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

Wintersemester 2018/19 Thomas Schneider

AG Theorie der könstlichen Intelligenz (TdRI)

http://tinyurl.com/ws1819-autom

Und nun ...

Semistrukturierte Daten sind .

 ein Datenmodell zur Beschreibung von Entititen und Attributen,

das weniger formale Struktur voraussetzt als z.B. relationale Datenbanken ein Vorläufer von XML

ein Vorläufer von XI
 gut geeignet, um

Dokumentansichten (z. B. Webseiten) und strukturierte Daten (z. B. Datenbank-Tabellen)

strukturierte Daten (z. B. Datenbank-Tabellen)
 zu repr
äsentieren und miteinander zu verbinden

Teil 2: endliche Bäume

Motivation: semistrukturierte Daten

Merkmale semistrukturierter Daten

8:32

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
- u Reihenfolge der Attribute kann eine Rolle spielen

Teil 2: endliche Bäume

-- Motivation: semistrukturierte Daten

-- Merkmale semistrukturierter Daten

Elbing you Franchism
 Orogiene Michigan
 Entitate densibles Crupes klosses
 Entitate densibles Crupes klosses
 Entitate Resibles Crupes klosses
 inchending der Artiches kann eine Rolle geiden
 inchending der Artiches kann eine Rolle geiden
 inchending der Artiches kann eine Rolle geiden
 inchending der Artiches kann eine Rolle geiden

Persons (Sines) (Tit: "Titsman", III: "Schaesden"),
 Totar "SASSZ,
 Rolle "ATTREL,
 Rolle "ATT

Merkmale semistrukturierter Daten

Markierung von Entitäten durch Tags

Daten werden in Entitäten (Einheiten) zusammengefasst

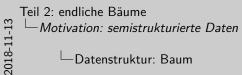
Teil 2: endliche Bäume

Motivation: semistrukturierte Daten

Datenstruktur: Baum

7/41

Datenstruktur: Baum



Datenstruktur: Baum

Person; { (bos.: {(Pi. "Tromes", Bi. "Eduardes")}, State: (4642), State: (4642), State: (477626), State:

Teil 2: endliche Bäume

Motivation: semistrukturierte Daten

Datenstruktur: Baum



8:34

2018-11-13

Automaten auf endlichen Bäumen

- ... sind wichtig für semistrukturierte Daten, weil sie ...

 w XML-Schemasprachen und -validierung zugrunde liegen
- XML-Anfragesprachen auf ihnen aufgebaut sind

Teil 2: endliche Bäume

Motivation: semistrukturierte Daten

XML-Schemasprachen und -validierung

8:35

XML-Schemasprachen und -validierung

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

8:35

XML-Schemasprachen und -validierung

- Schemasprachen zur Beschreibung gültiger Bäume
 Välidierung = Erkennen gültiger und ungültiger Bäume
- vanderung = Erkernen gurüger und ungutüger Baume

 gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben



8:35

XML-Schemasprachen und -validierung

- Schemasprachen zur Beschreibung gültiger Bäume
 Välidierung = Erkennen gültiger und ungültiger Bäume
- vanderung = Erkernen gurüger und ungutüger Baume

 gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen haben



XML-Schemasprachen und -validierung

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

8:35

XML-Schemasprachen und -validierung

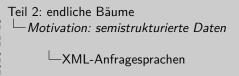
- Schemasprachen zur Beschreibung gültiger Bäume
 Validierung = Erkennen gültiger und ungültiger Bäume
- gängige Schemasprachen benutzen dafür Baumautomaten

Beispiel: jeder Name muss einen Vor- und Nachnamen habe Peraon Peraon



XML-Anfragesprachen

• beantworten Anfragen mit Daten aus gegebenen Bäumen



XML-Anfragesprachen

• basehontes Anfragen mit Dates aus gegebeen Blomes
Beiget gib alle Hames von Prozone zurück

Termin

Prozone

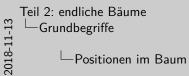
Termin

Prozone

Termin

**Termin*

Und nun . . .



Positionen im Baum $\begin{array}{c} \bullet \text{ position examines } X_{\theta} \\ \bullet \text{ Position: Wort } p \in \mathbb{N}_{+}^{*} \\ \bullet \text{ Position: Wort } p \in \mathbb{N}_{+}^{*} \\ \bullet \text{ MacColliger von } p \text{ int } pj \\ \bullet \text{ Balaples:} \\ \end{array}$

8:38

Alphabet mit Stelligkeit

hier: r-Alphabet Σ (auf Englisch: ranked alphabet)
 nichtbere endliche Menge von Symbolen;
 jedem Symbol ist eine Stelligkeit ∈ N zugeordnet

Σ = Menen der Symbole mit Stelligkeit in

 $\mathbf{x} = \mathbf{x} =$

Teil 2: endliche Bäume
Grundbegriffe
Alphabet mit Stelligkeit

8:38

Alphabet mit Stelligkeit

• hier: ϵ -Alphabet Σ (auf Englisch: ranked alphabet) • nichtsere endliche Menge von Symbolen: jedem Symbol ist eine Stelligkeit $\in \mathbb{N}$ zugeordnet • $\Sigma_m =$ Menge der Symbole mit Stelligkeit m• Schreibweise: $\Sigma = \{z_1, \ldots, z_n/c_p\}$ hellt: Σ enthitt die Symbole a im Stelligkeit $r_i := 1, \ldots, n$

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



Teil 2: endliche Bäume Grundbegriffe -Alphabet mit Stelligkeit

8:38

Alphabet mit Stelligkeit

 hier: r-Alphabet Σ (auf Englisch: ranked alphabet) a nichtleere endliche Menge von Symbolen: jedem Symbol ist eine Stelligkeit ∈ N zugeordnet $\mathbf{u} \; \Sigma_m = \mathsf{Menge} \; \mathsf{der} \; \mathsf{Symbole} \; \mathsf{mit} \; \mathsf{Stelligkeit} \; m$ v Schreibweise: $\Sigma = \{a_1/r_1, \dots, a_n/r_n\}$ heißt:

 Σ enthält die Symbole a: mit Stelliekeit r.: i = 1, ..., nBeispiel: $\Sigma = \{a/2, b/3, c/0, d/0\}$



Teil 2: endliche Bäume
└─Grundbegriffe
└─Was ist nun ein Baum?



Was ist nun ein Baum?

8:42

2018-11-13

1

Teil 2: endliche Bäume
Grundbegriffe
Was ist ein Baum?

8:42

Was ist ein Baum?

Definition 2.1 Ein endlicher geordneter Baum über dem r-Alphabet Σ ist ein Paar T=(P,t), wobei

P ⊆ N^{*}₊ eine nichtbere endl. präför-abgeschlossene Menge ist,
 t: P → Σ eine Funktion ist mit den folgenden Eigenschaften.
 (a) Wenn t(p) ∈ Σ_n, dann {j | pj ∈ P} = ∅.
 (b) Wenn t(p) ∈ Σ_m, m ≥ 1, dann {j | pj ∈ P} = {1, ..., m}.

Teil 2: endliche Bäume
Grundbegriffe
Was ist ein Baum?

8:42

Was ist ein Baum?

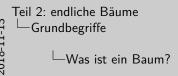
[Definition 2.1] Ein endlicher geordnoter Baum über dem r-Alphabet Σ ist ein Paar T=(P,t), wobei $(1)\ P \subseteq \mathbb{N}^n, \ \text{eine nichtleere endl. präfor-abgeschlossene Menge ist,}$

(2) t : P → Σ eine Funktion ist mit den folgenden Eigenschaften.
 (a) Wenn t(p) ∈ Σ₀, dann {j | pj ∈ P} = ∅.
 (b) Wenn t(p) ∈ Σ_m, m≥ 1, dann {j | pj ∈ 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 e) (2) (a) und (b) sagen: Stelligkeit des Zeichens an Position p muss mit der Anzahl der Kinder von p übereinstimmen.

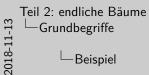


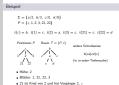
8:45

Was ist ein Baum?

Bezeichnungen

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





Teil 2: endliche Bäume
Grundbegriffe
Bottom-up-Baumautomaten

(Solimators 22): En sixties, Estima op Antienet and soof, growth Ellaness (MERA): tein Quadright $A = (Q, \Sigma, \Delta, F)$, which A = Q and $A = (Q, \Sigma, \Delta, F)$, which A = Q are solition inclusion Estimation large A at A in an Margor to the different purpose the form A and Margor to the different purpose the form A and A in an A in A

Bottom-up-Baumautomaten

8:50

Keine Anfangszustände – wir sehen gleich, warum.

Teil 2: endliche Bäume
Grundbegriffe
Bottom-up-Baumautomaten: Intuitionen

Bottom-up-Baumautomaten: Intuitionen Bedasteng der Überführungsregeln $a(q_1,\dots,q_n) \to q$: \bullet Wenn A in Fusition p Zéchen a lisst a und in p Kindern Zutände q_1,\dots,q_n eingenommen hat, dann darf A in p Zustand q einnehmen.

8:52 bis 8:56

Fragen: Was sind dann die Anfangszustände?

Teil 2: endliche Bäume
Grundbegriffe

 igspace Bottom-up-Baumautomaten: Intuitionen

8:52 bis 8:56

Fragen: Was sind dann die Anfangszustände?

Bottom-up-Baumautomaten: Intuitionen

Bedeutung der Überführungsregeln $a(q_1, \dots, q_m) \rightarrow q$: • Wenn A in Position p Zeichen a liest a und in p's Kindern Zustände q_1, \dots, q_m eingenommen hat,

dann darf A in p Zustand q einnehmen.

→ Andere Betrachtungsweise:

A markiert Eingabebaum T bottom-up mit Zuständen
 A akzeptiert T, wenn A in der Wurzel einen akzeptierenden

Zustand einnimmt

Teil 2: endliche Bäume

CTGrundbegriffe

Bottom-up-B

—Bottom-up-Baumautomaten: Intuitionen

Bioducture for Uberlühmungseignis $a(q, \dots, q_0) \rightarrow q$: • Wenn A in Positicis p Zichens Positisto p Zichens in Statistic a und in p's Kindern Zustände q_1, \dots, q_m eingenommen hut, dann darf A in p Zostand q einnehmen. 72.1 • Andrew Bestzchlüngsweisie: • A muskiert Einglachsum T bettam- up mit Zuständen q als zugestiert, vann A in der Wurzel einen aksugeierenden

Zustand einnimmt

Was sind dann die Anfanoszustände?

Bottom-up-Baumautomaten: Intuitionen

8:52 bis 8:56

Fragen: Was sind dann die Anfangszustände?

Teil 2: endliche Bäume
Grundbegriffe
Bottom-up-B

Bottom-up-Baumautomaten: Intuitionen

8:52 bis 8:56

Fragen: Was sind dann die Anfangszustände?

Bottom-up-Baumautomaten: Intuitionen
Bedeutung der Überführungsregeln $a(q_1, ..., q_m) \rightarrow q$:

Wenn A in Position p Zeichen a liest
 a und in p's Kindern Zustände q₁,..., q_m eingenommen hat,

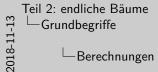
dann darf A in p Zustand q einnehmen. T2. \sim Andere Betrachtungsweise:

A markiert Eingabebaum T bottom-up mit Zuständen
 A akzeptiert T, wenn A in der Wurzel einen akzeptierenden

Was sind dann die Anfangszustände? a 0-Regeln $a() \rightarrow q$ deklarieren "zeichenspezifische" AZ:

Zustand einnimmt

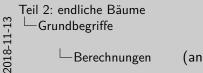
a 0-Regein $a() \to q$ deklarieren "zeichenspezifische" A darf in mit a markierten Blättern in q starten u Kurzschreibweise: $a \to q$



(analog zu NEAs)

Berechnungen (analog zv MEA) (Distincte Ξ So $A = (\Omega, T, \Delta, F)$ on NEBA and T = (P, t) sin Σ -Basen. \bullet En Rau von A and T is sin FBi. $t: P \to Q$ mit: \bullet Where $t(p) = t \in T_t$ and t(p) = t does $s \to q \in \Delta$.

