

Charakt.

• Baumautomat $\mathcal{A} = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit

$$\Delta = \{ c \to q_0, b(q_0) \to q_1, a(q_0, q_0) \to q_1, b(q_1) \to q_0, a(q_1, q_1) \to q_0 \}.$$

$$\sim L(A) = \{ T \mid \text{alle Wurzel-Blatt-Pfade in } T \text{ haben gerade Länge} \}.$$

$$\neq \{ T \mid T \text{ hat gerade H\"ohe} \}$$
T 2.4

Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

Beispiel:

• r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$

Charakt.

• Baumautomat $\mathcal{A} = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit

$$\Delta = \{ c \to q_0, b(q_0) \to q_1, a(q_0, q_0) \to q_1, b(q_1) \to q_0, a(q_1, q_1) \to q_0 \}.$$

$$\sim L(A) = \{ T \mid \text{alle Wurzel-Blatt-Pfade in } T \text{ haben gerade Länge} \}.$$

$$\neq \{ T \mid T \text{ hat gerade H\"ohe} \}$$

$$T 2.4$$

Frage: Sind die folgenden Baumsprachen (über Σ) erkennbar?

$$L_1 = \{ T \mid T \text{ hat gerade H\"ohe} \}$$

 $L_2 = \{ T \mid T \text{ ist vollst\"andiger Bin\"arbaum} \}$ T 2.4 Forts.

Nichterkennbare Baumsprachen: Intuitionen

Beispiel:

- r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$
- Baumautomat $\mathcal{A} = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit

$$\Delta = \{ c \to q_0, b(q_0) \to q_1, a(q_0, q_0) \to q_1, b(q_1) \to q_0, a(q_1, q_1) \to q_0 \}.$$

$$\sim$$
 $L(A) = \{ T \mid \text{alle Wurzel-Blatt-Pfade in } T \text{ haben gerade Länge} \}.$
 $\neq \{ T \mid T \text{ hat gerade H\"ohe} \}$ T 2.4

Frage: Sind die folgenden Baumsprachen (über Σ) erkennbar?

$$L_1 = \{ T \mid T \text{ hat gerade H\"ohe} \}$$

 $L_2 = \{ T \mid T \text{ ist vollst\"andiger Bin\"arbaum} \}$ T 2.4 Forts.

T 2.4 Forts. Antwort: Nein.

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen



Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Pumping-Lemma: Hilfsbegriffe

Einsetzen von Bäumen ineinander:

- Variable: zusätzliches nullstelliges Symbol $x \notin \Sigma_0$
- (unärer) Kontext: Baum über $\Sigma \cup \{x\}$, in dem ein Blatt mit x markiert ist **T 2.5**
- trivialer Kontext C_0 : Kontext der Höhe 0 (\Rightarrow nur Wurzel)



Pumping-Lemma: Hilfsbegriffe

Wir müssen Bäume ineinander "stecken" können. Dafür brauchen wir eine Markierung für die "Andockstelle" (Variable) und den Begriff des Kontexts (Baum mit Andockstelle).

Top-down-BAs

Abschlusseig.

Entscheid.-probl

Pumping-Lemma: Hilfsbegriffe

Variable: zusätzliches nullstelliges Symbol x ∉ Σ

- Electron una Diumon/Vontouton in Vontouto C[T] = der Baum/Kontext, den man aus C erhält,

Baum über $\Sigma \cup \{x\}$, in dem ein Blatt mit x markiert ist T2.5 trivialer Kontext Cs: Kontext der Höhe 0 (⇒ nur Wurze)

 $C^{*+1} = C^*[C]$

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Pumping-Lemma: Hilfsbegriffe

Einsetzen von Bäumen ineinander:

Pumping-Lemma: Hilfsbegriffe

Charakt.

- Variable: zusätzliches nullstelliges Symbol $x \notin \Sigma_0$
- (unärer) Kontext: Baum über $\Sigma \cup \{x\}$, in dem ein Blatt mit x markiert ist **T 2.5**
- trivialer Kontext C_0 : Kontext der Höhe 0 (\Rightarrow nur Wurzel)
- Einsetzen von Bäumen/Kontexten in Kontexte:
 - C[T] = der Baum/Kontext, den man aus C erhält, indem man die Position von x mit Baum/Kontext T ersetzt T 2.5 Forts.
 - Cⁿ induktiv definiert:

$$C^0 = C_0$$
$$C^{n+1} = C^n[C]$$

Wir müssen Bäume ineinander "stecken" können. Dafür brauchen wir eine Markierung für die "Andockstelle" (Variable) und den Begriff des Kontexts (Baum mit Andockstelle).

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Pumping-Lemma

Satz 2.7 (Pumping-Lemma)

Sei L eine NEBA-erkennbare Baumsprache über dem r-Alphabet Σ .

Dann gibt es eine Konstante $k \in \mathbb{N}$,

so dass für alle Bäume $T \in L$ mit Höhe $(T) \geqslant k$ gilt:

Es gibt Kontexte C, D mit $D \neq C_0$ und Baum V mit T = C[D[V]], so dass $C[D^i[V]] \in L$ für alle $i \in \mathbb{N}$.

Teil 2: endliche Bäume
Charakterisierungen erkennbarer Baumsprachen
Pumping-Lemma

Pumping-Lemma

Sei L eine NEBA-erkennbare Baumsprache über dem r-Alphabet : Dann gibt es eine Konstante $k \in \mathbb{N}$, so dass für alle Bäums $T \in L$ mit Höhe $(T) \geqslant k$ gilt: Es gibt Kontexte C, D mit $D \not= C_0$ und Baum V mit T = C[D[V]

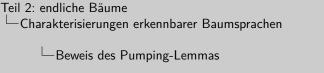
so dass $C[D^i[V]] \in L$ für alle $i \in \mathbb{N}$.

8:30

ation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML Toil 2: and light Päymes

Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA mit $L(\mathcal{A}) = L$.



Beweis des Pumping-Lemmas

8:32

Grundbegriffe

Beweis des Pumping-Lemmas

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA mit $L(\mathcal{A}) = L$.

Wir wählen k = |Q|.

Grundbegriffe

Top-down-BAs

Abschlusseig.

Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA mit $\mathcal{L}(\mathcal{A}) = \mathcal{L}$.

Wir wählen k = |Q|.

Sei $T = (P, t) \in L$ ein Baum mit Höhe > k, und sei r ein erfolgreicher Run von \mathcal{A} auf \mathcal{T} .

Charakt.

8:32

und sei r ein erfolgreicher Run von A auf T

Wegen Höhe $\geq k$ gibt es in T einen Pfad mit $\geq k + 1$ Knoten. Darauf gibt es also zwei Positionen p1 # p2 mit demselben Zustand, d.h. $r(p_1) = r(p_2) = a$ für ein $a \in Q$.

Charakt.

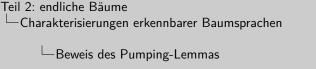
Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA mit $L(\mathcal{A}) = L$.

Wir wählen k = |Q|.

Sei $T = (P, t) \in L$ ein Baum mit Höhe > k, und sei r ein erfolgreicher Run von \mathcal{A} auf \mathcal{T} .

Wegen Höhe > k gibt es in T einen Pfad mit > k + 1 Knoten. Darauf gibt es also zwei Positionen $p_1 \neq p_2$ mit demselben Zustand, d. h. $r(p_1) = r(p_2) = q$ für ein $q \in Q$.



8:32

Beweis. Sei L eine erkennbare Baumsprache, und sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEBA mit $L(\mathcal{A}) = L$.

Wir wählen k = |Q|.

Sei $T = (P, t) \in L$ ein Baum mit Höhe > k, und sei r ein erfolgreicher Run von \mathcal{A} auf \mathcal{T} .

Wegen Höhe > k gibt es in T einen Pfad mit > k + 1 Knoten. Darauf gibt es also zwei Positionen $p_1 \neq p_2$ mit demselben Zustand, d. h. $r(p_1) = r(p_2) = q$ für ein $q \in Q$.

O. B. d. A. ist $p_2 = p_1 p_3$ für ein $p_3 \neq \varepsilon$.

T 2.6

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas

Reserve Sei / eine erkennhare Raumsnrach

und sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA mit L(A) = LSei $T = (P, t) \in L$ ein Baum mit Höhe $\geq k$. und sei r ein erfolgreicher Run von A auf T

Wegen Höhe $\geq k$ gibt es in T einen Pfad mit $\geq k + 1$ Knoten. Darauf gibt es also zwei Positionen p1 of p2 mit demselben Zustand, $d.h. r(p_1) = r(p_2) = a f ur e in a \in Q$ O. B. d. A. ist $p_2 = p_1p_2$ für ein $p_2 \neq \epsilon$

8:32

Motivation

Grundbegriffe

Beweis des Pumping-Lemmas

Charakt.

Entscheid.-probl.

Teil 2: endliche Bäume

2019-02-01

Top-down-BAs Abschlusseig.

 $U = T_{\Phi}$ C = derjenige Kontext mit C[U] = TD = derjenige Kontext mit U = D[V]

Beweis des Pumping-Lemmas

Seien nun:

$$U = T_{p_1}$$

$$C = \text{derjenige Kontext mit } C[U] = T$$

$$V = T_{p_2}$$

$$D = \text{derjenige Kontext mit } U = D[V]$$

8:38

-Charakterisierungen erkennbarer Baumsprachen

Beweis des Pumping-Lemmas

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

Beweis des Pumping-Lemmas

Seien nun:

$$U = T_{p_1}$$
 $C = \text{derjenige Kontext mit } C[U] = T$
 $V = T_{p_2}$

D = derjenige Kontext mit U = D[V]

Weil $p_1 \neq p_2$, ist D nichttrivial, also $D \neq C_0$ wie gefordert.

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas $U = T_{p_1}$ C = derjenige Kontext mit C[U] = TD = derjenige Kontext mit U = D[V]Weil $p_1 \neq p_2$, ist D nichttrivial, also $D \neq C_0$ wie gefordert

8:38

2019-02-01

Beweis des Pumping-Lemmas

Seien nun:

$$U = T_{p_1}$$

$$C = \text{derjenige Kontext mit } C[U] = T$$

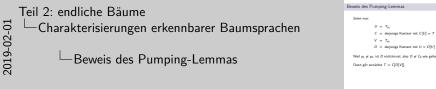
$$V = T_{p_2}$$

$$D = \text{derjenige Kontext mit } U = D[V]$$

Weil $p_1 \neq p_2$, ist D nichttrivial, also $D \neq C_0$ wie gefordert.

Dann gilt zunächst
$$T = C[D[V]]$$
.

T 2.6 Forts.



$$U = T_{p_1}$$

$$C = \text{derjenige Kontext mit } C[U] = T$$

$$V = T_{p_2}$$

$$D = \text{derjenige Kontext mit } U = D[V]$$

Weil $p_1 \neq p_2$, ist D nichttrivial, also $D \neq C_0$ wie gefordert.

Dann gilt zunächst T = C[D[V]].

T 2.6 Forts.

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen 2019-02-01 Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas $U = T_{p_1}$ C = derjenige Kontext mit C[U] = TD = derienize Kontext mit U = D[V]Dann gilt zunächst T = C[D[V]]Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

Grundbegriffe

Charakt.

1. Fall: i = 0, also $T_0 = C[V]$.

Top-down-BAs

Entscheid.-probl.

T 2.6 Forts.

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

1. Fall:
$$i = 0$$
, also $T_0 = C[V]$.

T 2.6 Forts.

Definieren Run r_0 positionsweise:

$$r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_2 p') & \text{falls } p = p_1 p' \end{cases}$$
 (*)

8:42

Teil 2: endliche Bäume

Charakterisierungen erkennbarer Baumsprachen

Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

1. Fall: i = 0, also $T_0 = C[V]$.

T 2.6 Forts.

Definieren Run r_0 positionsweise:

$$r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_2 p') & \text{falls } p = p_1 p' \end{cases}$$
 (*)

Leicht zu prüfen: r_0 ist ein **Run** von \mathcal{A} auf T_0 .

Beweis des Pumping-Lemmas Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas Leicht zu neifen: er ist ein Run von 4 auf To-

Charakt.

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

1. Fall: i = 0, also $T_0 = C[V]$.

T 2.6 Forts.

Definieren Run r_0 positionsweise:

$$r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_2 p') & \text{falls } p = p_1 p' \end{cases}$$
 (*)

Leicht zu prüfen: r_0 ist ein **Run** von \mathcal{A} auf T_0 .

 r_0 ist **erfolgreich**: wegen (*) ist $r_0(\varepsilon) = r(\varepsilon)$.

Beweis des Pumping-Lemmas Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

 $\begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_1p_1^{\prime\prime}) & \text{falls } p = p_1p_1^{\prime\prime} \end{cases}$ (*)

8:42

2019-02-01

Beweis des Pumping-Lemmas

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

Charakt.

1. Fall: i = 0, also $T_0 = C[V]$.

T 2.6 Forts.

Definieren Run r_0 positionsweise:

$$r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_2 p') & \text{falls } p = p_1 p' \end{cases}$$
 (*)

Leicht zu prüfen: r_0 ist ein **Run** von \mathcal{A} auf T_0 .

 r_0 ist **erfolgreich**: wegen (*) ist $r_0(\varepsilon) = r(\varepsilon)$.

Also $T_0 \in L$.

Beweis des Pumping-Lemmas Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0. $\begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_1p_1^{\prime\prime}) & \text{falls } p = p_1p_1^{\prime\prime} \end{cases}$ (*) Beweis des Pumping-Lemmas

8:42

2019-02-01

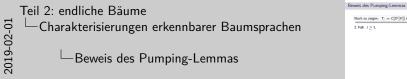
Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

2. Fall: $i \ge 1$.

T 2.6 Forts.



8:47

Fallunterscheidung nicht vorlesen, sondern anzeichnen!

2. Fall: $i \ge 1$.

Definieren Run r_i positionsweise:

Beweis des Pumping-Lemmas

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

T 2.6 Forts.

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas

8:47

Fallunterscheidung nicht vorlesen, sondern anzeichnen!

 $r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_1p') & \text{falls } p = p_1p_3^jp', \ p' \text{ kein NF von } p_3 \text{ und} \\ & p \text{ kein NF von } p_1p_3^i \\ r(p_2p') & \text{falls } p = p_1p_3^ip' \end{cases}$

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

2. Fall: i > 1.

T 2.6 Forts.

Definieren Run r_i positionsweise:

$$r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_1p') & \text{falls } p = p_1p_3^jp', \ p' \text{ kein NF von } p_3 \text{ und} \\ & p \text{ kein NF von } p_1p_3^i \\ r(p_2p') & \text{falls } p = p_1p_3^ip' \end{cases}$$

Wie im 1. Fall: r_i ist **erfolgreicher Run** von \mathcal{A} auf \mathcal{T}_i .

Beweis des Pumping-Lemmas Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Beweis des Pumping-Lemmas Wie im 1 Fall: r: ist orfoloroicher Bun von 4 auf T:

8:47

Fallunterscheidung nicht vorlesen, sondern anzeichnen!

Beweis des Pumping-Lemmas

Definieren Run r_i positionsweise:

Noch zu zeigen: $T_i := C[D^i[V]] \in L$ für alle i > 0.

Wie im 1. Fall: r_i ist **erfolgreicher Run** von \mathcal{A} auf \mathcal{T}_i .

T 2.6 Forts.

Beweis des Pumping-Lemmas

Beweis des Pumping-Lemmas

8:47

Fallunterscheidung nicht vorlesen, sondern anzeichnen!

2. Fall: i > 1.

 $r_0(p) = \begin{cases} r(p) & \text{falls } p \text{ kein Nachfolger von } p_1 \text{ ist} \\ r(p_1 p') & \text{falls } p = p_1 p_3^j p', \ p' \text{ kein NF von } p_3 \text{ und} \\ & p \text{ kein NF von } p_1 p_3^i \\ r(p_2 p') & \text{falls } p = p_1 p_3^i p' \end{cases}$

Motivation

Grundbegriffe

Charakt.

Anwendung des Pumping-Lemmas

Wenn es für alle Konstanten $k \in \mathbb{N}$

ein $i \in \mathbb{N}$ gibt mit $C[D^i[V]] \notin L$,

dann ist *L* keine erkennbare Baumsprache.

Entscheid.-probl.

Benutzen Kontraposition (siehe Kapitel "endliche Wörter"):

einen Baum $T \in L$ mit Höhe $(T) \ge k$ gibt, so dass es

-Anwendung des Pumping-Lemmas

Benutzen Kontraposition (siehe Kapitel ..endliche Wörter") für alle Kontexte C, D mit $D \neq C_0$ und Bäume V mit T = C[D[V]]dann ist L keine erkennbare Baumsprache

Anwendung des Pumping-Lemmas

8:52

für alle Kontexte C, D mit $D \neq C_0$ und Bäume V mit T = C[D[V]]

T 2.7

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Der Satz von Myhill-Nerode für Baumsprachen

Der Satz von Myhill-Nerode für Baumsprachen

Ziel: notwendige **und** hinreichende Bedingung für Erkennbarkeit

Definition 2.8

Sei L eine Baumsprache über Σ .

Zwei Σ -Bäume T_1 , T_2 sind L-äquivalent (Schreibw.: $T_1 \sim_L T_2$), wenn für alle Σ -Kontexte C gilt:

$$C[T_1] \in L$$
 genau dann, wenn $C[T_2] \in L$

Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Der Satz von Myhill-Nerode für Baumsprachen

Sei L eine Baumsprache über Σ. Zwei Σ-Bäume T1, To sind L-ätuivalent (Schreibw.: T1 ~1 To). wenn für alle Σ-Kontexte C gilt:

 $C[T_1] \in L$ genau dann, wenn $C[T_2] \in L$

9:14

Satz ohne Beweis. An Tafel nur Bsp.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Der Satz von Myhill-Nerode für Baumsprachen

Charakt.

Ziel: notwendige **und** hinreichende Bedingung für Erkennbarkeit

Definition 2.8

Sei L eine Baumsprache über Σ .

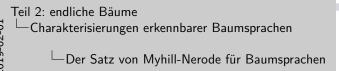
Zwei Σ -Bäume T_1 , T_2 sind L-äquivalent (Schreibw.: $T_1 \sim_L T_2$), wenn für alle Σ -Kontexte C gilt:

$$C[T_1] \in L$$
 genau dann, wenn $C[T_2] \in L$

Satz 2.9

 $L \subseteq \Sigma^*$ is NEBA-erkennbar gdw. \sim_I endlichen Index hat.

