

Theoretische Informatik

Zusammenfassung

SoSe2024

Inhaltsverzeichnis

1	Allgemein	3
1.1	Alphabete und Wörter	3
1.2	Grammatiken	3
2	Chomsky-Hierarchie	4
2.1	Typ 0 (\mathcal{L}_0) - Phrasenstrukturgrammatiken	4
2.2	Typ 1 (\mathcal{L}_1) - Kontextsensitive Grammatiken	4
2.3	Typ 2 (\mathcal{L}_2) - Kontextfreie Grammatiken	4
2.4	Typ 3 (\mathcal{L}_3) - Reguläre Grammatik	4
3	Deterministischer Endlicher Automat (DEA)	5
4	Nicht-deterministischer Endlicher Automat (NEA)	6
5	Äquivalenz von DEA und NEA	7
5.0.1	Satz von Rabin und Scott	7
5.0.2	Potenzmengenkonstruktion	7
6	Regex	8
6.0.1	Satz von Kleene	8
7	Pumping Lemma	9
8	Satz von Myhill und Nerode	10
9	Minimalautomaten	11
9.1	Table-Filling-Algorithmus	11
10	Kontextfreie Sprachen (\mathcal{L}_2)	12
10.1	Chomsky Normalform (CNF)	12
10.2	Greibach Normalform	12
10.3	Konvertierung	12

11 Kellerautomaten	13
12 CYK-Algorithmus	14

1 Allgemein

1.1 Alphabete und Wörter

- Ein Alphabet Σ ist eine endliche Menge unterscheidbarer Symbole
- Element $\sigma \in \Sigma$ ist ein Zeichen des Alphabets Σ
- Jedes Element $\omega \in \Sigma^*$ ist ein Wort über Σ
- ε = Leeres Wort
- Σ^* : Menge aller Wörter über Σ
- Σ^+ : Menge aller Wörter über Σ mit mind. 1 Element
- $|\omega|$: Länge eines Wortes ($|\varepsilon| = 0$)

1.2 Grammatiken

Eine Grammatik G ist ein 4-Tupel (V, Σ, P, S) :

- V : endliche Menge an Nicht-Terminal-Symbolen
- Σ : endliche Menge an Terminal-Symbolen ($V \cap \Sigma = \emptyset$)
- P : endliche Menge an Produktionsregeln
- S : Startsymbol ($S \in V$)

2 Chomsky-Hierarchie

2.1 Typ 0 (\mathcal{L}_0) - Phrasenstrukturgrammatiken

- Beliebige Kombination aus T- und NT-Symbolen

2.2 Typ 1 (\mathcal{L}_1) - Kontextsensitive Grammatiken

- $|l| \leq |r|$
- Länge des Wortes steigt
- $S \rightarrow \varepsilon$ erlaubt, wenn S auf **keiner** rechten Seite einer Regel steht!

Beispiel:

$$S \rightarrow S' \mid \varepsilon$$

$$S' \rightarrow aS'Bc \mid abc$$

$$cB \rightarrow Bc$$

$$bB \rightarrow bb$$

Das Nichtterminal S' braucht man nur, damit die Bedingung der Sonderregel erfüllt ist. Das Nichtterminal B wird mal zur Satzform Bc und mal zu bb, je nachdem ob B im **Kontext** c oder b steht.

2.3 Typ 2 (\mathcal{L}_2) - Kontextfreie Grammatiken

Beim Ableiten in Typ-1-Grammatiken muss man immer aufpassen, dass das Nichtterminal auch im richtigen Kontext steht. Das Erzeugen von Sätzen ist viel leichter, wenn die Grammatik kontextfrei ist.

Eine Grammatik G ist vom Typ 2, wenn sie vom Typ 1 ist und zusätzlich auf der linken Seite jeder Regel genau **ein** Nichtterminal steht!

- $l \in V$
- $X \rightarrow \varepsilon$ immer erlaubt

2.4 Typ 3 (\mathcal{L}_3) - Reguläre Grammatik

Eine Grammatik G ist vom Typ 3, wenn sie vom Typ 2 ist und zusätzlich folgende Regeln hat:

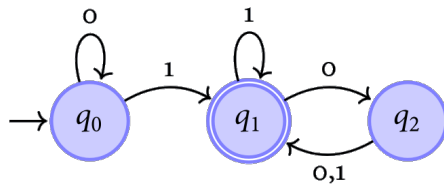
- $A \rightarrow b$
- $A \rightarrow bC$
- $A \rightarrow \varepsilon$

3 Deterministischer Endlicher Automat (DEA)

Eine DEA M ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$:

