Karlsruher Institut für Technologie

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Nachklausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2011/2012

Hier Aufkleber mit Name und Matrikelnr. anbringen

Vorname: ____________________

Nachname: ____________________

Matrikelnummer: ____________________

Beachten Sie:

- Bringen Sie den Aufkleber mit Ihrem Namen und Ihrer Matrikelnummer auf diesem Deckblatt an und beschriften Sie jedes Aufgabenblatt mit Ihrem Namen und Ihrer Matrikelnummer.
- Schreiben Sie die Lösungen auf die Aufgabenblätter und Rückseiten. Zusätzliches Papier erhalten Sie bei Bedarf von der Aufsicht.
- Zum Bestehen der Klausur sind **20** der möglichen **60** Punkte hinreichend.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte					Erreichte Punkte				
	a	b	c	d	Σ	a	b	c	d	Σ
1	1	3	-	-	4			-	-	
2	1	4	1	-	6				-	
3	2	4	-	-	6			-	-	
4	1	4	-	-	5			-	-	
5	4	-	-	-	4		-	-	-	
6	2	1	1	-	4				-	
7	1	1	1	1	4					
8	2	1	3	3	9					
9	2	3	3	-	8				-	
10	10x1				10					
Σ					60					

Aufgabe 1: (1 + 3 = 4 Punkte)

Gegeben sei die Grammatik $G = (\Sigma, V, S, R)$ mit Terminalen $\Sigma = \{a, b, e, f\}$, Nichtterminalen $V = \{S, A, B, E, F\}$, Startsymbol S und Produktionen

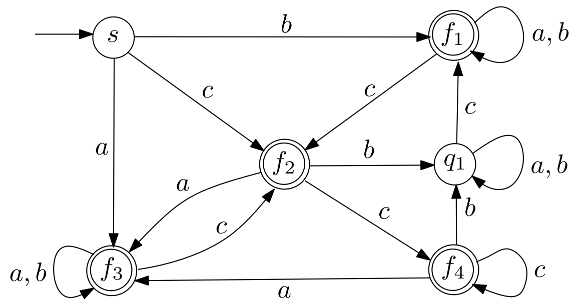
$$\begin{aligned} R = \{S &\rightarrow EB \\ A &\rightarrow EF \mid a \mid f \\ B &\rightarrow AE \mid b \\ E &\rightarrow FA \mid b \mid e \\ F &\rightarrow AB \mid f\}. \end{aligned}$$

(a) Welchen maximalen Chomsky-Typ hat G? Begründen Sie Ihre Antwort.

(b) Überprüfen Sie mit dem Cocke-Younger-Kasami Algorithmus, ob das Wort $fabfe$ in der Sprache $L(G)$ enthalten ist. Geben Sie dazu alle Zwischenergebnisse an.

Aufgabe 2: (1 + 4 + 1 = 6 Punkte)

Gegeben sei folgender endlicher Automat $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b, c\}$.



(a) Ist $\mathcal{A}$ determinisitisch? Begründen Sie Ihre Antwort.

(b) Konstruieren Sie den Minimalautomaten zu $\mathcal{A}$. Geben Sie die daraus folgenden Äquivalenzklassen der Zustände in $\mathcal{A}$ explizit an. Geben Sie das Ergebnis als Zustandsübergangsdiagramm an und bezeichnen Sie die Zustände mit den Äquivalenzklassen, die sie repräsentieren.

(c) Geben Sie für jede der folgenden Aussagen an, ob diese gilt.

- $[a(a^+ \cup c^* \cup b^*a)]^+ \subseteq L(\mathcal{A})$
- $[c(c^*b^* \cup a^+)]^+ \subseteq L(\mathcal{A})$

Aufgabe 3:

(2 + 4 = 6 Punkte)

(a) Geben Sie eine kontextfreie Grammatik $G = (\Sigma = \{a, b\}, V, S, R)$ an,
die die Sprache $L := \{a^n b^{2m} a^{m-n} \mid m \geq n \text{ und } n, m \in \mathbb{N}_0\}$ erzeugt. Verwenden Sie **maximal vier** Variablen, d.h. $|V| \leq 4$, und **maximal zehn** Regeln, d.h. $|R| \leq 10$.

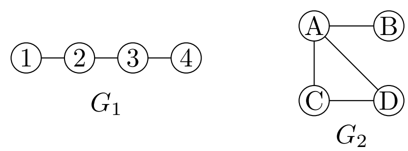
(b) Zeigen Sie, dass die Sprache L aus Teilaufgabe (a) maximal vom Chomsky-Typ 2 ist,
d.h., dass L nicht regulär ist.