T 2.8





9:14

Satz ohne Beweis. An Tafel nur Bsp.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Der Satz von Myhill-Nerode für Baumsprachen

Charakt.

Ziel: notwendige **und** hinreichende Bedingung für Erkennbarkeit

Definition 2.8

Sei L eine Baumsprache über Σ .

Zwei Σ -Bäume T_1 , T_2 sind L-äquivalent (Schreibw.: $T_1 \sim_L T_2$), wenn für alle Σ -Kontexte C gilt:

 $C[T_1] \in L$ genau dann, wenn $C[T_2] \in L$

Satz 2.9

 $L \subseteq \Sigma^*$ is NEBA-erkennbar gdw. \sim_I endlichen Index hat.

T 2.8

Auch für Baumsprachen gilt: endlicher Index n von \sim_{I} = minimale Anzahl von Zuständen in einem DEBA, der L erkennt Teil 2: endliche Bäume Charakterisierungen erkennbarer Baumsprachen Der Satz von Myhill-Nerode für Baumsprachen

9:14

Satz ohne Beweis. An Tafel nur Bsp.



Grundbegriffe

Abschlusseig.

2019-02-01

Teil 2: endliche Bäume Top-down-Baumautomaten

└─Und nun ...

Und nun . Top-down-Baumautomaten

- 2 Grundbegriffe

Automatentheorie u. i. A. WiSe 2018/19

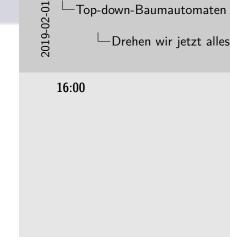
- 4 Top-down-Baumautomaten
- 5 Abschlusseigenschaften

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XMLTeil 2: endliche Bäume

Drehen wir jetzt alles um? ©







☐ Drehen wir jetzt alles um? ⑤



Top-down-Baumautomaten

- » Σ ein r-Alnhahet ist μ Δ eine Menge von Überführungsregeln der Form
- $(s, q) \rightarrow (q_1, \dots, q_m)$ ist mit $m \ge 0$. $a \in \Sigma_m$. $a, a_1, \dots, a_m \in Q$. und

I ⊂ Q die Menge der Anfangszustände ist

16:00

Teil 2: endliche Bäume

-Top-down-Baumautomaten

Top-down-Baumautomaten

... weisen der Wurzel einen Startzustand zu und arbeiten sich dann von oben nach unten zu den Blättern durch:

Definition 2.10

Ein nichtdet. Top-down-Automat auf endl. geord. Bäumen (NETDBA) ist ein Quadrupel $A = (Q, \Sigma, \Delta, I)$, wobei

- Q eine endliche nichtleere Zustandsmenge ist,
- Σ ein r-Alphabet ist,
- Δ eine Menge von Überführungsregeln der Form

$$(a,q) \rightarrow (q_1,\ldots,q_m)$$

ist mit $m \ge 0$, $a \in \Sigma_m$, $q, q_1, \ldots, q_m \in Q$, und

• $I \subset Q$ die Menge der Anfangszustände ist.

• $r(\varepsilon) \in I$

Top-down-BAs

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

• Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:

16:03

Automatentheorie u. i. A. WiSe 2018/19

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.11

Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - $r(\varepsilon) \in I$
 - Wenn $t(p) = a \in \Sigma_m \ (m \geqslant 1) \ \text{und} \ r(p) = q$ und wenn $r(p1) = q_1, \ldots, r(pm) = q_m$

Teil 2: endliche Bäume -Top-down-Baumautomaten Berechnungen, Akzeptanz und erkannte Sprache

• Wenn $t(p) = a \in \Sigma_m$ $(m \ge 1)$ and r(p) = qand wenn $r(p1) = a_1, \dots, r(pm) = a_m$

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - $r(\varepsilon) \in I$
 - Wenn $t(p) = a \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m,$ dann gibt es eine Regel $(a, q) \rightarrow (q_1, \dots, q_m) \in \Delta$.

Teil 2: endliche Bäume -Top-down-Baumautomaten Berechnungen, Akzeptanz und erkannte Sprache

16:03

dann gibt es eine Regel $(a,q) \rightarrow (q_1,\ldots,q_m) \in \Delta$

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

Top-down-BAs

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - $r(\varepsilon) \in I$
 - Wenn $t(p) = a \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann gibt es eine Regel $(a, q) \rightarrow (q_1, \dots, q_m) \in \Delta$.
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $(a, q) \to () \in \Delta$.

16:03

Automatentheorie u. i. A. WiSe 2018/19

Charakt.

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - $r(\varepsilon) \in I$
 - Wenn $t(p) = a \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann gibt es eine Regel $(a, q) \rightarrow (q_1, \dots, q_m) \in \Delta$.
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $(a, q) \to () \in \Delta$.
- \mathcal{A} akzeptiert T, wenn es einen Run von \mathcal{A} auf T gibt.

Teil 2: endliche Bäume -Top-down-Baumautomaten Berechnungen, Akzeptanz und erkannte Sprache

Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum

Run von A auf T ist eine Fkt. r : P → O mit • Wenn $t(p) = a \in \Sigma_m \ (m \ge 1) \ \text{und} \ r(p) = q$ dann gibt es eine Regel $(a,q) \to (q_1,\ldots,q_m) \in \Delta$

 $L(A) = \{ T \text{ über } \Sigma \mid A \text{ akzeptiert } T \}.$

Berechnungen, Akzeptanz und erkannte Sprache

Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum Run von A auf T ist eine Fkt. r : P → O mit • Wenn $t(p) = a \in \Sigma_m \ (m \ge 1) \ \text{und} \ r(p) = q$ dann gibt es eine Regel $(a,q) \to (q_1,\ldots,q_m) \in \Delta$ Wenn $t(p) = a \in \Sigma_1$ and r(p) = a, dann $(a, a) \rightarrow (i) \in \Delta$.

16:03

```
Berechnungen, Akzeptanz und erkannte Sprache
```

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum.

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:
 - $r(\varepsilon) \in I$
 - Wenn $t(p) = a \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann gibt es eine Regel $(a, q) \rightarrow (q_1, \dots, q_m) \in \Delta$.
 - Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $(a, q) \to () \in \Delta$.
- \mathcal{A} akzeptiert T, wenn es einen Run von \mathcal{A} auf T qibt.
- Die von \mathcal{A} erkannte Sprache ist $L(A) = \{ T \text{ "uber } \Sigma \mid A \text{ akzeptiert } T \}.$

Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA und T = (P, t) ein Σ -Baum Run von A auf T ist eine Fkt. r: P → Q mit • Wenn $t(p) = a \in \Sigma_m \ (m \ge 1) \ \text{und} \ r(p) = q$ dann gibt es eine Regel $(a,q) \to (q_1,\ldots,q_m) \in \Delta$ Wenn $t(p) = a \in \Sigma_1$ and r(p) = a, dann $(a, a) \rightarrow (i) \in \Delta$. A akzeptiert T. wenn es einen Run von A auf T oibt Die von A erkannte Sprache ist

 $L(A) = \{T \text{ über } \Sigma \mid A \text{ akzeptiert } T\}.$

Beachte: Keine Endzustände nötig - die Regeln in Δ müssen nur erlauben, von der Wurzel bis zu allen Blättern "durchzukommen

Teil 2: endliche Bäume -Top-down-Baumautomaten

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.11

Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und $\mathcal{T} = (P, t)$ ein Σ -Baum.

Top-down-BAs

• Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit:

Berechnungen, Akzeptanz und erkannte Sprache

- $r(\varepsilon) \in I$
- Wenn $t(p) = a \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wenn $r(p1) = q_1, \ldots, r(pm) = q_m$ dann gibt es eine Regel $(a, q) \rightarrow (q_1, \dots, q_m) \in \Delta$.
- Wenn $t(p) = a \in \Sigma_0$ und r(p) = q, dann $(a, q) \to () \in \Delta$.
- A akzeptiert T, wenn es einen Run von A auf T gibt.
- Die von \mathcal{A} erkannte Sprache ist $L(A) = \{ T \text{ "uber } \Sigma \mid A \text{ akzeptiert } T \}.$

Beachte: Keine Endzustände nötig – die Regeln in Δ müssen nur erlauben, von der Wurzel bis zu allen Blättern "durchzukommen".

XML

 \bullet Sei $\Sigma = \{a/2, b/0, c/0\}$ und

 $A = (\{q_0, q_1, q_2, q_4\}, \Sigma, \Delta, \{q_0\}) \text{ mit}$ $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$ $(a, q_0) \rightarrow (q_x, q_y)$

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

$$\Delta = \{ (a, q_0) \to (q_1, q_1), \ (b, q_2) \to (),$$

$$(a, q_1) \to (q_2, q_2), \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$

Top-down-BAs

Teil 2: endliche Bäume Top-down-Baumautomaten Beispiel 1

XML

 \bullet Sei $\Sigma = \{a/2, b/0, c/0\}$ und

u Dann gibt es auf dem Baum

 $A = (\{q_0, q_1, q_2, q_4\}, \Sigma, \Delta, \{q_0\})$ mit $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$

Beispiel 1

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

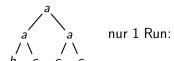
$$\Delta = \{ \ (a, q_0) \to (q_1, q_1), \ \ (b, q_2) \to (),$$

$$(a, q_1) \to (q_2, q_2), \ \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$
 }

Top-down-BAs

• Dann gibt es auf dem Baum



Teil 2: endliche Bäume Top-down-Baumautomaten Beispiel 1

 $A = (\{q_0, q_1, q_2, q_4\}, \Sigma, \Delta, \{q_0\})$ mit $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$

a nur 1 Run: qu qu

u Dann gibt es auf dem Baum

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

$$\Delta = \{ (a, q_0) \to (q_1, q_1), \ (b, q_2) \to (),$$

$$(a, q_1) \to (q_2, q_2), \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$
 }

• Dann gibt es auf dem Baum



nur 1 Run:





XML

. Dann eiht es auf dem Baum

 $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$

Beispiel 1

Beispiel 1

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

$$\Delta = \{ \ (a, q_0) \to (q_1, q_1), \ \ (b, q_2) \to (),$$

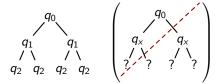
$$(a, q_1) \to (q_2, q_2), \ \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$
 }

Top-down-BAs

• Dann gibt es auf dem Baum





Teil 2: endliche Bäume -Top-down-Baumautomaten Beispiel 1

XML

Beispiel 1

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

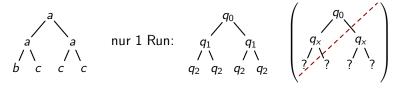
$$\Delta = \{ (a, q_0) \to (q_1, q_1), \ (b, q_2) \to (),$$

$$(a, q_1) \to (q_2, q_2), \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$
 }

Top-down-BAs

• Dann gibt es auf dem Baum



•
$$L(A) = ?$$

Teil 2: endliche Bäume -Top-down-Baumautomaten Beispiel 1

 \bullet Sei $\Sigma = \{a/2, b/0, c/0\}$ und $A = (\{q_0, q_1, q_2, q_4\}, \Sigma, \Delta, \{q_0\})$ mit $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$. Dann eiht es auf dem Baum • L(A) = ?

Beispiel 1

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und

$$\mathcal{A} = (\{q_0, q_1, q_2, q_x\}, \ \Sigma, \ \Delta, \ \{q_0\}) \text{ mit}$$

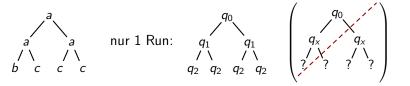
$$\Delta = \{ (a, q_0) \to (q_1, q_1), \ (b, q_2) \to (),$$

$$(a, q_1) \to (q_2, q_2), \ (c, q_2) \to (),$$

$$(a, q_0) \to (q_x, q_x)$$

Top-down-BAs

• Dann gibt es auf dem Baum



• $L(A) = \{ T \text{ "uber } \Sigma \mid \text{ alle Pfade in } T \text{ haben L"ange 2} \}$

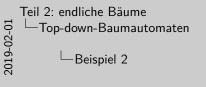
Teil 2: endliche Bäume -Top-down-Baumautomaten Beispiel 1

• Sei $\Sigma = \{a/2, b/0, c/0\}$ und $A = (\{q_0, q_1, q_2, q_4\}, \Sigma, \Delta, \{q_0\}) \text{ mit}$ $\Delta = \{(a, q_0) \rightarrow (q_1, q_1), (b, q_2) \rightarrow (),$ $(a, q_1) \rightarrow (q_2, q_2), (c, q_2) \rightarrow (),$ u Dann gibt es auf dem Baum v L(A) = { T über Σ | alle Pfade in T haben Länge 2}

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NETDBA erkennt $L_{cd} = \{T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister}\}$?



Beispiel 2 $Sai \ \Gamma = \{ p/2, \ b/1, \ c/3, \ c/3 \} \ .$ Whicher BETDER observe $L_{ain} = \{ T \ daw \ \Sigma \ | \ plac \ c'disht but an earlies \ d'Gardweiser \}^{\gamma}$

Motivation

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NETDBA erkennt $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \} ?$

NETDBA
$$\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\}) \text{ mit}$$

$$\Delta = \{ (a, q_0) \to (q_0, q_0), \quad b(q_0) \to q_0, \quad (c, q_c) \to (),$$

$$(a, q_0) \to (q_c, q_d), \qquad (d, q_d) \to (),$$

$$(d, q_0) \to () \}$$

Teil 2: endliche Bäume -Top-down-Baumautomaten Beispiel 2

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NETDBA erkennt $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \} ?$

NETDBA
$$\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\}) \text{ mit}$$

$$\Delta = \{ (a, q_0) \to (q_0, q_0), \quad b(q_0) \to q_0, \quad (c, q_c) \to (),$$

$$(a, q_0) \to (q_c, q_d), \qquad (d, q_d) \to (),$$

$$(d, q_0) \to () \}$$

Vergleiche mit dem NEBA $\mathcal{A} = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\})$ mit $\Delta = \{ a(q_f, q_f) \rightarrow q_f, b(q_f) \rightarrow q_f, c \rightarrow q_c,$ $a(q_c, q_d) \rightarrow q_f$ $d \rightarrow q_d$ $d \rightarrow q_f$ }

Teil 2: endliche Bäume -Top-down-Baumautomaten Beispiel 2

 $(d, q_d) \rightarrow ().$ Reiche mit dem NEBA $A = (\{a_r, a_\ell, a_r\}, \Sigma, \Delta, \{a_r\})$ mit

Grundbegriffe

Charakt.

Abschlusseig.

Top-down-BAs

Entscheid.-probl.

Beispiel 2

Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$. Welcher NETDBA erkennt $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \} ?$

NETDBA $\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\})$ mit $\Delta = \{ (a, q_0) \rightarrow (q_0, q_0), b(q_0) \rightarrow q_0, (c, q_c) \rightarrow (), \}$ $(a, q_0) \rightarrow (q_c, q_d),$ $(d, q_d) \rightarrow (),$ $(d, q_0) \rightarrow ()$

Vergleiche mit dem NEBA $\mathcal{A} = (\{q_c, q_d, q_f\}, \Sigma, \Delta, \{q_f\})$ mit $\Delta = \{ a(q_f, q_f) \rightarrow q_f, b(q_f) \rightarrow q_f, c \rightarrow q_c,$

 $a(q_c, q_d) \rightarrow q_f$ $d \rightarrow q_d$ $d \rightarrow q_f$ }

Was sagt uns das über das Verhältnis NETDBAs : NEBAs?

Teil 2: endliche Bäume -Top-down-Baumautomaten

Beispiel 2

 $(d, q_d) \rightarrow ().$ eiche mit dem NEBA $A = (\{a_r, a_\ell, a_\ell\}, \Sigma, \Delta, \{a_\ell\})$ mit

Motivation Grundbegriffe

Charakt.

Abschlusseig.

Entscheid.-probl.

Teil 2: endliche Bäume

Top-down-BAs

NETDBAs vs. NEBAs

NETDBAs vs. NEBAs

NETDBAs und **NEBAs** sind gleichmächtig!

Satz 2.12

 $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$

└─NETDBAs vs. NEBAs

Top-down-Baumautomaten

NETDBAs und NEBAs sind gleichmächtig!

 $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$ $\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (s, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$

NETDBAs und NEBAs sind gleichmächtig!

Charakt.

Satz 2.12

$$\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$$

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $\mathcal{A}^{\uparrow} = (Q, \Sigma, \Delta^{\uparrow}, F^{\uparrow})$ mit:

$$\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \to q) \mid (a, q) \to (q_1, \dots, q_m) \in \Delta\}$$
 $F^{\uparrow} = I$

Teil 2: endliche Bäume -Top-down-Baumautomaten └─NETDBAs vs. NEBAs

auch ein erfolgreicher Run von \mathcal{A}^{\uparrow} auf T

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtig!

Satz 2.12

$$\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$$

Top-down-BAs

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $\mathcal{A}^{\uparrow} = (Q, \Sigma, \Delta^{\uparrow}, F^{\uparrow})$ mit:

$$\Delta^{\uparrow} = \{ a(q_1, \dots, q_m) \to q) \mid (a, q) \to (q_1, \dots, q_m) \in \Delta \}$$

$$F^{\uparrow} = I$$

Dann ist jeder Run von \mathcal{A} auf einem Σ -Baum \mathcal{T} auch ein **erfolgreicher** Run von \mathcal{A}^{\uparrow} auf Tund umgekehrt.

Teil 2: endliche Bäume -Top-down-Baumautomaten └─NETDBAs vs. NEBAs

 $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}$ Konstruieren NEBA $A^{\dagger} = (Q, \Sigma, \Delta^{\dagger}, F^{\dagger})$ mit $\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (s, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$ Dann ist ieder Run von A auf einem Σ-Baum T

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtig!

Satz 2.12

$$\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$$

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $\mathcal{A}^{\uparrow} = (Q, \Sigma, \Delta^{\uparrow}, F^{\uparrow})$ mit:

$$\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \to q) \mid (a, q) \to (q_1, \dots, q_m) \in \Delta\}$$
 $F^{\uparrow} = I$

Dann ist jeder Run von \mathcal{A} auf einem Σ -Baum \mathcal{T} auch ein **erfolgreicher** Run von \mathcal{A}^{\uparrow} auf Tund umgekehrt.

Daraus folgt $L(A^{\uparrow}) = L(A)$.

