Berlin, 17.07.2013

Name: .... **Matr.-Nr.: ....**

Klausur TheGI 2 – Automaten und Komplexität

(Niedermeier/Hartung/Nichterlein, Sommersemester 2013)

1	
2	
3	
4	
5	
6	
7	
8	
Σ	

Bearbeitungszeit: 60 min.
max. Punktezahl: 60 Punkte

Allgemeine Hinweise:

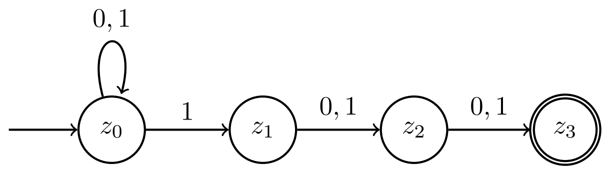
- Es sind keinerlei Hilfsmittel erlaubt.
- Benutzen Sie keinen Bleistift, sondern einen Kugelschreiber oder Füller in der Farbe schwarz oder blau.
- Beschriften Sie jedes Blatt mit Vor- und Nachnamen sowie Matrikelnummer.
- **Falls in der Aufgabenstellung nicht explizit ausgeschlossen, sind alle Antworten zu begründen! Antworten ohne Begründung erhalten 0 Punkte.**

Viel Erfolg!

Name: .... **Matr.-Nr.:** ....

Aufgabe 1: **Endliche Automaten** (1+6 Punkte)

Sei $M = (\{z_0, z_1, z_2, z_3\}, \{0,1\}, \delta, \{z_0\}, \{z_3\})$ mit δ gegeben durch:



(a) Geben Sie einen regulären Ausdruck an, der $T(M)$ beschreibt (ohne Begründung).

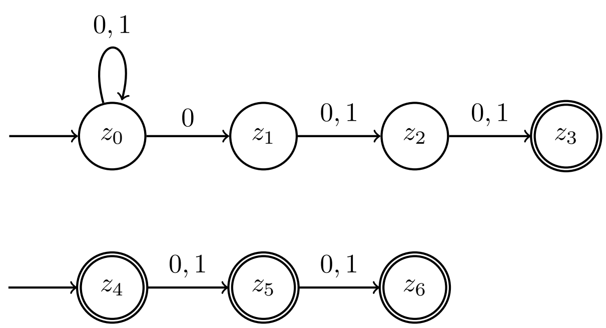
——————Lösung——————

$(0|1)^* \cdot 1 \cdot (0|1) \cdot (0|1)$

(b) Geben Sie einen NFA oder DFA M' mit $T(M') = \{w \in \{0,1\}^* \mid w \notin T(M)\}$ an.

——————Lösung——————

Wir geben einen NFA $M' = (\{z_0, z_1, z_2, z_3, z_4, z_5, z_6\}, \{0,1\}, \delta', \{z_0, z_4\}, \{z_3, z_4, z_5, z_6\})$ mit folgendem δ' an:



Begründung: Das Komplement der Sprache $T(M)$ ist die Sprache, bei der der drittletzte Buchstabe keine 1 ist. Der NFA M' besitzt zwei Startzustände. Für alle Wörter der Sprache mit mehr als 2 Buchstaben existiert eine akzeptierende Berechnung welche in z_0 beginnt. Mit den Zuständen z_4, z_5 und z_6 werden alle Wörter mit weniger als drei Buchstaben akzeptiert.

Matr.-Nr.: ....

Aufgabe 2: **Rechtskongruenz** (6 Punkte)

Sei $L \subseteq \Sigma^* = \{0,1\}^*$ eine Sprache. Angenommen es gilt $(11,1) \in R_L$ und $(0,1) \notin R_L$. Beweisen oder widerlegen Sie folgende Aussage: Es existiert eine natürliche Zahl $n \geq 1$ mit $(1^n, 0) \in R_L$.

Hinweis: Die Rechtskongruenz R_L ist definiert vermöge

$$(x,y) \in R_L \Leftrightarrow (\forall w \in \Sigma^* : xw \in L \Leftrightarrow yw \in L).$$

——————Lösung——————

Die Aussage ist falsch:

Wegen $(11,1) \in R_L$ gilt für jedes Wort $w' = 1^{k-1}w$ mit $w \in \Sigma^*$ und $k \in \mathbb{N}, k \geq 1$:

$$\begin{aligned} & 11w' \in L \Leftrightarrow 1w' \in L \\ \Leftrightarrow \quad & \forall w \in \Sigma^* : 1^{k+1}w \in L \Leftrightarrow 1^k w \in L \end{aligned}$$

Es folgt also $(1^{k+1}, 1^k) \in R_L, k \in \mathbb{N}, k \geq 1$. Durch die Transitivität und Symmetrie der Äquivalenzrelation R_L folgt auch für alle $i, j \geq 1$ gilt $(1^i, 1^j) \in R_L$. Insbesondere gilt für jedes $n \geq 1$: $(1^n, 1) \in R_L$.

Aus $(1^n, 1) \in R_L$ und $(1, 0) \notin R_L$ folgt, dass $(1^n, 0) \notin R_L$ für alle $n \geq 1$. $\square$

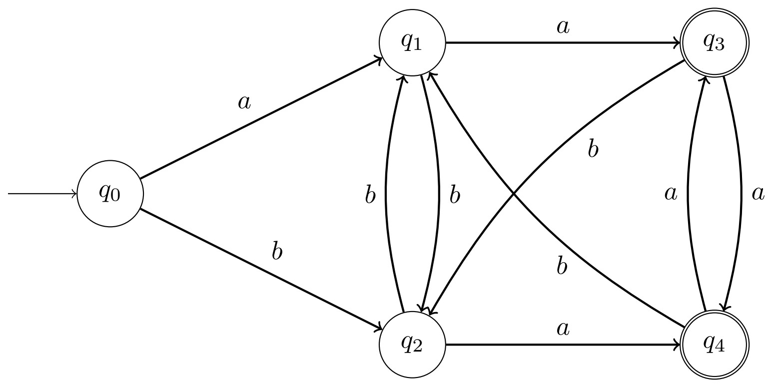
Alternativer Beweis:

Sei $M = (Z, \Sigma, \delta, z_0, E)$ der Minimalautomat für L, d.h. für jede Äquivalenzklasse aus R_L enthält M genau einen Zustand. Sei z der Zustand für $[1]_{R_L}$, d.h. $\hat{\delta}(z_0, 1) = z$. Da $(11,1) \in R_L$ gilt $11 \in [1]_{R_L}$ und somit ist $\delta(z,1) = z$. Für jedes $n \geq 1$ gilt nun $\hat{\delta}(z_0, 1^n) = z$ und somit $1^n \in [1]_{R_L}$, d.h. $(1^n, 1) \in R_L$. Aus $(1^n, 1) \in R_L$ und $(1,0) \notin R_L$ folgt, dass $(1^n, 0) \notin R_L$ für alle $n \geq 1$. $\square$

Matr.-Nr.: ....

Aufgabe 3: **Minimierung endlicher Automaten** (6 Punkte)

Gegeben sei ein DFA $M = (Z, \Sigma, \delta, q_0, E)$, wobei $Z = \{q_0, q_1, q_2, q_3, q_4\}$, $\Sigma = \{a, b\}$ und $E = \{q_3, q_4\}$. Die Überführungsfunktion δ sei wie folgt gegeben:



Geben Sie einen minimalen DFA M' mit $T(M') = T(M)$ an.

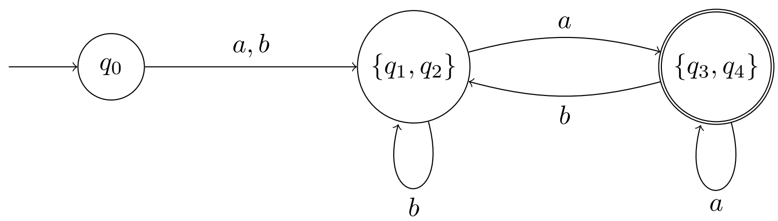
Sie können für die Minimierung von M die folgende Tabelle benutzen.

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1					
q_2					
q_3					
q_4					

————————Lösung————————

	q_0	q_1	q_2	q_3	q_4
q_0					
q_1	$\star$				
q_2	$\star$				
q_3	$\star$	$\star$	$\star$		
q_4	$\star$	$\star$	$\star$		

Ein minimaler DFA ist entsprechend der obigen Tabelle ist $M' = (Z', \Sigma, \delta', q_0, E)$ wobei $Z' = \{q_0, \{q_1, q_2\}, \{q_3, q_4\}\}$, $\Sigma = \{a, b\}$ und $E' = \{\{q_3, q_4\}\}$. Die Überführungsfunktion δ' ist wie folgt definiert:


q_0
a, b
$\{q_1, q_2\}$
a
b
$\{q_3, q_4\}$
b
a

Matr.-Nr.: ....

