FgdI - Klausurzusammenfassung

Mengen:

Der *Durchschnitt* $A \cap B = \{c : c \in A \text{ und } c \in B\} = \{a \in A : a \in B\} = \{b \in B : b \in A\}$. Zwei Mengen heißen *disjunkt*, falls ihr Durchschnitt die leere Menge ist.

Die *Vereinigung* $A \cup B = \{c : c \in A \text{ oder } c \in B\}$ (beachte, dass das logische "*oder*" nicht exklusiv ist: die Elemente von $A \cup B$ sind genau diejenigen, die Element von mindestens einer der Mengen A oder B sind).

Mengendifferenz: $A \setminus B = \{a \in A : a \notin B\}$. Beispiel: $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$.

Relationen:

Einige wichtige Eigenschaften, die eine zweistellige Relation $R \subseteq A^2$ haben *kann*:

Reflexivität R heißt reflexiv gdw. für alle $a \in A$ gilt aRa.

Symmetrie R heißt symmetrisch gdw. für alle $a, b \in A$ gilt: $aRb \Leftrightarrow bRa$.

Transitivität R heißt transitiv gdw. für alle $a, b, c \in A$ gilt: $(aRb \text{ und } bRc) \Rightarrow aRc$.

Ordnungsrelation: reflexiv, antisymmetrisch, transitiv
Äquivalenzrelation: reflexiv, symmetrisch, transitiv

Funktionen:

Definition 1.1.14 Sei $f : A \to B$ eine Funktion.

(i) f heißt *surjektiv*, falls $f[A] = B$ (jedes Element des Bildbereichs wird mindestens einmal als Wert angenommen).

(ii) f heißt *injektiv*, falls für alle $a, a' \in A$ mit $a \neq a'$ gilt dass $f(a) \neq f(a')$ (jedes Element des Bildbereichs wird höchstens einmal als Wert angenommen).

(iii) f heißt *bijektiv*, falls f injektiv und surjektiv ist.

Boolesche Algebra:

Definition 1.1.22 Eine Struktur $\mathcal{B} = (B, \cdot, +, ', 0, 1)$ heißt *Boolesche Algebra*, falls die folgenden Axiome erfüllt sind:

(1) $\cdot$ und $+$ sind assoziativ und kommutativ.

(2) *Idempotenz*; für alle $b \in B$ gilt: $b \cdot b = b + b = b$

(3) *Distributivgesetze*:
  (i) für alle $a, b, c \in B$ gilt: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
  (ii) für alle $a, b, c \in B$ gilt: $a + (b \cdot c) = (a + b) \cdot (a + c)$

(4) *de Morgan Gesetze*:
  (i) für alle $a, b \in B$ gilt: $(a \cdot b)' = a' + b'$
  (ii) für alle $a, b \in B$ gilt: $(a + b)' = a' \cdot b'$

(5) *Absorption*; für alle $a, b \in B$ gilt: $a \cdot (a + b) = a + (a \cdot b) = a$

(6) $'$ ist *involutiv*: für alle $b \in B$ gilt $(b')' = b$

(7) für alle $b \in B$ gilt: $b \cdot 0 = 0,\ b + 1 = 1$

(8) für alle $b \in B$ gilt: $b \cdot 1 = b + 0 = b$

(9) $1 \neq 0$, und für alle $b \in B$ gilt: $b \cdot b' = 0$ und $b + b' = 1$

$\cdot$	0	1
0	0	0
1	0	1

$+$	0	1
0	0	1
1	1	1

	0	1
$'$	1	0

Endliche Automaten – reguläre Sprache:

- $\Sigma \neq \emptyset$ endliches Alphabet; Σ^* Menge aller Σ-Wörter;
- Teilmengen $L \subseteq \Sigma^*$ heißen Σ-*Sprachen*.
- $\varepsilon \in \Sigma^*$ das leere Wort.
- $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ die Menge der nicht-leeren Σ-Wörter;
  für $n \in \mathbb{N}$: $\Sigma^n = \{w \in \Sigma^n : |w| = n\}$ Menge der Wörter der Länge n.
- $\begin{array}{rcll} \cdot : \Sigma^* \times \Sigma^* & \longrightarrow & \Sigma^* & \text{Konkatenation von Wörtern;} \\ (u, v) & \longmapsto & uv & \end{array}$
- $(\Sigma^*, \cdot, \varepsilon)$ ist das zugehörige Wort-Monoid.

Durchschnitt von zwei Σ-Sprachen, $L_1 \cap L_2$,

Vereinigung von zwei Σ-Sprachen, $L_1 \cup L_2$,

Komplement einer Σ-Sprache, $\overline{L} = \Sigma^* \setminus L$.

(i) $L(L_1 \cup L_2) = (LL_1) \cup (LL_2)$ und $(L_1 \cup L_2)L = (L_1L) \cup (L_2L)$.

(ii) $L^* = \{\varepsilon\} \cup L \cdot L^*$.

(iii) $(L_1^* \cup L_2^*)^* = (L_1 \cup L_2)^*$.

Reguläre Ausdrücke:

(i) $\emptyset$ ist ein regulärer Ausdruck.