Aufgabe 4: (1 + 4 = 5 Punkte)

Seien $G = (V_G, E_G)$ und $H = (V_H, E_H)$ ungerichtete Graphen mit $|V_G| = |V_H|$. Ein bijektiver Graphhomomorphismus von G nach H ist eine bijektive Funktion $f : V_G \to V_H$, so dass gilt:

$$\{u, v\} \in E_G \Rightarrow \{f(u), f(v)\} \in E_H$$

(a) Geben Sie einen bijektiven Graphhomomorphismus f von G_1 nach G_2 an. Geben Sie dazu explizit für jeden Knoten aus G_1 an, auf welchen Knoten aus G_2 er abgebildet wird.



(b) **Problem** BijHom

Gegeben: Ungerichtete Graphen $G = (V_G, E_G)$ und $H = (V_H, E_H)$
Frage: Gibt es einen bijektiven Graphhomomorphismus von G nach H?

Zeigen Sie, dass das Problem BijHom $\mathcal{NP}$-vollständig ist.
Sie dürfen dazu benutzen, dass das Problem Clique $\mathcal{NP}$-vollständig ist:

Problem Clique

Gegeben: Ungerichteter Graph $G = (V, E)$ und ein Parameter $k \leq |V|$
Frage: Gibt es in G eine Clique der Größe mindestens k?

Aufgabe 5: (4 Punkte)

Zeigen Sie, dass das Post'sche Korrespondenzproblem semi-entscheidbar ist.

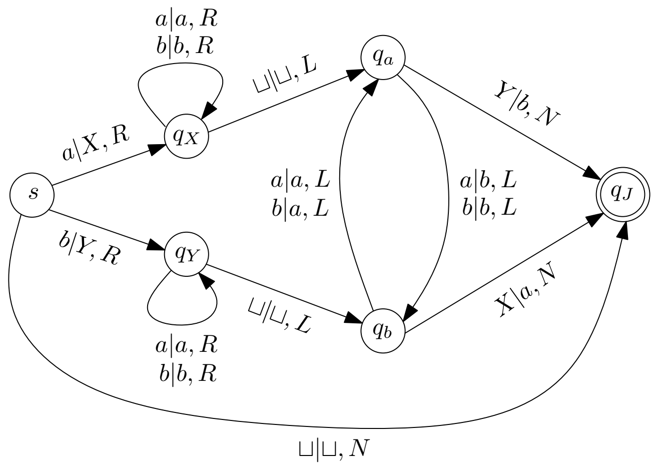
POST'SCHES KORRESPONDENZPROBLEM

Gegeben: Endliche Folge von Wortpaaren $K = ((x_1, y_1), \ldots, (x_n, y_n))$ über einem endlichen Alphabet Σ, mit $x_i \neq \varepsilon$ und $y_i \neq \varepsilon$ für alle $i \in \{1, \ldots n\}$.

Frage: Gibt es eine endliche Folge von Indizes $i_1, \ldots, i_k \in \{1, \ldots, n\}$, so dass $x_{i_1} \ldots x_{i_k} = y_{i_1} \ldots y_{i_k}$ gilt?

Aufgabe 6: (2 + 1 + 1 = 4 Punkte)

Gegeben sei folgende deterministische Turingmaschine $\mathcal{M} = (Q = \{s, q_X, q_Y, q_a, q_b, q_J\}, \Sigma = \{a, b\}, \Gamma = \Sigma \cup \{\sqcup, X, Y, c\}, \delta, s, F = \{q_J\})$. Die Übergangsfunktion δ von $\mathcal{M}$ ist durch untenstehendes Zustandsübergangsdiagramm dargestellt.



Für $\mathcal{M}$ gelten folgende Konventionen: $\mathcal{M}$ hält und akzeptiert sobald ein Endzustand erreicht wird. Alle Übergänge, die aus Endzuständen heraus führen, werden demnach nie benutzt. $\mathcal{M}$ hält und **akzeptiert nicht** für nicht dargestellte Zustandsübergänge.

(a) Dokumentieren Sie die Berechnung der Worte *abab*, *baaa* und *aba* durch die angegebene Turingmaschine $\mathcal{M}$. Geben Sie dazu für jeden Schritt die aktuelle **Konfiguration** an.
Geben Sie außerdem die **Sprache** $L(\mathcal{M}) \subseteq \Sigma^*$ an, die $\mathcal{M}$ akzeptiert.