Teil 2: endliche Bäume -Top-down-Baumautomaten └─NETDBAs vs. NEBAs NETDBAs vs. NEBAs NETDBAs und NEBAs sind gleichmächtig! $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}$ Konstruieren NEBA $A^{\dagger} = (Q, \Sigma, \Delta^{\dagger}, F^{\dagger})$ mit $\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (s, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$

> Dann ist ieder Run von A auf einem D-Baum T auch ein erfolgreicher Run von A^{\dagger} auf TDaraus folgt $L(A^{\dagger}) = L(A)$.

Daraus folgt $L(A^{\dagger}) = L(A)$. Rückrichtung analog.

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtig!

Satz 2.12

$$\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$$

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $\mathcal{A}^{\uparrow} = (Q, \Sigma, \Delta^{\uparrow}, F^{\uparrow})$ mit:

$$\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \to q) \mid (a, q) \to (q_1, \dots, q_m) \in \Delta\}$$
 $F^{\uparrow} = I$

Dann ist jeder Run von \mathcal{A} auf einem Σ -Baum \mathcal{T} auch ein **erfolgreicher** Run von \mathcal{A}^{\uparrow} auf Tund umgekehrt.

Daraus folgt $L(A^{\uparrow}) = L(A)$.

Rückrichtung analog.

Teil 2: endliche Bäume -Top-down-Baumautomaten

└─NETDBAs vs. NEBAs

Motivation

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Determinisierung von NETDBAs

Erinnerung an Beispiel 2: Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$ und $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \}.$



16:20

 $(d, q_d) \rightarrow (),$ $(d, q_0) \rightarrow ()$

Erinnerung an Beispiel 2: Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$ und

 $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \}.$

 $\Delta = \{ (a, q_0) \rightarrow (q_0, q_0), b(q_0) \rightarrow q_0, (c, q_c) \rightarrow (), \}$

NETDBA $\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\})$ mit

 $(a, q_0) \rightarrow (q_c, q_d),$

Determinisierung von NETDBAs

16:20

Determinisierung von NETDBAs

Erinnerung an Beispiel 2: Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$ und $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \}.$

NETDBA
$$\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\})$$
 mit

$$\Delta = \{ (a, q_0) \to (q_0, q_0), b(q_0) \to q_0, (c, q_c) \to (), (a, q_0) \to (q_c, q_d), (d, q_d) \to (), \}$$

▲ Nichtdeterminismus!

 $(d, q_0) \rightarrow ()$

16:20

Determinisierung von NETDBAs

Erinnerung an Beispiel 2: Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$ und $L_{cd} = \{ T \text{ ""uber } \Sigma \mid \text{ jedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister} \}.$

NETDBA
$$\mathcal{A} = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_0\})$$
 mit
$$\Delta = \{ \begin{array}{c} (a, q_0) \rightarrow (q_0, q_0), & b(q_0) \rightarrow q_0, & (c, q_c) \rightarrow (), \\ (a, q_0) \rightarrow (q_c, q_d), & (d, q_d) \rightarrow (), \\ & &$$

Wir wissen ja, wie man Nichtdeterminismus "loswird".

16:20

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Determinisierung von NETDBAs? u die erkennbare Baumsprache $L = \{a(bc), a(cb)\}$ (denke an die alternative Schreibweise von Folie 18)

Determinisierung von NETDBAs?

Charakt.

Betrachte

- $\Sigma = \{a/2, b/0, c/0\}$ und
- die erkennbare Baumsprache $L = \{a(bc), a(cb)\}.$ (denke an die alternative Schreibweise von Folie 18)

Frage: Welcher DETDBA erkennt *L*?

16:23 bis 16:30

Teil 2: endliche Bäume

-Top-down-Baumautomaten

-Determinisierung von NETDBAs?

Konsequenz: DETDBAs sind schwächer als DEBAs (und NE(TD)BAs); NEBAs bleiben das Standardmodell!

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Determinisierung von NETDBAs?

Charakt.

Betrachte

• $\Sigma = \{a/2, b/0, c/0\}$ und

• die erkennbare Baumsprache $L = \{a(bc), a(cb)\}.$

(denke an die alternative Schreibweise von Folie 18)

Frage: Welcher DETDBA erkennt *L*?

Anwort: Keiner!

Lemma 2.13

L wird von keinem DETDBA erkannt.

Beweis: siehe Tafel. T 2.9 Teil 2: endliche Bäume -Top-down-Baumautomaten -Determinisierung von NETDBAs?



16:23 bis 16:30

Konsequenz: DETDBAs sind schwächer als DEBAs (und NE(TD)BAs); NEBAs bleiben das Standardmodell!

Top-down-BAs

Abschlusseig.

• die erkennbare Baumsprache $L = \{a(bc), a(cb)\}.$

(denke an die alternative Schreibweise von Folie 18)

Frage: Welcher DETDBA erkennt *L*?

Anwort: Keiner!

Lemma 2.13

L wird von keinem DETDBA erkannt.

Beweis: siehe Tafel.

Korollar 2.14

Es gibt erkennbare Baumsprachen, die nicht von einem DETDBA erkannt werden. Teil 2: endliche Bäume -Top-down-Baumautomaten

–Determinisierung von NETDBAs?

Determinisierung von NETDBAs? u die erkennbare Baumsprache $L = \{a(bc), a(cb)\}$ I wird von keinem DETDRA erkannt Es gibt erkennbare Baumsprachen, die nicht von einem DETDBA erkannt werder

16:23 bis 16:30

Konsequenz: DETDBAs sind schwächer als DEBAs (und NE(TD)BAs); NEBAs bleiben das Standardmodell!

T 2.9

Grundbegriffe Charakt. Top-down-BAs Abschlusseig.

2019-02-01

Motivation

Abschlusseigenschaften

Und nun ...

2 Grundbegriffe

5 Abschlusseigenschaften

Automatentheorie u. i. A. WiSe 2018/19

Teil 2: endliche Bäume

47

Grundbegriffe

Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl.

Operationen auf Baumsprachen

w Vereinigung, falls eilt:

Falls L1, L2 erkennbar, so auch L1 U L2

a Komplement, falls eilt: Falls I erkennhar on such T

Schnitt, falls eilt: Falls L_1, L_2 erkennbar, so such $L_1 \cap L_2$

Operationen auf Baumsprachen

Zur Erinnerung: die Menge der erkennbaren Baumsprachen heißt abgeschlossen unter . . .

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

16:30

2019-02-01

Teil 2: endliche Bäume

Abschlusseigenschaften

-Operationen auf Baumsprachen

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Operationen auf Baumsprachen

Zur Erinnerung: die Menge der erkennbaren Baumsprachen heißt abgeschlossen unter ...

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

Quiz

Unter welchen Operationen sind die NEBA-erkennbaren Sprachen abgeschlossen?

Vereinigung? Komplement? Schnitt?

Teil 2: endliche Bäume 2019-02-01 Abschlusseigenschaften -Operationen auf Baumsprachen



16:30

abgeschlossen unter ...

• Schnitt, falls gilt:

• Vereinigung, falls gilt:

• Komplement, falls gilt:

Falls L erkennbar, so auch \overline{L} .

Charakt.

Entscheid.-probl.

Zur Erinnerung: die Menge der erkennbaren Baumsprachen heißt

Falls L_1, L_2 erkennbar, so auch $L_1 \cup L_2$.

Falls L_1, L_2 erkennbar, so auch $L_1 \cap L_2$.

2019-02-01

Teil 2: endliche Bäume Abschlusseigenschaften

-Operationen auf Baumsprachen



Operationen auf Baumsprachen

16:30

abgeschlossen?

Quiz

Unter welchen Operationen sind die NEBA-erkennbaren Sprachen

Vereinigung? Komplement? Schnitt?

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Operationen auf Baumsprachen

Zur Erinnerung: die Menge der erkennbaren Baumsprachen heißt abgeschlossen unter ...

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

Falls L_1, L_2 erkennbar, so auch $L_1 \cap L_2$.

Quiz

Unter welchen Operationen sind die NEBA-erkennbaren Sprachen abgeschlossen?

Vereinigung? Komplement? ✓ Schnitt?

Teil 2: endliche Bäume Abschlusseigenschaften -Operationen auf Baumsprachen

w Vereinigung, falls gilt: Falls I erkennhar on such T Schnitt, falls eilt: Falls L_1, L_2 erkennbar, so auch $L_1 \cap L_2$ Unter welchen Operationen sind die NEBA-erkennbaren Sprache Vereinigung? ✓ Komplement? ✓

Operationen auf Baumsprachen

16:30

2019-02-01

Zur Erinnerung: die Menge der erkennbaren Baumsprachen heißt abgeschlossen unter ...

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

Falls L_1, L_2 erkennbar, so auch $L_1 \cap L_2$.

Quiz

Unter welchen Operationen sind die NEBA-erkennbaren Sprachen abgeschlossen?

Vereinigung?

Komplement? Schnitt?

Entscheid.-probl.

Teil 2: endliche Bäume Abschlusseigenschaften

-Operationen auf Baumsprachen

w Vereinigung, falls gilt: Falls I erkennhar on such T Schnitt, falls eilt: Falls L_1, L_2 erkennbar, so auch $L_1 \cap L_2$ Unter welchen Operationen sind die NEBA-erkennbaren Sprache Vereinigung? ✓ Komplement? ✓ Schnitt? ✓

Operationen auf Baumsprachen

16:30

2019-02-01

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Abgeschlossenheit

Satz 2.15

Die Menge der NEBA-erkennbaren Sprachen ist abgeschlossen unter den Operationen \cup , \cap , $\overline{}$.

Direkte Konsequenz aus den folgenden Lemmata.

XML

Motivation Grundbegriffe

Lemma 2.16

Charakt.

Seien A_1, A_2 NEBAs über Σ .

Abschlusseig.

Entscheid.-probl.

Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$.

Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$.

Abgeschlossenheit unter Vereinigung

16:34 bis 16:38

Letzte Zeile:

Motivation

Grundbegriffe

Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

2019-02-01

Abgeschlossenheit unter Vereinigung

Lemma 2.16

Seien A_1, A_2 NEBAs über Σ .

Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$.

Beweis. analog zu NEAs:

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

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

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

Teil 2: endliche Bäume -Abschlusseigenschaften

—Abgeschlossenheit unter Vereinigung

Abgeschlossenheit unter Vereinigung Seien A1. A1 NEBAs über X. Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$. Konstruieren $A_3 = (O_3, \Sigma, \Delta_3, F_3)$ wie folgt.

16:34 bis 16:38

Letzte Zeile:

Abgeschlossenheit unter Vereinigung

Charakt.

Lemma 2.16

Seien A_1 , A_2 NEBAs über Σ .

Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$.

Beweis. analog zu NEAs:

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

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

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

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

Teil 2: endliche Bäume -Abschlusseigenschaften

—Abgeschlossenheit unter Vereinigung

Abgeschlossenheit unter Vereinigung Seien A1. A1 NEBAs über X. Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$. O. B. d. A. gelte $Q_1 \cap Q_2 = \emptyset$. Konstruieren $A_3 = (Q_3, \Sigma, \Delta_3, F_3)$ wie folgt. $Q_1 = Q_1 \cup Q_2$ \bullet $\Delta_1 = \Delta_1 \cup \Delta_2$ $F_1 = F_1 \cup F_2$

16:34 bis 16:38

Letzte Zeile:

2019-02-01

Abgeschlossenheit unter Vereinigung

Seien A., A. NEBAs über X.

—Abgeschlossenheit unter Vereinigung

Lemma 2.16

Seien A_1, A_2 NEBAs über Σ . Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cup L(A_2)$.

Beweis. analog zu NEAs:

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

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

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

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

Dann gilt: $L(A_3) = L(A_1) \cup L(A_2)$

16:34 bis 16:38

Letzte Zeile:

Motivation

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Abgeschlossenheit unter Komplement

Abgeschlossenheit unter Komplement

Lemma 2.17

Sei \mathcal{A} ein NEBA über Σ .

Dann gibt es einen NEBA \mathcal{A}^c mit $L(\mathcal{A}^c) = \overline{L(\mathcal{A})}$.

16:38 bis 16:40

Motivation

Grundbegriffe

Top-down-BAs

Abgeschlossenheit unter Komplement

Lemma 2.17

Sei \mathcal{A} ein NEBA über Σ . Dann gibt es einen NEBA \mathcal{A}^c mit $L(\mathcal{A}^c) = \overline{L(\mathcal{A})}$.

Beweis: analog zu NEAs:

- Umwandlung in DEBA
- Vertauschen von akzeptierenden und nicht-akz. Zuständen

Teil 2: endliche Bäume -Abschlusseigenschaften

-Abgeschlossenheit unter Komplement

16:38 bis 16:40

Umwandlung in DEBA

Teil 2: endliche Bäume Abschlusseigenschaften

-Abgeschlossenheit unter Komplement

Beweis: analog zu NEAs Umwandlung in DEBA w Vertauschen von akzeptierenden und nicht-akz. Zuständer

Nach Satz 2.6 gibt es DEBA $A^d = (Q^d, \Sigma, \Delta^d, F^d)$ mit $L(A^d) = L(A)$ und genau einem Run pro Eingabebaum

Dann gibt es einen NEBA A^c mit $L(A^c) = \overline{L(A)}$.

Abgeschlossenheit unter Komplement

Lemma 2.17

Sei \mathcal{A} ein NEBA über Σ . Dann gibt es einen NEBA \mathcal{A}^c mit $L(\mathcal{A}^c) = L(\mathcal{A})$.

Beweis: analog zu NEAs:

- Umwandlung in DEBA
- Vertauschen von akzeptierenden und nicht-akz. Zuständen

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

Nach Satz 2.6 gibt es **D**EBA $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, F^d)$ mit $L(\mathcal{A}^d) = L(\mathcal{A})$ und **genau einem** Run pro Eingabebaum.

16:38 bis 16:40

2019-02-01

-Abgeschlossenheit unter Komplement

Sei A ein NEBA über Σ.

Beweis: analog zu NEAs

Umwandlung in DEBA

Dann erkennt $A^c = (Q^d, \Sigma, \Delta^d, Q^d \setminus F^d)$ die Sprache $\overline{L(A)}$.

w Vertauschen von akzeptierenden und nicht-akz. Zuständer Nach Satz 2.6 gibt es DEBA $A^d = (Q^d, \Sigma, \Delta^d, F^d)$ mit $L(A^d) = L(A)$ und genau einem Run pro Eingabebaum

Abgeschlossenheit unter Komplement

Lemma 2.17

Sei \mathcal{A} ein NEBA über Σ . Dann gibt es einen NEBA \mathcal{A}^c mit $L(\mathcal{A}^c) = L(\mathcal{A})$.

Beweis: analog zu NEAs:

- Umwandlung in DEBA
- Vertauschen von akzeptierenden und nicht-akz. Zuständen

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

Nach Satz 2.6 gibt es **D**EBA $\mathcal{A}^d = (Q^d, \Sigma, \Delta^d, F^d)$ mit $L(\mathcal{A}^d) = L(\mathcal{A})$ und **genau einem** Run pro Eingabebaum.

Dann erkennt $\mathcal{A}^c = (Q^d, \Sigma, \Delta^d, Q^d \setminus F^d)$ die Sprache $\overline{L(\mathcal{A})}$.

16:38 bis 16:40

2019-02-01

XML

Teil 2: endliche Bäume -Abschlusseigenschaften ☐ Abgeschlossenheit unter Schnitt

Seien A_1 , A_2 NEBAs über Σ . Dann gibt es einen NEBA A_3 mit $L(A_1) = L(A_1) \cap L(A_2)$. folgt direkt aus der Abgeschlossenheit unter ∪ und = $L_1 \cap L_2 = \overline{L_1 \cup L_2}$

Lemma 2.18

Seien A_1, A_2 NEBAs über Σ . Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cap L(A_2)$.

... folgt direkt aus der Abgeschlossenheit unter ∪ und ¯:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

16:40

Abgeschlossenheit unter Schnitt

Lemma 2.18

Seien A_1, A_2 NEBAs über Σ . Dann gibt es einen NEBA A_3 mit $L(A_3) = L(A_1) \cap L(A_2)$.

... folgt direkt aus der Abgeschlossenheit unter ∪ und ¯:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Top-down-BAs

Alternative: Konstruktion des Produktautomaten wie für NEAs (vermeidet exponentielle "Explosion")



16:40

2019-02-01

Abschlusseig.

Abgeschlossenheit unter Verkettungsoperationen

Randbemerkung:

Man kann Analoga zu • und * für Baumsprachen definieren:

Seien L, L_1, L_2 Baumsprachen.

Bezeichne Con(L) die Menge aller Kontexte, die man aus Bäumen in L erhält, indem man ein Blattsymbol durch x ersetzt.

•
$$L_1L_2 = \{C[T] \mid T \in L_1, C \in Con(L_2)\}$$

•
$$L^* = \{ C_1[C_2[\dots [C_n[T]] \dots]] \mid T \in L, C_1, \dots, C_n \in Con(L), n \ge 0 \}$$

Abgeschlossenheit unter •, * kann man dann wie für NEAs zeigen, aber mit mehr technischem Aufwand (Eliminierung ε -Kanten ...)

16:41 bis 16:44; 5min Pause, 1min Puffer \rightarrow 16:50

Beachte: Bäume sind nach Def. nichtleer. Deshalb schließt die Def. von L* den leeren Baum auch aus.

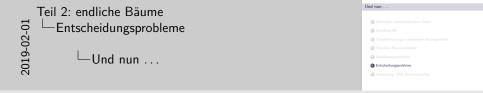
-Abgeschlossenheit unter Verkettungsoperationen

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Und nun ...

Motivation

- Motivation: semistrukturierte Daten
- 2 Grundbegriffe
- 3 Charakterisierungen erkennbarer Baumsprache
- 4 Top-down-Baumautomater
- 5 Abschlusseigenschaften
- 6 Entscheidungsprobleme
- Anwendung: XMI -Schemasprache



XML

Motivation Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

d.h. $LP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = \emptyset\}$ (analog für DEBAs)

Das Leerheitsproblem

Eingabe: NEBA (oder DEBA) \mathcal{A}

Frage: Ist $L(A) = \emptyset$?

d.h. $LP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = \emptyset\}$ (analog für DEBAs)

Teil 2: endliche Bäume -Entscheidungsprobleme

Das Leerheitsproblem

16:50

Satz 2.19

Grundbegriffe

Frage: Ist $L(A) = \emptyset$?

Das Leerheitsproblem

Abschlusseig.

Eingabe: NEBA (oder DEBA) \mathcal{A}

d. h. $LP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = \emptyset\}$ (analog für DEBAs)

LP_{NEBA} und LP_{DEBA} sind entscheidbar und P-vollständig.