Aufgabe 4: **Reguläre Sprachen** (9 Punkte)

Sei $L \subseteq \Sigma^*$. Für ein Symbol $a \in \Sigma$ definieren wir

$$L/a := \{w \in \Sigma^* \mid wa \in L\}.$$

Beweisen oder widerlegen Sie:

(a) Wenn L/a regulär ist, so ist auch L regulär.

———————Lösung———————

(2 Punkte) Falsch: $L = \{a^n b^n \mid n \geq 1\}$ ist bekanntermaßen nicht regulär. Allerdings ist $L/a = \emptyset$ regulär.

(b) Wenn L regulär ist, so ist auch L/a regulär.

———————Lösung———————

(7 Punkte) Wahr. Beweis via Satz von Myhill-Nerode. Seien $x, y \in \Sigma^*$ sodass $(x, y) \in R_L$. Wir zeigen $(x, y) \in R_{L/a}$: Sei $z \in \Sigma^*$ beliebig. Es gilt:

$$\begin{aligned} xz \in L/a &\iff xza \in L && \text{Definition von } L/a \\ &\iff yza \in L && \text{da } (x,y) \in R_L \\ &\iff yz \in L/a && \text{Definition von } L/a. \end{aligned}$$

Somit ist $R_L \subseteq R_{L/a}$. Aus der Regularität von L folgt damit, dass der Index von $R_{L/a}$ endlich ist und somit L/a regulär ist.

Alternativer Beweis über Automatenmodifikation:

Sei $M = (Z, \Sigma, \delta, z_0, E)$ ein DFA mit $T(M) = L$ und sei $Z' = \{z \in Z \mid \delta(z, a) \in E\}$ die Menge der Zustände aus denen mit Lesen eines a's in einen Endzustand übergegangen wird. Wir zeigen nun, dass $M' = (Z, \Sigma, \delta, z_0, Z')$ die Sprache L/a akzeptiert, d.h. $T(M') = L/a$.

„$\Rightarrow$“: Sei $w \in T(M')$, d.h. $\hat{\delta}(z_0, w) = z$ mit $z \in Z'$. Somit ist $\hat{\delta}(z_0, wa) = \delta(\hat{\delta}(z_0, w), a) = \delta(z, a) \in E$ und daher $wa \in L$. Nach Definition von L/a gilt nun $w \in L/a$.

„$\Leftarrow$“: Sei $w \in L/a$. Also ist $wa \in L = T(M)$. Somit gilt $\hat{\delta}(z_0, wa) \in E$. Sei $z = \hat{\delta}(z_0, w)$ der vorletzte besuchte Zustand. Es gilt $\delta(z, a) \in E$ und somit $z \in Z'$. Daher ist $\hat{\delta}(z_0, w) \in Z'$ und somit $w \in T(M')$. $\square$

Matr.-Nr.: ....

Aufgabe 5: **Büchi-Automaten** (4 Punkte)

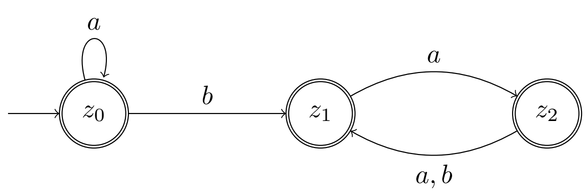
Geben Sie für die folgende Sprache einen nichtdeterministischen Büchi-Automaten an (ohne Begründung).

$$L_a = \{x \in \{a,b\}^\omega \mid \text{in } x \text{ kommen zwischen je zwei } b\text{'s ungerade viele } a\text{'s vor.}\}$$