(b) Geben Sie die Funktion $f_{\mathcal{M}} : L(\mathcal{M}) \longrightarrow \Gamma^*$ an, die $\mathcal{M}$ auf $L(\mathcal{M})$ realisiert.

(c) Erweitern Sie $\mathcal{M}$ so, dass $\mathcal{M}$ alle Worte aus Σ^* akzeptiert und nach Verarbeitung eines Wortes $w \in \Sigma^* \setminus L(\mathcal{M})$ gerade das Wort $w' := c^k$ auf dem Band steht. Dabei sei $k := |w|$ die Anzahl der Zeichen in w. Benutzen Sie für diese Erweiterung **maximal zwei** zusätzliche Zustände. Sie dürfen die Erweiterung in die vorige Abbildung zeichnen.

Aufgabe 7: (1 + 1 + 1 + 1 = 4 Punkte)

Sei $\mathcal{A} = (Q, \Sigma, \Gamma, s, Z, \delta, \{q_1\})$ der Kellerautomat mit Zustandsmenge $Q = \{q_0, q_1, q_2\}$, Eingabealphabet $\Sigma = \{0, 1, 2\}$, Stack-Alphabet $\Gamma = \{X, Z\}$, Anfangszustand q_0, Stack-Initialisierung Z, einzigem Endzustand q_1 und der folgenden Übergangsfunktion δ:

$$\delta(q_0, 0, Z) = \{(q_0, Z), (q_1, Z)\}, \quad \delta(q_1, 1, Z) = \{(q_1, X)\}, \quad \delta(q_1, 1, X) = \{(q_1, XX)\},$$
$$\delta(q_1, 2, X) = \{(q_2, \varepsilon)\}, \quad \delta(q_2, 2, X) = \{(q_2, \varepsilon)\}$$

(a) Ist $\mathcal{A}$ determinninistisch? Begründen Sie Ihre Antwort.

(b) Dokumentieren Sie eine durch Endzustand akzeptierende Berechnung des Wortes 00111. Geben Sie dazu für jeden Schritt die aktuelle Konfiguration an.

(c) Welche Sprache akzeptiert $\mathcal{A}$ durch akzeptierenden Endzustand?

(d) Welche Sprache akzeptiert $\mathcal{A}$ durch leeren Stack?

Aufgabe 8:

(2 + 1 + 3 + 3 = 9 Punkte)

Gegegeben sei eine endliche Menge $\mathcal{U}$ von booleschen Variablen und eine endliche, nicht leere Menge $\mathcal{C}$ von Klauseln über $\mathcal{U}$. Eine Klausel ist ein boolescher Ausdruck der Form $x_1 \vee \cdots \vee x_k$ mit $x_i \in \{u \mid u \in \mathcal{U}\} \cup \{\overline{u} \mid u \in \mathcal{U}\}$. Das Optimierungsproblem MAXSAT besteht darin, eine Wahrheitsbelegung für $\mathcal{U}$ zu finden, so dass möglichst viele Klauseln aus $\mathcal{C}$ den Wert `wahr` annehmen, also erfüllt sind.

Betrachten Sie folgenden Algorithmus:

Algorithmus 1: MAXSAT APPROXIMATION

Eingabe : Endliche Menge $\mathcal{U}$ von booleschen Variablen, nicht leere Menge $\mathcal{C}$ von Klauseln über $\mathcal{U}$
Ausgabe : Wahrheitsbelegung der Variablen aus $\mathcal{U}$
Solange $\mathcal{U} \neq \emptyset$ **wiederhole**
  $u \leftarrow$ wähle ein Element $u \in \mathcal{U}$;
  $\mathcal{C}_w(u) \leftarrow \{c \in \mathcal{C} \mid c$ enthält Literal $u\}$;
  $\mathcal{C}_f(u) \leftarrow \{c \in \mathcal{C} \mid c$ enthält Literal $\overline{u}\}$;
  wenn $|\mathcal{C}_w(u)| \geq |\mathcal{C}_f(u)|$ **dann**
    $u \leftarrow$ `wahr` ;
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}_w(u)$;
  sonst
    $u \leftarrow$ `falsch` ;
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}_f(u)$;
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$;