- Q : endliche Zustandsmenge
- Σ : endliches Alphabet
- $\delta: Q \times \Sigma \rightarrow Q$ Übergangsfunktionen
- q_0 : Startzustand
- F : Menge der akzeptierten Endzustände

Beispiel:



- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0, 1\}$
- $q_0 = q_0$
- $F = q_2$
- δ :

$$\delta(q_0, 0) = q_0$$

$$\delta(q_0, 1) = q_1$$

$$\delta(q_1, 0) = q_2$$

$$\delta(q_1, 1) = q_1$$

$$\delta(q_2, 0) = q_1$$

$$\delta(q_2, 1) = q_1$$

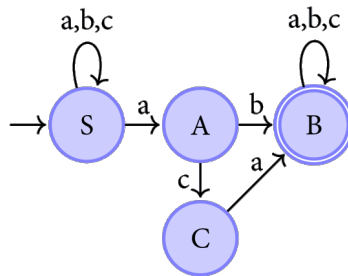
4 Nicht-deterministischer Endlicher Automat (NEA)

Eine NEA M ist ein 5-Tupel $(Q, \Sigma, \delta, q_0, F)$:

- Q : endliche Zustandsmenge
- Σ : endliches Alphabet
- $\delta: Q \times \Sigma \rightarrow Q$ Übergangsfunktionen
- q_0 : Menge der Startzustände
- F : Menge der akzeptierten Endzustände

Beispiel:

$S \rightarrow aS \mid bS \mid cS \mid aA$
 $A \rightarrow bB \mid cC$
 $B \rightarrow aB \mid bB \mid cB \mid \varepsilon$
 $c \rightarrow aB$



5 Äquivalenz von DEA und NEA

5.0.1 Satz von Rabin und Scott

Jede von einem NEA akzeptierte Sprache L ist auch von einem DEA akzeptierbar.

5.0.2 Potenzmengenkonstruktion

!!!TODO!!!

6 Regex

!!!TODO!!!

6.0.1 Satz von Kleene

Die Menge der durch reguläre Ausdrücke (Regex) beschreibbaren Sprachen ist genau die Menge der regulären Sprachen.

→ Alle endlichen Sprachen sind durch reguläre Ausdrücke beschreibbar

7 Pumping Lemma

Das Pumping-Lemma wird verwendet, um zu beweisen, dass eine Sprache sicher nicht regulär ist.

!!!TODO!!!

8 Satz von Myhill und Nerode

Eine Sprache L ist genau dann regulär, wenn der Index R_L endlich ist!

9 Minimalautomaten

!!!TODO!!!

9.1 Table-Filling-Algorithmus

10 Kontextfreie Sprachen ($\mathcal{L}2$)

10.1 Chomsky Normalform (CNF)

Regeln müssen folgende Formen haben:

- $A \rightarrow BC$
- $A \rightarrow a$
- $S \rightarrow \varepsilon$

10.2 Greibach Normalform

Eine ε -freie, kontextfrei Grammatik mit folgenden Regeln:

- $A \rightarrow aB_1B_2B_3\dots B_k$
- $k \geq 0$

10.3 Konvertierung

!!!TODO!!!

11 Kellerautomaten

Ein Kellerautomat (PDA) M ist ein 6-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, \#)$:

- Q : endliche Zustandsmenge
- Σ : endliches Bandalphabet
- Γ : endliches Kelleralphabet
- δ : Übergangsfunktionen
- q_0 : Startzustand ($q_0 \in Q$)
- $\#$: Ursprüngliches Kellersymbol ($q_0 \in \Gamma$)

Akzeptanz:

- Kein akzeptierender Endzustand!
- Akzeptanzkriterien für Wörter $x \in |\Sigma|^*$:
 1. Wort komplett gelesen
 2. Keller (Stack leer)

Nicht-Determinismus:

- Mehrere simultane Übergänge möglich
- Spontane Übergänge ($a = \varepsilon$) möglich

Konfiguration eines PDA gegeben durch 3-Tupel (Q, Σ^*, Γ^*) :

- $q \in Q$: Momentaner Zustand
- $w' \in \Sigma^*$: Noch zu lesender Anteil der Eingabe
- $\gamma \in \Gamma^*$: Aktueller Kellerinhalt

Übergangsfunktion:

- $\delta(q, a, A) \ni (q', B_1 B_2 \dots B_k)$
- Wenn Automat in Zustand q ist, das Symbol a liest und A oben auf Stack liegt, wechselt er in Zustand q' und ersetzt das A auf dem Stack durch $B_1 B_2 \dots B_k$

12 CYK-Algorithmus