Das Leerheitsproblem

Eingabe: NEBA (oder DEBA) A

16:50

Frage: Ist $L(A) = \emptyset$?

Satz 2.19

Beweis.

Das Leerheitsproblem

LP_{NEBA} und LP_{DEBA} sind entscheidbar und P-vollständig.

Entscheidbarkeit in Polyzeit analog zu NEAs:

d. h. $\mathsf{LP}_{\mathsf{NEBA}} = \{ \mathcal{A} \mid \mathcal{A} \; \mathsf{NEBA}, \; \mathcal{L}(\mathcal{A}) = \emptyset \}$ (analog für DEBAs)

prüfe, ob ein akz. Zustand erreichbar ist (nächste Folie)

Eingabe: NEBA (oder DEBA) \mathcal{A}

Das Leerheitsproblem

Eingabe: NEBA (oder DEBA) A

-Das Leerheitsproblem

16:50

Automatentheorie u. i. A. WiSe 2018/19

Frage: Ist $L(A) = \emptyset$?

Eingabe: NEBA (oder DEBA) \mathcal{A}

Das Leerheitsproblem

Eingabe: NEBA (oder DEBA) A

d.h. LPscna = $\{A \mid A \text{ NEBA}, L(A) = \emptyset\}$ (analog für DEBAs

Entscheidungsprobleme

-Das Leerheitsproblem

- d. h. $\mathsf{LP}_{\mathsf{NEBA}} = \{ \mathcal{A} \mid \mathcal{A} \; \mathsf{NEBA}, \; \mathcal{L}(\mathcal{A}) = \emptyset \}$ (analog für DEBAs)
- Satz 2.19

LP_{NEBA} und LP_{DEBA} sind entscheidbar und P-vollständig.

Beweis.

- Entscheidbarkeit in Polyzeit analog zu NEAs: prüfe, ob ein akz. Zustand erreichbar ist (nächste Folie)
- P-Härte: Reduktion von "Solvable Path Systems" $(\approx$ Erreichbarkeit in Hypergraphen mit ternärer Kantenrelation), siehe [Comon et al. 2008, Exercise 1.19]

Motivation Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

· Prüfe, ob ein akzeptierender Zustand darin ist

Das Leerheitsproblem

Polynomialzeitalgorithmus:

- Berechne Menge der erreichbaren Zustände
- Prüfe, ob ein akzeptierender Zustand darin ist.

Teil 2: endliche Bäume Entscheidungsprobleme ☐ Das Leerheitsproblem

16:53 bis 17:18

2019-02-01

"in Polyzeit:" weil R monoton wachsend und durch Q beschränkt!

Letzte Äquivalenz: erkannte Sprache leer gdw. kein akz. Zust. erreichbar

XML

2019-02-01

Sei $A = (O \Sigma \land F)$

☐ Das Leerheitsproblem

Das Leerheitsproblem

Polynomialzeitalgorithmus:

- Berechne Menge der erreichbaren Zustände
- Prüfe, ob ein akzeptierender Zustand darin ist.

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

Konstruieren Menge $R \subseteq Q$ wie folgt:

- $R := \{ q \mid a \to q \in \Delta \text{ für ein } a \in \Sigma_0 \}$
- Wenn es $q_1, \ldots, q_m \in R$ und $a \in \Sigma_m$ gibt mit $a(q_1,\ldots,q_m)\to q\in\Delta$ und $q\notin R$, dann $R:=R\cup\{q\}$.
- Wiederhole letzten Schritt, bis sich R nicht mehr ändert.

16:53 bis 17:18

"in Polyzeit:" weil R monoton wachsend und durch Q beschränkt!

Letzte Äquivalenz: erkannte Sprache leer gdw. kein akz. Zust. erreichbar

Top-down-BAs

Abschlusseig.

Entscheid.-probl

Das Leerheitsproblem

u Berechne Menge der erreichbaren Zustände Prüfe, ob ein akzeptierender Zustand darin ist

Sei $A = (O \Sigma \land F)$

Konstruieren Menge R C Ø wie folgt: • $R := \{q \mid a \rightarrow q \in \Delta \text{ für ein } a \in \Sigma_n\}$

a Wenn es $q_1, \dots, q_m \in R$ und $a \in \Sigma_m$ gibt mit $a(\sigma_1, \dots, \sigma_m) \rightarrow \sigma \in \Delta \text{ und } \sigma \notin R, \text{ dann } R := R \cup \{\sigma\}.$ w Wiederhole letzten Schritt, bis sich R nicht mehr ändert Leicht zu sehen: Berechnung endet nach < |O| vielen Schritten → R ist in Polyzeit berechenbar

Das Leerheitsproblem

Polynomialzeitalgorithmus:

- Berechne Menge der erreichbaren Zustände
- Prüfe, ob ein akzeptierender Zustand darin ist.

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

Konstruieren Menge $R \subseteq Q$ wie folgt:

- $R := \{ q \mid a \to q \in \Delta \text{ für ein } a \in \Sigma_0 \}$
- Wenn es $q_1, \ldots, q_m \in R$ und $a \in \Sigma_m$ gibt mit $a(q_1,\ldots,q_m)\to g\in\Delta$ und $g\notin R$, dann $R:=R\cup\{g\}$.
- Wiederhole letzten Schritt, bis sich R nicht mehr ändert.

Leicht zu sehen: Berechnung endet nach < |Q| vielen Schritten \rightarrow R ist in Polyzeit berechenbar.

16:53 bis 17:18

2019-02-01

Teil 2: endliche Bäume

Entscheidungsprobleme

☐ Das Leerheitsproblem

"in Polyzeit:" weil R monoton wachsend und durch Q beschränkt!

Letzte Äquivalenz: erkannte Sprache leer gdw. kein akz. Zust. erreichbar

Top-down-BAs Abschlusseig.

Entscheid.-probl

Teil 2: endliche Bäume Entscheidungsprobleme

Sei $A = (O, \Sigma, \Lambda, F)$ Konstruieren Menge R C Ø wie folgt: • $R := \{q \mid a \rightarrow q \in \Delta \text{ für ein } a \in \Sigma_n\}$ a Wenn es $q_1, \dots, q_m \in R$ und $a \in \Sigma_m$ gibt mit $a(\sigma_1, \dots, \sigma_m) \rightarrow \sigma \in \Delta \text{ und } \sigma \notin R, \text{ dann } R := R \cup \{\sigma\}.$ w Wiederhole letzten Schritt, bis sich R nicht mehr ändert Leicht zu sehen: Berechnung endet nach < |O| vielen Schritten ~ R ist in Polyzeit berechenbar

Noch zu zeigen: $L(A) = \emptyset$ adw. $R \cap F = \emptyset$ T2.10

u Berechne Menge der erreichbaren Zustände

Prüfe, ob ein akzeptierender Zustand darin ist

Das Leerheitsproblem

☐ Das Leerheitsproblem

Polynomialzeitalgorithmus:

- Berechne Menge der erreichbaren Zustände
- Prüfe, ob ein akzeptierender Zustand darin ist.

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

Konstruieren Menge $R \subseteq Q$ wie folgt:

- $R := \{ q \mid a \to q \in \Delta \text{ für ein } a \in \Sigma_0 \}$
- Wenn es $q_1, \ldots, q_m \in R$ und $a \in \Sigma_m$ gibt mit $a(q_1,\ldots,q_m)\to g\in\Delta$ und $g\notin R$, dann $R:=R\cup\{g\}$.
- Wiederhole letzten Schritt, bis sich R nicht mehr ändert.

Leicht zu sehen: Berechnung endet nach < |Q| vielen Schritten

 \rightarrow R ist in Polyzeit berechenbar.

Noch zu zeigen: $L(A) = \emptyset$ gdw. $R \cap F = \emptyset$

T 2.10

16:53 bis 17:18

2019-02-01

"in Polyzeit:" weil R monoton wachsend und durch Q beschränkt!

Letzte Äquivalenz: erkannte Sprache leer gdw. kein akz. Zust. erreichbar

Grundbegriffe

Top-down-BAs

Abschlusseig.

(analog f. DEBAs)

Charakt.

Eingabe: NEBA (oder DEBA) A, Baum T über Σ

Frage: Ist $T \in L(A)$?

d. h.
$$WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$$

Teil 2: endliche Bäume -Entscheidungsprobleme

Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zugehörigkeitsproblem d.h. WParna = $\{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$ (analog f. DEBAs

17:18 bis 17:22

"alias": Bäume sind eben keine Wörter. Werde trotzdem weiter "WP" benutzen.

LOGCFL:

Probleme, die sich in logspace aufs Wortproblem für kfS reduzieren lassen

LOGDCFL:

dito, aber Red. auf determinist. kfS (d. h. durch DPDA erkennbar)

$$\mathsf{L} \subseteq \frac{\mathsf{LogDCFL}}{\mathsf{NL}} \subseteq \mathsf{LogCFL} \subseteq \mathsf{P}$$

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Das Wortproblem alias Zugehörigkeitsproblem

Charakt.

Eingabe: NEBA (oder DEBA) A, Baum T über Σ

Frage: Ist $T \in L(A)$?

d.h.
$$WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$$
 (analog f. DEBAs)

Satz 2.20

WP_{NEBA} und WP_{DEBA} sind entscheidbar und in P.

Teil 2: endliche Bäume Entscheidungsprobleme

Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zugehörigkeitsproblem WP_{NEBA} und WP_{DEBA} sind entscheidbar und in P.

17:18 bis 17:22

"alias": Bäume sind eben keine Wörter. Werde trotzdem weiter "WP" benutzen.

LOGCFL:

Probleme, die sich in logspace aufs Wortproblem für kfS reduzieren lassen

LOGDCFL:

dito, aber Red. auf determinist. kfS (d. h. durch DPDA erkennbar)

$$\mathsf{L} \subseteq \frac{\mathsf{LogDCFL}}{\mathsf{NL}} \subseteq \mathsf{LogCFL} \subseteq \mathsf{P}$$

Frage: Ist $T \in L(A)$?

Das Wortproblem alias Zugehörigkeitsproblem

Eingabe: NEBA (oder DEBA) A, Baum T über Σ

d.h. $WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$

WP_{NFBA} und WP_{DFBA} sind entscheidbar und in P.

Beweis. analog zu NEAs, per Reduktion zum LP:

Satz 2.20

(analog f. DEBAs)

Teil 2: endliche Bäume Entscheidungsprobleme

Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zugehörigkeitsproblem d.h. $WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$ (analog f. DEBA) WPNESS und WPNESS sind entscheidbar und in P. wohei $A = \sin DERA mit I (A =) = IT ist (konstruiere selbst!)$

17:18 bis 17:22

"alias": Bäume sind eben keine Wörter. Werde trotzdem weiter "WP" benutzen.

LOGCFL:

Probleme, die sich in logspace aufs Wortproblem für kfS reduzieren lassen

LOGDCFL:

dito, aber Red. auf determinist. kfS (d. h. durch DPDA erkennbar)

$$\mathsf{L} \;\subseteq\; \frac{\mathsf{Log}\mathsf{DCFL}}{\mathsf{NII}} \;\subseteq\; \mathsf{Log}\mathsf{CFL} \;\subseteq\; \mathsf{P}$$

 $T \in L(A)$ gdw. $L(A) \cap L(A_T) \neq \emptyset$,

wobei A_T ein DEBA mit $L(A_T) = \{T\}$ ist (konstruiere selbst!)

Das Wortproblem alias Zugehörigkeitsproblem

Eingabe: NEBA (oder DEBA) A, Baum T über Σ

Frage: Ist $T \in L(A)$?

d. h.
$$WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$$
 (analog f. DEBAs)

Satz 2.20

WP_{NEBA} und WP_{DEBA} sind entscheidbar und in P.

Beweis. analog zu NEAs, per Reduktion zum LP:

$$T \in L(A)$$
 gdw. $L(A) \cap L(A_T) \neq \emptyset$,

wobei A_T ein DEBA mit $L(A_T) = \{T\}$ ist (konstruiere selbst!)

WP_{NFBA} ist LOGCFL-vollständig. (zwischen **NL** und P) WP_{DEBA} ist in LOGDCFL. (Genaue Komplexität ist offen!) WP_{DFTDBA} ist **L**-vollständig.

Teil 2: endliche Bäume Entscheidungsprobleme Das Wortproblem alias Zugehörigkeitsproblem Das Wortproblem alias Zugehörigkeitsproblem d.h. $WP_{NEBA} = \{(A, T) \mid A \text{ NEBA}, T \in L(A)\}$ (analog f. DEBA WPNESS und WPNESS sind entscheidbar und in P. nobel A_T ein DEBA mit $L(A_T) = \{T\}$ ist (konstruiere selbst!)

WPNena ist LOGCFL-vollständig. (zwischen NL und P)

WPocznoa ist L-vollständie

17:18 bis 17:22

"alias": Bäume sind eben keine Wörter. Werde trotzdem weiter "WP" benutzen.

LOGCFL:

Probleme, die sich in logspace aufs Wortproblem für kfS reduzieren lassen

LOGDCFL:

dito, aber Red. auf determinist. kfS (d. h. durch DPDA erkennbar)

$$\mathsf{L} \subseteq \mathsf{Log}\mathsf{DCFL} \subseteq \mathsf{Log}\mathsf{CFL} \subseteq \mathsf{P}$$

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

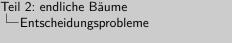
d.h. $AP_{MEDA} = \{(A_1, A_2) \mid A_1, A_2 \text{ NEBAs}, L(A_1) = L(A_2)\}$ etc

Das Äquivalenzproblem

Eingabe: NEBAs (oder DEBAs) A_1, A_2

Frage: Ist $L(A_1) = L(A_2)$?

d. h.
$$\ddot{\mathsf{AP}}_{\mathsf{NEBA}} = \{ (\mathcal{A}_1, \mathcal{A}_2) \mid \mathcal{A}_1, \mathcal{A}_2 \; \mathsf{NEBAs}, \; L(\mathcal{A}_1) = L(\mathcal{A}_2) \} \; \; \mathsf{etc}.$$



□Das Äguivalenzproblem

Untere Schranke für DEBAs: weiß nicht ...

17:22 bis 17:25

 \triangle = symmetrische Differenz zweier Mengen; ausdrückbar mittels \cup , \cap , $\bar{\cdot}$ \rightarrow Abschlusseigenschaften!

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Das Äguivalenzproblem

d.h. $AP_{MEBA} = \{(A_1, A_2) \mid A_1, A_2 \text{ NEBAs}, L(A_1) = L(A_2)\}$ etc

AP_{NEDA} und AP_{DEDA} sind entscheidbar. AP_{NEDA} ist ExpTime-vollständig; AP_{DEDA} ist in P.

Das Äquivalenzproblem

Eingabe: NEBAs (oder DEBAs) A_1, A_2

Frage: Ist $L(A_1) = L(A_2)$?

d. h.
$$\mathsf{\ddot{A}P}_{\mathsf{NEBA}} = \{(\mathcal{A}_1, \mathcal{A}_2) \mid \mathcal{A}_1, \mathcal{A}_2 \; \mathsf{NEBAs}, \; \mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)\} \;\; \mathsf{etc}.$$

Satz 2.21

ÄP_{NEBA} und ÄP_{DEBA} sind entscheidbar.

ÄP_{NEBA} ist **ExpTime**-vollständig; ÄP_{DEBA} ist in P.

17:22 bis 17:25

Teil 2: endliche Bäume

Entscheidungsprobleme

□ Das Äguivalenzproblem

Untere Schranke für DEBAs: weiß nicht ...

 \triangle = symmetrische Differenz zweier Mengen; ausdrückbar mittels \cup , \cap , $\bar{\cdot}$ \rightarrow Abschlusseigenschaften! lotivation

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Das Äquivaler

Das Äquivalenzproblem

Eingabe: NEBAs (oder DEBAs) A_1, A_2

Charakt.

Frage: Ist $L(A_1) = L(A_2)$?

d. h. $\mathsf{\ddot{A}P}_{\mathsf{NEBA}} = \{(\mathcal{A}_1, \mathcal{A}_2) \mid \mathcal{A}_1, \mathcal{A}_2 \; \mathsf{NEBAs}, \; \mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)\} \;\; \mathsf{etc}.$

Satz 2.21

ÄP_{NEBA} und ÄP_{DEBA} sind entscheidbar.

 AP_{NEBA} ist **ExpTime**-vollständig; AP_{DEBA} ist in P.

Beweis.

• Entscheidbarkeit analog zu NEAs, per Reduktion zum LP:

$$L(A_1) = L(A_2)$$
 gdw. $L(A_1) \triangle L(A_2) = \emptyset$

Teil 2: endliche Bäume
—Entscheidungsprobleme

└─Das Äquivalenzproblem

Das Aquivalentzproblem
Englac REBAS (oder DEBA) A_1 , A_2
Frage int $(A_1) = (L(\lambda)^2)$
d. h. $M_{\rm REBA} = \{(L_1) \times (1 + L_2) \times (1 + L_3) \times (1 + L_4)\}$ etc.
Since 22:
Amena, and APreas, one destination:
APreas, int Explime-ordinating. APreas, int in P.
Brook.
6. Enchandituriant analog in MEAn, par Rediction zum LP: $(L(\lambda) = L(\Lambda))$
gin. $(L(\lambda)) \times (L(\Lambda)) = \emptyset$

17:22 bis 17:25

 $\triangle = \text{symmetrische Differenz zweier Mengen;} \\ \text{ausdr\"{u}ckbar mittels } \cup, \cap, \bar{\boldsymbol{\cdot}} \quad \leadsto \text{Abschlusseigenschaften!} \\$

Untere Schranke für DEBAs: weiß nicht ...

Entscheidungsprobleme

Teil 2: endliche Bäume

—Das Äguivalenzproblem

AParens und APorens sind enterheidhar APNEDA ist ExpTime-vollständig: APnena ist in P . Entscheidbarkeit analog zu NEAs, per Reduktion zum LP $L(A_1) = L(A_2)$ sides, $L(A_1) \wedge L(A_2) = \emptyset$ a obere Schranken: Automat für $L(A_1) \triangle L(A_2)$ ist exponentiell in der Größe der Eingabe-NEBAs / polynomiell für DEBAs

Eingabe: NEBAs (oder DEBAs) A_1, A_2

Charakt.

Frage: Ist $L(A_1) = L(A_2)$?

d.h. $\mathsf{\ddot{A}P}_{\mathsf{NFRA}} = \{(\mathcal{A}_1, \mathcal{A}_2) \mid \mathcal{A}_1, \mathcal{A}_2 \mid \mathsf{NEBAs}, L(\mathcal{A}_1) = L(\mathcal{A}_2)\}$ etc.

