Theoretische Informatik

Zusammenfassung

24.07.2024

Inhaltsverzeichnis

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
- $\varepsilon =$ 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 ($V \cap \Sigma = \varnothing$)
- 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
- $S \to \varepsilon$ erlaubt, wenn S auf **keiner** rechten Seite einer Regel steht!

Beispiel:
$S \to S' \mid \varepsilon$
$S' \to aS'Bc \mid abc$
$cB \to Bc$
$bB \to bb$

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!

- $l \in V$
- $X \to \varepsilon$ 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:

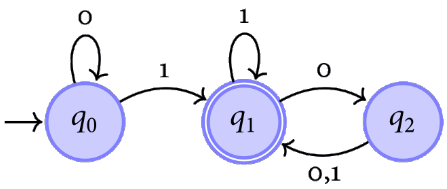
- $A \to b$
- $A \to bC$
- $A \to \varepsilon$

3 Deterministischer Endlicher Automat (DEA)

Eine DEA M ist ein 5-Tupel (Q, Σ, δ, q_0, F):

- Q: endliche Zustandsmenge
- Σ: endliches Alphabet
- δ: $Q \times \Sigma \to Q$ 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:

$$\begin{aligned}
&\delta(q_0, 0) = q_0\\
&\delta(q_0, 1) = q_1\\
&\delta(q_1, 0) = q_2\\
&\delta(q_1, 1) = q_1\\
&\delta(q_2, 0) = q_1\\
&\delta(q_2, 1) = q_1
\end{aligned}$$

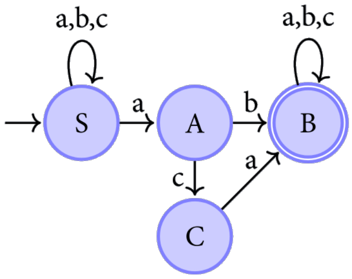
4 Nicht-deterministischer Endlicher Automat (NEA)

Eine NEA M ist ein 5-Tupel (Q, Σ, δ, q_0, F):

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

Beispiel:

$S \rightarrow aS$ | bS | cS | aA
$A \rightarrow bB$ | cC
$B \rightarrow aB$ | bB | cB | ε
$c \rightarrow aB$



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.

$\rightarrow$ 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 ($\mathcal{L}2$)