(ii) für $a \in \Sigma$ ist $\mathbf{a}$ ein regulärer Ausdruck.

(iii) für $\alpha, \beta \in \mathrm{REG}(\Sigma)$ ist $(\alpha + \beta) \in \mathrm{REG}(\Sigma)$

(iv) für $\alpha, \beta \in \mathrm{REG}(\Sigma)$ ist $(\alpha\beta) \in \mathrm{REG}(\Sigma)$

(v) für $\alpha \in \mathrm{REG}(\Sigma)$ ist $\alpha^* \in \mathrm{REG}(\Sigma)$

(i) $L(\emptyset) := \emptyset$.

(ii) $L(\mathbf{a}) := \{a\}$ für jedes $a \in \Sigma$.

(iii) $L(\alpha + \beta) := L(\alpha) \cup L(\beta)$.

(iv) $L(\alpha\beta) := L(\alpha) \cdot L(\beta)$.

(v) $L(\alpha^*) := (L(\alpha))^*$.

Deterministischer endlicher Automat (DFA):

Definition 2.2.2 [DFA] Ein *deterministischer endlicher* Σ-*Automat* ist spezifiziert als

$$\mathcal{A} = (\Sigma, Q, q_0, \delta, A).$$

Dabei ist	Q	die endliche, nicht-leere Zustandsmenge
	$q_0 \in Q$	der *Anfangszustand*
	$A \subseteq Q$	die Menge der *akzeptierenden Zustände*
	$\delta \colon Q \times \Sigma \to Q$	die *Übergangsfunktion*.

Die *von* $\mathcal{A}$ *akzeptierte Sprache* oder *erkannte Sprache* ist

$$L(\mathcal{A}) := \{w \in \Sigma^* : \mathcal{A} \text{ akzeptiert } w\}.$$

Erweiterung der Funktion: $\hat{\delta} \colon Q \times \Sigma^* \longrightarrow Q$,

Bemerkung: Die von $\mathcal{A}$ akzeptierte Sprache ist dann $L(\mathcal{A}) = \{w \in \Sigma : \hat{\delta}(q_0, w) \in A\}$.

$\hat{\delta}(q, w)$ ist rekursiv definiert über $w \in \Sigma^*$ durch: $\hat{\delta}(q, \varepsilon) := q$
$\hat{\delta}(q, wa) := \delta(\hat{\delta}(q, w), a)$.

Nichtdeterministischer endlicher Automat (NFA):

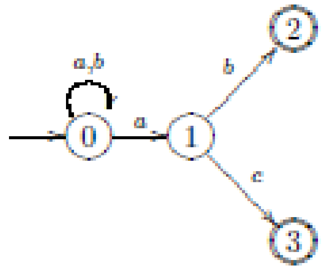
$$\mathcal{A} = (\Sigma, Q, q_0, \Delta, A)$$

mit Q, $q_0 \in Q$ und $A \subseteq Q$ wie bei DFA, aber mit einer Transitions*relation*

$$\Delta \subseteq Q \times \Sigma \times Q.$$

$L(\mathcal{A}) = \{w \in \Sigma^* : \mathcal{A} \text{ hat mindestens eine akzeptierende Berechnung auf } w\}$.

Vom NFA zum DFA (Potenzmengentrick);



δ	a	b	c
$\{0\}$	$\{0,1\}$	$\{0\}$	$\emptyset$
$\{0,1\}$	$\{0,1\}$	$\{0,2\}$	$\{3\}$
$\{0,2\}$	$\{0,1\}$	$\{0\}$	$\emptyset$
$\{3\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

Neuer Zustand {Q}

Die akzeptierenden Zustände sind $\{0,2\}$ und $\{3\}$.

Schritt 1: Tabelle erstellen mit : Zustand | e1 | e2 | … | en
Schritt 2: Startzustand (hier 0) in Zeile eintragen
Schritt 3: Ergebnisse der Übergangsfunktionen d(e1...) für die Zeile eintragen
Schritt 4: Neue Zustände zusammenfassen und in Spalte d eintragen.. → Schritt 3
Schritt 5: Neue Zustände mit akzeptierenden Zustand sind neue akzept. Zustände
Schritt 6: Zustände mit neuen Übergangsfunktionen zeichnen

Minimierung von DFA:

q_3	☐			
q_2	☐	☐		
q_1	☐	☐	☐	
q_0	☐	☐	☐	☐
	q_4	q_3	q_2	q_1

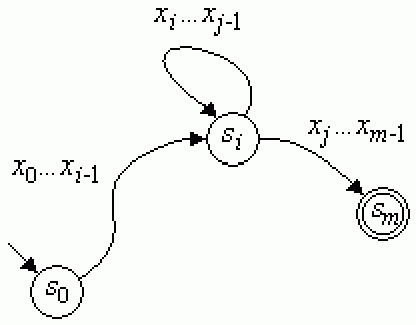
Schritt 1: Zustände löschen welche vom Startpunkt aus nicht erreicht werden können
Schritt 2: Tabelle für den Minimierungsalgorithmus erstellen (Bild oben)
Schritt 3: 1. Unterscheiden zwischen akzept. Und nicht akzept. Zuständen
2. Unterscheiden zwischen Übergängen durch Buchstaben in unterschiedliche Zustände (z.B. akzept. und nicht akzept. Zustände)
3. Immer längere Wörter zum Unterscheiden benutzen
Schritt 4: Gleiche Zustände zu einem Zustand zusammenfassen

