Einführung in die Theoretische Informatik
Zusammenfassung

Ali, Mihir, Noah

April 28, 2022

Contents

1 Formale Sprachen

1.1 Grundbegriffe

- Alphabet Σ (endliche Menge) z.B. $\{1, 0\}$
- Wort/String über Σ ist eine endliche Folge von Zeichen aus Σ
- $|w|$ länge des Wortes w
- Leeres Wort ϵ
- uv konkatenation der Wörter u und w
- Ist w ein Wort so ist $w^0 = \epsilon$ und $w^{n+1} = ww^n$

- Σ^* Menge aller Wörter über Σ
- (formale) Sprache $L \subseteq \Sigma^*$

1.1.1 Operationen auf Sprachen

Seien $A, B \subseteq \Sigma^*$

- Konkatenation:

$AB = \{uv | u \in A \wedge v \in B\}$

- Konkatenation mit sich selbst:

$A^n = \{w_1...w_n | w_1, ..., w_n \in A\} = A...A$

- $A^* = \{w_1...w_n | n \geq 0 \wedge w_1, ..., w_n \in A\} = \bigcup_{n\in\mathbb{N}} A^n$
- $A^+ = AA^* = \bigcup_{n\geq 1} A^n$

1. Sonderfälle:
   - $\forall A : \epsilon \in A^*$
   - $\emptyset^* = \{\epsilon\}$
   - $\emptyset A = \emptyset$
   - $\{\epsilon\}A = A$
   - $A^*A^* = A^* = (A^*)^*$

1.1.2 Grammatiken

4-Tupel $G = (V, \Sigma, P, S)$

- V ist endliche Menge von Nichtterminalzeichen
- Σ ist endliche Menga von Terminalzeichen (= Alphabet)
- $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ ist Menge von Produktionen
- $S \in V$ ist das Startsymbol

Die Sprache von G ist die Menge aller Wörter, die von G erzeugt werden. Sie wird mit $L(G)$ bezeichnet. Also jedes Wort, dass die Grammatik erzeugt muss in der Sprache erhalten sein und jedes Wort in der Sprache muss von der Grammatik erzeugt werden.

1. Reflexve transitive Hülle

    - $\alpha \to_G^0 \alpha$
    - $\alpha \to_G^{n+1} \gamma : \exists\beta.\alpha \to_G^n \to_G \gamma$
    - $\alpha \to_G^* \beta : \exists n.\alpha \to_G^n \beta$
    - $\alpha \to_G^+ \beta : \exists n > 0.\alpha \to_G^n \beta$

1.1.3 Chomsky Hierarchie

Eine Grammatik G ist vom

- Typ 0 immer
- Typ 1 falls fpr jede Produktion $\alpha \to \beta$ außer $S \to \epsilon$ gilt $|\alpha| \leq |\beta|$
- Typ 2 Falls G vom typ 1 ist und für jede Produktion $\alpha\beta$ gilt $\alpha \in V$
- Typ 3 falls G vom Typ 2 ist und für jede Produktion $\alpha \to \beta$ außer $S \to \epsilon$ gilt $\beta \in \Sigma \cup \Sigma V$

1. Grmmatiken und Sprachklassen:

Typ 3	Rechtslineare Grammatiken	Reguläre Sprachen
Typ 2	Kontextfreie Grammatik	Kontextfreie Sprachen
Typ 1	Kontextsensitive Grammatik	Kontextsens. Sprachen
Typ 0	Phrasenstrukturgrammatik	Rekursiv aufzählbare Sprachen

2. Satz 2.13 $L(Typ3) \subset L(Typ2) \subset L(Typ1) \subset L(Typ0)$

1.1.4 Wortproblem

Gegeben: eine Grammatik G, ein Wort $w \in \Sigma^*$ Frage: Ist das Wort in w enthalten ($w \in L(G)$)?

2 Reguläre Sprachen

2.1 Deterministische endliche Automaten

- Beispiel:
    - Eingabewort $baba \to$ Zustandsfolge 0,0,1,2,2

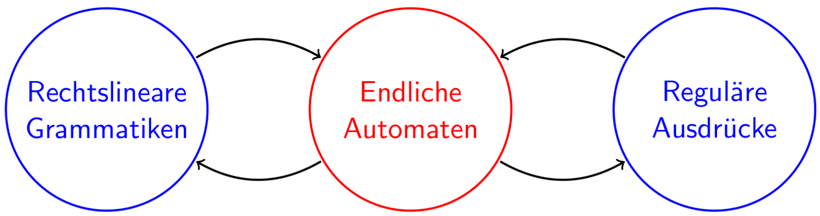

Figure 1: Reguläre Sprachen Schema

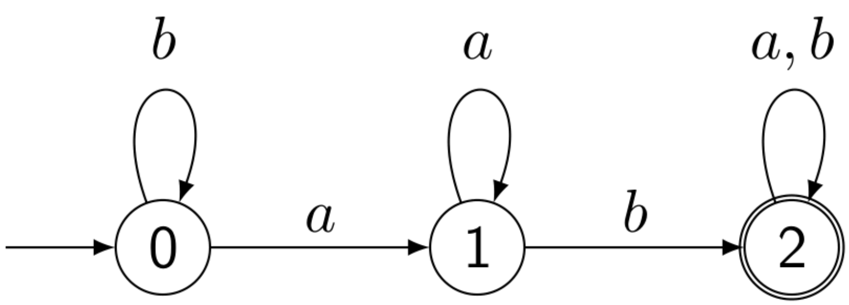

Figure 2: Beispiel Automat

- “Bei dieser Grammatik muss mindestens nach einem a ein b kommen”
- Die Sprache des DFA ist die Menge aller Wörter über $\{a, b\}$, die ab enthalten

Erkannte Sprache: Menge der Wörter, die vom Startzustand in einen Endzustand führen. Recognizer, die nur einmal das Wort durchläuft und in linearer Zeit es akzeptiert oder ablehnt.

2.1.1 Definition

Ein deterministischer endlicher Automat $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- endliche Menge von Zuständen Q
- endlichem Eingabealphabet Σ
- einer totalen Übergangsfunktion $\delta : Q \times \Sigma \to Q$
- eienm Startzustand $q_0 \in Q$
- einer Menge $F \subset Q$ von Endzuständen

2.1.2 Akzeptierte Sprachen (Definition 3.2)

Von M akzeptierte Sprache $L(M) := \{w \in \Sigma^* | \hat{\delta}(q_0, w) \in F\}$ wobei
$\hat{\delta} : Q \times \Sigma^* \to Q$ induktiv definiert ist:
$\hat{\delta}(q, \epsilon) = q$
$\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$, für $a \in \Sigma, w \in \Sigma^*$
($\hat{\delta}(q, w)$ bezeichnet den Zustand, den man aus q mit w erreicht.)
Eine Sprache ist regulär **gdw** sie von einem DFA akzeptiert wird.