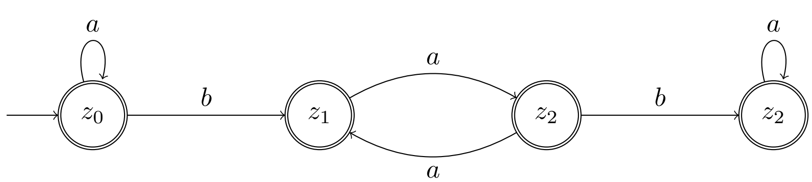
——————Lösung——————

$$M = (\{z_0, z_1, z_2\}, \{a, b\}, \delta, \{z_0\}, \{z_0, z_1, z_2\})$$

δ :



Obiger Automat ist für den Fall zwischen je zwei *aufeinanderfolgenden* b's ungerade viele a's vorkommen. Die Sprache L_a kann natürlich auch so interpretiert werden, das zwischen allen Paaren von b's ungerade viele a's vorkommen. In diesem Fall kann es max. 2 b's geben. Der Automat sieht in dem Fall wie folgt aus:



Hinweis: Büchi Automaten sind im SoSe 2014 nicht prüfungsrelevant.

Matr.-Nr.: ....

Aufgabe 6: **Grammatiken** (1+2+6 Punkte)

Sei $G = (\{S, A, B, C\}, \{a, b, c\}, P, S)$ eine Grammatik mit

$$\begin{aligned} P = \{ & S \to AC, \\ & A \to aABC \mid aBC, \\ & CB \to BC, \\ & aBB \to ab,\ bBB \to bb \\ & bC \to bcc,\ cC \to ccc\}. \end{aligned}$$

(a) Geben Sie den größtmöglichen Typ der Grammatik G bezüglich der Chomsky-Hierarchie an.
Hinweis: Der größtmögliche Typ einer Grammatik ist i, wenn sie vom Typ i ist aber nicht vom Typ $i + 1$.

(b) Geben Sie die von der Grammatik G erzeugte Sprache $L(G)$ an (ohne Begründung).

(c) Zeigen Sie, dass $L(G)$ nicht kontextfrei ist, indem Sie das Pumping-Lemma für kontextfreie Sprachen benutzen.

Hinweis: Pumping-Lemma für kontextfreie Sprachen:

Ist eine Sprache L kontextfrei, so existiert eine natürliche Zahl n derart, dass sich alle $z \in L$ der Länge $|z| \geq n$ so in der Form $z = uvwxy$ für gewisse $u, v, w, x, y \in \Sigma^*$ zerlegen lassen, dass gilt:

(1) $|vx| \geq 1$,

(2) $|vwx| \leq n$ und

(3) für alle $i \geq 0$ ist $uv^iwx^iy \in L$.

——————Lösung——————

(a) Die Grammatik G ist vom Typ 0, da es die verkürzende Regel $aBB \to ab$ gibt.

(b) $L(G) = \{a^{2m}b^mc^{4m+2} \mid m \geq 1\}$.

(c) Angenomen, $L(G)$ wäre kontextfrei, so existierte ein n mit den Eigenschaften des Pumping-Lemmas für kontextfreie Sprachen. Wähle $z = a^{2n}b^nc^{4n+2}$. Wegen $|z| = 7n + 2 > n$ müsste es eine Zerlegung $z = uvwxy$ mit $|vwx| \leq n$ und $|vx| \geq 1$ geben. Beachte, dass weder v noch x zwei oder drei verschiedene Buchstaben enthält, weil sonst in v^i oder x^i eine Wiederholung von Teilwörtern entsteht und somit das so gebildete Wort uv^iwx^iy nicht in der Sprache liegt. Wenn v und x ausschließlich aus einer Art von Buchstabe bestehen, dann liegt das Wort uwy auch nicht in der Sprache, denn mindestens ein Buchstabe von einer Art fehlt und mindestens eine Art Buchstabe bleibt komplett erhalten im Vergleich zu $uvwxy$. Somit ist die Bedingung $\#_a(uwy) = 2 \cdot \#_b(uwy)$ oder $\#_c(uwy) = 4 \cdot \#_b(uwy) + 2$ verletzt.

Aufgabe 7: **Polynomzeitreduktionen** (2+8 Punkte)

Das DOMINATING SET-Problem ist bereits aus der Vorlesung bekannt.

DOMINATING SET
Eingabe: Ein ungerichteter Graph $G = (V, E)$ und ein $k \in \mathbb{N}$.
Frage: Besitzt G eine dominierende Knotenmenge V' der Größe höchstens k, d. h.
$\exists V' \subseteq V, |V'| \leq k : \forall u \in V : u \in V' \vee$ (u hat mindestens einen Nachbarn in V')?

Ein Graph $G = (V, E)$ wird *bipartit* genannt, falls es eine Zerlegung von V in V_1 und V_2 mit $V_1 \cap V_2 = \emptyset$ gibt, sodass jede Kante einen Endknoten in V_1 und einen in V_2 hat. Entsprechend ist das BIPARTITE DOMINATING SET-Problem wie folgt definiert:

BIPARTITE DOMINATING SET
Eingabe: Ein bipartiter Graph $G = (V_1 \cup V_2, E)$ und ein $k \in \mathbb{N}$.
Frage: Besitzt G eine dominierende Knotenmenge der Größe höchstens k?

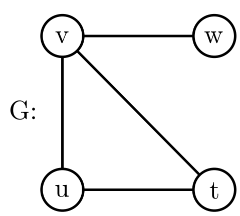
Aus der Vorlesung ist bekannt, dass DOMINATING SET NP-vollständig ist. Betrachten Sie folgende Reduktion von DOMINATING SET auf BIPARTITE DOMINATING SET.

Reduktion: Sei der Graph $G = (V, E)$ und $k \in \mathbb{N}$ eine Eingabe für DOMINATING SET. Wir konstruieren einen Graph $G' = (V_1 \cup V_2, E')$ in 3 Schritten:

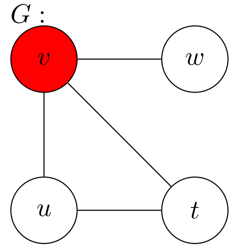
1. Füge in V_1 und V_2 jeweils eine Kopie von V ein. Bezeichne mit v^1 bzw. v^2 die Kopie des Knotens $v \in V$ in V_1 bzw. V_2.
2. Für jede Kante $\{u, v\} \in E$ füge die Kanten $\{u^1, v^2\}$ und $\{v^1, u^2\}$ ein. Füge zusätzlich die Kante $\{u^1, u^2\}$ für alle Knoten $u \in V$ ein.
3. Füge einen Knoten z_1 zu V_1 hinzu und auch einen Knoten z_2 zu V_2. Füge die Kante $\{z_1, z_2\}$ hinzu und auch die Kante $\{u^1, z_2\}$ für alle $u \in V$.

Wir definieren nun die Polynomzeitreduktion f durch $f(G, k) = (G', k + 1)$.

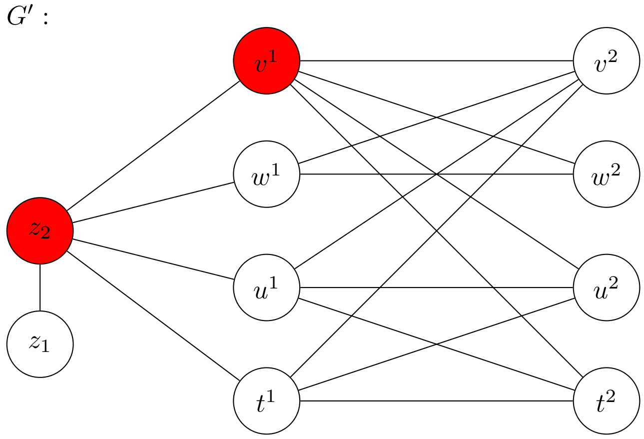
(a) Geben Sie denjenigen Graph G' an, welcher durch obige Reduktion für den wie folgt gegebenen Graphen G konstruiert wird. Geben Sie in beiden Graphen eine dominierende Menge minimaler Größe an (ohne Begründung, einzeichnen genügt).



Lösung



G' :



(b) Überprüfen Sie die obige Reduktion f auf Korrektheit und korrigieren Sie diese gegebenenfalls. Beweisen Sie anschließend die Korrektheit der (eventuell korrigierten) Reduktion, d. h. zeigen Sie

$$\forall(G,k) : (G,k) \in \text{Dominating Set} \Leftrightarrow f(G,k) \in \text{Bipartite Dominating Set}.$$

Beweisen Sie darauf aufbauend, dass Bipartite Dominating Set NP-vollständig ist.

——————Lösung——————

Die Reduktion f ist korrekt.