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



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

(analog zu NEAs)

Berechnungen (axiog z_0 NEA)

[Definition 2.3

Sid $A = (Q, \Sigma, A, F)$ on NEBA and T = (P, r) oin Σ -Blaum.

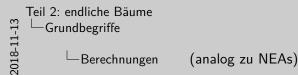
Let Be Bar was A and T at since $\text{Fit}, r: P \to Q$ mit.

Also $(p) = x \in \Sigma$, and $(p) = x \in \Sigma$, and $(p) = x \in \Sigma$.

When $(p) = x \in \Sigma$, and $(p) = x \in \Sigma$.

When $(p) = x \in \Sigma$, and $(p) = x \in \Sigma$.

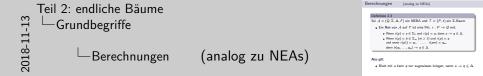
and when $(p) = x \in \Sigma$, and $(p) = x \in \Sigma$.



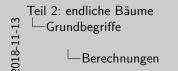
(analog zu NEAs) Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum. • Ein Run von A auf T ist eine Fkt. $r: P \rightarrow Q$ mit: • Wenn $t(p) = a \in \Sigma_0$ and r(p) = q, dann $a \to q \in \Delta$. . Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ \mathrm{und} \ r(p) = q$ und wenn $r(p1) = q_1, \dots, r(pm) = q_m$ dann $b(q_1, \dots, q_m) \rightarrow q \in \Delta$.

Berechnungen

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



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



(analog zu NEAs)

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

Berechnungen (analog zu NEAs)

Definition 2.3 Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum.

Also gilt:

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

Berechnungen, Akzeptanz und erkannte Sprache Teil 2: endliche Bäume -Grundbegriffe Sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA und T = (P, t) ein Σ -Baum. \bullet Ein Run von $\mathcal A$ auf T ist eine Funktion $r:P\to Q$ mit: • Wenn $t(p) = a \in \Sigma_0$ and r(p) = q, dann $a \rightarrow q \in \Delta$. • Wenn $t(p) = b \in \Sigma_m \ (m \ge 1) \ und \ r(p) = q$ und wenn $r(p1) = q_1, \dots, r(pm) = q_m$ dann $b(q_1, \dots, q_m) \rightarrow q \in \Delta$. -Berechnungen, Akzeptanz und erkannte Sprache \bullet Ein Run r von A auf T ist erfolgreich, wenn $r(\varepsilon) \in F$. ${\bf u}$, ${\cal A}$ altoptiert ${\cal T}$, wenn es einen erfolgreichen Run von ${\cal A}$ auf ${\cal T}$

9:04

u Die von "A erkannte Sprache ist $L(A) = \{T \text{ über } \Sigma \mid A \text{ akzeptiert } T\}.$

Beispiel 1
$$\begin{split} & \text{Set } \Sigma = \{ \rho | 2, b | 0, c | 0 \} \text{ and } A = \{ \{ q_0, \ldots, q_t \}, \Sigma, \Delta, \{ q_t \} \} \\ & \text{mit } \Delta = \{ b \rightarrow q_t, c \rightarrow q_t, A_{(q_t, q_t)} \rightarrow q_t, A_{(q_t, q_t)} \rightarrow q_t, A_{(q_t, q_t)} \rightarrow q_t, A_{(q_t, q_t)} \rightarrow q_t \}. \end{split}$$

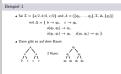
9:05

2018-11-13

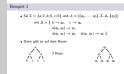
Teil 2: endliche Bäume



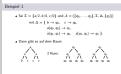
9:05



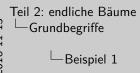
9:05

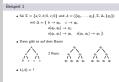


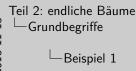
9:05

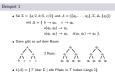


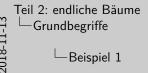
9:05

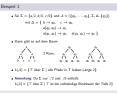


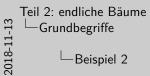












Sei $\Sigma = \{a/2,\ b/1,\ c/0,\ d/0\}$. Welcher NEBA erkennt $\{T \text{ liber } \Sigma \mid \text{ jedes c-Blatt hat ein rechtes } d\text{-Geschwister}\}$?

Beispiel 2

9:08 bis 9:15

Beispiel 2 Sai $\Sigma = (a/2, b/1, c/0, d/0)$. Withour NEBA orients (Γ dies Σ | joins cillist that sin resolute of Garchinister)? $A = (\{q_0, q_0, \psi\}, \Sigma, \Delta, \{\phi\})$ not $\Delta = \{q_0, q_0, \psi\}, \Delta, \{\phi\})$ orients $A \to \phi$, $A \to \phi$

9:08 bis 9:15

Beigniel 2 $\begin{aligned} & \text{Sis} \ \Gamma - \{a/2, b/1, c/0, d/0\}. \ \text{Wolker NEBA released} \\ & \text{T disor} \ \Gamma \ | \text{ index of Blain that sis reclass a Grandwissed} \}^2 \\ & A = \{\{c, u, w, y\}, \Gamma, \Delta, \{\psi\}\} \text{ max} \\ & A = \{\{c, u, w, y\}, \Gamma, \Delta, \{\psi\}\} \text{ max} \\ & A = \{c, u, w, v\}, \Delta, d, v\}, \psi, \\ & A = \{c, w, v\}, \Delta, v\}, \psi, \lambda \in A \rightarrow \psi, \lambda \in A$

9:08 bis 9:15

Beispiel 2 $\begin{aligned} & \text{Sis } \mathbb{T} = \{a/2, b/1, c/0, d/0\}. & \text{Widels NEBA element} \\ & \text{C filter } \mathbb{E} \mid \text{poinc cBlist hat als release a Gauchinister}\} \end{aligned}$ $& A = \{\{c, c, c, \phi\}, \mathbb{E}, \Delta, \{\phi\}\} \text{ mix}$ $& A = \{\{c, c, d, \phi\}, \mathbb{E}, \Delta, \{\phi\}\} \text{ mix}$ $& A = \{c, c, d, \phi\}, \mathcal{E}, \Delta, \{\phi\}, \phi\}, \phi\}, \quad A = \{c, c, \phi\}, \mathcal{E}, \mathcal{E},$

Beispielbaum und -run: siehe Tafel

9:08 bis 9:15

9:15, 5 min Pause bis 9:20

Erkennbare Baumsprache

ist eine erkennbare Baumsprache, wenn es einen NEBA A eibt mit L(A) = L. Teil 2: endliche Bäume
Grundbegriffe
Determinismus

9:20

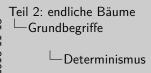
Determinismus, "höchstens eine Regel ...":

Papierkorbzustand q^- ist hier umständlicher als bei NEAs:

- ullet mehr Kombis $a(q_1,\ldots,q_m) o q^-$
- ullet man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) o q^-$

Beispiele: beide Aut. sind NEBAs.

Frage stellen!



Determinismus

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

- → Nachfolgezustand f
 ür jedes (m + 1)-Tupel a(q₁,...,q_m) ist eindeutig bestimmt (wenn er existiert)
- a Jeder DEBA ist ein NEBA, aber nicht umgekehrt (z.B. die vergangenen 2 Beispiele).

¹bler "höchstens eine" statt "genau eine": vermeldet Papierkorbzustand

9:20

Determinismus, "höchstens eine Regel ...":

Papierkorbzustand q^- ist hier umständlicher als bei NEAs:

- ullet mehr Kombis $a(q_1,\ldots,q_m) o q^-$
- \bullet man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) \to q^-$

Beispiele: beide Aut. sind NEBAs.

Frage stellen!

Teil 2: endliche Bäume
Grundbegriffe

Determinismus

1hler "höchstens eine" statt "genau eine": vermeidet Papierkorbzustan

9:20

Determinismus, "höchstens eine Regel ...":

Papierkorbzustand q^- ist hier umständlicher als bei NEAs: • mehr Kombis $a(q_1, \dots, q_m) \to q^-$

• man brauchte bei m-stelligen Zeichen a lange Regel $a(q^-,\ldots,q^-) \to q^-$

Beispiele: beide Aut. sind NEBAs.

Frage stellen!



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

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

Teil 2: endliche Bäume
Grundbegriffe

—Potenzmengenkonstruktion

Potent mengen konstruktion heriott. Mar her

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

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

Teil 2: endliche Bäume
Grundbegriffe
Potenzmengenkonstruktion

Potent mengenkonstruktion histori tal histori tal A_i^{ab} mit $A_i^{c,d}$ = $A_i^{c,d}$ mit

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

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

Teil 2: endliche Bäume

CH-Charakterisierungen erkennbarer Baumsprachen

Und nun ...

Und nun ...

Teil 2: endliche Bäume

└─Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

Beliphic: $\mathbf{v} \cdot \mathsf{Alphabet} \ \Sigma = \{s/2, \ b/1, \ c/0\}$ $\diamond \ \mathsf{Baumautomat} \ \mathcal{A} = \{\{q_0, q_1\}, \Sigma, \Delta, \{q_0\}\} \ \mathsf{mit}$ $\Delta = \{c \rightarrow q_1, \ b(q_1) \rightarrow q_1, \ s(q_0, q_0) \rightarrow q_1, \ b(q_1) \rightarrow q_0, \ s(q_1, q_1) \rightarrow q_0\}.$ $\sim L(A) =$

	Teil 2: endliche Bäume — Charakterisierungen erkennbarer Baumspracher
;	·

-Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

Bisepair.

• Alphabet $\Sigma = \{s/2, b/1, c/0\}$ • Baumautomat $A = \{q_0, q_1\}, \Sigma, \Delta, \{q_0\}\}$ mit $\Delta = \{c \rightarrow q_0, b(q_0) \rightarrow q_1, s(q_0, q_0) \rightarrow q_1, s(q_0, q_0) \rightarrow q_1, s(q_0, q_0) \rightarrow q_0, s(q_0, q_1) \rightarrow q_0\}.$ $\sim L(A) = \{T \mid \text{alle Wirral Buller Pades in T hadan sexule Lines}\}.$

eil 2: endliche Bäume		
—Charakterisierungen	erkennbarer	Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

w s-Alphabet $\Sigma = \{a/2, b/1, c/0\}$ ø Baumantomat $A = \{\{a_0, \eta_1\}, \Sigma, \Delta, \{q_0\}\} \text{ mit}$ $\Delta = \{c \rightarrow q_1, \delta(q_0) \rightarrow q_1, A(q_0, q_0) \rightarrow q_1, \delta(q_0) \rightarrow q_0, \delta(q_1, q_1) \rightarrow q_0\}$. $\sim L(A) = \{T \mid \text{alle Wurst-Bitts-Pfade in } T \text{ haben gurade Lings}\},$ $\neq \{T \mid T \text{ hat greade Hilbs}\}$

Teil 2: endliche Bäume -Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

Nichterkennbare Baumsprachen: Intuitionen

 \mathbf{v} r-Alphabet $\Sigma = \{a/2, b/1, c/0\}$ • Baumautomat $A = (\{q_0, q_1\}, \Sigma, \Delta, \{q_0\})$ mit $\Delta = \{c \rightarrow q_0, b(q_0) \rightarrow q_1, a(q_0, q_0) \rightarrow q_1,$ $b(q_1) \rightarrow q_0$, $a(q_1, q_1) \rightarrow q_0$ } → L(A) = {T | alle Wurzel-Blatt-Pfade in T haben gerade Länge}. ≠ {T | T hat gerade Höhe}

Frage: Sind die folgenden Baumsprachen (über Σ) erkennbar? $L_1 = \{T \mid T \text{ hat gerade Höhe}\}$

L2 = {T | T ist vollständiger Binärbaum} T2.4 Forts

2018-11-13

Teil 2: endliche Bäume

Charakterisierungen erkennbarer Baumsprachen

Nichterkennbare Baumsprachen: Intuitionen

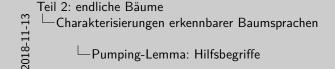
Nichterkennbare Baumsprachen: Intuitionen

• c-Alphabet $\Sigma = \{s/2, b/1, c/0\}$ • Baumautomat $A = \{\{q_0, q_1\}, \Sigma, \Delta, \{q_0\}\} \text{ mit}$ $\Delta = \{c \rightarrow q_0, b(q_0) \rightarrow q_1, s(q_0, q_0) \rightarrow q_1, b(q_1) \rightarrow q_2, s(q_1, q_1) \rightarrow q_0\}.$ $\sim L(A) = \{T \mid \text{alle Warral-Butt-Pfade in } T \text{ haben gerade Lingus}\}$ $\neq f \mid T \mid T \text{ hat gended Helbar}$ T2.4

Frage: Sind die folgenden Baumsprachen (über Σ) erkennbar? $L_1 = \{T \mid T \text{ hat gerade Höhe}\}$ $L_2 = \{T \mid T \text{ ist vollständiger Binlerbaum}\}$ T2.4 Forts

Antwort: Nein.