→ Beispiel Übungsblatt 4 – G1 und H2

Pumping Lemma: Beweis durch Widerspruch einer nichtregulären Sprache
Sei A ein Alphabet und L Teilmenge von A* eine reguläre Sprache. Dann lassen sich alle Wörter x € L ab einer gewissen Länge |x| >= p (der Pumping Länge) darstellen als:

x = uvw mit u,v,w € A*
wobei
|v| > 0 und |uv| <= p
sodass gilt:
uv^kw € L für alle k € $\mathbb{N}_0$ (also inklusive 0!!)

!!!!!Beispiele im Anhang!!!!!!!!!!Beispiele im Anhang!!!!!!!!!!Beispiele im Anhang!!!!!



Grammatiken:

Definition 3.1.1 [Grammatik] Eine *Grammatik* G ist spezifiziert als

$$G = (\Sigma, V, P, X_0)$$

Dabei ist	Σ	das endliche *Terminalalphabet*, $\Sigma \neq \emptyset$,
	V	die endliche Menge von *Variablen*, $V \cap \Sigma = \emptyset$
	$X_0 \in V$	die *Startvariable*/das *Startsymbol*
	$P \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$	die endliche Menge der *Produktionen/Regeln*

Eine Grammatik ist kontextfrei wenn es keine Produktionen der Art
xA → b gibt.
Sie ist kontextfrei, wenn alle Variablen auf der linken Seite alleinstehend sind, also
A → bBa | bb
B → aBa | ab

Chomsky-Hierarchie:

Typ 3 **regulär**	alle Produktionen rechtslinear, d.h. von der Form $X \to \varepsilon$, $X \to a$ oder $X \to aY$.
Typ 2 **kontextfrei**	nur Produktionen von der Form $X \to v$.
Typ 1 **kontextsensitiv**	nur harmlose ε-Produktionen; alle anderen Produktionen nicht verkürzend: $v \to v'$ mit $\lvert v'\rvert \geqslant \lvert v\rvert$.
Typ 0 **allgemein**	keine Einschränkungen.

!!!!!!Siehe Anhang!!!!!!!!!!!!Siehe Anhang!!!!!!!!!!!!Siehe Anhang!!!!!!!!!!!!Siehe Anhang!!!!!!

Kontextfreie Sprachen (Chomsky Normalform für Typ 2 Grammatiken):

Definition 3.3.1 Eine kontextfreie (Typ 2) Grammatik G ohne ε-Produktionen ist in *Chomsky-Normalform*, falls sie nur Produktionen von der Form $X \to YZ$ und $X \to a$ hat ($X, Y, Z \in V, a \in \Sigma$).

Umformen einer kontextfreien Grammatik in die Chomsky-Normalform:

!!!!Siehe Übungsblatt 5 – H1 !!!!!

Eine Grammatik ist in der Normalform, wenn alle Produktionen nur folgende Formen haben:

X → YZ oder A → a..

Alle anderen Formen sind unzulässig!

Schritt 1: Epsilon Abbildungen Streichen, ABER alle Produktionen die diese Abbildung benutzen müssen angepasst werden: z.B.:

S → AB
A → DE | epsilon

daraus folgt nach dem Streichen von A → epsilon:

S → B
UND!! S → **DE**B (da dies auch von A benutzt wird)

ALSO: alle möglichen Kombinationen ohne epsilon aufschreiben!

Schritt 2: Variablen für Buchstaben einfügen (A → a, B → b, …)

Schritt 3: Kettenproduktionen eliminieren (X → Y, A → B, …)

Schritt 4: Mehr als 2 Variablen aufteilen:

A → ABC

wird zu:

A → AD
D → BC

Nicht kontextfreie Sprachen: Pumping Lemma

Satz 3.3.8 (Pumping Lemma) *Für jede kontextfreie Sprache $L \subseteq \Sigma^*$ gibt es ein $n \in \mathbb{N}$, sodass sich jedes $x \in L$ mit $|x| \geqslant n$ zerlegen lässt als $x = y \cdot u \cdot v \cdot w \cdot z$, wobei $uw \neq \varepsilon$, sodass für alle $m \in \mathbb{N}$*

$$y \cdot u^m \cdot v \cdot w^m \cdot z = y \cdot \underbrace{u \cdots u}_{m \text{ mal}} \cdot v \cdot \underbrace{w \cdots w}_{m \text{ mal}} \cdot z \in L.$$

Man kann dabei u, v, w so wählen, dass $|uvw| \leqslant n$.

!!!!!Beispiele im Anhang!!!!!!!!!!Beispiele im Anhang!!!!!!!!!!Beispiele im Anhang!!!!!

CYK – Algorithmus: