Lösung

Einführung in die theoretische Informatik – Klausur

Beachten Sie: Soweit nicht anders angegeben, ist stets eine Begründung bzw. der Rechenweg anzugeben!

Hinweise

- $\mathbb{N}$ bezeichnet die natürlichen Zahlen **einschließlich** der 0.
- $|w|_a$ mit $a \in \Sigma$ bezeichnet die Anzahl der as in w, z.B. $|aabab|_a = 3$ und $|aabab|_b = 2$.

Aufgabe 1 Quiz 5P

Begründen Sie, ob die jeweilige Aussage korrekt oder inkorrekt ist. Beziehen Sie sich auf die entsprechenden Ergebnisse aus der Vorlesung.

Beispiel:

- **Behauptung**: Sei A ein DFA mit n Zuständen. Dann hat die Myhill-Nerode-Relation zu $L(A)$ genau n Äquivalenzklassen.
- **Antwort**: Inkorrekt. Sei A ein DFA. Konstruiere A' aus A durch Hinzufügen eines neuen, unerreichbaren Zustands q mit $\delta(q, a) = q$ für alle Alphabetszeichen a. Dann gilt $L(A) = L(A')$.

Behauptungen:

(a) Die Sprache $L = L(ab(a|b)^*)$ hat fünf Äquivalenzklassen bezüglich der Äquivalenzrelation $\equiv_L$.

Erinnerung: $u \equiv_L v \Leftrightarrow (\forall w \in \{a, b\}^*.uw \in L \Leftrightarrow vw \in L)$.

(b) Sei A ein minimaler DFA mit n Zuständen. Dann hat der minimale DFA zu $\overline{L(A)}$ ebenfalls n Zustände.

(c) Sei L eine kontextfreie Sprache und L' eine deterministisch kontextfreie Sprache. Dann ist $L \cap L'$ eine kontextfreie Sprache.

(d) Seien G_1, G_2 kontextfreie Grammatiken. Dann ist es entscheidbar, ob $L(G_1) \setminus L(G_2) \neq \emptyset$ gilt.

(e) Sei G eine kontextfreie Grammatik. Dann gilt $L(G) \leq_p SAT$.

Lösung:

(a) Inkorrekt. Der minimale DFA für L hat vier Zustände. Somit gibt es genau vier Äquivalenzklassen.

(b) Korrekt. Da DFAs durch invertieren der Endzustände komplementiert werde können, hat der minimale DFA für $\overline{L(A)}$ maximal n Zustände. Angenommen er hätte $m < n$ Zustände, dann hätte der minimale DFA für $L(A)$ maximal $m < n$ Zustände. Widerspruch. Somit hat der minimale DFA für $\overline{L(A)}$ genau n Zustände.

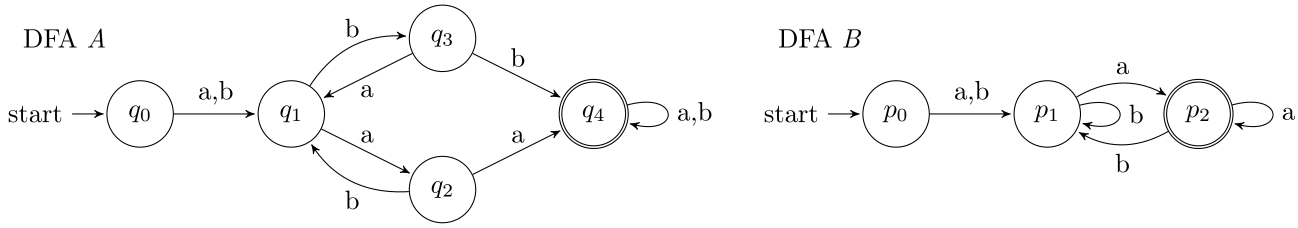
(c) Inkorrekt. Sei $L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$ und $L_2 = \{a^n b^m c^m \mid n, m \in \mathbb{N}\}$ und beide Sprachen sind sogar DCFL. Der Schnitt ist aber nicht CFL (laut VL).

(d) Inkorrekt: $L(G_1) \setminus L(G_2) \neq \emptyset \Leftrightarrow L(G_1) \subseteq L(G_2)$, letzteres ist nach VL unentscheidbar für kontextfreie Grammatiken G_1, G_2.

(e) Korrekt. Umwandlung in CNF und CYK sind in PTIME. Führe Algorithmus aus und bestimme $w \in L(G)$ und reduziere dementsprechend auf `true` und `false`.

Aufgabe 2 Endliche Automaten 7P

(a) Konstruieren Sie zu folgenden DFAs A, B einen DFA C mit $L(C) = L(A) \cap L(B)$. Verwenden Sie die Produktkonstruktion aus der Vorlesung, um C zu konstruieren.



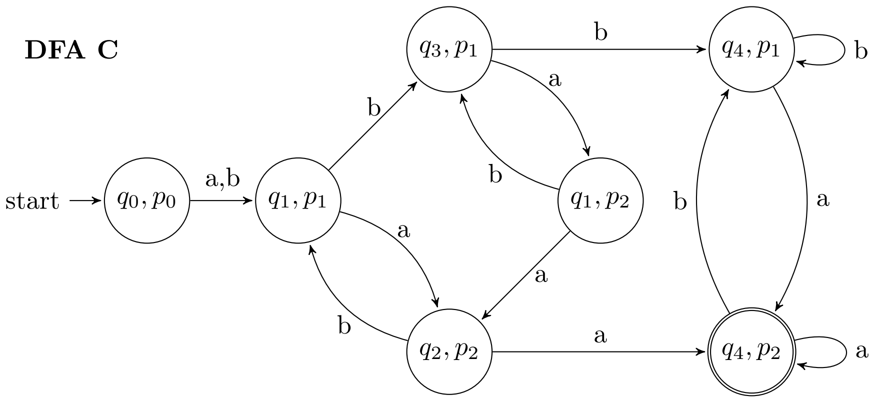
Geben Sie C graphisch an. Konstruieren Sie **nur** den vom Startzustand von C aus tatsächlich erreichbaren Teil von C.

(b) Beschreiben Sie, wie man mittels der Produktkonstruktion für zwei DFAs A, B entscheiden kann, ob $L(A) \subseteq L(B)$ gilt.

Wenden Sie dann Ihr Verfahren auf die Automaten aus (a) an. Sollte $L(A) \not\subseteq L(B)$ gelten, geben Sie ein kürzestes Wort $w \in L(A) \setminus L(B)$ an.

Lösung:

(a) Produktkonstruktion:



(b) $L(A) \subseteq L(B)$ gdw. für jeden (vom Startzustand erreichbaren [nur die Zustände sollten ja angegeben werden]) Zustand (q_A, q_B) von C gilt $q_A \in F_A \Rightarrow q_B \in F_B$

Negiert: $L(A) \setminus L(B) \neq \emptyset$ gdw. es existiert ein Zustand (q_A, q_B) in C mit $q_A \in F_A \wedge q_B \notin F_B$

Ein kürzestes Wort in $L(A) \setminus L(B)$ – soweit existent – findet man somit durch Anwendung z.B. von Breitensuche, um einen kürzesten Pfad vom Startzustand zu einem Zustand aus $F_A \times (Q_B \setminus F_B)$ zu bestimmen.

Im obigen Beispiel gilt $F_A \times (Q_B \setminus F_B) = \{(q_4, p_1), (q_4, p_0)\}$ mit $abb, bbb \in L(A) \setminus L(B)$.

Aufgabe 3 Pumping-Lemma 4P

Zeigen Sie mittels des Pumping-Lemmas für reguläre Sprachen: $L = \{w \in \{a,b\}^* \colon |w|_a = |w|_b\}$ ist nicht regulär.

Lösung: Sei L regulär und n eine dann existierende Pumping-Lemma-Konstante für L.

Wähle $z = a^n b^n \in L$. Offensichtlich gilt $z \in L$ und $|z| \geq n$ und somit existiert eine Zerlegung $z = uvw$ mit (1) $|uv| \leq n$, (2) $v \neq \varepsilon$ und (3) $\forall i \in \mathbb{N} \colon uv^i w \in L$.

Wegen (1) muss $uv = a^k$ mit $k \leq n$ gelten. Wegen (2) folgt $v = a^j$ mit $1 \leq j \leq n$. Mit (3) folgt schließlich der Widerspruch $L \ni uw = a^{n-j} b^n$, da $n - j < n$.

L ist somit nicht regulär.

Aufgabe 4 Kontextfreie Sprachen 6P

Geben Sie jeweils eine kontextfreie Grammatik für die folgende Sprachen an. Die Grammatiken **müssen** jeweils **höchstens** 10 Variablen und 10 Produktionen besitzen, und für jede Produktion $X \to w$ muss $|w| \leq 3$ gelten. Sie müssen **nicht** begründen, warum die Grammatik korrekt ist.

(a) $L_1 = \{w \in \{a,b,c,d\}^* \mid |w|_a + |w|_b = |w|_c + |w|_d\}$

(b) $L_2 = \{a^{256}\}$

(c) $L_3 = \{w_1 w_2 \in \{0,1\}^* \mid |w_1| = |w_2| \wedge w_1 \neq w_2\}$

Lösung:

(a)

$$S \to XSY \mid YSX \mid SS \mid \varepsilon \qquad X \to a \mid b \qquad Y \to c \mid d$$

Begründung: Zunächst können a, b bzw. c, d als jeweils ein Zeichen X bzw. Y aufgefasst werden (daher: $X \to a|b$ bzw. $Y \to c|d$). Jedes Wort $w \in \{X, Y\}^*$ lässt sich eindeutig (vlg. korrekte Klammerausdrücke über einem Klammertyp) in minimale Blöcke mit $|w|_X = |w|_Y$ faktorisieren (daher $S \to SS$), im Gegensatz zu den Klammerausdrücken, kann jedoch auch mit ")" begonnen werden (daher sowohl $S \to XSY$ als auch $S \to YSX$). Zzgl. noch $S \to \varepsilon$.

(b)

$$X_{256} \to X_{128}X_{128} \qquad X_{128} \to X_{64}X_{64} \qquad X_{64} \to X_{32}X_{32} \qquad \ldots \qquad X_2 \to X_1X_1 \qquad X_1 \to a$$

Begründung: Iteriertes Quadrieren entsprechend schneller Exponentation bzw. direktes Codieren eines perfekten Binärbaums der Höhe 8 (ohne Blätter für Terminale) als CFG.

(c)

$$S \to AB \mid BA \qquad A \to XAX \mid 0 \qquad B \to XBX \mid 1 \qquad X \to 0 \mid 1$$

Begründung: Satzformen sind von der Form $X^iAX^iX^jBX^j = X^iAX^{i+j}BX^j = X^iAX^jX^iBX^j$ bzw. $X^iBX^iX^jAX^j = \ldots = X^iBX^jX^iAX^j$ mit $i, j \in \mathbb{N}$.

Jedes erzeugte Wort der Länge $2(i + j + 1)$ unterscheidet sich damit zumindest in den beiden Zeichen an den Positionen $i + 1$ und $(i + j + 1) + (i + 1)$ (wobei die Positionen ab 1 nummeriert werden).

Aufgabe 5 CYK 6P

Wir betrachten die Grammatik $G = (\{S, T, U, X, Y, Z\}, \{x, y, z\}, P, S)$ in CNF mit den folgenden Produktionen P:

$$S \to SZ \mid TS \mid ZT \mid z \qquad T \to UT \mid XU \mid y \qquad U \to SY \mid XY$$
$$X \to x \qquad Y \to y \qquad Z \to z$$

Bestimmen Sie mit dem CYK-Algorithmus, ob folgende Wörter w_1, w_2 in $L(G)$ liegen. Der CYK-Algorithmus **muss** entsprechend der Vorlesung dargestellt werden. Geben Sie weiterhin für jedes der beiden Wörter an, wie viele verschiedene Ableitungsbäume es bezüglich G für das jeweilige Wort gibt.

(a) $w_1 = yzzzy$

(b) $w_2 = xxyzz$

Lösung:

(a) Es gilt $yzzzy \notin L(G)$, da $S \notin V_{1,5}$. Somit gibt es keine Ableitungsbäume.

15 U				
14 S	25 U			
13 S	24 S	35 U		
12 S	23 S	34 S	45 S, U	
11 T, Y	22 Z, S	33 Z, S	44 Z, S	55 T, Y
y	z	z	z	y

(b) Es gilt $xxyzz \in L(G)$, da $S \in V_{1,5}$.

15 S				
14 S	25			
13 T	24	35 S		
12	23 U	34 S	45 S	
11 X	22 X	33 T, Y	44 Z, S	55 Z, S
x	x	y	z	z

Es gibt zwei verschiedene Ableitungsbäume, was man durch Anpassen des CYK-Algorithmus direkt mitberechnen kann (vgl. Erweiterung des CYK zur Bestimmung der Ableitungsbäume selbst).

15 $(S,2)$				
14 $(S,1)$	25			
13 $(T,1)$	24	35 $(S,2)$		
12	23 $(U,1)$	34 $(S,1)$	45 $(S,1)$	
11 $(X,1)$	22 $(X,1)$	33 $(T,1)$, $(Y,1)$	44 $(Z,1)$, $(S,1)$	55 $(Z,1)$, $(S,1)$
x	x	y	z	z

Aufgabe 6 Primitiv rekursive Funktionen 4P

Zeigen Sie: Wenn $f, g\colon \mathbb{N} \to \mathbb{N}$ primitiv rekursiv sind, dann sind auch die folgenden Funktionen $h_1, h_2\colon \mathbb{N} \to \mathbb{N}$ primitiv rekursiv.

(a) $h_1(x) = \begin{cases} 1 & \text{falls } f(x) = g(x) \\ 0 & \text{sonst} \end{cases}$ (b) $h_2(x) = \sum_{i=0}^{f(x)} g(i)$

Beachten Sie: h_1, h_2 **müssen** unter Verwendung des (erweiterten) Schemas der primitiven Rekursion und der (erweiterten) Komposition jeweils explizit aus den primitiv rekursiven Basisfunktionen zzgl. der primitiv rekursiven Funktionen $+$, $\dot{-}$ und $\times$ konstruiert werden.

Lösung:

(a) Wir definieren die Funktion $eq(x, y)$:

$$eq(x, y) := 1 \dot{-} ((x \dot{-} y) + (y \dot{-} x))$$

Und somit ist $h_1(x)$:

$$h_1(x) := eq(f(x), g(x))$$

(b) Setze $g_{\text{sum}}(0) := 0$ und $g_{\text{sum}}(m + 1) := g_{\text{sum}}(m) + g(m)$ (primitive Rekursion) und somit gilt:

$$g_{\text{sum}}(m + 1) = \sum_{k=0}^{m} g(k)$$

Und somit ist $h_2(x)$:

$$h_2(x) := g_{\text{sum}}(f(x) + 1) \quad = \sum_{i=0}^{f(x)} g(i)$$

Aufgabe 7 Entscheidbarkeit 4P

Begründen Sie entsprechend Aufgabe 1, ob die jeweilige Aussage korrekt oder inkorrekt ist. Wenn die Sprache L entscheidbar (semi-entscheidbar) ist, beschreiben Sie einen Algorithmus, der die charakteristische Funktion χ_L (die Funktion χ'_L) berechnet. Wenn L unentscheidbar ist, leiten Sie einen Widerspruch zu einem Ergebnis der Vorlesung ab.

Behauptungen:

(a) Wenn A und B entscheidbare Sprachen sind, dann ist $A \cap B$ entscheidbar.

(b) Wenn A und $A \cup B$ entscheidbar sind, dann ist B entscheidbar.

(c) Das Problem, ob $L(M) \neq \emptyset$ für eine gegebene Turingmaschine M gilt, ist semi-entscheidbar.

(d) Das Problem, ob $L(M) = \emptyset$ für eine gegebene Turingmaschine M gilt, ist semi-entscheidbar.