T 2.4 Forts.



nestens von Blamen inisiaande:

• Wariable: zusätzlichen nullstelligen Symbol $x \notin \Sigma_0$ • (univer) Kontect:

Baum über $\Sigma \cup \{x_i\}$, in dem ein Blatt mit x markiert ist $\mathbf{T2.5}$ • brückler Kontect G_V Kontect der Hölse $\mathbb O$ (\Rightarrow nur Wurzelf)

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). Teil 2: endliche Bäume

Charakterisierungen erkennbarer Baumsprachen

Pumping-Lemma: Hilfsbegriffe

Pumping Lemma: Hillbegriffe
Einstran on Blamm instander

(width: radiction milestiffer
(width: radiction milestiffer
(width) Texture
Blam ider X. (i, i), in den in Blatt mix i marken in it 72.3

(which radiction i), which will blatt mix i marken in 172.3

(which radiction i), Konten der Filler i) or i marken in it 72.3

(which radiction i), Konten der Filler i) on an of white interest i), and i marken in the Filler i), when i me of Filler i) is a considerable in Kontentia.

(i) of i) and i me of Filler i me of i me of

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

Pumping-Lemma

Beweis des Pumping-Lemmas Brauck. Sei L eine erkennbare Baumsprache, und sei $A=(Q,\Sigma,\Delta,F)$ ein NEBA mit L(A)=L.

8:32

Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA mit L(A) = L. Wir wählen k = |Q|. Teil 2: endliche Bäume
Charakterisierungen erkennbarer Baumsprachen
Beweis des Pumping-Lemmas

Beweit. Sei L eine erkennbare Burmsprache, 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 $\geq k$, und sei τ ein alzespteirender Run von \mathcal{A} auf T.

Beweis des Pumping-Lemmas

8:32

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

8:32

Beweis des Pumping-Lemmas

Bewels. 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 $\geq k$, und sei r ein akzeptierender 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 $p_1 \neq p_2$ mit demselben Zustand, d.h. $r(p_1) = r(p_2) = q$ für ein $q \in Q$.

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

Beweis des Pumping-Lemmas

Beweis. Sei L eine erkennbare Baumsprache, und sei $A = (Q, \Sigma, \Delta, F)$ ein NEBA mit L(A) = L. Wir wählen k = |Q|

Sei $T = (P, t) \in L$ ein Baum mit Höhe $\geq k$. und sei r ein akzeptierender 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. $r(p_1) = r(p_2) = q$ für ein $q \in Q$.

O. B. d. A. ist $\rho_2=\rho_1\rho_3$ für ein $\rho_3\neq\varepsilon$.

8:32

Beweis des Pumping-Lemmas $U = T_{p_1}$ $U = T_{p_2}$ C = dispinsipe Kontant mit <math>C[U] = T $V = T_{p_1}$ D = dispinsipe Kontant mit <math>U = D[V]

Severis des Pumping-Lemmas $\begin{array}{ll} U = T_n \\ C = \operatorname{dispinige} \operatorname{Kontext} \operatorname{mix} \mathbb{C}[U] = T \\ V = T_n \\ D = \operatorname{dispinige} \operatorname{Kontext} \operatorname{mix} U = D[V] \\ \end{array}$ (Weil $p_1 \neq p_2$, ist D michterhold, also $D \neq C_0$ wie gefordert.

Solinn num: $U = T_h$ C = dispingly Kontont mit C[U] = T $V = T_h$ D = dispingly Kontont mit U = D[V] $\text{Well } p_1 \neq p_2 \text{ in D orbitroinal, also } D \neq G_0 \text{ wis genforter.}$ Date of it and chat T = C[G[V]]. T2.6 Fort.

Beweis des Pumping-Lemmas

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

Beweis des Pumping-Lemmas $\begin{aligned} &\text{Noch is a singles} \quad T_i = \zeta[P[V]] \in L \text{ for alls } i \geq 0. \end{aligned}$ $1. \text{ Fat } i = 0, \text{ sind } T_i = \zeta[V]. \qquad \qquad \text{T 2.6 Forts.}$ $Definisers Ran a particionesiste: \\ &n(s) = \begin{cases} r(s) & \text{fith p also including one p, let} \\ c(s)s^{-1} & \text{fith p also including one p, let} \end{cases}$



Browels des Pumping: Lemmas $\begin{aligned} & \operatorname{Hoch} \text{ as a signe. } \ T_i = \mathcal{L}[P]/V] \in L \text{ for alls } i \geq 0. \\ & L \text{ Rel } I = \ell \text{ also } T_i = \mathbb{C}[V]. \end{aligned} \qquad \begin{array}{c} & \text{T2.6 Feets.} \\ & \text{Defineme Run on positionswises} \\ & \sigma(\rho) = \binom{\rho(\rho)}{\ell(\rho)} \quad \text{disk } p \text{ aim Nuchtdage view } \rho_i \text{ six} \\ & \varepsilon(\rho) = \binom{\rho(\rho)}{\ell(\rho)} \quad \text{disk } p \text{ aim Nuchtdage view } \rho_i \text{ six} \\ & \varepsilon(\rho) = \binom{\rho(\rho)}{\ell(\rho)} \quad \text{disk } \rho \text{ aim And All } \mathcal{I}_i \\ & \text{ aim in disligation.} \end{aligned}$

Beweis des Pumping: Lemmas $\begin{aligned} &\text{Noch as asign} \quad T_i = \zeta[D^i[V]] \in L \text{ for alls } i \geq 0. \end{aligned}$ $1. \text{ Fat } i = 0. \text{ sin } T_i = \zeta[V] \qquad \qquad \text{T2.6 Forts.}$ Definisers fin n positionersities $a(p) = \begin{cases} (p) & \text{ sin } p \text{ sin Nucleotyre on } p_i \text{ sit } \\ \zeta(p_i) & \text{ sin } p \text{ sin Nucleotyre on } p_i \text{ sit } \end{cases}$ (*) Lector as police v_i is the finance of and T_i on its ordiginals. The expect v_i is the following v_i (*) at $u_i(p) = (p_i)$.

Also $T_0 \in L$

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

8:47

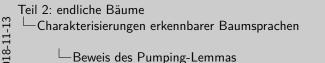
Noch zu zeigen: $T_j:=C[D[V]]\in L$ für alle $i\geq 0$. 2. Fals: $i\geq 1$. T.2.6 Fc. Definieren Run r_j positionsowies: $r_0(p)=\begin{cases} r_0(p) & \text{falls } p \text{ ion Nachdolger von } p_j \text{ ist} \\ r_0(p_j)^2 & \text{falls } p \text{ ion Nachdolger von } p_j \text{ int} \\ r_0(p_j)^2 & \text{falls } p = p_j p_j^2 p_j^2 \text{ } p^2 \text{ Asin NF von } p_j p_j^2 \\ r_0(p_j)^2 & \text{falls } p = p_j p_j^2 p_j^2 \end{cases}$

Beweis des Pumping-Lemmas

8:47

Beweis des Pumping-Lemmas

8:47



Note he a seigen: $T_i := C[D[V]] \in L$ für alls $i \ge 0$. 2. Palt: $i \ge 1$. Definitions Run c positionsessis: $a(p) = \begin{cases} c(p) & \text{falls } p \text{ sain Nachfolger von } p_i \text{ six} \\ c(p) & \text{falls } p \text{ m. } p_i p_i^{i} p_i^{i} \text{ of sin NF von } p_i \text{ sur} \\ c(p) & \text{falls } p \text{ m. } p_i p_i^{i} p_i^{i} \text{ of sin NF von } p_i \text{ six} \end{cases}$ Wu in 1. Palt: c_i is orbipricher Run von A and T_i . Also $T_i \in L$.

Beweis des Pumping-Lemmas

8:47

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

Anwendung des Pumping-Lemmas

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

dann ist L keine erkennbare Baumsprache.

Benutzen Kontraposition (siehe Kapitel "endliche Wörter"): Wenn es für alle Konstanten $k \in \mathbb{N}$ einen Baum $T \in L$ mit Höhe $(T) \ge k$ gibt, so dass es

Der Satz von Myhill-Nerode für Baumsprachen Ziel: notwerfige und himrichende Bedingung für Erksenharkeit Delminson 28. Sie 4 den Baumsprach über Σ . Zomi Σ Blamm T_1, T_2 und 4, 3 glankeitet (Schweber: $T_1 \sim_4 T_2$), weren für all Σ Kontrott C gitt: $C[T_1] \in L$ genau dann, wenn $C[T_2] \in L$

9:14

Satz ohne Beweis. An Tafel nur Bsp.

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 T1, To sind L-ätuivalent (Schreibw.: T1 ~1 To). wenn für alle Σ-Kontexte C gilt:

 $C[T_1] \in L \quad \text{genau dann, wenn} \quad C[T_2] \in L$

T 2.8

 $L \subset \Sigma^*$ is NEBA-erkennbar gdw. \sim_L endlichen Index hat

9:14

Satz ohne Beweis. An Tafel nur Bsp.

Ziel: notwendige und hinreichende Bedingung für Erkennbarkeit Definition 2.8 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: $C[T_1] \in L \quad \text{genau dann, wenn} \quad C[T_2] \in L$

Der Satz von Myhill-Nerode für Baumsprachen

 $L \subset \Sigma^*$ is NEBA-erkennbar gdw. \sim_L endlichen Index hat

Auch für Baumsprachen gilt: endlicher Index n von \sim_L = minimale Anzahl von Zuständen in einem DEBA, der L erkennt

9:14

Satz ohne Beweis. An Tafel nur Bsp.

Teil 2: endliche Bäume

Top-down-Baumautomaten

Und nun ...

Und nun ...

Teil 2: endliche Bäume

Top-down-Baumautomaten

−Drehen wir jetzt alles um? ©

Teil 2: endliche Bäume

Top-down-Baumautomaten

Top-down-Baumautomaten

16:00

Top-down-Baumautomaten

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

Ein nichtdet. Top-down-Automat auf endl. geord. Bäumen (NETDBA) ist ein Quadrupel $\mathcal{A} = (Q, \Sigma, \Delta, I)$, wobei \square Q eine endliche nichtleere Zustandsmense ist.

Q eine endliche nichtleere Zustandsmenge ist,
 Σ ein r-Alphabet ist.

Δ eine Menge von Überführungsregeln der Form

 $(a,q)\to (q_1,\dots,q_m)$ ist mit $m\geqslant 0, \quad a\in \Sigma_m, \quad q,q_1,\dots,q_m\in Q, \quad \text{und}$

u / ⊆ Q die Menge der Anfangszustände ist.

Teil 2: endliche Bäume

Top-down-Baumautomaten

Serechnungen, Akzeptanz und erkannte Sprache

[Ontenze 231]
St 4 = (O. E.A.) + on NETORA + on T = (P. 1) on E Elemente (Ren) + on A and T ist eine Filt. f | P → O onte

Teil 2: endliche Bäume

Top-down-Baumautomaten

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache Dolleisen 211 Sei A=(D,Z,L) ein NETDBA und T=(P,t) ein S.-Baum. \bullet Boschmag (Ban) von A auf T ist eine Fei. $r:P\to Q$ mix $e^{i}(P)$ = I . When $e^{i}(p)$ = $a\in \Sigma$. I ($a\in X$) I is the feil i . When $e^{i}(p)$ = $a\in \Sigma$. I ($a\in X$) I is an $e^{i}(p)$ = $a\in X$ and when $e^{i}(p)$ = $a\in X$. I ($a\in X$) I is an $e^{i}(p)$ = $a\in X$.

Teil 2: endliche Bäume

Top-down-Baumautomaten

-Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA and T = (P, t) ein Σ -Baum. • Boechnung (Run) von A auf T ist eine Fkt. $r : P \rightarrow Q$ mit: • $r(r) \in I'$ • When $t(p) = x \in \Sigma_m$ $(m \ge 1)$ and r(p) = qund were $r(p)1 = q_0, \ldots, r(pm) = q_m$ dan light is a sim Rugh $(A, Q) \rightarrow (q_0, \ldots, q_m) \in \Delta$

Teil 2: endliche Bäume

Top-down-Baumautomaten

Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Interstance 2.11

So $A = \{0, \Sigma, \Delta, \ell\}$ sin NETDBA and $T = \{P, t\}$ sin Σ -Baum.

Benchmary (Rim) von A and T ist sine Fat. $t : P \to Q$ mix: $s : \ell_1 \geq t = \ell$ When $t(\rho) = s \in \Sigma_m (m \geq 1)$ and $t(\rho) = q$ and when $t(\rho) = q$. $t_m (m \geq 1)$ and $t_m (\rho) = q$ dues given $t_m (\rho) = q$. $t_m (\rho) = q$.

dang bit $t_m (\rho) = q$. $t_m (\rho) = q$. $t_m (\rho) = q$. When $t(\rho) = s \in \Sigma_m (n \mid \ell_1) = q$. $t_m (n \mid \ell_1) = q$.

Berechnungen, Akzeptanz und erkannte Sprache

Detection 2.11

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

Beschnung (Run) von A auf T ist eine Fkt. $r : P \rightarrow Q$ mit:

• $r(r) \in I$ When $t(p) = s \in \Sigma_m$ ($m \ge 1$) und r(p) = qund wern $r(p1) = q_0, \dots, r(pm) = q_m$ dan light se nim Ruga $(A, p) + (q_0, \dots, q_m) \in \Delta$.

When $t(\rho)=a\in \Sigma_0$ and $r(\rho)=q$, dann $(a,q)\to ()\in \Delta$. • $\mathcal A$ algorithm $\mathcal T$, when is either Run von $\mathcal A$ auf $\mathcal T$ gibt.

Berechnungen, Akzeptanz und erkannte Sprache

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

• Berechtung (Run) von A auf T ist eine Fkt. $r : P \rightarrow Q$ mit:

• $r(r) \in I$ • When $t(p) = s \in \Sigma_m \ (m \ge 1)$ und r(p) = qund wern $r(n) 1 = 0, \dots, r(nm) = q$.

Wenn t(p) = a ∈ Σ_m (m ≥ 1) and τ(p) = q
 und worm τ(p1) = q, ... τ(pm) = qm
 dann gibt as sinn Regal (a, q) → (q₁, ..., q_m) ∈ Δ.
 Wenn t(p) = a ∈ Σ_q and τ(p) = q, dann (a, q) → () ∈ Δ.
 A alxeption T, wonn as sinon Run von A auf T gibt.

a Die von A erkannte Sprache ist $L(A) = \{T \text{ über } \Sigma \mid A \text{ akzeptiert } T\}.$

Definition 2.11

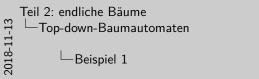
Berechnungen, Akzeptanz und erkannte Sprache

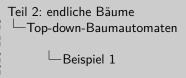
Definition 2.11 Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA und $\mathcal{T} = (P, t)$ ein Σ -Baum. • Borechnung (Run) von \mathcal{A} auf \mathcal{T} ist eine Fkt. $r: P \rightarrow Q$ mit: • $r(t) \in I$ • When $t(p) = s \in \Sigma_m$ ($m \ge 1$) and r(p) = q

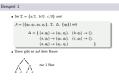
und weren $r(p1) = q_1, \dots, r(pm) = q_m$ dann gibt en eine Regul $(x, q) \mapsto (q_1, \dots, q_m) \in \Delta$. When $t(p) = a \in \Sigma_q \text{ und } r(p) = q, \text{ dann } (a, q) \to \{\} \in \Delta$. A Akceptiert T, ween es einem Run von A auf T gibt. a Die von A erkkante Sprache ist

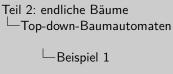
L(A) = { T über $\Sigma \mid A$ akzeptiert T}.

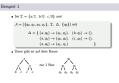
Beachte: Keine Endzustände nötig – die Regeln in Δ müssen nur erlauben, von der Wurzel bis zu allen Blättern "durchzukommen".

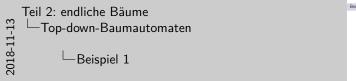








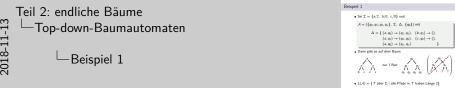


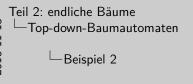


Beispiel 1

• Su $\Gamma = \{a/2, b/0, c/0\}$ and $A = \{(a_0, a_1, a_2, a_3), \Gamma, \Delta, \{a_3\}\} \text{ mix}$ $\Delta = \{(a, a_3) + (a_0, a_1), (b, a_2) + (b, a_3) + (b, a_3)$

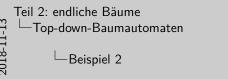






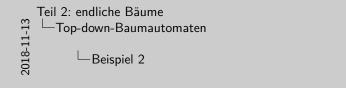
Beispiel 2

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



Beispiel 2 $Sa \ \Sigma = \{a(2,b), 1, c(0,d)0\}. \ \ Widther NETDBA enhance \\ L_{d} = \{T \ dow \ \Sigma \ | \ plan c \ of Hun har do welfase $\mathcal{L}(a,d) \text{-width} \}$ $NETDBA \ \mathcal{A} = \{\{a, a_0, a_0, b_1, \Sigma, \mathcal{L}, \{a_0\}\} \text{ osc } \\ \Delta = \{(a, a_0 - (a_0, a_0), K_0) \rightarrow a_0, (c, a_0) \rightarrow 1\}, \\ (a, a_0) \rightarrow \{a_0, c_0\}, K_0 \rightarrow a_0, (c, a_0) \rightarrow 1\}, \\ (a, a_0) \rightarrow \{a_0, c_0\}, K_0 \rightarrow a_0, (c, a_0) \rightarrow 1\}$





Beispiel 2

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

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

Was sart uns das über das Verhältnis NETDBAs : NEBAs?

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

 $d \rightarrow q_d$, $d \rightarrow q_f$ }

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

NETDBAs vs. NEBAs NETDBAs and gloid-micrology Sees 222 See 222 See 222 See 222 See 242 See 24

 $16{:}15 \rightarrow \text{bis } 16{:}20$

 $16{:}15 \rightarrow \text{bis } 16{:}20$

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtigl

 $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}$ Beweis. Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruisren NEBA $A^{\dagger} = (Q, \Sigma, \Delta^{\dagger}, F^{\dagger})$ mit:

 $\begin{array}{lll} \Delta^{\uparrow} &=& \{a(q_1,\ldots,q_m) \rightarrow q) &|& (a,q) \rightarrow (q_1,\ldots,q_m) \in \Delta \} \\ F^{\uparrow} &=& I \end{array}$

 $16{:}15 \rightarrow \text{bis } 16{:}20$

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtigl

Satz 2.12 $\{L(A) \mid A \text{ ist ein NETDBA}\} = \{L(A) \mid A \text{ ist ein NEBA}\}.$ Beweik. Sei $A = (Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $A^{\uparrow} = (Q, \Sigma, \Delta^{\uparrow}, F^{\uparrow})$ mit: $\Delta^{\uparrow} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (a, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$ $E^{\uparrow} = I$

Dann ist jeder Run von A auf einem Σ -Baum Tauch ein **crfolgreicher** Run von A^{\dagger} auf Tund umgekehrt.

 $16{:}15 \rightarrow \text{bis } 16{:}20$

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 $A=(Q, \Sigma, \Delta, I)$ ein NETDBA.

Konstruieren NEBA $A^{\dagger} = (Q, \Sigma, \Delta^{\dagger}, F^{\dagger})$ mit: $\Delta^{\dagger} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (a, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$

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

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

 $16:15 \to bis 16:20$

NETDBAs vs. NEBAs

NETDBAs und NEBAs sind gleichmächtig!

Beweis. Sei $\mathcal{A} = (Q, \Sigma, \Delta, I)$ ein NETDBA. Konstruieren NEBA $\mathcal{A}^{\dagger} = (Q, \Sigma, \Delta^{\dagger}, F^{\dagger})$ mit: $\Delta^{\dagger} = \{a(q_1, \dots, q_m) \rightarrow q) \mid (a, q) \rightarrow (q_1, \dots, q_m) \in \Delta\}$

 $\Delta^{+} = \{a(q_1, ..., q_m) \rightarrow q) \mid (a, q) \rightarrow (q_1, ..., q_m) \in F^{\dagger} = I$ Dann ist ieder Run von A auf einem Σ -Baum T

Dann ist jeder Run von A auf einem Σ -Baum Tauch ein erfolgreicher Run von A^{\dagger} auf Tund umgeloehet. Daraus folgt $L(A^{\dagger}) = L(A)$.

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

0

Determinisierung von NETDBAs

16:20

Schauen wir uns auch hier noch die deterministische Variante von Top-down-Baumautomaten an.

-Determinisierung von NETDBAs

Determinisierung von NETDBAs Erinnerung an Beispiel 2: Sei $\Sigma = \{a/2, b/1, c/0, d/0\}$ und $L_{cl} = \{T \text{ ider } \Sigma \mid \text{ iedes } c\text{-Blatt hat ein rechtes } d\text{-Geschwister}\}$ NETDBA $A = (\{q_0, q_c, q_d\}, \Sigma, \Delta, \{q_b\})$ mit $\Delta = \{ (a, q_0) \rightarrow (q_0, q_0), b(q_0) \rightarrow q_0, (c, q_c) \rightarrow (), \}$ $(a, q_0) \rightarrow (q_c, q_d)$, $(d, q_d) \rightarrow ()$. $(d, q_0) \rightarrow ()$

16:20

Schauen wir uns auch hier noch die deterministische Variante von Top-down-Baumautomaten an.



16:20

Schauen wir uns auch hier noch die deterministische Variante von Top-down-Baumautomaten an.

Determinisierung von NETDBAS $\begin{aligned} & \text{Einsense} & \text{a Reight 2.} & \text{ Set } \Sigma_{-}\left(x_{i}^{2}, h_{i}^{2}, c_{i}^{2}, d_{i}^{2}\right) \text{ and} \\ & L_{ii} = \left\{7 \text{ Beb T }\right\} \text{ piles clittle but in relates of Gendensterl} \\ & \text{NETDBA} & -1\left((a_{0}, a_{0}^{2}), \Sigma_{i}, \delta_{i}, a_{0}^{2}\right) \text{ int} \\ & -2\left(a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) \\ & -2\left(a_{0}^{2}, a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) + \left(a_{0}^{2}, a_{0}^{2}\right) \\ & \text{We wisse } j_{i}, \text{ with me N-Matheteristicates } \end{aligned}$

16:20

Schauen wir uns auch hier noch die deterministische Variante von Top-down-Baumautomaten an.

Teil 2: endliche Bäume Top-down-Baumautomaten

-Determinisierung von NETDBAs?

Determinisierung von NETDBAs? • $\Sigma = \{a/2, b/0, c/0\}$ und die erkennbare Baumsprache L = {a(bc), a(cb)}. (denke an die alternative Schneibweise von Folie 18) Frage: Welcher DETDBA erkennt L?

16:23 bis 16:30

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

-Determinisierung von NETDBAs?



16:23 bis 16:30

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

16:23 bis 16:30

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

Teil 2: endliche Bäume

CHAbschlusseigenschaften

Und nun ...

Und nun ...

Teil 2: endliche Bäume

Abschlusseigenschaften

Operationen auf Baumsprachen

16:30

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 $L_1 \cap L_2$. Teil 2: endliche Bäume

Abschlusseigenschaften

Operationen auf Baumsprachen

16:30

Operationen auf Baumsprachen

Zur Einnenung, die Mineg der einenbarer Baumsprachen heißt.
abgeschissen und zur

Bellen der Bernellen der Bellen der Bellen

Komplement? Schnitt? Teil 2: endliche Bäume

—Abschlusseigenschaften

—Operationen auf Baumsprachen



Teil 2: endliche Bäume

Abschlusseigenschaften

Operationen auf Baumsprachen

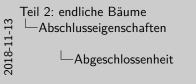


Teil 2: endliche Bäume

Abschlusseigenschaften

Operationen auf Baumsprachen





16:33

Abgeschlossenheit

Satz 2.55

De Mege der NEBA-erkenberen Sprachen ist abgischlossen unter
den Operationen L./. n. –

Driekte Konsequenz aus den folgenden Lemmatia.

Lemma 2.16 Seien A_1 , A_2 NEBAs über Σ . Dann gibt es einen NEBA A_2 mit $L(A_1) = L(A_1) \cup L(A_2)$.

Abgeschlossenheit unter Vereinigung

16:34 bis 16:38

Letzte Zeile:

Zeige: jeder Run von \mathcal{A} auf T entspricht einem Run von \mathcal{A}_1 oder \mathcal{A}_2 , und umgekehrt.

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

16:34 bis 16:38

Letzte Zeile:

Zeige: jeder Run von A auf T entspricht einem Run von A_1 oder A_2 , und umgekehrt.

16:34 bis 16:38

Letzte Zeile:

Zeige: jeder Run von \mathcal{A} auf T entspricht einem Run von \mathcal{A}_1 oder \mathcal{A}_2 , und umgekehrt.

16:34 bis 16:38

Letzte Zeile:

Zeige: jeder Run von \mathcal{A} auf \mathcal{T} entspricht einem Run von \mathcal{A}_1 oder \mathcal{A}_2 , und umgekehrt.

16:38 bis 16:40

 -Abgeschlossenheit unter Komplement

16:38 bis 16:40

Earma 2.17
Set A ein KEBA über Σ .
Dann gibt se seinen NEBA A'' mix $L(A'') = \overline{L(A)}$.

Bereik: analog zu NEAs.

• Ummondlung in DEBA
• Vertauschen von akkastierenden und einbt-akz. Zuständen

Abgeschlossenheit unter Komplement

Teil 2: endliche Bäume
—Abschlusseigenschaften

—Abgeschlossenheit unter Komplement

16:38 bis 16:40

Abgeschlossenheit unter Komplement (Samus, 23):

Samus, 23):
So A in NESA size X.
Done gilt is sizes NEBA A' mit $L(A') = \overline{L(A)}$.

Benick: analog in NEAE:
• Unmonfliel in OEBA A' with in initial size Z. Zodiaden
• Vertucksher vor abapterenen in nicht size Z. Zodiaden
• Vertucksher vor abapterenen Z.
Nach Sizes Z is give in OEBA $A' = (Q', X, \Delta^{A'}, F')$ in L(A') = L(A) in Z give in other in Z.

-Abgeschlossenheit unter Komplement

16:38 bis 16:40

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

Beweis: analog zu NEAs: Umwandlung in DEBA

w Vertauschen von akzeptierenden und nicht-akz. Zuständen

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

-Abgeschlossenheit unter Schnitt

16:40

Abgeschlossenheit unter Schnitt

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 ¬:

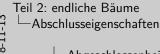
 $L_1 \cap L_2 = \overline{L_1 \cup L_2}$

16:40

Lemma 2.18 Some β_{L} is NEEAs idear Σ . Does give some NEEAs A_{L} and $L(A_{L}) = L(A_{L}) \cap L(A_{L})$. Using given some NEEAs A_{L} and $L(A_{L}) = L(A_{L}) \cap L(A_{L})$.

In fig. dimits as our A_{L} Reproductions that start $-U = ud \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L})$. Alternative A_{L} is $A_{L} \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L})$. Alternative $A_{L} \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L})$. Alternative $A_{L} \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L}) \cap L(A_{L})$.

Abgeschlossenheit unter Schnitt



-Abgeschlossenheit unter Verkettungsoperationen

Abgeschlossenheit unter Verkettungsoperationen Man kann Analoga zu - und * für Baumsprachen definieren

Randbemerkung:

• $L^* = \{C_1[C_2[...[C_n[T]]...]] \mid$

Seien L. L. L. Baumsprachen. Bezeichne Con(L) die Menge aller Kontexte, die man aus Bäumen in L erhält, indem man ein Blattsymbol durch x ersetzt. $u L_1 L_2 = \{C[T] \mid T \in L_1, C \in Con(L_2)\}$

 $T \in L$, $C_1, \dots, C_n \in Con(L)$, $n \ge 0$ }

Abreschlossenheit unter .. * kann man dann wie für NEAs zeigen. aber mit mehr technischem Aufwand (Eliminierung «-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.

Teil 2: endliche Bäume $\stackrel{\text{El}}{\sqsubseteq}$ Entscheidungsprobleme $\stackrel{\text{Und}}{\sqsubseteq}$ Und nun . . .

Und nun ...

16:50

Das Leerheitsproblem $\begin{aligned} & \text{Eighe: NEBA (oder DEBA)} , A \\ & \text{Finge: in } L(A) &= 0 \text{ F} \\ & \text{d. h. } L^{\bullet}_{\text{NEBA}} = \{A \mid A \text{ NEBA. } (L(A) = 0\} \end{aligned}$ (and/og for DEBAs) $\begin{aligned} & \text{Sing 2-19} \\ & \text{Li}_{\text{DEBA}} & \text{old statisholds und } P \text{-odital-odig.} \end{aligned}$

Teil 2: endliche Bäume
—Entscheidungsprobleme

—Das Leerheitsproblem

16:50

Das Leerheitsproblem

Englan, REBA (oder DEBA), A
Frege: Int (I.A.) = 87

d. h. Uraca = 4 | A NEBA, (I.A) = 8) (analog für DEBA)

Säüt 2:39

Elevan, and Piccus, sind witscheidhar und P-volksändig.

Brank

Entscheidshalt in Physical analog zu NEA:
prids, du ein aks. Zustand erneichbar ist (sickste Folio)

Teil 2: endliche Bäume Entscheidungsprobleme -Das Leerheitsproblem

16:50

Das Leerheitsproblem Eingabe: NEBA (oder DEBA) A Frage: lst $L(A) = \emptyset$? 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. · Entscheidbarkeit in Polyzeit analog zu NEAs:

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

Reduktion von "Solvable Path Systems" (≈ Erreichbarkeit in Hypergraphen mit ternärer Kantenrelation), siehe [7, Exercise 1.19]

Das Leerheitsproblem
Polymotisch algorithmis:

- Biersches Mengs der erreichbren Zustände
- Prüfe, ob en abzupderender Zustand darin ist.

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

Teil 2: endliche Bäume
—Entscheidungsprobleme

—Das Leerheitsproblem

Polynomials dispositions: $\theta \mbox{ Boxes Mergy determined Total sides} \mbox{ Polis, do in a Negotierneder Total sides in it. } \mbox{ Sid. } A = (\Omega, \Sigma, \Delta, E). $$$ Sid. $A = (\Omega, \Sigma, \Delta, E). $$$ Sid. $A = (\Omega, \Sigma, \Delta, E). $$$ Side $A = (\Omega, \Sigma, \Delta, E). $$$ When so $(\Omega, \Sigma, \Sigma, \Delta, E). $$$ Side $A = (\Omega, \Sigma, E). $$$ Side

Das Leerheitsproblem

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

Teil 2: endliche Bäume

—Das Leerheitsproblem

Das Leerheitsproblem
Polynomialzeitalsorithmus:

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

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

Konstruieren Menge $R \subseteq Q$ wie folgt:

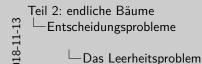
• $R := \{q \mid a \rightarrow q \in \Delta \text{ für ein } a \in \Sigma_0\}$ • Wenn es $q_1, \dots, q_m \in R$ und $a \in \Sigma_m$ gibt mit

Ween es q;..., q_n ∈ R and s ∈ L_m gist mit.
 s(q₁,...,q_n) → q ∈ Δ and q ∈ R, dann R := R ∪ {q}.
 Winderhole letzten Schritt, bis sich R nicht mehr ändert.
 Leicht zu sehen: Berechnung endet nach ≤ [Q] vielen Schritten → R ist in Polyzeit berocherbar.

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



Das Leerheitsproblem Polynomialzeitaloorithmus

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

Sei $A = (Q, \Sigma, \Delta, F)$. Konstruieren Menge $R \subset Q$ wie folgt: $a R := \{a \mid a \rightarrow a \in \Delta \text{ für ein } a \in \Sigma_0\}$

 \mathbf{u} Wenn es $q_1,\ldots,q_m\in R$ und $\mathbf{a}\in\Sigma_m$ gibt mit $a(q_1, ..., q_m) \rightarrow q \in \Delta \text{ und } q \notin R, \text{ dann } R := R \cup \{q\}.$

w Wiederhole letzten Schritt, bis sich R nicht mehr ändert Leicht zu sehen: Berechnung endet nach < |Q| vielen Schritten ~ R ist in Polyzeit berechenbar.

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

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

Das Wortproblem alias Zugehörigkeitsproblem Eingabe: NEBA (oder DEBA) \mathcal{A} . Baum T über Σ Frage: Ist $T \in L(A)$? d.h. WP_{MEBA} = $\{(\mathcal{A}, T) \mid \mathcal{A} \text{ NEBA}, T \in L(A)\}$ (analog f. DEBAs)

—Das Wortproblem alias Zugehörigkeitsproblem

17:18 bis 17:22

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

LOGCEL:

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

LOGDCFL:

– Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zugehörigkeitsproblem Eispele: NERA (dan DERA), A. Beam τ über Σ Fegre: Int $\tau \in L(A)$ and τ über Σ of the Σ Fegre: Int $\tau \in L(A)$ (τ) A. NERA, $\tau \in L(A)$) (under t DEBA) Sides 2.28 Whose, and WPossa, sind enterbuilder and 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:

– Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zügehörigkeitsproblem Eingleir. NEBA (odur DEBA) A. Baum T übr Σ Finge: in $T \in L(A, T) \mid A$ NEBA $T \in L(A)$ (swing I DEBA) S Sain 2 20 Wingsan M Wingsan solid estenhible and in P.

Brank, analog as NEA, ap Pladdelstan zum L^p $T \in L(A)$ give $L(A) \cap L(A, T) \neq \emptyset$, while A_T and DEBA int $L(A) \cap L(A, T) \neq \emptyset$.

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:

Entscheidungsprobleme

-Das Wortproblem alias Zugehörigkeitsproblem

Das Wortproblem alias Zugehörigkeitsproblem Eingabe: NEBA (oder DEBA) A, Baum T über Σ Frage: lst $T \in L(A)$? d.h. $WP_{NEBA} = \{(A, T) \mid A \text{ NEBA, } T \in L(A)\}$ (analog f. DEBAs) WPNESS und WPNESS sind entscheidbar und in P. Beweis. analog zu NEAs, per Reduktion zum LP: $T \in L(A)$ wdw. $L(A) \cap L(A_T) \neq \emptyset$. wobei A_T ein DEBA mit $L(A_T) = \{T\}$ ist (konstruiere selbst!) WPNERA ist LOGCFL-vollständig. (zwischen NL und P)

17:18 bis 17:22

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

LOGCEL:

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

LOGDCFL:

Das Äquivalenzproblem Eingabe: NEBAs (oder DEBAs) \mathcal{A}_1 , \mathcal{A}_2 Frage: Ist $L(\mathcal{A}_2) = L(\mathcal{A}_2)$? d. h. $AP_{MRM} = \{(\mathcal{A}_1, \mathcal{A}_2) \mid \mathcal{A}_1, \mathcal{A}_2 \text{ NEBAs}, L(\mathcal{A}_1) = L(\mathcal{A}_2)\}$ etc.

17:22 bis 17:25

 $\label{eq:delta} \begin{array}{l} \triangle = \text{symmetrische Differenz zweier Mengen;} \\ \text{ausdr\"{u}ckbar mittels } \cup, \cap, \bar{\boldsymbol{\cdot}} \quad \leadsto \text{Abschlusseigenschaften!} \end{array}$

Das Aquivalenceproblem Engabe: NESAs (oder DEBA), A_1 , A_2 Fuge: 1st $\{A_1, b = \{A_2\}\}$? d. h. A^{Pagas} = $\{\{A_1, A_2\} \mid A_1, A_2\}$ NEBAs, $\{A_1, b = A(A_2)\}$ sec. SSR2 21 APuras, and AP_{CRAS}, sind extributibles APuras, in ESP This concluded in EAPURAS, in ESP This APuras, in ESP This concluded in EAPURAS, in ESP This APURAS IN ESP THIS CONCLUDED.

17:22 bis 17:25

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

– Das Äquivalenzproblem

Das Aquivalentrycoblem

Englain: RURAS (soon EESRa), A_1 , A_2 Frage: Its $L(A_2) = L(A_3)$ Frage: Its $L(A_3) = L(A_3)$ An Parasa: $R(A_3) = L(A_3)$ Since 2.3

Since 2.3

APParasa: And Parasa: said semechatiflus:
APparasa: Its Englain APparasa: and a Parasa: and a Semechatiflus:
APparasa: Its Englain APparasa: And APparasa: A

17:22 bis 17:25

Teil 2: endliche Bäume Entscheidungsprobleme –Das Äquivalenzproblem

Das Äguivalenzproblem Eingabe: NEBAs (oder DEBAs) A1. A2 Frage: lst $L(A_1) = L(A_2)$? d.h. $AP_{AEBA} = \{(A_1, A_2) \mid A_1, A_2 \text{ NEBAs}, L(A_1) = L(A_2)\}$ etc APvena und APvena sind entscheidbar. APNERA ist ExpTime-vollständig: APDERA ist in P. w Entscheidbarkeit analog zu NEAs, per Reduktion zum LP $L(A_1) = L(A_2)$ gdw. $L(A_1) \triangle L(A_2) = \emptyset$ g obere Schranken: Automat für L(A₁) ∧ L(A₂) 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!

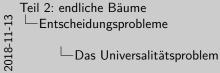
–Das Äquivalenzproblem

Eingabe: NEBAs (oder DEBAs) A1. A2 Frage: lst $L(A_1) = L(A_2)$? d.h. $AP_{AEBA} = \{(A_1, A_2) \mid A_1, A_2 \text{ NEBAs}, L(A_1) = L(A_2)\}$ etc APvena und APvena sind entscheidbar. APNERA ist ExpTime-vollständig: APDERA ist in P. w Entscheidbarkeit analog zu NEAs, per Reduktion zum LP $L(A_1) = L(A_2)$ gdw. $L(A_1) \triangle L(A_2) = \emptyset$ g obere Schranken: Automat für L(A₁) ∧ L(A₂) ist exponentiell in der Größe der Eingabe-NEBAs / polynomiell für DEBAs u ExpTime-Härte: Reduktion vom Universalitätsproblem (F. 59)

Das Äguivalenzproblem

17:22 bis 17:25

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



17:25 bis 17:27

Das Universalitätsproblem

Eisgabe: NEBA (oder DEBA). A

Frage: Ist $L(A) = T(\Sigma)^T$ d. b. $UP_{MEM} = F(A \mid A \mid NEBA, L(A) = T(\Sigma)$) (anslog f DEBA)

Teil 2: endliche Bäume
—Entscheidungsprobleme

—Das Universalitätsproblem

17:25 bis 17:27

Das Universalitätsproblem Einjahr. EIRA (oder DEBA) A Figur ist (A, D = T(1), T = T(1)) = $(T \mid T \text{ int ein } L \text{ flowe})$ d. b. $UP_{\text{BERR}} = \{A \mid A \text{ NEBA}, L(A) = T(\Sigma)\}$ (melog f. DEBA) Sinz 22 UP_{BERR} and UP_{DEBA} sind extscheidher. UP_{BERR} and UP_DEBA sind extscheidher. UP_BERRA EIR_TENTER SIND SINCE SIND SINCE SIND SINCE SINCE

Teil 2: endliche Bäume Entscheidungsprobleme ☐ Das Universalitätsproblem

17:25 bis 17:27

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

UPNERA und UPDERA sind entscheidbar.

UPwrna ist ExpTime-vollständig: UPvrna ist in P.

u Entscheidbarkeit & obere Schranken per Red. zum AP: $L(A) = T(\Sigma)$ gdw. $L(A) = L(A_{\Sigma})$. wobei A_{Σ} DEBA mit $L(A_{\Sigma}) = \mathcal{T}(\Sigma)$ (konstruiere selbst!) Teil 2: endliche Bäume Entscheidungsprobleme └─Das Universalitätsproblem

17:25 bis 17:27

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

UPNERA und UPDERA sind entscheidbar. UPwrna ist ExpTime-vollständig: UPvrna ist in P.

Entscheidbarkeit & obere Schranken per Red. zum AP: $L(A) = T(\Sigma)$ gdw. $L(A) = L(A_{\Sigma})$.

wobei A_{Σ} DEBA mit $L(A_{\Sigma}) = \mathcal{T}(\Sigma)$ (konstruiere selbst!) v ExpTime-Härte: Red. vom WP für lin. platzbeschränkte alternierende TM (s. a. [?, §1.7])

Teil 2:	endliche Bäume)
└─En [.]	tscheidungsprobl	eme

—Überblick Entscheidungsprobleme für NEBAs/DEBAs

Überblick Entscheidungsprobleme für NEBAs/DEBAs

nachweisicht (da Exprime # P)

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

Nächstes Mal: XML, Kapitel zuende

Zum Literaturverzeichnis blättern

Und nun ...

Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

— Zur Erinnerung: semistrukturierte Daten

Zur Erinnerung: semistrukturierte Daten

Daten werden in Entitäten (Einheiten) zusammengefasst

Markierung von Entitäten durch Tags
 Bildung von Hierarchien

Gruppieren ähnlicher Entitäten

Gruppieren ähnlicher Entitäten

 Entitäten derselben Gruppe können verschiedene (oder keine) Attribute haben

u Reihenfolge der Attribute kann eine Rolle spielen

Teil 2: endliche Bäume

Anwendung: XML-Schemasprachen

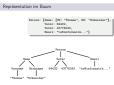
Zur Erinnerung: semistrukturierte Daten

Bildung von Hierarchien

Daten werden in Entitäten (Einheiten) zusammengefasst a Märkierung von Entitäten durch Tags

└─Zur Erinnerung: semistrukturierte Daten

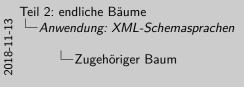




Transmiss of the Control of the Cont

Größeres Bsp.: XML-Dokument für Konferenzprogramm

—Größeres Bsp.: XML-Dokument für Konferenzprogramm



conference track

sentin break sentin

chair talk salk sentin

title space track sathers

sen to demonstration to the sentin track

Ab joint air backwake mar de Straktur, quadene da Dates

Zugehöriger Baum

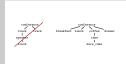
Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen Was ist ein gültiges Konferenzdokument?

018-11-13

Teil 2: endliche Bäume —— Anwendung: XML-Schemasprachen

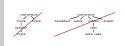
Was ist ein gültiges Konferenzdokument?

conference track track speaker track



Was ist ein gültiges Konferenzdokument?

Was ist ein gültiges Konferenzdokument?



Was ist ein gültiges Konferenzdokument?

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen

-Mögliche Anforderungen an gültige Konferenzdokumente

8:38

Mögliche Anforderungen an gültige Konferenzdokumente

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

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

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

Gültige Dokumente als Baumsprachen!

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

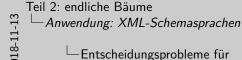
Eine solche Beschreibung wird auch Schema genannt.

Ein Dokument ist eiltig für ein Schema.

wenn sein Baum zur Baumsprache des Schemas gehört

Ziele dieses Abschnitts

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



XML-Schemasprachen

Entscheidungsprobleme für XML-Schemasprachen

Die bekannten Entscheidungsprobleme entsprechen natürlichen
Frasen 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

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

XML-Schemasprachen

Die bekanntes Entscheidungsprobleme entsprachen natürlichen Forgen für XML-Oulswerten und -Sohmungsruchen: Zupströngstradelme Ist ein gegebenes Sohmun? Ist ein gegebenes Oulswerte gältig für ein gegebenes Sohmun? (im Bur eithlich im gebenes Kond-dedument des Anforderungen?) Lenfoltsproblem. Gült ist für ein ein erselbenes Sohmun gellen Delumentet?

Entscheidungsprobleme für XML-Schemasprachen

(Enthält das gegebene Schema keinen "Widenspruch"?)

8:42

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

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 "Widenspruch"?) Äquivalenzproblem

Haben zwei Schemata dieselben gültigen Dokumente?

(Wichtig bei der Vereinfachung von Schemata)

8:42

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

Dokumenttypdefinitionen (DTDs)

DTDs ind ein Standard zur Beschnibung giltiger Dokuments
Eine DTD ist eine kontentfrie Germantik (457),
eine nichte Regleinen regleine Ausleiche serbalten klönen
Abletungskämer der VS bilden des Einsengrache,
die druch der DTD seistem wird

-Beispiel-DTD für Konferenzdokumente

< IDOCTYPE CONFERENCE [<!ELEMENT conference (track+|(session,break?)+)> <!ELEMENT track ((session,break?)+)> < SELECTION Newsion (chair,talk+)> <!ELEMENT talk ((title,authors)|(title,speaker))> <!ELEMENT chair (*PCDATA)> <!ELEMENT break (*PCDATA)> <!ELEMENT title (*PCDATA)> <!ELEMENT authors (#PCDATA)> <!ELEMENT speaker (#PCDATA)> .

Verkettung

Beispiel-DTD für Konferenzdokumente

< IDOCTYPE CONFERENCE [<!ELEMENT conference (track+|(session,break?)+)> <!ELEMENT track ((session,break?)+)> <!ELEMENT mession (chair,talk+)> <!ELEMENT talk ((title,authors)|(title,speaker))> <!ELEMENT chair (*PCDATA)> <!ELEMENT break (epcnara)> <!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 senzion-Kinder hat, zwischen denen einzelee break-Geschwister stehen dürfen

Beispiel-DTD für Konferenzdokumente

Beispiel-DTD für Konferenzdokumente

Beispiel-DTD für Konferenzdokumente

```
< !DOCTYPE COMPENSAGE [
 <!ELEMENT conference (track+|(session,break?)+)>
 <!ELEMENT track ((session,break?)+)>
 <!ELEMENT mession
                     (chair,talk+)>
 <!ELEMENT talk</pre>
                     ((title,suthors)|(title,speaker))>
 <!ELEMENT chair
                      (*PCDATA)>
 <!ELEMENT break
                      (eprmara)>
 <!ELEMENT title
                      (#PCDATA)>
 <!ELEMENT authors
                      (#PCDATA)>
 <!ELEMENT speaker
                      (#PCDATA)>
                                           . 

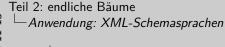
Verkettung
```

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

ein oder mehren track-Kinder hat oder
 ein oder mehrene session-Kinder hat,
 zwischen denen einzelne break-Geschwister stehen dürfen

Wir brauchen Symbole mit beliebiger Stelligkeit!

Wir brauchen Symbole mit beliebiger Stelligkeit! Erweiten unser r-Alphabet: • U: Menge von Symbolen ohne Stelligkeit • $\Sigma = U \cup \bigcup_{i \ge 0} \Sigma_i$



└─Wir brauchen Symbole mit beliebiger Stelligkeit!

Wir brauchen Symbole mit beliebiger Stelligkeit!

Erwitern unser (-Alphabet:

• U: Menge von Symbolen ohne Stelligkeit • $\Sigma = U \ \cup \ \bigcup_{i \ge 0} \Sigma_i$

Endlicher geordneter Baum T=(P,t) über Σ : • $P\subseteq \mathbb{N}^*_+$ nichtliere endl. präfix-abgeschlossene Menge

 \bullet $P\subseteq \mathbb{N}_+^*$ nichtleere endl. präfix-abgeschlossene Menge a $t:P\to \Sigma$ Funktion mit

 $oldsymbol{\Phi}$ Wenn $t(\rho) \in \Sigma_m$, dann $\{j \mid \rho j \in P\} = \{1, \dots, m\}$. $oldsymbol{\Phi}$ Wenn $t(\rho) \in U$, dann $\{j \mid \rho j \in P\} = \{1, \dots, k\}$ für ein $k \ge 0$

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

8:50

Weitere Begriffe

- u Höhe, Tiefe, Teilbaum: wie für Bäume mit Stelligkeit v $a(T_1 \cdots T_n)$: Baum mit a in Wurzel und Teilbäumen T_1, \dots, T_n direkt darunter
- Hecke (Hedge): Folge T₁ · · · T_n von Bäumen
 - leare Hecke: a
- • $H(\Sigma)$: Menge aller Hecken über Σ

Teil 2: endliche Bäume

CT

Anwendung: XML-Schemasprachen

Weitere Begriffe

8:50

Weitere Begriffe

u Höhe, Tiefe, Teilbaum: wie für Bäume mit Stelligkeit
v $a(T_1 \cdots T_n)$: Baum mit a in Wurzel und
Teilbäumen T_1, \dots, T_n direkt darunter

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

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

induktive Charakterisierung von Bäumen:
 a Jede Folge von Bäumen ist eine Hecke.
 u Wenn h eine Hecke und a ∈ Σ ein Symbol ist,

dann ist a(h) ein Baum

 $h=\varepsilon \ \leadsto \ {
m schreiben} \ a \ {
m statt} \ a(\varepsilon)$

Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

— Beispiele für Hecken und Bäume

8:53 bis 8:58

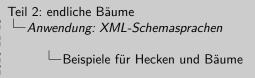
Beispiele für Hecken und Bäume

Sei $\Sigma = \{a, b, c\}$. ε ist eine Hecke $\sim a = a()$ ist ein Baum $\sim aa$ ist eine Hecke

→ b(aa) ist ein Baum
→ ab(aa)c ist eine Hecke

→ ab(aa)c ist eine Hecke
→ a(ab(aa)c) ist ein Baum

T 2.11



8:53 bis 8:58

Beispiele für Hecken und Bäume

Sei $\Sigma = \{a, b, c\}$. ε ist eine Hecke $\sim a = a()$ ist ein Baum $\sim aa$ ist eine Hecke

→ b(aa) ist ein Baum

→ ab(aa)c ist eine Hecke

→ ab(aa)c ist eine Hecke
→ a(ab(aa)c) ist ein Baum
(a(c(b)cb(ab)) ist ein Baum

aum T2.11 aum T2.11 Forts. Teil 2: endliche Bäume

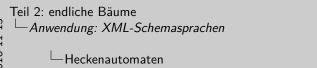
— Anwendung: XML-Schemasprachen

— Heckenautomaten

8:58 bis 9:06

Heckenautomaten

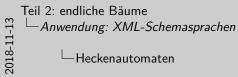
...sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit Können sie analog zu NEBAs definiert werden?



8:58 bis 9:06

Heckenautomaten
... sind Bottom-up-Automaten auf Bäumen ohne Stelligkeit
Konene sie anvlog zu NEBAs definiert werden?
Neit: Woll Stelligheit von a E. E. nicht festgelegt ist,

brauchten wir 1 Regel $a(q_1, \dots, q_m) \rightarrow q$ pro $m \geqslant 0$. T 2.12



8:58 bis 9:06

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 $a(q_1,\dots,q_n)\to q$ per $m\geqslant 0$. T 2.12 Abhlie: Natzen reoulier Ausdickle über O in linken Reusleiden

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Heckenautomaten

8:58 bis 9:06

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 $a(q_1, \dots, q_m) \rightarrow q$ pro $m \geqslant 0$. T 2.12

Abhilfe: Nutzen reguläre Ausdrücke über Q in linken Regelseiten

Ein nichtdeterministischer endlicher Heckenautomat (NEHA)

ist ein Quadrupel $A = (Q, \Sigma, \Delta, F)$, wobei u Q, Σ, F wie für NEBAs definiert sind und u Δ eine Menge von Überführungsregeln der Form $a(R) \rightarrow q$

ist, wobei $a \in \Sigma$ und $R \subset Q^*$ eine reg. Sprache über Q ist.

-Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Sei $\mathcal{A}=(Q,\Sigma,\Delta,F)$ ein NEHA und $\mathcal{T}=(P,t)$ ein Σ -Baum o. S. \bullet Berechtung (Run) von \mathcal{A} auf \mathcal{T} ist eine Fist. $r:P\to Q$ mit: Wenn t(p)=a, r(p)=q und m=Anzahl von p's Kindern, dann gibt es $s(R)\to q$ in Δ mit $r(p1)\cdots r(pm)\in R$.

Teil 2: endliche Bäume

Anwendung: XML-Schemasprachen

-Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

Sei $A = (Q, \Sigma, \Delta, F)$ ein NEHA und T = (P, t) ein Σ -Baum o. S. \bullet Berechnung (Run) von A auf T ist eine Fixt. $r: P \rightarrow Q$ mit: Wens t(p) = a, t(p) = q and m = A rank) von p is Kindern, dann gist en $a(R) \rightarrow q$ in Δ mit $t(p1) \cdots t(pm) \in R$.

Anmerkungen

 Wenn p Blattposition mit Markierung a (d.h. t(p) = a), dam darf a(R) → q nur angewendet werden, wenn ε ∈ R.
 Repräsentation der reg. Sprache R ⊆ Q*: NEAs, DEAs oder reg. Ausdrücke

das ist egal für die Mächtigkeit von NEHAs,
 aber nicht für Entscheidungsverfahren und deren Komplexität!

Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

-Berechnungen, Akzeptanz und erkannte Sprache

Berechnungen, Akzeptanz und erkannte Sprache

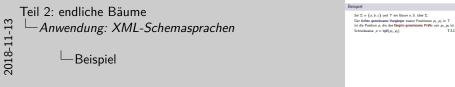
Sei $A = (Q, \Sigma, \Delta, F)$ ein NEHA und T = (P, t) ein Σ -Baum o. S. • Berechnung (Ran) von A auf T ist eine Fkt. $r: P \rightarrow Q$ mit: Wens $t(\rho) = x, r(\rho) = q$ und $m = Anzahl von <math>\rho$'s Kindern, dans gibt as $s(R) \rightarrow q$ in Δ mit $t(p)1 \cdots t(pm) \subseteq R$.

■ Ein Run r von A auf T ist erfolgreich, wenn $r(\varepsilon) \in F$.

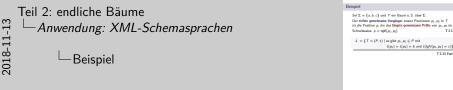
▼ A akzeptiert T, wenn es erfolgreichen Run von A auf T gibt.

A akzeptiert T, wenn es erfolgreichen Run von A au
 Die von A erkannte Sprache ist
 (14) = (T über X | A skrootiert T)

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

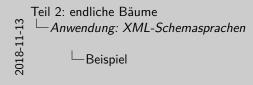


9:08 bis 9:15



 $t(p_1) = t(p_2) = b \text{ und } t(tgV(p_1, p_2) = c)$ T 2.13 Forts.

9:08 bis 9:15



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 Tist die Position p, die das längste gemeinsame Präffx von p_1 , p_2 ist. Schreibweise: $p = \operatorname{tgV}(p_1, p_2)$ T 2.13

 $L = \{T = (P, t) \mid \text{es gibt } \rho_1, \rho_2 \in P \text{ mit} \\ t(\rho_1) = t(\rho_2) = b \text{ und } t(\text{tgV}(\rho_1, \rho_2) = c)\}$ T2.13 Forts.

Idee für einen Baumautomaten:

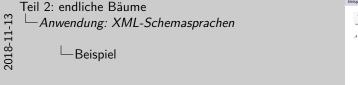
u Gehe in q₀ , sobald *b* gesehen.

Propagiere q₀ nach oben.

Propagiere q_b nach oben. • Gehe in q_c, wenn c gesehen und in 2 Kindern q_b

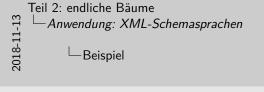
Propagiere q_c nach oben.

Akzeptiere, wenn Wurzel mit qc markiert.



9:15 bis 9:21; 5min Pause \rightarrow bis 9:26





9:15 bis 9:21; 5min Pause \rightarrow bis 9:26



Beispiel

(1)

9:26

Wollen jetzt sehen, wie man DTDs in NEHAs umwandelt, um Automatentechniken auf Validierung usw. anwenden zu können.

-Umwandlung der Beispiel-DTD in einen NEHA

1. Schritt: erweiterte kfG, die der DTD entspricht (hier am Bsp.) ("erweitert": mit regulären Ausdrücken auf rechten Regelseiten)

(1)

Umwandlung der Beispiel-DTD in einen NEHA (1)

- energe centrament (
- energe centrament

→ DATA

(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)

Enachter's contestficie Grammatic (WMMg.):

conference

Trac + (massion (break + f))*

conference

Trac + (massion (break + f))*

conference

**Conference

**C

Umwandlung der Beispiel-DTD in einen NEHA (2)

9:29 bis 9:35

(2)

Wie funktionieren eigentlich erweiterte kfGs? Hier keine formale Def., sondern Vorgehen bei einzelnem Ableitungsschritt

-Umwandlung der Beispiel-DTD in einen NEHA

Teil 2: endliche Bäume
— Anwendung: XML-Schemasprachen

Executor to totactivis Grammath (Widths):

conference → trad* * (seasion (transk* + t))*
conference → trad* * (seasion (transk* + t))*
conference → that tradit
takis → (takis extraor) + (takis spandar)
colair → shift
title → SATH

Startsymbol. Non-conference

Wähle Regel ℓ → R (R: reg. Sprache über Σ, Inhaltsmodell)
 Wähle a₁ · · · a_n ∈ R und füre Kinder a₁ · · · · a_n zu ℓ hinzu

Umwandlung der Beispiel-DTD in einen NEHA (2)

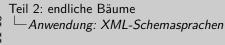
Ableitungsschritt:

• Wähle mit ℓ beschriftetes Blatt, $\ell \in \Sigma$

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



conference	-	track* + (session (break + r))*
track	\rightarrow	$(session (break + \varepsilon))^{+}$
mension.	\rightarrow	chair talk*
talk	\rightarrow	(title authors) + (title speaker)
chair	\rightarrow	DATA
title		DATA

Wähle Regel ℓ → R (R: reg. Sprache über Σ, Inhaltsmodell)
 Wähle a····a·∈ R und füre Kinder a·····a·zu ℓ hinzu

Umwandlung der Beispiel-DTD in einen NEHA (2)

Ableitungsschritt:

• Wähle mit ℓ beschriftetes Blatt, $\ell \in \Sigma$

-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

	ren 2. endiiche Daume
13	—Anwendung: XML-Schemasprachen
≟	
7	

Umwandlung der Beispiel-DTD in einen NEHA (3)

Erwötote kottottfreis Grammatk (Wdblg.):

conference → track* + (sassin (break + r))*

track → (seastin (break + r))*

::

tttls → batta

9:35 bis 9:38

3.JJ DIS 3.JC

Schritt 2: NEHA aus Grammatik bauen

 \bullet Zustände = Alphabet

Tail Or andliaha Därman

(3)

ullet Δ : aus den Produktionen (Zustände kursiv, Zeichen aufrecht)

-Umwandlung der Beispiel-DTD in einen NEHA

9:35 bis 9:38

Schritt 2: NEHA aus Grammatik bauen

- \bullet Zustände = Alphabet
- Δ: aus den Produktionen (Zustände kursiv, Zeichen aufrecht)

Präzise Definition DTD & zugehöriger NEHA

Eine Dokumanttypdefinition (DTD) ist ein Tupel $D=(\Sigma,s,\Delta)$ mit \blacksquare einem Alphabet Σ (ohne Stelfigkeit) \blacksquare einem Startsymbol $s\in\Sigma$ und \blacksquare einem Albhidung $\Delta:\Sigma\to \text{regulire}$ Ausdrücke über Σ

 $(\Delta$ entspricht einer Menge von Regeln – die Folge der Symbole in den Kindern jedes Knotens mit $s\in \Sigma$ muss in $L(\Delta(s))$ sein.)

Präzise Definition DTD & zugehöriger NEHA

Teil 2: endliche Bäume
— Anwendung: XML-Schemasprachen

Präzise Definition DTD & zugehöriger NEHA

Präzise Definition DTD & zugehöriger NEHA

Eine Dokumenttypdefinition (DTD) ist ein Tupel $D=(\Sigma,s,\Delta)$ mit u einem Alphabet Σ (ohne Stelligkeit) v einem Startsymbol $s\in \Sigma$ und

einer Abbildung Δ : Σ → reguläre Ausdrücke über Σ
 (Δ entspricht einer Menze von Rezeln – 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

ugenunger werse: $A_D = (Q_D, L, \Delta_D, P_D)$ mit u $Q_D = \Sigma$

• $F_D = \{s\}$ • $\Delta_D = \{a(\Delta(a)) \rightarrow a \mid a \in \Sigma\}$

Teil 2: endliche Bäume
— Anwendung: XML-Schemasprachen
— Lokale Sprachen

9:41 bis 9:52

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.

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Lokale Sprachen

9:41 bis 9:52

Lokale Sprachen Definition 2.26

 Die von einer DTD D erzeugte Sprache ist L(An). Eine Baumsprache über Σ heißt lokal.

wenn sie von einer DTD über E erzeugt wird.

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.?

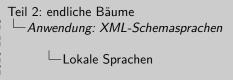
Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen -Lokale Sprachen

9:41 bis 9:52

Lokale Sprachen Definition 2.26

- Die von einer DTD D erzeugte Sprache ist L(An). v Eine Baumsprache über Σ heißt lokal.
- wenn sie von einer DTD über E erzeugt wird.

Wie verhalten sich lokale Sprachen zu NEHA-erkennbaren Spr.? (Gibt es NEHAs, die keine DTD beschreiben?)



9:41 bis 9:52



Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

— Lokale Sprachen

9:41 bis 9:52



Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

— Lokale Sprachen

9:41 bis 9:52



"alle Sitzungen jeder Konf. haben zusammen > 5 Vortragende")

(Nicht ausdrückbar:

Lokale Sprachen

Deterministische Inhaltsmodelle

Die W3C*-Empfeihlung für XML fordert, dass Inhaltsmodelle deterministische reguläre Ausdrücke sind. "World Wole Web Consortiem, int. Agentur für WWW-Standards

Deterministische Inhaltsmodelle

9:52

 $\ensuremath{\mathsf{DTDs}}$ werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.

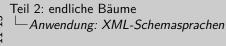
Die W3C²-Empfehlung für XML fordert, dass Inhaltsmodelle deterministische regeläre Ausdrücke sind. "World Wide Web Consortion, int. Agentur für WWW-Standards

Deterministische Inhaltsmodelle

Regulärer Ausdruck r über Σ ist deterministisch, falls • für jedes Wort $w \in \Sigma^*$ und jeden Buchstaben a in whöchstens ein Vorkommen von a in r existiert, auf das a passt

9:52

DTDs werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.



Deterministische Inhaltsmodelle

Die W3C²-Empfehlung für XML fordert, dass Inhaltsmodelle deterministische reguläre Ausdrücke sind. "World Wide Web Consortien, Int. Agnetur für WWW-Standards

Regulärer Ausdruck r über Σ ist deterministisch, falls

Deterministische Inhaltsmodelle

u 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.

9:52

DTDs werden aber laut W3C-Standard nochmal eingeschränkt, nämlich wie folgt.



9:54

-Deterministische Inhaltsmodelle – Beispiel

Betrachte die Zole

((title_authers)(title_openhar))

und die zogekhrige Regel

tit → (title authers) + (title spanker)

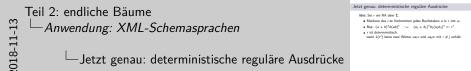
Für Wörter über X. die mit dem Buchstaben title beginnen

dieser Buchstabe entspricht!

ist nicht klar, welchem Vorkommen von title im Inhaltsmodell

Deterministische Inhaltsmodelle - Beispiel

9:54



9:55 bis 10:00 o Ende? (Rest dauert noch $pprox 5 \, \text{min}$)

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen

9:55 bis $10:00 \rightarrow \text{Ende}$? (Rest dauert noch $\approx 5 \text{ min}$)

-Jetzt genau: deterministische reguläre Ausdrücke

Jetzt genau: deterministische reguläre Ausdrücke Idee: Sei r ein RA über D. Markiere das i-te Vorkommen iedes Buchstaben a in r mit a:.

 \bullet Bsp: $(a + b)^*b(ab)^* \longrightarrow (a_1 + b_1)^*b_2(a_2b_3)^* =: r'$. wenn L(r') keine zwei Wörter ua_iv und ua_jw mit $i \neq j$ enhält.

RA r über Σ → markierter RA r' über Σ'

u wie üblich: $L(r) \subset \Sigma^*$ und $L(r') \subset \Sigma'^*$

- Jetzt genau: deterministische reguläre Ausdrücke

when $L(r^i)$ being zool Wörter stay and says with $i \neq j$ exhibit. Base Notation:

a RA r diber Σ — markierter RA r^i then Σ^* — we in thich: $L(r) \subseteq \Sigma^*$ and $L(r^i) \subseteq \Sigma^*$.

Definition 2.27:

En deterministischer RA (DRA) ats ein RA r ober Σ , so dass first all Worter α , we $C\Sigma^*$ and Σ -form $\alpha \subseteq \Sigma$.

T 2.16

Markiere das i-te Vorkommen iedes Buchstaben a in r mit a:.

 \bullet Bsp: $(a + b)^*b(ab)^* \longrightarrow (a_1 + b_1)^*b_2(a_2b_3)^* =: r'$.

Jetzt genau: deterministische reguläre Ausdrücke

Idee: Sei r ein RA über D.

r ist deterministisch

mit ua_iv , $ua_iw \in L(r')$ gilt: i = j.

9:55 bis $10:00 \rightarrow \text{Ende}$? (Rest dauert noch $\approx 5 \text{ min}$)

Was nützen uns nun DRAs?

(10:00 bis 10:01)

2018-11-13

Was nützen uns nun DRAs?

Satz 2.280 Zu jedem DRA r kann man in Polynomialzeit einen DEA A mit L(A) = L(r) konstruieren. (Ohne Beweis.)

(10:00 bis 10:01)

Teil 2: endliche Bäume - Anwendung: XML-Schemasprachen

─Was nützen uns nun DRAs?

(10:00 bis 10:01)

Was nützen uns nun DRAs? Zu jedem DRA r kann man in Polynomialzeit einen DEA A mit L(A) = L(r) konstruieren. (Ohne Beweis.) 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?

(10:00 bis 10:01)

Was nitzen uns nun DRAs? Sez 225 per den DRAs nitzen uns nun DRAs nitzen man in Polymonializat drien DRA A and $L(\phi) = L(\phi)$ banchieren. (Owns Benna). Filipment 229 $L(\phi) = L(\phi)$ banchieren DTD kann man in Polymonializat crima injunction BRAN(CRA) bonchrainen. BRAN(CRA) bonchrainen. BRAN(CRA) bonchrainen. BRAN(CRA) $L(\phi) = L(\phi) = L(\phi)$ be due for for all $u(R) \rightarrow q \in \Delta$ R at DRAs grades on R are for R and R and R are the form of R and R and R are the form of R and R are the R and R are the form of R and R are the R are the form of R and R are the R and R are the R are the R and R are the R are the R and R are the R and R are the R are the R and R are the R are the R and R are the R are the R are the R and R are the R are the R are the R and R are the R are the R are the R and R are the R are the R are the R and R are the R and R are the R are t

Deterministische DTDs sind effizient!

Sauz 200

Für deterministische DTDs sind in Polynomialzeit lösbar

a das Zugehörigknistproblem

a das Keninistproblem

(Ohne Bessels,)

(10:01 bis 10:02)

"Ohne Beweis": eins davon voraussichtlich auf Ü-Blatt 3

-Deterministische DTDs sind effizient!

Teil 2: endliche Bäume

— Anwendung: XML-Schemasprachen

—Deterministische DTDs sind effizient!

■ Zogehörigkeitsproblem (Gülzigkeit)
Int ein gegebenen Dokument gülzig für ein gegebenen Schuma?

♦ Leerheitsproblem (Widenpruchafveheit)
Gülzt en für ein gegebenen Schema gülzige Dokumente?

• Äquivalengspeblem

Haben zwei Schemata dieselben gültigen Dokumente?

Für deterministische DTDs sind in Polynomialzeit lösbar

Deterministische DTDs sind effizient!

u das Zugehörigkeitsproblem
u das Leerheitsproblem
u das Äquivalenzproblem
(Ohne Beweis.)
Zur Erimmerune:

(10.01 L:- 10.00)

"Ohne Beweis": eins davon voraussichtlich auf Ü-Blatt 3

(10:01 bis 10:02)

Teil 2: endliche Bäume

CT

Anwendung: XML-Schemasprachen

Sind deterministische DTDs schwächer als allgemeine?

(10:02 bis 10:03)

 Im Allgemeinen ja,
 aber es ist entscheidbar, ob eine gegebene DTD äquivalent zu einer deterministischen DTD ist:

Teil 2: endliche Bäume —— Anwendung: XML-Schemasprachen

Sind deterministische DTDs schwächer als allgemeine?

(10:02 bis 10:03)

Sind deterministische DTDs schwächer als allgemeine?

(10:02 bis 10:03)

Sind deterministische DTDs schwächer als allgemeine?

 Im Allgemeinen ja,
 aber es ist entscheidbar, ob eine gegebene DTD äquivalent zu einer deterministischen DTD ist:

Nicht jede reg. Sprache wird durch einen DRA beschrieben:

 $\{L(r) \mid r \text{ ist DRA}\} \subset \{L(r) \mid r \text{ ist RA}\}$ \bigcirc Das folgende Problem ist in Polynomialzeit entscheidbar.

 Das totgende Problem ist in Polynomialzeit entscheidbar. Gegeben: DEA A Frage: Gibt es einen DRA r mit L(r) = L(A)?

Wenn ein solcher DRA existiert, dann kann er in Exponentialzeit konstruiert werden Zusammenfassung für deterministische DTDs

Zusammenfassung für deterministische DTDs

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.

u Dafür sind die wichtigen Entscheidungsprobleme effizient lösbar.

Dafür sind die wichtigen Entscheidungsprobleme effizient lösbar.

(10:03 bis 10:04)

Teil 2: endliche Bäume

Anwendung: XML-Schemasprachen

Ausblick: Lockern der Einschränkungen

(10:04 bis 10:05)

Ausblick: Lockern der Einschränkungen

Extended DTDs (EDTDs)

 führen durch eine einfache syntaktische Erweiterung aus den lokalen Sprachen heraus

- us den lokalen Sprachen heraus v sind fast äquivalent zu NEHAs
- (beschränkt auf Sprachen, in denen alle Bäsme dasselbe
- haben ein in Polynomialzeit lösbares
- Zugehörigkeits- und Leerheitsproblem

Teil 2: endliche Bäume

CT

Anwendung: XML-Schemasprachen

—Ausblick: Lockern der Einschränkungen

(10:04 bis 10:05)

Ausblick: Lockern der Einschränkungen

Extended DTDs (EDTDs)

führen durch eine einfache syntaktische Erweiterung

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

a liegt XML Schoma zugrunde

-Damit sind wir am Ende dieses Kapitels.



(10:05) Ende \rightarrow nochmal zur Literatur

Literatur für diesen Teil (Basis)

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.

Automata on Infinite Words and Trees.

Literatur für diesen Teil (weiterführend)

Literatur für diesen Teil (weiterführend)

Anne Brüggemann-Klein, Derick Wood. One-Unambiguous Regular Languages.