Satz 2.21

ÄP_{NEBA} und ÄP_{DEBA} sind entscheidbar.

ÄP_{NFRA} ist ExpTime-vollständig; ÄP_{DFRA} ist in P.

Beweis.

• Entscheidbarkeit analog zu NEAs, per Reduktion zum LP:

$$L(A_1) = L(A_2)$$
 gdw. $L(A_1) \triangle L(A_2) = \emptyset$

• obere Schranken: Automat für $L(A_1) \triangle L(A_2)$ ist exponentiell in der Größe der Eingabe-NEBAs / polynomiell für DEBAs

17:22 bis 17:25

 \triangle = symmetrische Differenz zweier Mengen; ausdrückbar mittels \cup , \cap , $\bar{\cdot}$ \rightarrow Abschlusseigenschaften!

Untere Schranke für DFBAs: weiß nicht

Top-down-BAs Charakt.

Abschlusseig.

Entscheid.-probl.

Das Äguivalenzproblem

Eingabe: NEBAs (oder DEBAs) A_1, A_2

Frage: Ist $L(A_1) = L(A_2)$?

d. h.
$$\mbox{\Bar{A}P}_{\mbox{\scriptsize NEBA}} = \{(\mathcal{A}_1,\mathcal{A}_2) \mid \mathcal{A}_1,\mathcal{A}_2 \mbox{\ NEBAs}, \ \mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)\}$$
 etc.

Satz 2.21

ÄP_{NEBA} und ÄP_{DEBA} sind entscheidbar.

ÄP_{NFRA} ist ExpTime-vollständig; ÄP_{DFRA} ist in P.

Beweis.

• Entscheidbarkeit analog zu NEAs, per Reduktion zum LP:

$$L(A_1) = L(A_2)$$
 gdw. $L(A_1) \triangle L(A_2) = \emptyset$

- obere Schranken: Automat für $L(A_1) \triangle L(A_2)$ ist exponentiell in der Größe der Eingabe-NEBAs / polynomiell für DEBAs
- ExpTime-Härte: Reduktion vom Universalitätsproblem (F. 59)

d.h. $AParma = \{(A_1, A_2) \mid A_1, A_2 \text{ NEBAs}, L(A_1) = L(A_2)\}$ etc. AParens und APorens sind enterheidhar APNEDA ist ExpTime-vollständig: APnena ist in P . Entscheidbarkeit analog zu NEAs, per Reduktion zum LP $L(A_1) = L(A_2)$ adw. $L(A_1) \wedge L(A_2) = \emptyset$ a obere Schranken: Automat für $L(A_1) \triangle L(A_2)$ ist exponentiell in der Größe der Eingabe-NEBAs / polynomiell für DEBAs

■ ExpTime-Härte: Reduktion vom Universalitätsproblem (F.59)

Das Äguivalenzproblem

Frage: lst $L(A_1) = L(A_2)$?

Eingabe: NEBAs (oder DEBAs) A1. A5

17:22 bis 17:25

 \triangle = symmetrische Differenz zweier Mengen; ausdrückbar mittels \cup , \cap , $\bar{\cdot}$ \rightarrow Abschlusseigenschaften!

Untere Schranke für DFBAs: weiß nicht

—Das Äguivalenzproblem

Motivation Grundbegriffe

Charakt.

Das Universalitätsproblem

Top-down-BAs

Frage: Ist $L(A) = T(\Sigma)$? $(T(\Sigma) = \{T \mid T \text{ ist ein } \Sigma\text{-Baum}\})$ d. h. $UP_{NEBA} = \{ A \mid A \text{ NEBA}, L(A) = T(\Sigma) \}$ (analog f. DEBAs)

Eingabe: NEBA (oder DEBA) \mathcal{A}

Teil 2: endliche Bäume -Entscheidungsprobleme

☐ Das Universalitätsproblem

Das Universalitätsproblem

d.h. $UP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = T(\Sigma)\}$ (analog f. DEBAs)

Grundbegriffe

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

Das Universalitätsproblem

Eingabe: NEBA (oder DEBA) \mathcal{A}

Frage: Ist
$$L(A) = T(\Sigma)$$
? $(T(\Sigma) = \{T \mid T \text{ ist ein } \Sigma\text{-Baum}\})$

d. h. $UP_{NFBA} = \{ A \mid A \text{ NEBA}, L(A) = T(\Sigma) \}$ (analog f. DEBAs)

Satz 2.22

UP_{NEBA} und UP_{DEBA} sind entscheidbar.

UP_{NEBA} ist **ExpTime**-vollständig; UP_{DEBA} ist in P.

Teil 2: endliche Bäume Entscheidungsprobleme

☐ Das Universalitätsproblem

Das Universalitätsproblem

Eingabe: NEBA (oder DEBA) A

Frage: Ist
$$L(A) = T(\Sigma)$$
?

Frage: Ist
$$L(A) = T(\Sigma)$$
? $(T(\Sigma) = \{T \mid T \text{ ist ein } \Sigma\text{-Baum}\})$

d. h.
$$\mathsf{UP}_\mathsf{NEBA} = \{\mathcal{A} \mid \mathcal{A} \; \mathsf{NEBA}, \; \mathit{L}(\mathcal{A}) = \mathcal{T}(\Sigma)\}$$
 (analog f. DEBAs)

Satz 2.22

UP_{NEBA} und UP_{DEBA} sind entscheidbar.

UP_{NFBA} ist ExpTime-vollständig; UP_{DFBA} ist in P.

Beweis:

• Entscheidbarkeit & obere Schranken per Red. zum ÄP:

$$\label{eq:loss_loss} \textit{L}(\mathcal{A}) = \mathcal{T}(\Sigma) \;\; \text{gdw.} \;\; \textit{L}(\mathcal{A}) = \textit{L}(\mathcal{A}_{\Sigma}),$$

wobei A_{Σ} DEBA mit $L(A_{\Sigma}) = \mathcal{T}(\Sigma)$ (konstruiere selbst!)

Teil 2: endliche Bäume Entscheidungsprobleme

□Das Universalitätsproblem

· Entscheidbarkeit & obere Schranken per Red. zum AP: $L(A) = T(\Sigma)$ gdw. $L(A) = L(A_{\Sigma})$ wobei A_{Γ} DEBA mit $L(A_{\Gamma}) = T(\Sigma)$ (konstruiere selbst!

d.h. $UP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = T(\Sigma)\}$ (analog f. DEBAs UPNERA und UPDERA sind entscheidbar. UPNEBA ist ExpTime-vollständig: UPDEBA ist in P.

Frage: Ist
$$L(A) = T(\Sigma)$$
? $(T(\Sigma) = \{T \mid T \text{ ist ein } \Sigma\text{-Baum}\})$

d.h.
$$\mathsf{UP}_\mathsf{NEBA} = \{ \mathcal{A} \mid \mathcal{A} \; \mathsf{NEBA}, \; \mathit{L}(\mathcal{A}) = \mathcal{T}(\Sigma) \}$$
 (analog f. DEBAs)

Satz 2.22

UP_{NEBA} und UP_{DEBA} sind entscheidbar.

UP_{NEBA} ist ExpTime-vollständig; UP_{DEBA} ist in P.

Beweis:

• Entscheidbarkeit & obere Schranken per Red. zum ÄP:

$$L(A) = \mathcal{T}(\Sigma)$$
 gdw. $L(A) = L(A_{\Sigma})$,

wobei A_{Σ} DEBA mit $L(A_{\Sigma}) = \mathcal{T}(\Sigma)$ (konstruiere selbst!)

• ExpTime-Härte: Red. vom WP für lin. platzbeschränkte alternierende TM (s. a. [Comon et al. 2008, § 1.7])

Teil 2: endliche Bäume Entscheidungsprobleme

□ Das Universalitätsproblem

UPNERA ist ExpTime-vollständig: UPDERA ist in P. · Entscheidbarkeit & obere Schranken per Red. zum AP $L(A) = T(\Sigma)$ gdw. $L(A) = L(A_{\Sigma})$ wobei A_{Γ} DEBA mit $L(A_{\Gamma}) = T(\Sigma)$ (konstruiere selbst alternierende TM (s. a. [Comon et al. 2008, 61.7])

d.h. $UP_{NEBA} = \{A \mid A \text{ NEBA}, L(A) = T(\Sigma)\}$ (analog f. DEBAs UPwrna und UPnrna sind entscheidbar

Das Universalitätsproblem

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Überblick Entscheidungsprobleme für NEBAs/DEBAs

		für DEBAs	für NEBAs
Problem	entscheidbar?	effizient lösbar?	effizient lösbar?
LP	✓	✓	✓
WP	✓	✓	✓
ÄP	✓	✓	× *
UP	\checkmark	\checkmark	X *

^{*} nachweislich! (da $ExpTime \neq P$)

Teil 2: endliche Bäume
LEntscheidungsprobleme

Überblick Entscheidungsprobleme für NEBAs/DEBAs

Überblick Entscheidungsprobleme für NEBAs/DEBAs

17:27 bis 17:29 → 1 min Reserve
Hier nochmal für Euch als Überblick

Nächstes Mal: XML, Kapitel zuende

Zum Literaturverzeichnis blättern

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Und nun ...

Motivation

- 2 Grundbegriffe

- 6 Abschlusseigenschaften
- Anwendung: XML-Schemasprachen



Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Zur Erinnerung: semistrukturierte Daten

Daten werden in Entitäten (Einheiten) zusammengefasst w Markierung von Entitäten durch Tags

- Bildung von Hierarchien
- a Gruppieren ähnlicher Entitäten
- verschiedene (oder keine) Attribute haben

Reihenfolge der Attribute kann eine Rolle spielen -Zur Erinnerung: semistrukturierte Daten

8:30

2019-02-01

Teil 2: endliche Bäume

-Anwendung: XML-Schemasprachen

- Zur Erinnerung: semistrukturierte Daten
 - Daten werden in Entitäten (Einheiten) zusammengefasst
 - Markierung von Entitäten durch Tags
 - Bildung von Hierarchien
 - Gruppieren ähnlicher Entitäten
 - Entitäten derselben Gruppe können verschiedene (oder keine) Attribute haben
 - Reihenfolge der Attribute kann eine Rolle spielen

Grundbegriffe

Entscheid.-probl

Top-down-BAs

Abschlusseig.

Charakt. Zur Erinnerung: semistrukturierte Daten

Daten werden in Entitäten (Einheiten) zusammengefasst

- Markierung von Entitäten durch Tags
- Bildung von Hierarchien
- Gruppieren ähnlicher Entitäten
- Entitäten derselben Gruppe können verschiedene (oder keine) Attribute haben
- Reihenfolge der Attribute *kann* eine Rolle spielen (Mengen oder Listen z. B. von Telefonnummern?)

Person: {Name: {VN: "Thomas", NN: "Schneider"}, Telnr: 64432, Beispiel: Telnr: 43776243, Email: "ts@informatik..."}

Zur Erinnerung: semistrukturierte Daten Teil 2: endliche Bäume Daten werden in Entitäten (Einheiten) zusammengefasst -Anwendung: XML-Schemasprachen Bildung von Hierarchien a Gruppieren ähnlicher Entitäten -Zur Erinnerung: semistrukturierte Daten

8:30

2019-02-01

- verschiedene (oder keine) Attribute haben
- Reihenfolge der Attribute kann eine Rolle spielen (Mengen oder Listen z. B. von Telefonnummern?)

Person: {Name: {VN: "Thomas", NN: "Schneider" Telnr: 43776243. Enail: "te@informatik . . . "}

Motivation

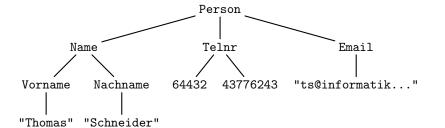
Grundbegriffe Charakt. Top-down-BAs Abschlusseig.

XML

Entscheid.-probl.

Repräsentation im Baum

Person: {Name: {VN: "Thomas", NN: "Schneider"}, Telnr: 64432, Telnr: 43776243, Email: "ts@informatik..."}



Repräsentation im Baum Teil 2: endliche Bäume Person: {Name: {VN: "Thomas", NN: "Schneider" Telnr: 64432. - Anwendung: XML-Schemasprachen Enail: "te@informatik..."} Repräsentation im Baum

Motivation

<conference>

<talk>

</talk> <talk>

</talk> </session>

<session> </session> </track> <track> </track>

<break> Coffee </preak>

<track> <session>

Grundbegriffe

<chair> F. Angorn </chair>

Charakt.

<title> The Pushdown Hierarchy </title> <speaker> D.J. Gaugal </speaker>

<title> Trees Everywhere </title> <authors> B. Aum, T. Rees </authors> Abschlusseig.

Entscheid.-probl.

aus Tree Automata Techni-

ques and Applications, S. 230

- Anwendung: XML-Schemasprachen

-Größeres Bsp.: XML-Dokument für Konferenzprogramm



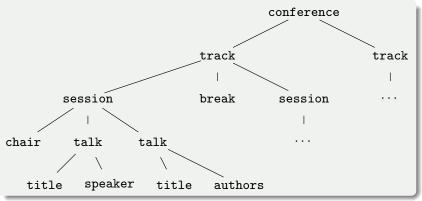
8:33

2019-02-01

</conference>

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Zugehöriger Baum



aus Tree Automata Techniques and Applications, S. 230

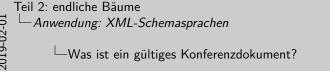
▶ Ab jetzt: wir beschreiben nur die Struktur, ignorieren die Daten

8:35

XML

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML Tol. 20 and links Discussed.

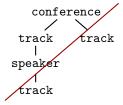
Was ist ein gültiges Konferenzdokument?

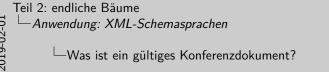


nference k track er

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML Tail 2. and links Päyres

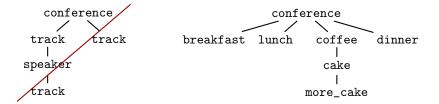
Was ist ein gültiges Konferenzdokument?

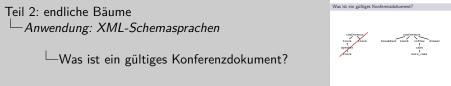




Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

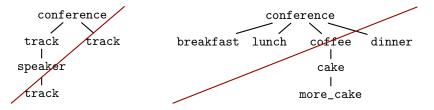
Was ist ein gültiges Konferenzdokument?

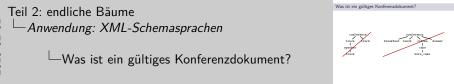




Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Was ist ein gültiges Konferenzdokument?





Abschlusseig

Entscheid.-probl

Top-down-BAs Mögliche Anforderungen an gültige Konferenzdokumente

- Eine Konferenz kann in mehrere Blöcke (Tracks) geteilt sein.
- Jeder Block (oder die Konf. selbst, wenn sie keine Blöcke hat) ist in Sitzungen aufgeteilt.
- Jede Sitzung hat einen oder mehrere Vorträge.
- Jede Sitzung wird von einer Person geleitet (Chair).
- Jeder Vortrag hat einen Titel und
 - Autor_innen (falls es sich um einen Konferenzbeitrag handelt)
 - oder Vortragende n (falls es ein eingeladener Vortrag ist).
- Zwischen den Sitzungen kann es Pausen geben.

2019-02-01

Teil 2: endliche Bäume

-Anwendung: XML-Schemasprachen

Konferenzdokumente

–Mögliche Anforderungen an gültige

Motivation

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Gültige Dokumente als Baumsprachen!

Die gelisteten Anforderungen beschreiben eine Baumsprache über dem Alphabet {conference, track, session, ... }.

Eine solche Beschreibung wird auch **Schema** genannt.

Ein Dokument ist gültig für ein Schema, wenn sein Baum zur Baumsprache des Schemas gehört. Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Gültige Dokumente als Baumsprachen!

8:40

2019-02-01

Top-down-BAs

Entscheid.-probl.

XML

Abschlusseig.

Die gelisteten Anforderungen beschreiben eine Baumsprache über dem Alphabet {conference, track, session, ... }.

Eine solche Beschreibung wird auch **Schema** genannt.

Ein Dokument ist gültig für ein Schema, wenn sein Baum zur Baumsprache des Schemas gehört.

Ziele dieses Abschnitts

- Vorstellen von XML-Schemasprachen
- Diskutieren von Verbindungen zur Automatentheorie
- Untersuchen der Ausdrucksstärke von Schemasprachen
- und ihre Entscheidungsprobleme

Teil 2: endliche Bäume 2019-02-01 Anwendung: XML-Schemasprachen -Gültige Dokumente als Baumsprachen!

8:40

Gültige Dokumente als Baumsprachen!

Die gelisteten Anforderungen beschreiben eine Baumsprache über dem Alphabet (conference, track, session...)

Eine solche Beschreibung wird auch Schema genannt

Ein Dokument ist gültig für ein Schema. wenn sein Baum zur Baumsprache des Schemas gehör

Vorstellen von XML-Schemaspracher

 Diskutieren von Verbindungen zur Automatentheorie Untersuchen der Ausdrucksstärke von Schemasprachen • und ihre Entscheidungsprobleme

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML Entscheidungsprobleme für XML-Schemasprachen Teil 2: endliche Bäume

-Entscheidungsprobleme für XML-Schemasprachen

Anwendung: XML-Schemasprachen

Fragen für XML-Dokumente und -Schemasprachen

Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bap.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?)

Entscheidungsprobleme für XML-Schemasprachen

Die bekannten Entscheidungsprobleme entsprechen natürlichen Fragen für XML-Dokumente und -Schemasprachen:

Zugehörigkeitsproblem

Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bsp.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?) 8:42

2019-02-01

Zum ÄP, "Vereinfachung von Schemata": Ist das Resultat der Vereinfachung äquivalent zum ursprünglichen Schema?

Top-down-BAs

Charakt.

Abschlusseig

Entscheid.-probl.

XML

Entscheidungsprobleme für XML-Schemasprachen

Die bekannten Entscheidungsprobleme entsprechen natürlichen Fragen für XML-Dokumente und -Schemasprachen:

Zugehörigkeitsproblem

Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bsp.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?)

Leerheitsproblem

Gibt es für ein gegebenes Schema gültige Dokumente? (Enthält das gegebene Schema keinen "Widerspruch"?)