(a) Gegeben sei die folgende MAXSAT Instanz $I = (\mathcal{U}, \mathcal{C})$ mit Variablenmenge $\mathcal{U}$ und Klauselmenge $\mathcal{C}$:

$$\mathcal{U} = \{u_1, u_2, u_3\}$$

$$\mathcal{C} = \{u_1 \vee u_2 \vee \overline{u_3}, \quad u_1 \vee \overline{u_2}, \quad \overline{u}_1 \vee u_2 \vee \overline{u_3}, \quad \overline{u_2} \vee u_3, \quad u_2 \vee \overline{u_3}, \quad \overline{u_2} \vee \overline{u_3}\}.$$

Geben Sie eine optimale Lösung für I an. Welchen Wert hat jede optimale Lösung?

Geben Sie zusätzlich die Lösung an, die Algorithmus 1 bei Eingabe von I berechnet. Nehmen Sie an, dass in jedem Schritt die Variable $u_i \in \mathcal{U}$ mit dem kleinsten Index i gewählt wird. Welchen Wert hat diese Lösung?

(b) Sei F die Menge der nicht erfüllten Klauseln bezüglich der Ausgabe von Algorithmus 1 und sei F_i die Menge der Klauseln, die im i-ten Schritt der Schleife durch Belegung der Variable u_i den Wert `falsch` annehmen, also unerfüllbar werden. Zeigen Sie, dass gilt:

$$F_i \cap F_j = \emptyset \text{ für } i \neq j,\ i, j \in \{1, \ldots, |\mathcal{U}|\} \quad \text{und} \quad \bigcup_{i=1}^{|\mathcal{U}|} F_i = F.$$

(c) Zeigen Sie, dass Algorithmus 1 1-approximativ ist, d.h. dass Algorithmus 1 eine relative Gütegarantie von $1 + 1 = 2$ hat.

(d) Zeigen Sie: Falls $\mathcal{P} \neq \mathcal{NP}$, gibt es keinen absoluten Approximationsalgorithmus für MAXSAT. Sie dürfen benutzen, dass MAXSAT $\mathcal{NP}$-schwer ist.

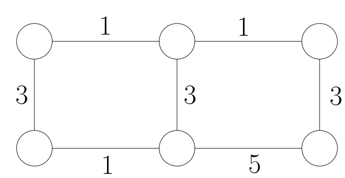
Aufgabe 9: (2 + 3 + 3 = 8 Punkte)

Gegeben sei ein einfacher, ungerichteter, positiv gewichteter Graph $G = (V, E, c)$ mit Gewichtsfunktion $c : E \to \mathbb{N}$. Eine Teilmenge $S \subseteq V$ induziert eine Zerteilung von V in zwei (möglicherweise leere) Mengen S und $V \setminus S$. Das Gewicht $c(S, V \setminus S)$ einer solchen Zerteilung ist die Summe der Gewichte aller Kanten, die zu genau einem Knoten in S inzident sind, d.h. $c(S, V \setminus S) := \sum_{\{u,v\} \in E : u \in S, v \notin S} c(\{u, v\})$.
Das Problem MaxCut fragt nach einer Teilmenge $S \subseteq V$, sodass $c(S, V \setminus S)$ maximal ist.
Betrachten Sie folgenden Algorithmus.

Algorithmus 2: Almost MaxCut

Eingabe : Einfacher, ungerichteter Graph $G = (V, E, c)$ mit Gewichtsfunktion $c : E \to \mathbb{N}$
Ausgabe : Teilmenge $S \subseteq V$
$S \leftarrow \emptyset$;
Solange *Knoten u in S oder $V \setminus S$ existiert, sodass ein Seitenwechsel von u $c(S, V \setminus S)$ vergrößert* **wiederhole**
  Führe Seitenwechsel von u durch, d.h. $S \leftarrow S \cup \{u\}$ falls $u \notin S$ und $S \leftarrow S \setminus \{u\}$ falls $u \in S$;
Gib S aus;

(a) Zeichnen Sie eine möglichst gute Lösung S_1 für den Graphen G_1 in untenstehende Abbildung ein. Markieren Sie die Knoten in S_1 deutlich.



(b) Zeigen Sie, dass die Schleife in Algorithmus 2 terminiert.

(c) Zeigen Sie, dass Algorithmus 2 eine **pseudo**-polynomielle Laufzeit hat, indem Sie zeigen, dass die Laufzeit polynomiell in der Anzahl der Kanten und Knoten in G und in der größten vorkommenden Zahl der Gewichtsfunktion ist.
Sie dürfen annehmen, dass für einen Knoten u in konstanter Zeit gepüft werden kann, ob ein Wechsel das Gewicht erhöht. Ebenso sei der Seitenwechsel eines Knotens in konstanter Zeit durchführbar.

