Theoretische Grundlagen der Informatik

Paul Züchner

14. März 2019

Inhaltsverzeichnis

1 Wiederholung

- Wörter
  - Eine Folge von Zeichen aus einem Alphabet Σ
  - Leeres Wort ε
  - Σ^* Menge aller Wörter über einem Alphabet
- Formale Sprachen
  - Eine formale Sprache L über einem Alphabet Σ ist eine Teilmenge $L \subseteq \Sigma^*$
  - Es gibt Präfix, Teilwort, Suffix
- Reguläre Ausdrücke
  - definieren Reguläre Sprachen
  - Aus Faulheit es ordentlich zu schreiben gehen wir schlampig
- Endliche Automaten $DEA = (Q, \Sigma, \delta, s, F)$
  - Q Menge an Zuständen
  - Σ endliche Menge von Eingabesymbolen
  - δ Übergansfunktion $\delta : Q \times \Sigma \to Q$
  - s Startzustand $s \in Q$
  - F eine Menge an Endzuständen $F \subseteq Q$
- Kontextfreie Grammatiken $G = (\Sigma, V, S, R)$
  - Σ endliches Alphabet (Terminalsymbole)
  - V eine endliche Menge von Variablen (Nicht-Terminalsymbole)
  - $S \in V$ Starsymbol
  - R eine endliche Menge an Ableitungregeln

2 DEA & NEA

- Reg & DEA
- NEA
  - ε -Übergang
  - Potenzmengenkonstruktion

3 Tutorium 26.10.18

TGI Tut Nr.24 maximilian.ruff@student.kit.edu
Alphabet: endliche Menge von Zeichen

Alphabet endliche Menge von Zeichen

Formale Sprache $L \subseteq \Sigma^*$

Konkatenation $L_1 * L_2 = \{uv | u \in L_1, v \in L_2\}$

3.1 Reguläre Ausdrücke

$\emptyset, \epsilon, a \in \Sigma$	sind reguläre Ausdrücke
$\rightarrow R_1 \cup R_2, R_1 * R_2, R_1^*$	sind reguläre Ausdrücke

4 Endliche Automaten

4.1 DEA

4.2 NEA

NEA akzeptier Wort w $\iff$ Es existiert Abarbeitung von w, die in einem akzeptierenen Zustand endet.
DEA und NEA können **genau** die regulären Sprachen erkennen.

4.3 Potenzmengen-Konstruktion

$$NEA \rightarrow DEA$$

Akzeptierender Zustand

Zustand	a	b
{s}	{s,1}	{s}
{s,1}	{s, 1, 2, 3}	{s, 4}
{s, 1, 2, 3}	{s, 1, 2, 3}	{s, 4}
{s, 4}	{s, 1}	{s}

Zustand	a	b
{s, 1, 3}	{1, 3}	{2, 3}
{1, 3}	{}	{2, 3}
{2, 3}	{3}	{3}
{3}	{}	{3}
{}	{}	{}

5 Tutorium 2.11.2018

Ist die Sprache regulär?
Ja? mit endlichen Automaten (NEA/DEA) oder regulären Audruck zeigen
Nein? Pumping Lemma

BSP: $L = \{a^n c b^n | n \in N\}$ ist nicht regulär

5.1 Pumping Lemma

Sei $L \subseteq \Sigma^*$ regulär.
Dann: $\exists n \in N : \forall w \in L$ mit $|w| > n$
$\exists u, v, x \in \Sigma^*$ mit $w = uvx, |uv| \leq n, v \neq \epsilon$:
$\forall i \in N_0 : uv^i x \in L$

6 Vorlesung 06.11.18

- Äquivalenz von Zuständen wiederholung
- Neroderelation

7 Tutorium 09.11.18

7.1 Aquivalenzklassenautomat

Finde Zeugen die die nicht-aquivalenz bezeugen

	S	1	2	3	4	5
5	ϵ	ϵ	ϵ	ϵ	ϵ	
4	b	b	b	b		
3	bb	bb	bb			
2	bbb	/				
1	bbb					
S						

$1 \equiv 2$

8 Tutorium 16.11.18

8.1 Turingmaschine

Turingmaschine startet immer auf dem ersten Zeichen von dem Eingabewort w.
Eine TM kann in einem akzeptierenden oder in einem nicht akzeptierenden Zustand hal-

ten oder gar nicht halten.

9 Tutorium 23.11.18

9.1 Entscheidbarkeit

9.2 NTM

10 Klausurenphase

11 Endliche Automaten

Endliche Automaten können nur Wörter akzeptieren oder ablehnen. Sie haben ein endliches Gedächtnis und können nicht zählen.

12 Äquivalenzklassenautomat

Um einen Äquivalenzklassenautomat aus einen gegebenen Automaten zu erstellen, müssen wir nach Zeugen/Wörter suchen, die ggf. die nicht-Äquivalenz bezeugen können.

-

DEA

NEA

Einen nicht-deterministischen endlichen Automaten kann man in einen deterministischen endlichen Automaten umformen in dem man die **Potenzmengenkonstruktion** anwendet:

Zustand	Übergang a	b
{s}	{s, q_1}	{q_2}
{s, q_1}	...	...
{q_2}	...	...

