Einführung in die Theoretische Informatik
Zusammenfassung

Efe Kamasoglu

May 6, 2023

1 Grundbegriffe und Grammatiken

1.1 Grundbegriffe

- Alpabet Σ, endliche Menge
- Wort/String w über Σ, endliche Menge von Zeichen aus Σ
- $|w|$, Länge des Wortes w
- ϵ, das leere Wort mit der Länge 0
- Wörter u und v, uv ist ihre Konkatenation
- Wort w, w^n definiert durch:
  - $w^0 = \epsilon$
  - $w^{n+1} = ww^n$
  - *Beispiel:* $(ab)^3 = ababab$
- Σ^*, Menge aller Wörter über Σ
- Teilmenge $L \subseteq \Sigma^*$, formale Sprache
  - *Beispiel:* $\emptyset,\ \{\epsilon\},\ L_1 = \{\epsilon, ab, aabb, aaabbb, ...\} = \{a^n b^n \mid n \in \mathbb{N}\}$

1.2 Operationen auf Sprachen

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

- **Konkatenation:** $AB = \{uw \mid u \in A \ \wedge \ w \in B\}$
  - *Beispiel:* $\{ab, b\}\{a, bb\} = \{aba, abbb, ba, bbb\}$
  - $A^n = \{w_1...w_n \mid w_1, ..., w_n \in A\} = \underbrace{A...A}_{\text{n}}$
  - $A^0 = \{\epsilon\},\ A^{n+1} = AA^n$
  - $A^* = \{w_1...w_n \mid n \geq 0 \ \wedge \ w_1, ..., w_n \in A\} = \bigcup_{n \in \mathbb{N}} A^n$
    - A^* enthält ϵ immer
  - $A^+ = AA^* = \bigcup_{n \geq 1} A^n$
    - $\epsilon \in A$ *gdw.* $\epsilon \in A^+$
- **Kartesisches Produkt:** $A \times B$
  - *Beispiel:* $\{ab, b\} \times \{a, bb\} = \{(ab, a), (ab, bb), (b, a), (b, bb)\}$
- **Rechenregeln über Sprachen:**
  - Für alle A: $\epsilon \in A^*$
  - $\epsilon \notin \emptyset$

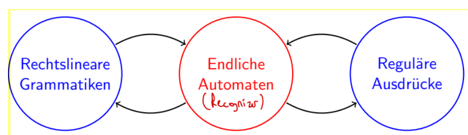
- $\emptyset^* = \{\epsilon\} = \emptyset^0$
- $A\{\epsilon\} = \{\epsilon\}A = A$
- $A\emptyset = \emptyset A = \emptyset$
- $A \times \emptyset = \emptyset$
- $A(B \cup C) = AB \cup AC$
- $(A \cup B)C = AC \cup BC$
- $A(B \cap C) = AB \cap AC$ gilt i.A. **nicht**
- $A(B \setminus C) = AB \setminus AC$ gilt i.A. **nicht**
- $A^*A^* = (A^*)^* = A^*$

1.3 Grammatiken

- Grammatik, 4-Tupel $G = (V, \Sigma, P, S)$
  - V, endliche Menge von **Nichtterminalen**
  - Σ, endliche Menge von **Terminalen**, disjunkt von V
  - $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$, Menge von **Produktionen**
  - $S \in V$, **Startsymbol**
- Eine Grammatik G induziert eine **Ableitungsrelation** $\rightarrow_G$ auf Wörtern über $V \cup \Sigma$:
  - $\alpha \rightarrow \alpha'$ gdw. es eine Regel $\beta \rightarrow \beta'$ in P und Wörter α_1, α_2 gibt, so dass $\alpha = \alpha_1 \beta \alpha_2 \ \wedge \ \alpha' = \alpha_1 \beta' \alpha_2$
- Eine **Sequenz** $\alpha_1 \rightarrow_G \alpha_2 \rightarrow_G ... \rightarrow_G \alpha_n$ ist eine **Ableitung** von α_n aus α_1.
- Wenn $\alpha_1 = S$ und $\alpha_n \in \Sigma^*$, dann **erzeugt** G das Wort α_n. Erzeugte Wörter bestehen nur aus **Terminalzeichen**.
- Die Sprache von G ist die Menge aller Wörter (Σ^*), die von G erzeugt werden: $L(G)$
- **Chomsky Hierarchie:** Eine Grammatik G ist vom
  - **Typ 0** immer
  - **Typ 1** falls für jede Produktion $\alpha \rightarrow \beta$ ausser $S \rightarrow \epsilon$ gilt $|\alpha| \leq |\beta|$
  - **Typ 2** falls G vom Typ 1 ist und für jede Produktion $\alpha \rightarrow \beta$ gilt $\alpha \in V$
  - **Typ 3** falls G vom Typ 2 ist und für jede Produktion $\alpha \rightarrow \beta$ ausser $S \rightarrow \epsilon$ gilt $\beta \in \Sigma \cup \Sigma V$
  - Typ 3 $\subset$ Typ 2 $\subset$ Typ 1 $\subset$ Typ 0
  - $L(\text{Typ 3}) \subset L(\text{Typ 2}) \subset L(\text{Typ 1}) \subset L(\text{Typ 0})$