Lösung:

(a) Korrekt.

Sei T_A DTM, die A entscheidet, T_B DTM, die B entscheidet.

DTM zu $A \cap B$: Gegeben x, berechne $T_A(x)$. Falls $T_A(x) = 0$, lehne x ab, ansonsten berechne $T_B(x)$. Gilt $T_B(x) = 0$, lehne x ab, sonst akzeptiere x. Da T_A, T_B die jeweiligen Mengen entscheiden, terminieren beide DTM immer, womit auch die DTM zu $A \cap B$ stets mit dem korrekten Ergebnis terminiert, womit $A \cap B$ entscheidbar ist.

(b) Inkorrekt.

Sei $B = H_0 \subseteq \{0,1\}^*$ das Halteproblem auf leere Eingabe und $A = \{0,1\}^*$. Dann sind A und $A \cup B$ entscheidbar, aber B nicht.

(c) Korrekt.

"Dove-Tailing": NTM, die ein Wort in $L(M)$ raten kann, soweit es existiert, determinisieren.

(d) Inkorrekt.

Kann nicht semi-entscheidbar sein, sonst wäre $L(M) \neq \emptyset$ entscheidbar, womit das Halteproblem auf leere Eingabe entscheidbar wäre mittels der Reduktion: Bilde TM-Codierung w auf Codierung w' der TM "Falls $x \neq \varepsilon$, lehne x ab, sonst warte, bis $M_w(\varepsilon)$ terminiert hat, und akzeptiere dann ε". Offensichtlich akzeptiert $M_{w'}$ höchstens ε und das genau dann, wenn M_w auf ε terminiert.

Aufgabe 8 NP-Schwere 4P

Zeigen Sie, dass folgendes Problem **NP**-schwer ist, indem Sie eine Reduktion von 3KNF-SAT auf das Problem angeben.

- **Gegeben**: Reguläre Ausdrücke $r_1, r_2, \ldots, r_n$.
- **Entscheide**: $L(r_1) \cap L(r_2) \cap \ldots \cap L(r_n) \neq \emptyset$.

Veranschaulichen Sie Ihre Reduktion anhand der Formel $(x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_1 \vee x_3 \vee \neg x_4)$.

Hinweis: Konstruieren Sie für jede Klausel C_i einer gegebenen Formel $\varphi = \bigwedge_{i=1}^{k} C_i$ in 3KNF einen geeigneten regulären Ausdruck r_i.

Lösung: Identifiziere minimale passende Belegungen β für gegebene 3KNF-Formel φ (oBda über den Variablen $\{x_1, \ldots, x_n\}$, insbesondere kommt jede Variable mindestens einmal in φ vor, d.h. $n \leq |\varphi|$) mit Wort $\beta(x_1)\beta(x_2)\ldots\beta(x_n) \in \{0,1\}^n$.

Bilde jede Klausel auf den regulären Ausdruck ab, der genau die Wörter/minimalen Belegungen akzeptiert, unter denen die Klausel erfüllt ist – Bsp: $n = 4$ und Klausel $x_1 \vee \neg x_2 \vee x_4$ wird auf $1(0|1)(0|1)(0|1) \mid (0|1)0(0|1)(0|1) \mid (0|1)(0|1)(0|1)1$ abgebildet.

(Bemerkung: $1(0|1)^n$ muss vollständig ausgeschrieben werden, ansonsten wird Problem schwieriger.)

Der reguläre Ausdruck hat dabei maximal die Länge $3 \cdot (5(n-1)+1)+2$ und kann damit in der Zeit $O(|\varphi|)$ konstruiert werden.

Damit lassen sich in Zeit $O(kn) \subseteq O(|\varphi|^2)$ zu einer Formel mit n Variablen und k Klausen k reguläre Ausdrücke konstruieren, deren Schnitt gerade alle Wörter/minimalen Belegungen codiert, welche die Formel erfüllen. Der Schnitt ist daher genau dann nicht leer, wenn die gegebene Formel erfüllbar ist. Damit ist das Problem NP-schwer.