ϵ-Übergänge können entfernt werden ohne die Anzahl der Zustände zu erhöhen. Dazu muss man sich anschauen welche Zustände sind erreichbar bei Eingabe x durch ϵ-Übergänge also: $\epsilon^* x \epsilon^*$

13 Pumping-Lemma

Das Pumping-Lemma ist eine Eigenschaft die jede reguläre Sprache hat, hat eine Sprache diese Eigenschaft nicht dann ist sie nicht regulär, hat eine Sprache diese Eigenschaft dann ist sie nicht zwingend regulär.

Für reguläre Sprachen

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, so dass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n \text{ und } v \neq \epsilon$$

existiert, bei der auch $uv^i x \in L$ ist für alle $i \in \mathbb{N}_0$
Mathematisch ausgedrückt:

$\forall L \subseteq \Sigma^*$	mit L regulär
$\exists n \in \mathbb{N}$	
$\forall w \in L$	mit $\|w\| > n$
$\exists u, v, x \in \Sigma^*$	mit $w = uvx, \|uv\| \leq n, v \neq \epsilon$
$\forall i \in \mathbb{N}_0$:	
$uv^i x \in L$	

Für kontextfreie Sprachen

Für jede kontextfreie Sprache L gibt es eine Konstante $n \in \mathbb{N}$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ so in $z = uvwxy$ zerlegen lässt, dass gilt:

- $|vx| \geq 1$
- $|vwx| \leq n$
- $uv^i wx^i y \in L$ für alle $i \geq 0$

14 Turingmaschine

Turingmaschine startet immer auf dem ersten Zeichen von dem Eingabewort w.
Eine TM kann in einem akzeptierenden oder in einem nicht akzeptierenden Zustand halten oder gar nicht halten.

15 Entscheidbarkeit

Eine Menge M heißt entscheidbar, wenn ihre charakteristische Funktion $\chi(m)$ berechenbar ist. $\chi(m)$ gibt an, ob ein Element m in M enthalten ist oder nicht:

$$\chi(m) = \begin{cases} 1 & \text{wenn } m \in M \\ 0 & \text{sonst.} \end{cases}$$

15.1 Semi-Entscheidbarkeit

Eine TM hält und akzeptiert wenn das Wort in der Sprache $L(TM)$ liegt. Liegt es nicht in der Sprache lehnt die TM das Wort ab oder hält nicht.
Unentscheidbarkeit

16 Berechenbarkeit

16.1 Churche These

Die Menge der (Turing-)berechenbaren Funktionen ist genau die Men- ge der im intuitiven Sinne überhaupt berechenbaren Funktionen.

16.2 Satz von Rice

17 Chomsky-Hierarchie

Typ 0 - rekursiv Aufzählbar

Keine Einschränkungen bis auf rekursiv aufzählbar (durch einen Algorithmus).

$$A \to \epsilon$$

ist erlaubt.

$$S \to \epsilon$$

erlaubt wenn in keiner Regel S rechts vorhanden.

Typ 1 - kontextsensitive

Regeln müssen die Länge des Wortes verlängern oder behalten, verkürzen ist **illegal**.

Beispiele:

$$bAc \to bac$$

ist okay. b oder c müssen nicht gleichzeitig auftreten einer von beiden geht auch.

$$S \rightarrow \epsilon$$

erlaubt wenn in keiner Regel S rechts vorhanden.

$$A \rightarrow \epsilon$$

ist illegal.

$$Ab \rightarrow b$$

auch illegal, weil Länge des Wortes wird gekürzt.

Typ 2 - kontextfrei

$$A \rightarrow b$$

b kann $\in V \vee T$

$$S \rightarrow \epsilon$$

erlaubt wenn in keiner Regel S rechts vorhanden.

Typ 3 - regulär

Alle Produktionen müssen entweder rechts- oder linkslinear sein.

$$A \rightarrow cB$$

rechtslinear.

$$A \rightarrow Bc$$

linkslinear.

$$S \rightarrow \epsilon$$

erlaubt wenn in keiner Regel S rechts vorhanden.

18 Chomsky-Normalform

Umformung

1. Nur $X \rightarrow Y^*$ oder $X \rightarrow a$ erlaubt, führe neu Variable ein um $X \rightarrow Ya$ in $X \rightarrow YZ_a, Z_a \rightarrow a$ zu ersetzen.
2. Rechte Seite ≤ 2 führe neue Variablen ein.

3. Ziel: Entferne $A \to \epsilon$. Finde alle Variablen die $A \xrightarrow{\star} \epsilon$ bilden und gebe sie in die Menge V. Füge zusätzliche Regel hinzu für $A \to BC$

$$A \to \begin{cases} B & \text{falls } C \in V \\ C & \text{falls } B \in V \text{`} \end{cases}$$

4. Identifiziere und eliminiere Kreise und Kettenregeln $A \to B \to C \to A$ setze $A := B := C$

5. Ausnahme $S \to \epsilon$ behandeln. Enthält die Grammatik G die Regel $S \to \epsilon$, wird ein neues Startsymbol S' für G' eingeführt. Anschließend erhält die neue Grammatik die Regeln $S' \to \epsilon | S$. Damit ist sichergestellt, dass die Grammatik weiterhin das leere Wort ermöglicht und das ursprüngliche Startsymbol weiterhin auf der rechten Seite verwendet werden kann.