- **Grammatiken und Sprachklassen:**
  - **Typ 3, Rechtslineare Grammatik, Reguläre Sprachen**
  - **Typ 2, Kontextfreie Grammatik, Kontextfreie Sprachen**
  - Typ 1, Kontextsensitive Grammatik, Kontextsensitive Sprachen
  - Typ 0, Phrasenstrukturgrammatik, Rekursiv aufzählbare Sprachen

2 Reguläre Sprachen



2.1 Deterministische endliche Automaten (DFA)

- DFA, 5-Tupel $M = (Q, \Sigma, \delta, q_0, F)$
  - Q, endliche Menge von **Zuständen**
  - Σ, endliches **Eingabealphabet**
  - $\delta : \; Q \times \Sigma \to Q$, totale **Übergangsfunktion**
  - $q_0 \in Q$, ein **Startzustand**
  - $F \subseteq Q$, endliche Menge von **Endzuständen**
- Die von DFA M **akzeptierte** Sprache ist $L(M) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$, wobei $\hat{\delta} : \; Q \times \Sigma^* \to Q$ induktiv definiert durch:
  - $\delta(q, a)$, Zustand, den man aus q mit einem **Zeichen** a erreicht
  - $\hat{\delta}(q, w)$, Zustand, den man aus q mit einem **Wort** w erreicht
  - $\hat{\delta}(q, \epsilon) = q$
  - $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ für $a \in \Sigma, w \in \Sigma^*$

2.2 Nicht-Deterministische endliche Automaten (NFA)

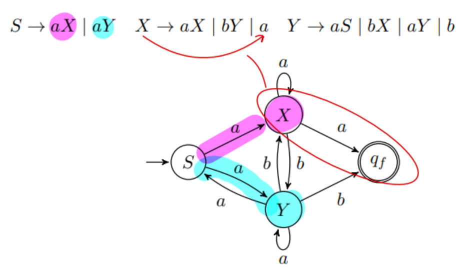
- NFA, 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$
  - $Q, \Sigma, q0$ und F wie beim DFA
  - $\delta : \; Q \times \Sigma \to \mathcal{P}(Q)$, wobei $\mathcal{P}(Q)$ Menge aller Teilmengen von Q

- Die von NFA N **akzeptierte** Sprache ist $L(N) = \{w \in \Sigma^* \mid \hat{\bar{\delta}}(\{q_0\}, w) \cap F \neq \emptyset\}$, wobei
  - $\bar{\delta}(S, a) = \bigcup_{q \in S} \delta(q, a)$, Menge aller Zustände, die man von einem Zustand in S aus mit einem **Zeichen** a erreicht
  - $\hat{\bar{\delta}}(S, w)$, Menge aller Zustände, die man von einem Zustand in S aus mit einem **Wort** w erreicht
  - $\bar{\delta}: \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$
  - $\hat{\bar{\delta}}: \mathcal{P}(Q) \times \Sigma^* \to \mathcal{P}(Q)$

2.3 Rechtsline are Grammatik $\to$ NFA

1. Füge einen Zustand für jedes Nichtterminal, Startsymbol wird zum Startzustand
2. Füge einen Endzustand für jedes Terminal, falls es $S \to \epsilon$ gibt, dann Startzustand ist auch ein Endzustand
3. Für jede Kombination $Y \to aX$ füge eine Kante von Y nach X mit dem Zeichen a
4. Für jede Kombination $Y \to a$ füge eine Kante von Y nach dem Endzustand mit dem Zeichen a

Beispiel:

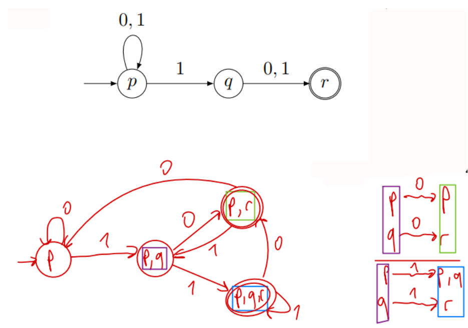


2.4 NFA $\to$ DFA, Potenzmengenkonstruktion

Für jede NFA mit n Zuständen kann der DFA max bis zu 2^n Zustände haben.

1. Für alle Zustände wiederhole (beginnend mit Startzustand q_0):
   (a) Bestimme wohin man mit welcher Kante geht
   (b) Erzeuge neue Zustände durch Vereinigung der auf der rechten Seite stehenden Zuständen mit der selben Kanten, verbinde diese
   (c) **Mindestens einer von den Zuständen**, die in dem neuen Zustand sind, ist ein **Endzustand** $\to$ der neue Zustand wird ein Endzustand

Beispiel:



2.5 ϵ-NFA

- Ein NFA mit **ϵ-Übergängen** ist ein NFA mit $\epsilon \notin \Sigma$ und $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$

2.6 ϵ-NFA $\to$ NFA

1. Lösche überflüssige Zustände:

   $$q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{\epsilon} q_2$$

2. Verbinde die Zustände in der Form mit einer einzigen Kante:

   $$q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{a} q_2 \xrightarrow{\epsilon} q_3$$

   wird zu

   $$q_0 \xrightarrow{a} q_3$$

3. Lösche nicht erreichbare Zustände

4. Falls ϵ in der Sprache ist, dann mache den Startzustand Endzustand

2.7 Reguläre Ausdrücke (REs)

- Reguläre Ausdrücke sind induktiv definiert:
  - $\emptyset$
  - ϵ
  - Für jedes $a \in \Sigma$
  - Wenn α, β RE, auch:

    - $\alpha\beta$
    - $\alpha \mid \beta = \alpha + \beta$
    - α^*

- $*$, Kleene'sche Iteration
- Für RE γ ist die Sprache induktiv definiert:
  - $L(\emptyset) = \emptyset$
  - $L(\epsilon) = \{\epsilon\}$
  - $L(a) = \{a\}$
  - $L(\alpha\beta) = L(\alpha)L(\beta)$
  - $L(\alpha \mid \beta) = L(\alpha) \cup L(\beta)$
  - $L(\alpha^*) = L(\alpha)^*$
- $\alpha \equiv \beta$ gdw. $L(\alpha) = L(\beta)$
- **Rechenregeln über REs:**
  - **Null und Eins Lemma:**
    - $\emptyset \mid \alpha \equiv \alpha \mid \emptyset \equiv \alpha$
    - $\emptyset\alpha \equiv \alpha\emptyset \equiv \emptyset$
    - $\epsilon\alpha \equiv \alpha\epsilon \equiv \alpha$
    - $\emptyset^* \equiv \epsilon$
    - $\epsilon^* \equiv \epsilon$
  - **Assoziativität:**
    - $(\alpha \mid \beta) \mid \gamma \equiv \alpha \mid (\beta \mid \gamma)$
    - $\alpha(\beta\gamma) \equiv (\alpha\beta)\gamma$
  - **Kommutativität:**
    - $\alpha \mid \beta \equiv \beta \mid \alpha$
  - **Distributivität:**
    - $\alpha(\beta \mid \gamma) \equiv \alpha\beta \mid \alpha\gamma$
    - $(\beta \mid \gamma)\alpha \equiv \beta\alpha \mid \gamma\alpha$
  - **Idempotenz:**
    - $\alpha \mid \alpha \equiv \alpha$
- **Stern Lemma:**
  - $\epsilon \mid \alpha\alpha^* \equiv \alpha^*$
  - $\alpha^*\alpha \equiv \alpha\alpha^*$
  - $(\alpha^*)^* \equiv \alpha^*$

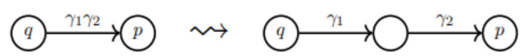
2.8 RE $\rightarrow$ ϵ-NFA

1. Wende folgende Ersetzungsregeln an

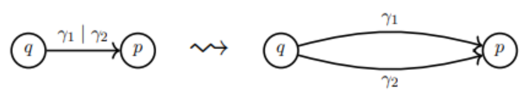
$$\gamma\,\emptyset \rightsquigarrow \emptyset \qquad \gamma \mid \emptyset \rightsquigarrow \gamma \qquad \emptyset^* \rightsquigarrow \epsilon$$
$$\emptyset\,\gamma \rightsquigarrow \emptyset \qquad \emptyset \mid \gamma \rightsquigarrow \gamma$$

2. Wende folgende Transformationsregeln an

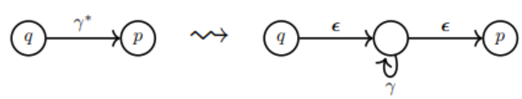
Konkatenation:



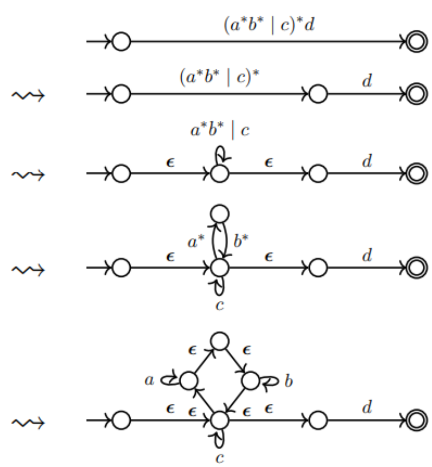
Auswahl:



Iteration:

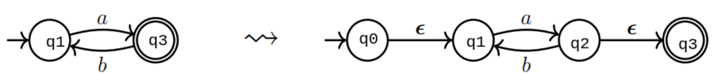


Beispiel:



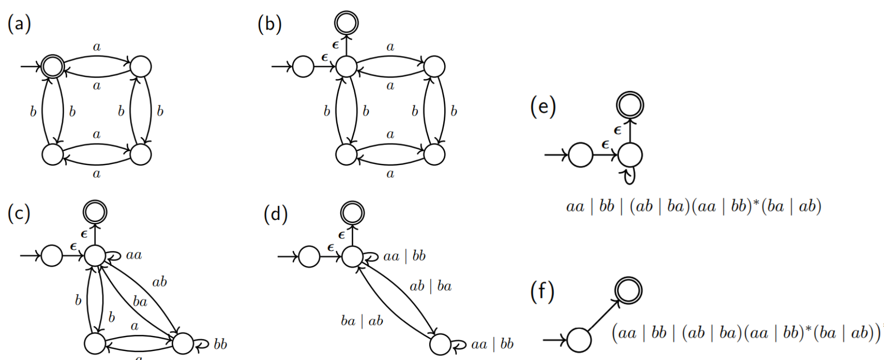
2.9 ϵ-NFA $\rightarrow$ RE

1. Hat Startzustand q_1 eingehende Übergänge, füge einen neuen Startzustand q_0 mit einem ϵ-Übergang nach q_1
2. Füge einen neuen Endzustand q_3 und ϵ-Übergänge nach q_3 von allen Endzuständen (q_2 in diesem Beispiel)



3. Wende die Transformationsregeln von 2.8 an, aber umgekehrt

Beispiel:



2.10 Ardens Lemma

- Sind A, B, X Sprachen mit $\epsilon \notin A$, so gilt:

$$\boxed{X = AX \cup B \Longrightarrow X = A^*B}$$

- Sind α, β, X REs mit $\epsilon \notin L(\alpha)$, so gilt:

$$\boxed{X \equiv \alpha X \mid \beta \Longrightarrow X \equiv \alpha^*\beta}$$

- **Bemerkungen:**
  - $X \equiv X\alpha \mid \beta \Longrightarrow X \equiv \beta\alpha^*$

- $X \equiv \alpha X \mid \beta$ für $\epsilon \in L(\alpha)$
  - hat **keine eindeutige Lösung**: jede Sprache $B \subseteq X$ ist Lösung
  - *Beispiel:* für $\alpha = \epsilon$ und $\beta = b$ kann $X = b$ oder $X = a \mid b$ oder $X = aba \mid b$
- $X \equiv X \mid aX$
  - hat **keine eindeutige Lösung:** $X = \emptyset$ oder $X = \Sigma^*$ oder $X = a^*$
- $X \equiv \alpha X$ für $\epsilon \notin L(\alpha)$
  - hat **eine eindeutige Lösung:** $X = \emptyset$
- $X \equiv \alpha X$ für $\epsilon \in L(\alpha)$
  - hat **keine eindeutige Lösung:** $X = \Sigma^*$ oder $X = ab^*a$
- $X \equiv aXb \mid \epsilon$
  - hat **keine reguläre Lösung:** $X = L$ für $L = \{a^n b^n, n \geq 0\}$
- $X \equiv abX \mid \epsilon$
  - hat **eine eindeutige Lösung:** $X = (ab)^*\epsilon = (ab)^*$

2.11 FA → RE mittels Ardens Lemma

1. FA als Gleichungssystem schreiben
   - für jeden Endzustand X_f füge $X_f \equiv \epsilon$ ein
   - für jeden Zustand X mit

     $$X \xrightarrow{a} X_1$$

     mache $X \equiv aX_1$
2. Gleichungen einsetzen und damit eliminieren
3. Rechenregeln über REs und Ardens Lemma verwenden

2.12 Konversionen bezüglich regulärer Sprachen

$$\begin{tikzcd}
\text{DFA} & \text{NFA} \arrow[l] \arrow[d] & \epsilon\text{-NFA} \arrow[l] \\
& \text{RE} \arrow[ur] \arrow[from=ul] &
\end{tikzcd}$$

RE $\rightarrow$ ϵ-NFA:	RE der Länge $n \rightsquigarrow O(n)$ Zustände
ϵ-NFA $\rightarrow$ NFA:	$Q \rightsquigarrow Q$
NFA $\rightarrow$ DFA:	n Zustände $\rightsquigarrow O(2^n)$ Zustände
NFA $\rightarrow$ RE:	n Zustände $\rightsquigarrow$ RE der Länge $O(3^n)$

2.13 Abschlusseigenschaften regulärer Sprachen

Seien $R, R_1, R_2 \subseteq \Sigma^*$ reguläre Sprachen, dann sind auch

- $R_1 R_2$
- $R_1 \cup R_2$, $R_1 \cap R_2$, $R_1 \setminus R_2$
- $\overline{R}$ bzw. $\Sigma^* \setminus R$
- R^*
- R^R (Spiegelung von R)

2.14 Komplementierung $\overline{R}$ bezüglich FAs

- **Für DFAs:** Vertauschen von Endzuständen und Nicht-Endzuständen
- **Für NFAs:** funktioniert das Vertauschen **nicht**

2.15 Schnitt zweier DFAs, Produktkonstruktion

- Sind M_1 und M_2 DFAs. Dann ist der **Produkt-Automat M** mit $L(M) = L(M_1) \cap L(M_2)$.
- **Produktkonstruktion für M_1 und M_2:**
    1. Erzeuge einen neuen Startzustand aus den Startzuständen der M_1 und M_2
    2. Bestimme wohin man mit welcher Kante geht
    3. Erzeuge neue Zustände durch Vereinigung der auf der rechten Seite stehenden Zuständen mit der selben Kanten, verbinde diese
    4. **Alle Zustände**, die in dem neuen Zustand sind, sind **Endzustände** $\rightarrow$ der neue Zustand wird ein Endzustand

2.16 Vereinigung zweier DFAs

- Sind M_1 und M_2 DFAs. Dann ist M mit $L(M) = L(M_1) \cup L(M_2)$.
- **Konstruktion für M_1 und M_2:** Gleich wie die Produktkonstruktion bis auf 4.
    4. **Mindestens einer von den Zuständen**, die in dem neuen Zustand sind, ist ein **Endzustand** $\rightarrow$ der neue Zustand wird ein Endzustand

2.17 Pumping Lemma