Entscheidungsprobleme für XML-Schemasprachen

Teil 2: endliche Bäume Anwendung: XML-Schemasprachen

-Entscheidungsprobleme für XML-Schemasprachen

Gibt es für ein gegebenes Schema gültige Dokumente? (Enthält das gegebene Schema keinen "Widenspruch")

Fragen für XML-Dokumente und -Schemasprachen Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bap.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?)

8:42

2019-02-01

Zum ÄP, "Vereinfachung von Schemata": Ist das Resultat der Vereinfachung äquivalent zum ursprünglichen Schema?

Top-down-BAs Abschlusseig

Entscheid.-probl.

XML Entscheidungsprobleme für XML-Schemasprachen Teil 2: endliche Bäume

Anwendung: XML-Schemasprachen

-Entscheidungsprobleme für XML-Schemasprachen

Die hakannten Entscheidungsmohleme entswerhen natürliche Fragen für XML-Dokumente und -Schemasprachen Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bap.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?) Gibt es für ein gegebenes Schema gültige Dokumente? (Enthält das gegebene Schema keinen "Widenspruch"?

> Haben zwei Schemata dieselben gültigen Dokumente? (Wichtig bei der Vereinfachung von Schemata)

8:42

2019-02-01

Zum ÄP, "Vereinfachung von Schemata": Ist das Resultat der Vereinfachung äquivalent zum ursprünglichen Schema?

Die bekannten Entscheidungsprobleme entsprechen natürlichen Fragen für XML-Dokumente und -Schemasprachen:

Zugehörigkeitsproblem

Ist ein gegebenes Dokument gültig für ein gegebenes Schema? (im Bsp.: erfüllt ein gegebenes Konf.-dokument die Anforderungen?)

Leerheitsproblem

Gibt es für ein gegebenes Schema gültige Dokumente? (Enthält das gegebene Schema keinen "Widerspruch"?)

Äguivalenzproblem

Haben zwei Schemata dieselben gültigen Dokumente? (Wichtig bei der Vereinfachung von Schemata)

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Dokumenttypdefinitionen (DTDs)

DTDs sind ein Standard zur Beschreibung gültiger Dokumente

Eine DTD ist eine kontextfreie Grammatik (kfG), deren rechte Regelseiten reguläre Ausdrücke enthalten können

Ableitungsbäume der kfG bilden die Baumsprache, die durch die DTD bestimmt wird

Dokumenttypdefinitionen (DTDs) Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen DTDs sind ein Standard zur Beschreibung gültiger Dokuments Eine DTD ist eine kontextfreie Grammatik (kfG) deren rechte Regelseiten reguläre Ausdrücke enthalten können -Dokumenttypdefinitionen (DTDs) Ableitungsbäume der kfG bilden die Baumsprache

Motivation

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Beispiel-DTD für Konferenzdokumente

Charakt.

```
<!DOCTYPE CONFERENCE [</pre>
  <!ELEMENT conference (track+|(session,break?)+)>
  <!ELEMENT track
                        ((session,break?)+)>
  <!ELEMENT session
                        (chair,talk+)>
                        ((title,authors)|(title,speaker))>
  <!ELEMENT talk
                        (#PCDATA)>
  <!ELEMENT chair
  <!ELEMENT break
                        (#PCDATA)>
  <!ELEMENT title
                        (#PCDATA)>
  <!ELEMENT authors
                        (#PCDATA)>
  <!ELEMENT speaker
                        (#PCDATA)>
]>
                                                , ≜ Verkettung
```

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Beispiel-DTD für Konferenzdokumente

Beispiel-DTD für Konferenzdokumente

<!ELEMENT conference (track+|(session.break?)+)>

XML

Beispiel-DTD für Konferenzdokumente

Beispiel-DTD für Konferenzdokumente

<!DOCTYPE CONFERENCE [</pre> <!ELEMENT conference (track+|(session,break?)+)> <!ELEMENT track ((session, break?)+)> <!ELEMENT session (chair,talk+)> ((title,authors)|(title,speaker))> <!ELEMENT talk <!ELEMENT chair (#PCDATA)> <!ELEMENT break (#PCDATA)> <!ELEMENT title (#PCDATA)> <!ELEMENT authors (#PCDATA)> <!ELEMENT speaker (#PCDATA)> , ≜ Verkettung

Beschreibt Bäume, in denen z.B. jeder conference-Knoten

- ein oder mehrere track-Kinder hat oder
- ein oder mehrere session-Kinder hat. zwischen denen einzelne break-Geschwister stehen dürfen

8:45

```
<!DOCTYPE CONFERENCE [</pre>
  <!ELEMENT conference (track+|(session,break?)+)>
  <!ELEMENT track
                        ((session,break?)+)>
  <!ELEMENT session
                        (chair,talk+)>
                        ((title,authors)|(title,speaker))>
  <!ELEMENT talk
  <!ELEMENT chair
                        (#PCDATA)>
  <!ELEMENT break
                        (#PCDATA)>
  <!ELEMENT title
                        (#PCDATA)>
  <!ELEMENT authors
                        (#PCDATA)>
  <!ELEMENT speaker
                        (#PCDATA)>
                                                , ≜ Verkettung
```

Beschreibt Bäume, in denen z.B. jeder conference-Knoten

- ein oder mehrere track-Kinder hat oder
- ein oder mehrere session-Kinder hat, zwischen denen einzelne break-Geschwister stehen dürfen.

→ beliebige Stelligkeit!

Abschlusseig.

Beispiel-DTD für Konferenzdokumente Teil 2: endliche Bäume

- Anwendung: XML-Schemasprachen

(!ELEMENT conference (track+|(session.break?)+)>

Beispiel-DTD für Konferenzdokumente

Motivation Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Wir brauchen Symbole mit beliebiger Stelligkeit!

Wir brauchen Symbole mit beliebiger Stelligkeit u U: Menge von Symbolen ohne Stelligkeit $\bullet \Sigma = U \cup \sqcup_{i > 0} \Sigma_i$

Erweitern unser r-Alphabet:

- *U*: Menge von Symbolen ohne Stelligkeit
- $\Sigma = U \cup \bigcup_{i \geq 0} \Sigma_i$

8:48

2019-02-01

u U: Menee von Symbolen ohne Stelligkeit $\bullet \Sigma = U \cup \sqcup_{i > 0} \Sigma_i$ P ⊂ N* nichtleere endl. präfix-abseschlossene Mense

 \bullet Wenn $t(p) \in \Sigma_m$, dann $\{j \mid pj \in P\} = \{1, \dots, m\}$ **Q** Wenn t(p) ∈ U, dann $\{j \mid pj \in P\} = \{1, ..., k\}$

8:48

Wir brauchen Symbole mit beliebiger Stelligkeit!

Erweitern unser r-Alphabet:

• *U*: Menge von Symbolen ohne Stelligkeit

Charakt.

• $\Sigma = U \cup \bigcup_{i \geq 0} \Sigma_i$

Endlicher geordneter Baum T = (P, t) über Σ :

- $P \subseteq \mathbb{N}_{+}^{*}$ nichtleere endl. präfix-abgeschlossene Menge
- $t: P \to \Sigma$ Funktion mit
 - Wenn $t(p) \in \Sigma_m$, dann $\{i \mid pi \in P\} = \{1, \ldots, m\}$.
 - ② Wenn $t(p) \in U$, dann $\{j \mid pj \in P\} = \{1, ..., k\}$

für ein $k \geqslant 0$.

Wir brauchen Symbole mit beliebiger Stelligkeit!

Erweitern unser r-Alphabet:

- *U*: Menge von Symbolen ohne Stelligkeit
- $\Sigma = U \cup \bigcup_{i \geq 0} \Sigma_i$

Endlicher geordneter Baum T = (P, t) über Σ :

- $P \subset \mathbb{N}^*_{\perp}$ nichtleere endl. präfix-abgeschlossene Menge
- $t: P \to \Sigma$ Funktion mit
 - Wenn $t(p) \in \Sigma_m$, dann $\{i \mid pi \in P\} = \{1, \ldots, m\}$.
 - ② Wenn $t(p) \in U$, dann $\{j \mid pj \in P\} = \{1, ..., k\}$ für ein $k \geqslant 0$.

Beschränken uns auf den Fall ohne Stelligkeit (o. S.): $\Sigma = U$

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Wir brauchen Symbole mit beliebiger Stelligkeit!

• Wenn $t(p) \in \Sigma_m$, dann $\{j \mid pj \in P\} = \{1, \dots, m\}$ Wenn $t(p) \in U$, dann $\{j \mid pj \in P\} = \{1, \dots, k\}$ Beschränken uns auf den Fall ohne Stelligkeit (o. S.): $\Sigma = U$

P ⊂ N* nichtleere endl. präfix-abseschlossene Mense

8:48

Motivation

Weitere Begriffe

Charakt.

Grundbegriffe

• $a(T_1 \cdots T_n)$: Baum mit a in Wurzel und

• Hecke (Hedge): Folge $T_1 \cdots T_n$ von Bäumen leere Hecke: ε

• $H(\Sigma)$: Menge aller Hecken über Σ

• Höhe, Tiefe, Teilbaum: wie für Bäume mit Stelligkeit

Top-down-BAs

Teilbäumen T_1, \ldots, T_n direkt darunter

XML

- u H(Σ): Menge aller Hecken über Σ

2019-02-01

Weitere Begriffe

- Anwendung: XML-Schemasprachen

Teil 2: endliche Bäume

8:50

Automatentheorie u. i. A. WiSe 2018/19

- Höhe, Tiefe, Teilbaum: wie für Bäume mit Stelligkeit
- $a(T_1 \cdots T_n)$: Baum mit a in Wurzel und Teilbäumen T_1, \ldots, T_n direkt darunter
- Hecke (Hedge): Folge $T_1 \cdots T_n$ von Bäumen leere Hecke: ε
- $H(\Sigma)$: Menge aller Hecken über Σ
- → induktive Charakterisierung von Bäumen:
 - Jede Folge von Bäumen ist eine Hecke.
 - Wenn h eine Hecke und $a \in \Sigma$ ein Symbol ist, dann ist a(h) ein Baum

$$h = \varepsilon \implies$$
 schreiben a statt $a(\varepsilon)$

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen

Hecke (Hedge): Folge T1 · · · Tn yon Bäumen u H(Σ): Menge aller Hecken über Σ · induktive Charakterisierung von Bäumen v Jede Folge von Bäumen ist eine Hecke. \bullet Wenn h eine Hecke und $a \in \Sigma$ ein Symbol ist,

 $h = \varepsilon \sim \text{schreiben a statt } a(\varepsilon)$

■ a(T₁···T_n): Baum mit a in Wurzel und

Weitere Begriffe

Weitere Begriffe dann ist a(h) ein Baum

8:50

2019-02-01

Motivation

Grundbegriffe

Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl.

XML

Beispiele für Hecken und Bäume

Sei $\Sigma = \{a, b, c\}$.

 ε ist eine Hecke

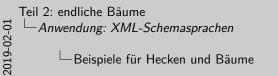
$$\rightarrow$$
 $a = a()$ ist ein Baum

$$\rightarrow b(aa)$$
 ist ein Baum

$$\rightarrow$$
 ab(aa)c ist eine Hecke

 \rightarrow a(ab(aa)c) ist ein Baum

T 2.11





Beispiele für Hecken und Bäume

8:53 bis 8:58

Motivation

Sei $\Sigma = \{a, b, c\}$.

 ε ist eine Hecke

 \rightarrow a = a() ist ein Baum \rightarrow aa ist eine Hecke $\rightarrow b(aa)$ ist ein Baum \rightarrow ab(aa)c ist eine Hecke \rightarrow a(ab(aa)c) ist ein Baum

(a(c(b)cb(ab)) ist ein Baum

Top-down-BAs

2019-02-01

T 2.11

T 2.11 Forts.

Beispiele für Hecken und Bäume

Sei $\Sigma = \{a, b, c\}.$

~ aa ist eine Hecke ~ b(aa) ist ein Baum Beispiele für Hecken und Bäume ~ ab(aa)c ist eine Hecke ~ a(ab(aa)c) ist ein Baur (a(c(b)cb(ab)) ist ein Baum

8:53 bis 8:58

Motivation Grundbegriffe Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Teil 2: endliche Bäume -Anwendung: XML-Schemasprachen

-Heckenautomaten

... sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit

Können sie analog zu NEBAs definiert werden?

Motivation Grundbegriffe

undbegriffe Charakt.

Top-down-BAs Abschli

Abschlusseig.

Entscheid.-probl.

XML

Heckenautomaten

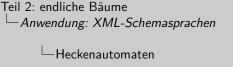
...sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit Können sie analog zu NEBAs definiert werden? Nein: Weil Stelligkeit von a ∈ Σ nicht festgelegt ist,

Heckenautomaten

...sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit

Können sie analog zu NEBAs definiert werden?

Nein: Weil Stelligkeit von $a \in \Sigma$ nicht festgelegt ist, brauchten wir 1 Regel $a(q_1, \ldots, q_m) \to q$ pro $m \ge 0$. T 2.12



Grundbegriffe

Charakt.

Abschlusseig.

Entscheid.-probl.

XML

Heckenautomaten

sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit Können sie analog zu NEBAs definiert werden? Abhilfe: Nutzen reguläre Ausdrücke über Q in linken Regelseiter

Top-down-BAs

Heckenautomaten

...sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit

Können sie analog zu NEBAs definiert werden?

Nein: Weil Stelligkeit von $a \in \Sigma$ nicht festgelegt ist, brauchten wir 1 Regel $a(q_1, \ldots, q_m) \to q$ pro $m \ge 0$. T 2.12

Abhilfe: Nutzen reguläre Ausdrücke über Q in linken Regelseiten

-Heckenautomaten

- Anwendung: XML-Schemasprachen

Teil 2: endliche Bäume

Heckenautomaten

...sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit

Top-down-BAs

Können sie analog zu NEBAs definiert werden?

Nein: Weil Stelligkeit von $a \in \Sigma$ nicht festgelegt ist, brauchten wir 1 Regel $a(q_1, \ldots, q_m) \to q$ pro $m \ge 0$. T 2.12

Abhilfe: Nutzen reguläre Ausdrücke über Q in linken Regelseiten

Definition 2.23

Ein nichtdeterministischer endlicher Heckenautomat (NEHA) ist ein Quadrupel $\mathcal{A} = (Q, \Sigma, \Delta, F)$, wobei

- Q, Σ , F wie für NEBAs definiert sind und
- Δ eine Menge von Überführungsregeln der Form

$$a(R) \rightarrow q$$

ist, wobei $a \in \Sigma$ und $R \subseteq Q^*$ eine reg. Sprache über Q ist.

Teil 2: endliche Bäume -Anwendung: XML-Schemasprachen

-Heckenautomaten

sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit Können sie analog zu NFRAs definiert werden Ein nichtdeterministischer endlicher Heckenautomat (NEHA) ist ein Quadrupel $A = (Q, \Sigma, \Delta, F)$, wobei

v Q, Σ, F wie für NEBAs definiert sind und Δ eine Menge von Überführungsregeln der Form ist, wobei $a \in \Sigma$ und $R \subseteq Q^*$ eine reg. Sprache über Q ist.

Heckenautomater

Grundbegriffe

Abschlusseig.

Entscheid.-probl.

XML

-Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.24

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEHA und $\mathcal{T} = (P, t)$ ein Σ -Baum o. S.

• Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit: Wenn t(p) = a, r(p) = q und m = Anzahl von p's Kindern,dann gibt es $a(R) \to q$ in Δ mit $r(p1) \cdots r(pm) \in R$.

9:06

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

A Run unn 4 auf T ist eine Flet r : P -> O mitdann gibt es $s(R) \rightarrow q$ in Δ mit $r(p1) \cdots r(pm) \in R$. Wenn p Blattposition mit Markierung a (d.h. t(p) = a),

- Wenn t(p) = a, r(p) = q und m = Anzahl von p's Kindern,dann darf $a(R) \rightarrow g$ nur angewendet werden, wenn $\epsilon \in R$ Repräsentation der reg. Sprache R C Q*
- das ist egal für die Mächtigkeit von NEHAs

· aber nicht für Entscheidungsverfahren und deren Komplexitä

9:06

Definition 2.24

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEHA und T = (P, t) ein Σ -Baum o. S.

• Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit: Wenn t(p) = a, r(p) = q und m = Anzahl von p's Kindern,dann gibt es $a(R) \rightarrow q$ in Δ mit $r(p1) \cdots r(pm) \in R$.

Anmerkungen

- Wenn p Blattposition mit Markierung a (d. h. t(p) = a), dann darf $a(R) \rightarrow q$ nur angewendet werden, wenn $\varepsilon \in R$.
- Repräsentation der reg. Sprache $R \subseteq Q^*$: NEAs, DEAs oder reg. Ausdrücke
 - das ist egal für die Mächtigkeit von NEHAs,
 - aber nicht für Entscheidungsverfahren und deren Komplexität!

 $L(A) = \{ T \text{ über } \Sigma \mid A \text{ akzeptiert } T \}.$

dann gibt es $s(R) \rightarrow o$ in Δ mit $r(o1) \cdots r(om) \in R$.

9:08

Berechnungen, Akzeptanz und erkannte Sprache

Definition 2.24 (Fortsetzung)

Sei $\mathcal{A} = (Q, \Sigma, \Delta, F)$ ein NEHA und T = (P, t) ein Σ -Baum o. S.

- Run von \mathcal{A} auf T ist eine Fkt. $r: P \to Q$ mit: Wenn t(p) = a, r(p) = q und m = Anzahl von p's Kindern,dann gibt es $a(R) \rightarrow q$ in Δ mit $r(p1) \cdots r(pm) \in R$.
- Ein Run r von \mathcal{A} auf T ist erfolgreich, wenn $r(\varepsilon) \in F$.
- A akzeptiert T, wenn es erfolgreichen Run von A auf T gibt.
- Die von \mathcal{A} erkannte Sprache ist $L(A) = \{ T \text{ "uber } \Sigma \mid A \text{ akzeptiert } T \}.$

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Berechnungen, Akzeptanz und erkannte Sprache

Motivation Grundbegriffe

Charakt.

Top-down-BAs Abschlusseig.

XML

Entscheid.-probl.

Beispiel

Sei $\Sigma = \{a, b, c\}$ und T ein Baum o. S. über Σ .

Der tiefste gemeinsame Vorgänger zweier Positionen p_1, p_2 in Tist die Position p, die das längste gemeinsame Präfix von p_1 , p_2 ist.