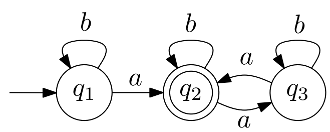
Aufgabe 10: (10 Punkte)

Jeder der folgenden Aussagenblöcke umfasst drei Einzelaussagen. Die sich unterscheidenden Aussagenbausteine sind durch Kästchen gekennzeichnet. Kreuzen Sie genau jene Bausteine an, die in einer wahren Einzelaussage enthalten sind. Jeder Aussagenblock enthält mindestens eine wahre Einzelaussage. Unvollständig oder falsch angekreuzte Aussagenblöcke werden mit null Punkten bewertet, Sie erhalten einen Punkt für jeden Aussagenblock, für den Sie genau die richtige Menge an Aussagenbausteinen angekreuzt haben.

Jede Sprache, die das Pumping-Lemma für reguläre Sprachen erfüllt

- ☐ ist regulär.
- ☐ erfüllt auch das verallgemeinerte Pumping-Lemma für reguläre Sprachen.
- ☐ erfüllt auch das Pumping-Lemma für kontextfreie Sprachen.

Für den folgenden deterministischen endlichen Automaten $\mathcal{A}$ über dem Alphabet $\Sigma = \{a, b\}$ gilt:



- ☐ b ist Zeuge für die Nichtäquivalenz von q_1 und q_3.
- ☐ q_1 und q_3 sind äquivalent.
- ☐ Der Index der Neroderelation der von $\mathcal{A}$ akzeptierten Sprache $L(\mathcal{A})$ ist 3.

Falls L_1 und L_2 semi-entscheidbare Sprachen sind, dann ist

- ☐ $L_1 \cup L_2$ entscheidbar.
- ☐ $L_1 \cup L_2$ semi-entscheidbar.
- ☐ L_1^c semi-entscheidbar.

Falls 2SAT $\mathcal{NP}$-vollständig ist, gilt:

- ☐ 3SAT $\in \mathcal{P}$.
- ☐ 2SAT $\notin \mathcal{P}$.
- ☐ für alle $L \in \mathcal{NP} : L \propto$ 2SAT.

Jede reguläre Sprache L ist

- ☐ kontextfrei.
- ☐ endlich.
- ☐ in $\mathcal{NTAPE}(n)$ enthalten.

Aus $\mathcal{P} \neq \mathcal{NP}$ folgt:

- ☐ 3SAT $\notin \mathcal{P}$.
- ☐ Es gibt keine deterministische 2-Band-Turingmaschine, die 3SAT in Polynomialzeit entscheidet.
- ☐ Es gibt keinen absoluten Approximationsalgorithmus für das Optimierungsproblem Clique.

Falls es ein polynomiales Approximationsschema (PAS) für ein Optimierungsproblem Π gibt, dann

- ☐ ist Π nicht $\mathcal{NP}$-schwer.
- ☐ gibt es einen Approximationsalgorithmus mit relativer Gütegarantie für Π.
- ☐ gibt es keinen Approximationsalgorithmus mit relativer Gütegarantie für Π.

Gegeben sei die Grammatik G mit Nichtterminalmenge $\{S, A, B\}$, Terminalmenge $\{a, b\}$, Startsymbol S und folgender Regelmenge:

$$S \to aBA|a$$
$$A \to aS|a$$
$$B \to bBBS|b$$

Die oben gegebene Grammatik G ist

☐ in Chomsky-Normalform.
☐ in Greibach-Normalform.
☐ kontextfrei.

Eine Sprache ist semi-entscheidbar genau dann, wenn es eine

☐ Grammatik gibt, die sie erzeugt.
☐ deterministische Turingmaschine gibt, die genau die Eingaben w akzeptiert, für die $w \in L$ gilt.
☐ deterministische Turingmaschine gibt, die auf allen Eingaben w hält, für die $w \in L$ gilt.

Aus $\mathcal{NP} \neq$ co-$\mathcal{NP}$ folgt:

☐ $\mathcal{P} \neq \mathcal{NP}$.
☐ TSP liegt nicht in co-$\mathcal{NP}$.
☐ $\mathcal{NPI} \neq \emptyset$.