Schreibweise: $p = \operatorname{tgV}(p_1, p_2)$ T 2.13 Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Beispiel

Sei $\Sigma = \{a, b, c\}$ und T ein Baum o. S. über Σ

Schreibweise: $\rho = tqV(\rho_1, \rho_2)$

Der tiefste gemeinsame Vorgänger zweier Positionen p_1, p_2 in T

ist die Position p_i die das längste gemeinsame Präfix von p_i , p_j ist.

9:08 bis 9:15

Sei $\Sigma = \{a, b, c\}$ und T ein Baum o. S. über Σ .

Der tiefste gemeinsame Vorgänger zweier Positionen p_1, p_2 in T ist die Position p, die das längste gemeinsame Präfix von p_1 , p_2 ist.

Schreibweise: $p = \operatorname{tgV}(p_1, p_2)$

T 2.13

XML

$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit}$$

 $t(p_1) = t(p_2) = b \text{ und } t(\text{tgV}(p_1, p_2) = c)\}$

T 2.13 Forts.

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Beispiel

ist die Position p, die das längste gemeinsame Präfix von p, p, ist $L = \{T = (P, t) \mid \text{as gibt } p_1, p_2 \in P \text{ mit }$

 $t(\rho_1) = t(\rho_2) = b \text{ und } t(tgV(\rho_1, \rho_2) = c)$

9:08 bis 9:15

Beispiel

Sei $\Sigma = \{a, b, c\}$ und T ein Baum o. S. über Σ .

Der tiefste gemeinsame Vorgänger zweier Positionen p_1, p_2 in T ist die Position p, die das längste gemeinsame Präfix von p_1 , p_2 ist.

T 2.13 Schreibweise: $p = \operatorname{tgV}(p_1, p_2)$

$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit}$$

$$t(p_1) = t(p_2) = b \text{ und } t(\operatorname{tgV}(p_1, p_2) = c)\}$$

T 2.13 Forts.

Idee für einen Baumautomaten:

- Gehe in q_b , sobald b gesehen. Propagiere q_b nach oben.
- Gehe in q_c , wenn c gesehen und in 2 Kindern q_b . Propagiere q_c nach oben.
- Akzeptiere, wenn Wurzel mit q_c markiert.

Der tiefste gemeinsame Vorgänger zweier Positionen p_1, p_2 in Tist die Position p, die das längste gemeinsame Präfix von p, p, ist $L = \{T = (P, t) \mid \text{as gibt } p_1, p_2 \in P \text{ mit }$ $t(\rho_1) = t(\rho_2) = b \text{ und } t(tgV(\rho_1, \rho_2) = c)$ w Gehe in as. sobald b sesehen Propagiere q_b nach oben. Gehe in a. . wenn c gesehen und in 2 Kindern a.

Sei $\Sigma = \{a, b, c\}$ und T ein Baum o. S. über Σ

a Akzeptiere, wenn Wurzel mit qc markiert

9:08 bis 9:15

-Beispiel

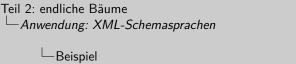
 $A = (Q, \Sigma, \Delta, F)$ mit $Q = \{q_0, q_b, q_c\}$, $F = \{q_c\}$ und

 $t(p_1) = t(p_2) = b \text{ und } t(tgV(p_1, p_2) = c)$

Beispiel

$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit}$$
$$t(p_1) = t(p_2) = b \text{ und } t(\text{tgV}(p_1, p_2) = c)\}$$

$$\mathcal{A} = (Q, \Sigma, \Delta, F)$$
 mit $Q = \{q_0, q_b, q_c\}, F = \{q_c\}$ und



9:15 bis 9:21: 5min Pause \rightarrow bis 9:26

(TODO: einarbeiten) A ist natürlich stark nichtdeterministisch: mehrere Übergänge mit demselben Buchstaben und Zustandsfolge, z. B. $a(q_b) \rightarrow q_0$ und $a(q_b) \rightarrow q_b$.

Das kann man beheben, indem man die Ausdrücke, die mit demselben Zeichen auftreten, paarweise disjunkt macht, z. B. hier:

- 1. Spalte: $a(q_0^*) \rightarrow q_0$, dito für c, $b((q_0 + q_c)^*) \rightarrow q_b$
- 2. Spalte: $a(q_0^*q_bq_0^*) \to q_b$, dito für c, $c((q_0 + q_c)^*q_b(q_0 + q_$ $q_c)^*) \rightarrow q_c$
- 3. Spalte: bleibt so

$$L = \{T = (P, t) \mid \text{es gibt } p_1, p_2 \in P \text{ mit}$$

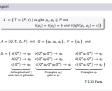
 $t(p_1) = t(p_2) = b \text{ und } t(\text{tgV}(p_1, p_2) = c)\}$

$$\mathcal{A} = (Q, \Sigma, \Delta, F)$$
 mit $Q = \{q_0, q_b, q_c\}$, $F = \{q_c\}$ und

$$\Delta = \{ \begin{array}{cccc} a(Q^*) \rightarrow q_0 & a(Q^*q_bQ^*) \rightarrow q_b & a(Q^*q_cQ^*) \rightarrow q_c \\ b(Q^*) \rightarrow q_b & c(Q^*q_bQ^*) \rightarrow q_b & b(Q^*q_cQ^*) \rightarrow q_c \\ \hline \\ c(Q^*) \rightarrow q_0 & c(Q^*q_bQ^*q_bQ^*) \rightarrow q_c & c(Q^*q_cQ^*) \rightarrow q_c \\ \hline \\ \\ \text{,,Anfangszustand''/} & \text{Propagiere } q_b/ & c(Q^*q_cQ^*) \rightarrow q_c \\ \hline \\ \text{noch kein } b \text{ gefunden} & \text{Propagiere } q_b/ & \text{Propagiere } q_c \\ \hline \end{array}$$

T 2.13 Forts.

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen



9:15 bis 9:21: 5min Pause \rightarrow bis 9:26

└─ Beispiel

(TODO: einarbeiten) A ist natürlich stark nichtdeterministisch: mehrere Übergänge mit demselben Buchstaben und Zustandsfolge, z. B. $a(q_b) \rightarrow q_0$ und $a(q_b) \rightarrow q_b$.

Das kann man beheben, indem man die Ausdrücke, die mit demselben Zeichen auftreten, paarweise disjunkt macht, z. B. hier:

- 1. Spalte: $a(q_0^*) \rightarrow q_0$, dito für c, $b((q_0 + q_c)^*) \rightarrow q_b$
- 2. Spalte: $a(q_0^*q_bq_0^*) \to q_b$, dito für c, $c((q_0 + q_c)^*q_b(q_0 + q_$ $q_c)^*) \rightarrow q_c$
- 3. Spalte: bleibt so

Grundbegriffe

. . .

Charakt.

<!DOCTYPE CONFERENCE [</pre>

<!ELEMENT session

<!ELEMENT speaker

<!ELEMENT track

<!ELEMENT talk

<!ELEMENT chair

((session, break?)+)>

(chair,talk+)>

(#PCDATA)>

(#PCDATA)>

Umwandlung der Beispiel-DTD in einen NEHA (1)

<!ELEMENT conference (track+|(session,break?)+)>

XML

Entscheid.-probl.

((title,authors)|(title,speaker))>

9:26

Wollen jetzt sehen, wie man DTDs in NEHAs umwandelt, um Automatentechniken auf Validierung usw. anwenden zu können.

1. Schritt: erweiterte kfG, die der DTD entspricht (hier am Bsp.) ("erweitert": mit regulären Ausdrücken auf rechten Regelseiten)

Charakt.

```
((title, authors)|(title, speaker)))
→ (title authors) + (title speaker
```

9:26

Wollen jetzt sehen, wie man DTDs in NEHAs umwandelt, um Automatentechniken auf Validierung usw. anwenden zu können.

1. Schritt: erweiterte kfG, die der DTD entspricht (hier am Bsp.) ("erweitert": mit regulären Ausdrücken auf rechten Regelseiten)

```
<!DOCTYPE CONFERENCE |
  <!ELEMENT conference (track+|(session,break?)+)>
  <!ELEMENT track
                        ((session, break?)+)>
  <!ELEMENT session
                        (chair,talk+)>
                        ((title,authors)|(title,speaker))>
  <!ELEMENT talk
  <!ELEMENT chair
                        (#PCDATA)>
  . . .
                        (#PCDATA)>
  <!ELEMENT speaker
```

Zugehörige erweiterte kontextfreie Grammatik:

```
track^+ + (session (break + \varepsilon))^+
conference
                     (session (break + \varepsilon))^+
track
                     chair talk+
session
talk
                     (title authors) + (title speaker)
chair
                     DATA
speaker
                     DATA
```

otivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML Toil 2: and links Päymes

Umwandlung der Beispiel-DTD in einen NEHA (2)

Erweiterte kontextfreie Grammatik (Wdhlg.):

```
\begin{array}{lll} {\rm conference} & \to & {\rm track}^+ + ({\rm session} \ ({\rm break} + \varepsilon))^+ \\ {\rm track} & \to & ({\rm session} \ ({\rm break} + \varepsilon))^+ \\ {\rm session} & \to & {\rm chair} \ {\rm talk}^+ \\ {\rm talk} & \to & ({\rm title} \ {\rm authors}) + ({\rm title} \ {\rm speaker}) \\ {\rm chair} & \to & {\rm DATA} \\ & \dots \\ {\rm title} & \to & {\rm DATA} \end{array}
```

```
Teil 2: endliche Bäume

Anwendung: XML-Schemasprachen

Umwandlung der Beispiel-DTD in einen NEHA

(2)
```

9:29 bis 9:35

2019-02-01

Wie funktionieren eigentlich **erweitert**e kfGs? Hier keine formale Def., sondern Vorgehen bei einzelnem Ableitungsschritt

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Umwandlung der Beispiel-DTD in einen NEHA (2)

Erweiterte kontextfreie Grammatik (Wdhlg.):

Charakt.

```
track^+ + (session (break + \varepsilon))^+
conference
                      (session (break + \varepsilon))^+
track
                     chair talk+
session
                      (title authors) + (title speaker)
talk
                     DATA
chair
title
                     DATA
                \rightarrow
```

Startsymbol: hier conference

Ableitungsschritt:

- Wähle mit ℓ beschriftetes Blatt. $\ell \in \Sigma$
- Wähle Regel $\ell \to R$ (R: reg. Sprache über Σ , Inhaltsmodell)
- Wähle $a_1 \cdots a_n \in R$ und füge Kinder a_1, \ldots, a_n zu ℓ hinzu

Umwandlung der Beispiel-DTD in einen NEHA (2) Teil 2: endliche Bäume 2019-02-01 - Anwendung: XML-Schemasprachen Umwandlung der Beispiel-DTD in einen NEHA

9:29 bis 9:35

(2)

Wie funktionieren eigentlich erweiterte kfGs? Hier keine formale Def., sondern Vorgehen bei einzelnem Ableitungsschritt

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Teil 2: endliche Bäume

(2)

- Anwendung: XML-Schemasprachen

Umwandlung der Beispiel-DTD in einen NEHA

Umwandlung der Beispiel-DTD in einen NEHA (2)

9:29 bis 9:35

Wie funktionieren eigentlich erweiterte kfGs? Hier keine formale Def., sondern Vorgehen bei einzelnem Ableitungsschritt

Umwandlung der Beispiel-DTD in einen NEHA (2)

Erweiterte kontextfreie Grammatik (Wdhlg.):

```
track^+ + (session (break + \varepsilon))^+
conference
                      (session (break + \varepsilon))^+
track
                     chair talk+
session
                      (title authors) + (title speaker)
talk
                     DATA
chair
title
                     DATA
                \rightarrow
```

Startsymbol: hier conference

Ableitungsschritt:

- Wähle mit ℓ beschriftetes Blatt. $\ell \in \Sigma$
- Wähle Regel $\ell \to R$ (R: reg. Sprache über Σ , Inhaltsmodell)
- Wähle $a_1 \cdots a_n \in R$ und füge Kinder a_1, \ldots, a_n zu ℓ hinzu
- Beispielableitung: siehe Tafel

T 2.14

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Umwandlung der Beispiel-DTD in einen NEHA (3)

Erweiterte kontextfreie Grammatik (Wdhlg.):

```
\begin{array}{lll} {\rm conference} & \to & {\rm track}^+ + ({\rm session} \ ({\rm break} + \varepsilon))^+ \\ {\rm track} & \to & ({\rm session} \ ({\rm break} + \varepsilon))^+ \\ & \vdots & & \\ {\rm title} & \to & {\rm DATA} \end{array}
```



9:35 bis 9:38

2019-02-01

Schritt 2: NEHA aus Grammatik bauen

• Zustände = Alphabet

(3)

 \bullet Δ : aus den Produktionen (Zustände kursiv, Zeichen aufrecht)

Umwandlung der Beispiel-DTD in einen NEHA (3)

Erweiterte kontextfreie Grammatik (Wdhlg.):

```
track^+ + (session (break + \varepsilon))^+
conference
                    \rightarrow (session (break + \varepsilon))<sup>+</sup>
track
title
                    \rightarrow
                            DATA
```

Zugehöriger NEHA: $A = (Q, \Sigma, \Delta, F)$ mit

```
{conference, track, session, talk, chair, ..., DATA}
{conference}
\{ conf(track^+ + (session (break + \varepsilon))^+) \}
                                                      conf,
  track((session (break + \varepsilon))^+)
                                                      track,
  title(DATA)
                                                       title,
  DATA()
                                                      DATA }
```

9:35 bis 9:38

Schritt 2: NEHA aus Grammatik bauen

• Zustände = Alphabet

(3)

Δ: aus den Produktionen (Zustände kursiv, Zeichen aufrecht)

Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Präzise Definition DTD & zugehöriger NEHA

XML Teil 2: endliche Bäume

Präzise Definition DTD & zugehöriger NEHA -Anwendung: XML-Schemasprachen

Präzise Definition DTD & zugehöriger NEHA

u einer Abbildung $\Delta : \Sigma \rightarrow reguläre Ausdrücke über <math>\Sigma$ Kindern iedes Knotens mit $x \in \Sigma$ muss in $L(\Delta(x))$ sein.)

Eine Dokumenttypdefinition (DTD) ist ein Tupel $D = (\Sigma, s, \Delta)$ mit einem Alphabet Σ (ohne Stelligkeit) a einem Startsymbol $s \in \Sigma$ und

9:38

Definition 2.25

Eine Dokumenttypdefinition (DTD) ist ein Tupel $D = (\Sigma, s, \Delta)$ mit

 \bullet einem Alphabet Σ (ohne Stelligkeit)

Charakt.

- einem Startsymbol $s \in \Sigma$ und
- \bullet einer Abbildung $\Delta : \Sigma \to \text{reguläre Ausdrücke über } \Sigma$

(Δ entspricht einer Menge von Regeln – die Folge der Symbole in den Kindern jedes Knotens mit $a \in \Sigma$ muss in $L(\Delta(a))$ sein.)

Präzise Definition DTD & zugehöriger NEHA

Definition 2.25

Eine Dokumenttypdefinition (DTD) ist ein Tupel $D = (\Sigma, s, \Delta)$ mit

- \bullet einem Alphabet Σ (ohne Stelligkeit)
- einem Startsymbol $s \in \Sigma$ und
- \bullet einer Abbildung $\Delta : \Sigma \to \text{reguläre Ausdrücke über } \Sigma$

(Δ entspricht einer Menge von Regeln – die Folge der Symbole in den Kindern jedes Knotens mit $a \in \Sigma$ muss in $L(\Delta(a))$ sein.)

Zugehöriger NEHA: $A_D = (Q_D, \Sigma, \Delta_D, F_D)$ mit

- $Q_D = \Sigma$
- $F_D = \{s\}$
- $\Delta_D = \{a(\Delta(a)) \rightarrow a \mid a \in \Sigma\}$

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Präzise Definition DTD & zugehöriger NEHA Präzise Definition DTD & zugehöriger NEHA Eine Dokumenttypdefinition (DTD) ist ein Tupel $D = (\Sigma, s, \Delta)$ mi einem Alphabet Σ (ohne Stelligkeit) a einem Startsymbol $s \in \Sigma$ und u einer Abbildung $\Delta : \Sigma \rightarrow reguläre Ausdrücke über <math>\Sigma$ Kindern iedes Knotens mit $x \in \Sigma$ muss in $L(\Delta(x))$ sein. $A_D = (Q_D, \Sigma, \Delta_D, F_D)$ mit $u F_0 = \{s\}$

• $\Delta_D = \{a(\Delta(a)) \rightarrow a \mid a \in \Sigma\}$

9:38

Motivation Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

-Anwendung: XML-Schemasprachen

Lokale Sprachen

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

Lokale Sprachen

8:30, nur Wdhlq. 9:41 bis 9:52 Motivation Grundbegriffe

Charakt.

Top-down-BAs

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

Frage:

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.?

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Lokale Sprachen

8:30, nur Wdhlq. 9:41 bis 9:52 Grundbegriffe

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Lokale Sprachen

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

Charakt.

Frage:

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.? (Gibt es NEHAs, die keine DTD beschreiben?) Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Lokale Sprachen

8:30, nur Wdhlq. 9:41 bis 9:52 (Gibt es NEHAs, die keine DTD beschreiben

Grundbegriffe Charakt. Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

Frage:

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.? (Gibt es NEHAs, die keine DTD beschreiben?)

Anwort:

Nicht jede NEHA-erkennbare Sprache ist lokal. (Ja.)

Weil . . .

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Lokale Sprachen

Lokale Sprachen v Eine Baumsprache über Σ heißt lokal. wenn sie von einer DTD über Σ erzeugt wird. (Gibt es NEHAs, die keine DTD beschreiben Nicht jede NEHA-erkennbare Sprache ist lokal.

8:30, nur Wdhlq. 9:41 bis 9:52

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Lokale Sprachen

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

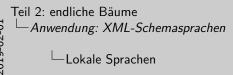
Frage:

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.? (Gibt es NEHAs, die keine DTD beschreiben?)

Anwort:

Nicht jede NEHA-erkennbare Sprache ist lokal. (Ja.)

Weil DTDs "immer nur eine Ebene nach unten schauen"



8:30, nur Wdhlq. 9:41 bis 9:52 Well DTDs ..immer nur eine Ebene nach unten schauen"

Grundbegriffe Lokale Sprachen

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

Definition 2.26

- Die von einer DTD D erzeugte Sprache ist $L(A_D)$.
- Eine Baumsprache über Σ heißt lokal, wenn sie von einer DTD über Σ erzeugt wird.

Frage:

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.? (Gibt es NEHAs, die keine DTD beschreiben?)

Anwort:

Nicht jede NEHA-erkennbare Sprache ist lokal. (Ja.)

Weil DTDs "immer nur eine Ebene nach unten schauen" T 2.15 (Nicht ausdrückbar:

"alle Sitzungen jeder Konf. haben zusammen ≥ 5 Vortragende")

Teil 2: endliche Bäume Anwendung: XML-Schemasprachen -Lokale Sprachen

8:30, nur Wdhlq. 9:41 bis 9:52 Motivation Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

XML

-Deterministische Inhaltsmodelle

Deterministische Inhaltsmodelle

Die W3C^a-Empfehlung für XML fordert, dass Inhaltsmodelle deterministische reguläre Ausdrücke sind.

^aWorld Wide Web Consortium, int. Agentur für WWW-Standards



DTDs werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.

Top-down-BAs

-Deterministische Inhaltsmodelle

*World Wide Web Consortium, Int. Agentur für WWW-Standards

Die W3C^a-Empfehlung für XML fordert, dass Inhaltsmodelle deterministische reguläre Ausdrücke sind.

^aWorld Wide Web Consortium, int. Agentur für WWW-Standards

Regulärer Ausdruck r über Σ ist deterministisch, falls

• für jedes Wort $w \in \Sigma^*$ und jeden Buchstaben a in w höchstens ein Vorkommen von a in r existiert, auf das a passt. 8:31, nur Wdhlg.

DTDs werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.

Deterministische Inhaltsmodelle

Charakt.

XML

Stellt sicher dass das Zusehöriekeitsnrohlem für DTDs in

Die W3C^a-Empfehlung für XML fordert, dass Inhaltsmodelle deterministische reguläre Ausdrücke sind.

^aWorld Wide Web Consortium, int. Agentur für WWW-Standards

Regulärer Ausdruck r über Σ ist deterministisch, falls

- für jedes Wort $w \in \Sigma^*$ und jeden Buchstaben a in w höchstens ein Vorkommen von a in r existiert, auf das a passt.
- Dann lässt sich in Polyzeit ein äquivalenter DEA konstruieren.
- → Stellt sicher, dass das Zugehörigkeitsproblem für DTDs in Polyzeit lösbar ist.

8:31, nur Wdhlg.

DTDs werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.

Motivation

Grundbegriffe Charakt.

Deterministische Inhaltsmodelle – Beispiel

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML Teil 2: endliche Bäume

- Anwendung: XML-Schemasprachen

Deterministische Inhaltsmodelle – Beispiel

Deterministische Inhaltsmodelle - Beispiel ((title.suthors))(title.sueaker))> talk → (title authors) + (title speaker)

Betrachte die Zeile

((title,authors)|(title,speaker))> <!ELEMENT talk

und die zugehörige Regel

(title authors) + (title speaker) $\mathtt{talk} \quad \rightarrow \quad$

8:32, nur Wdhlg. 9:54

Betrachte die Zeile

und die zugehörige Regel

talk

dieser Buchstabe entspricht!

<!ELEMENT talk

Charakt. Deterministische Inhaltsmodelle – Beispiel

Entscheid.-probl.

Top-down-BAs

((title,authors)|(title,speaker))>

(title authors) + (title speaker)

Abschlusseig.

XML

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen

Deterministische Inhaltsmodelle – Beispiel

talk → (title authors) + (title speaker) Für Wörter über Σ. die mit dem Buchstaben title beginnen dissar Ruchstaha entswicht!

((title.suthors))(title.speaker))>

Deterministische Inhaltsmodelle - Beispiel

8:32, nur Wdhlg. 9:54

Für Wörter über Σ , die mit dem Buchstaben title beginnen, ist nicht klar, welchem Vorkommen von title im Inhaltsmodell

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

2019-02-01

Jetzt genau: deterministische reguläre Ausdrücke

Charakt.

Idee: Sei r ein RA über Σ .

• Markiere das *i*-te Vorkommen jedes Buchstaben *a* in *r* mit *a_i*.

• Bsp.:
$$(a + b)^*b(ab)^* \rightarrow (a_1 + b_1)^*b_2(a_2b_3)^* =: r'$$
.

• r ist deterministisch, wenn L(r') keine zwei Wörter $ua_i v$ und $ua_i w$ mit $i \neq j$ enthält.

letzt genau: deterministische reguläre Ausdrücke Teil 2: endliche Bäume

wenn L(r') keine zwei Wörter ua v und ua w mit i ≠ i enthält

- Anwendung: XML-Schemasprachen

└─Jetzt genau: deterministische reguläre Ausdrücke

8:33, nur Wdhlq. 9:55 bis $10:00 \rightarrow \text{Ende}$? (Rest dauert noch $\approx 5 \text{ min}$)

TODO: Klarmachen, dass RAe über $+,\cdot,*$ definiert sind und kein \cdot^+ erlaubt ist!

Top-down-BAs

Abschlusseig.

XML

2019-02-01

Entscheid.-probl.

Jetzt genau: deterministische reguläre Ausdrücke

Charakt.

Idee: Sei r ein RA über Σ .

- Markiere das *i*-te Vorkommen jedes Buchstaben *a* in *r* mit *a_i*.
- Bsp.: $(a + b)^*b(ab)^* o (a_1 + b_1)^*b_2(a_2b_3)^* =: r'$.
- r ist deterministisch. wenn L(r') keine zwei Wörter $ua_i v$ und $ua_i w$ mit $i \neq j$ enthält.

Etwas Notation:

- RA r über $\Sigma \sim \text{markierter RA } r'$ über Σ'
- wie üblich: $L(r) \subset \Sigma^*$ und $L(r') \subset \Sigma'^*$

8:33, nur Wdhlq. 9:55 bis $10:00 \rightarrow \text{Ende}$? (Rest dauert noch $\approx 5 \text{ min}$)

└─Jetzt genau: deterministische reguläre Ausdrücke

TODO: Klarmachen, dass RAe über $+,\cdot,*$ definiert sind und kein \cdot^+ erlaubt ist!

8:33, nur Wdhlq. 9:55 bis $10:00 \rightarrow \text{Ende}$? (Rest dauert noch $\approx 5 \text{ min}$)

└─Jetzt genau: deterministische reguläre Ausdrücke

TODO: Klarmachen, dass RAe über $+,\cdot,*$ definiert sind und kein \cdot^+ erlaubt ist!

Jetzt genau: deterministische reguläre Ausdrücke

Idee: Sei r ein RA über Σ .

- Markiere das *i*-te Vorkommen jedes Buchstaben *a* in *r* mit *a_i*.
- Bsp.: $(a + b)^*b(ab)^* o (a_1 + b_1)^*b_2(a_2b_3)^* =: r'$.
- r ist deterministisch. wenn L(r') keine zwei Wörter ua_iv und ua_iw mit $i \neq j$ enthält.

Etwas Notation:

- RA r über $\Sigma \sim \text{markierter RA } r'$ über Σ'
- wie üblich: $L(r) \subset \Sigma^*$ und $L(r') \subset \Sigma'^*$

Definition 2.27

Ein deterministischer RA (DRA) ist ein RA r über Σ , so dass

für alle Wörter $u, v, w \in \Sigma'^*$ und Zeichen $a \in \Sigma$ mit ua_iv , $ua_iw \in L(r')$ gilt: i = j.

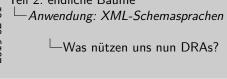
T 2.16

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Teil 2: endliche Bäume

Was nützen uns nun DRAs?

Motivation



8:34 (10:00 bis 10:01)

Motivation

Charakt.

Abschlusseig.

Entscheid.-probl.

8:34

(10:00 bis 10:01)

Satz 2.28

Zu jedem DRA r kann man in Polynomialzeit einen DEA $\mathcal A$ mit L(A) = L(r) konstruieren.

(Ohne Beweis.)

─Was nützen uns nun DRAs?

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Was nützen uns nun DRAs?

L(A) = L(r) konstruieren. (Ohne Beweis.)

Zu iedem DRA r kann man in Polynomialzeit einen DEA A mit

Zu ieder deterministischen DTD kann man in Polynomialzeit

einen ämisslenten NFHA/DFA) konstruieren

Was nützen uns nun DRAs?

Charakt.

Satz 2.28

Zu jedem DRA r kann man in Polynomialzeit einen DEA ${\cal A}$ mit L(A) = L(r) konstruieren.

(Ohne Beweis.)

Folgerung 2.29

Zu jeder deterministischen DTD kann man in Polynomialzeit einen äquivalenten NEHA(DEA) konstruieren.

NEHA(DEA): $A = (Q, \Sigma, \Delta, F)$, bei dem für alle $a(R) \rightarrow q \in \Delta$ R als DEA gegeben ist.

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen └─Was nützen uns nun DRAs?

8:34 (10:00 bis 10:01)

Charakt. Top-down-BAs Abschlusseig.

Entscheid.-probl.

Was nützen uns nun DRAs?

Was nützen uns nun DRAs?

Satz 2.28

Zu jedem DRA r kann man in Polynomialzeit einen DEA $\mathcal A$ mit L(A) = L(r) konstruieren.

(Ohne Beweis.)

Folgerung 2.29

Zu jeder deterministischen DTD kann man in Polynomialzeit einen äquivalenten NEHA(DEA) konstruieren.

NEHA(DEA): $A = (Q, \Sigma, \Delta, F)$, bei dem für alle $a(R) \rightarrow q \in \Delta$ R als DEA gegeben ist.

Und dieses Resultat garantiert nun ...?

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen └─Was nützen uns nun DRAs?

Zu ieder deterministischen DTD kann man in Polynomialzeit einen ämijvalenten NEHA/DEA) konstruieren R als DEA gegeben ist Und dieses Resultat garantiert nun ...

Zu iedem DRA r kann man in Polynomialzeit einen DEA A mit

L(A) = L(r) konstruieren.

8:34 (10:00 bis 10:01) Motivation

Grundbegriffe

Charakt.

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

XML

- Anwendung: XML-Schemasprachen

-Deterministische DTDs sind effizient!

Für deterministische DTDs sind in Polynomialzeit lösbar das Zuzehörigkeitsproblem das Leerheitsproblem a das Aquivalenzproblem (Ohne Beweis.)

8:35 (10:01 bis 10:02)

"Ohne Beweis": eins davon voraussichtlich auf Ü-Blatt 3

Satz 2.30

Für deterministische DTDs sind in Polynomialzeit lösbar:

- das Zugehörigkeitsproblem
- das Leerheitsproblem
- das Äquivalenzproblem

(Ohne Beweis.)

Charakt.

Deterministische DTDs sind effizient!

Top-down-BAs

Entscheid.-probl.

XML

2019-02-01

Abschlusseig.

-Deterministische DTDs sind effizient!

Deterministische DTDs sind effizient! Für deterministische DTDs sind in Polynomialzeit lösba a das Zuzehöriekeitsproblem a das Aquivalenzproblen (Ohne Beweis.) Zuschöriskeitsproblen

lst ein gegebenes Dokument gültig für ein gegebenes Schema?

8:35 (10:01 bis 10:02)

"Ohne Beweis": eins davon voraussichtlich auf Ü-Blatt 3

Satz 2.30

Für deterministische DTDs sind in Polynomialzeit lösbar:

- das Zugehörigkeitsproblem
- das Leerheitsproblem
- das Äguivalenzproblem

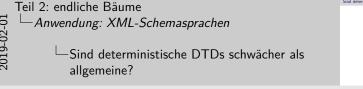
(Ohne Beweis.)

Zur Erinnerung:

- Zugehörigkeitsproblem (Gültigkeit) Ist ein gegebenes Dokument gültig für ein gegebenes Schema?
- Leerheitsproblem (Widerspruchsfreiheit) Gibt es für ein gegebenes Schema gültige Dokumente?
- Äquivalenzproblem Haben zwei Schemata dieselben gültigen Dokumente?

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML T.: 1.0. a. v. 11: a. p. 2. v. 11: a.

Sind deterministische DTDs schwächer als allgemeine?



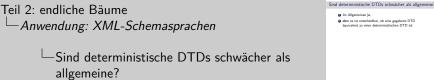
8:36 (10:02 bis 10:03)

Motivation

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Sind deterministische DTDs schwächer als allgemeine?

- 1 Im Allgemeinen ja,
- **②** aber es ist entscheidbar, ob eine gegebene DTD äquivalent zu einer deterministischen DTD ist:



8:36 (10:02 bis 10:03)

Sind deterministische DTDs schwächer als allgemeine?

- Im Allgemeinen ja,
- 2 aber es ist entscheidbar, ob eine gegebene DTD äguivalent zu einer deterministischen DTD ist:

Charakt.

Satz 2.31

1 Nicht jede reg. Sprache wird durch einen DRA beschrieben:

$${L(r) \mid r \text{ ist DRA}} \subset {L(r) \mid r \text{ ist RA}}$$

2 Das folgende Problem ist in Polynomialzeit entscheidbar.

Gegeben: DEA \mathcal{A}

Frage: Gibt es einen DRA r mit L(r) = L(A)?

Wenn ein solcher DRA existiert. dann kann er in Exponentialzeit konstruiert werden.

Sind deterministische DTDs schwächer als allgemeine Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen aber es ist entscheidbar, ob eine gegebene DTD äquivalent zu einer deterministischen DTD ist: Nicht jede reg. Sprache wird durch einen DRA beschrieben $\{L(r) \mid r \text{ ist DRA}\} \subset \{L(r) \mid r \text{ ist RA}\}$ -Sind deterministische DTDs schwächer als Das folgende Problem ist in Polynomialzeit entscheidba Frage: Gibt as einen DRA r mit L(r) = L(A)? allgemeine?

8:36 (10:02 bis 10:03)

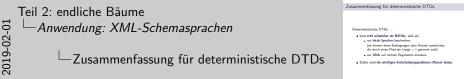
Top-down-BAs

Abschlusseig.

Charakt.

Deterministische DTDs

- sind echt schwächer als NEHAs, weil sie
 - nur lokale Sprachen beschreiben (sie können keine Bedingungen über Knoten ausdrücken, die durch einen Pfad der Länge > 1 getrennt sind);
 - nur DRAs auf rechten Regelseiten erlauben.
- Dafür sind die wichtigen Entscheidungsprobleme effizient lösbar.



(10:03 bis 10:04) 8:37

Top-down-BAs

Abschlusseig.

Entscheid.-probl.

Extended DTDs (EDTDs)

- führen durch eine einfache syntaktische Erweiterung aus den lokalen Sprachen heraus
- sind fast äquivalent zu NEHAs (beschränkt auf Sprachen, in denen alle Bäume dasselbe Wurzelsymbol haben)
- haben ein in Polynomialzeit lösbares Zugehörigkeits- und Leerheitsproblem

Teil 2: endliche Bäume 2019-02-01 -Anwendung: XML-Schemasprachen -Ausblick: Lockern der Einschränkungen Ausblick: Lockern der Einschränkungen

a führen durch eine einfache syntaktische Erweiterung aus den lokalen Sprachen heraus u sind fast äquivalent zu NEHAs (beschränkt auf Sprachen, in denen alle Bäume dasselbe Zugehörigkeits- und Leerheitsproblem

Extended DTDs (EDTDs)

8:38 (10:04 bis 10:05)

Charakt.

Top-down-BAs Abschlusseig.

Entscheid.-probl

XML

2019-02-01

Ausblick: Lockern der Einschränkungen

Extended DTDs (EDTDs)

- führen durch eine einfache syntaktische Erweiterung aus den lokalen Sprachen heraus
- sind fast äquivalent zu NEHAs (beschränkt auf Sprachen, in denen alle Bäume dasselbe Wurzelsymbol haben)
- haben ein in Polynomialzeit lösbares Zugehörigkeits- und Leerheitsproblem

Weitere Einschränkung von EDTDs

- garantiert auch ein in Polynomialzeit lösbares Äguivalenzproblem
- liegt XML Schema zugrunde

Ausblick: Lockern der Einschränkungen Teil 2: endliche Bäume Extended DTDs (EDTDs) Anwendung: XML-Schemasprachen a führen durch eine einfache syntaktische Erweiterung aus den lokalen Sprachen heraus u sind fast äquivalent zu NEHAs (beschränkt auf Sprachen, in denen alle Bäume dasselbe Wurzelsymbol haben) haben ein in Polynomialzeit lösbares Zugehörigkeits- und Leerheitsproblem -Ausblick: Lockern der Einschränkungen Weitere Einschränkung von EDTDs w garantiert auch ein in Polynomialzeit lösbares ■ liegt XML Schema zugrunde

8:38 (10:04 bis 10:05) Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XML

Damit sind wir am Ende dieses Kapitels.



Vielen Dank.

Teil 2: endliche Bäume

Damit sind wir am Ende dieses Kapitels.



8:39 ightarrow nochmal zur Literatur, bis 8:40 $\,$ (10:05) Ende ightarrow nochmal zur Literatur

Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl.

Literatur für diesen Teil (Basis)



Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, Marc Tommasi.

Tree Automata Techniques and Applications.

http://tata.gforge.inria.fr Nov. 2008.

Kapitel 1

Abschnitt 2.4 (Verbindung zu kontextfreien Wortsprachen) Abschnitte 8.2.1, 8.2.2, 8.7 (Heckenaut. und XML-Schemasprachen)



Meghyn Bienvenu.

Automata on Infinite Words and Trees.

Vorlesungsskript, Uni Bremen, WS 2009/10.

http://www.informatik.uni-bremen.de/tdki/lehre/ws09/automata/automata-notes.pdf

Kapitel 3

Teil 2: endliche Bäume

Literatur für diesen Teil (Basis)

Habert Carees, Mac Daschet, Birnt Gilbern, Florest Jongsamed, Denis Logic, Christi Lidegi, Sophia Tiene, Mar Tarmend, Tim Arizonaria Theoryteen of Registrations, Tim Arizonaria Theoryteen of Registrations, and the Company of the Co

Literatur für diesen Teil (Basis)

Motivation Grundbegriffe Charakt. Top-down-BAs Abschlusseig. Entscheid.-probl. XI

Literatur für diesen Teil (weiterführend)



 $\label{eq:Anne Brüggemann-Klein, Derick Wood.}$

One-Unambiguous Regular Languages.

Information and Computation, 142:1998, S. 182-206.

http://dx.doi.org/10.1006/inco.1997.2695

Grundlegende Resultate für deterministische reguläre Ausdrücke.

Teil 2: endliche Bäume

2019-02-01

Literatur für diesen Teil (weiterführend)

Literatur für diesen Teil (weiterführend)