Beweis der Korrektheit:

„$\Rightarrow$“: Sei $(G,k) \in$ Dominating Set, dann existiert eine dominierende Knotenmenge V' mit $|V'| \leq k$. Wir zeigen nun, dass $V_1' := \{v^1 \in V_1 \mid v \in V'\} \cup \{z_2\}$ eine dominierende Knotenmenge in G' ist. Durch Konstruktionsschritt 3 dominiert z_2 den Knoten z_1 sowie alle Knoten aus V_1. Für jeden Knoten $v \in V \setminus V'$ existiert ein Knoten $u \in V'$ sodass $\{u,v\} \in E$. Per Konstruktionsschritt 2 existiert eine Kante $\{u^1, v^2\} \in E'$ und somit ist u^2 dominiert. Weiterhin ist $\{u^1, u^2\} \in E'$, d.h. alle Kopien dominierender Knoten in V_2 werden durch V_1' dominiert. Somit dominiert V_1' alle Knoten in G'.

„$\Leftarrow$“: Sei $(G', k+1) \in$ Bipartite Dominating Set, dann existiert eine dominierende Menge V'' mit $|V''| \leq k+1$. Falls $z_1 \in V''$, ersetze ihn durch z_2. Falls $z_1 \notin V''$, so muss z_2 bereits in V'' sein, da z_1 sonst nicht dominiert würde. Damit werden (durch z_2) alle Knoten in $V_1 \cup \{z_1, z_2\}$ dominiert. Wählen wir nun die zugehörige Knotenmenge V' in G (also $V' = \{v \in V \mid v^1 \in V'' \vee v^2 \in V''\}$), so ist diese per Konstruktion nun auch für G eine dominierende Knotenmenge mit höchstens k Knoten: Sei $u \in V \setminus V'$ ein beliebiger Knoten. Da u^2 durch V'' dominiert wird und nach Konstruktion von V' gilt $u^2 \notin V''$, gibt es einen Knoten $v^1 \in V''$ mit $\{v^1, u^2\} \in E'$. Es gilt nun $v \in V'$ und nach Konstruktionsschritt 2 gilt $\{u,v\} \in E$, somit ist u in G durch v dominiert. $\square$

Beweis der NP-Vollständigkeit: Die obige Reduktion kann in $O(|V|^2)$ Schritten und somit in polynomieller Zeit ausgeführt werden. Aus der bekannten NP-Schwere von Dominating Set folgt mit der eben gezeigten polynomiellen Reduktion, dass Bipartite Dominating Set NP-schwer ist. Als Spezialfall von Dominating Set liegt auch Bipartite Dominating Set in NP, ist insgesamt also NP-vollständig.

Matr.-Nr.: ....

Aufgabe 8: **Vermischtes** (9 Punkte)

Beantworten Sie die folgenden Fragen bzw. bewerten Sie die Aussagen bzgl. ihres Wahrheitsgehaltes. Begründen Sie ihre Antworten jeweils kurz in 2-3 Sätzen.

a) Wäre das „P vs. NP"-Problem gelöst, wenn man zeigt, dass es eine kontextfreie Sprache L gibt, sodass $L \leq_m^p L'$ für alle $L' \in \mathrm{NP} \setminus \{\emptyset, \Sigma^*\}$?

b) Unter der Annahme P = NP gilt:

- Jede NP-schwere Sprache kann in Polynomzeit auf einer deterministischen Turingmaschine entschieden werden.
- Jede Sprache in P kann polynomiell auf Sat reduziert werden.
- Für jede endliche nichtleere Sprache A gilt, dass $L \leq_m^p A$ für alle $L \in \mathrm{NP}$.

c) Da Typ-2 Sprachen nicht unter Schnitt abgeschlossen sind, kann es keine zwei Typ-2 Sprachen A, B geben, sodass auch $A \cap B$ eine Typ-2 Sprache ist.

d) Es gibt unendlich viele Typ-1 Sprachen, die durch reguläre Ausdrücke beschrieben werden können.

——————Lösung——————

(a) Nein, denn mit dem CYK-Algorithmus existiert ein Algorithmus mit polynomieller Laufzeit für das Wortproblem kontextfreier Sprachen. Also lassen sich alle kontextfreien Sprachen in Polynomzeit auf alle nichttrivialen Sprachen in NP reduzieren. Dies gilt unabhängig von der Frage P vs. NP.

(b)
- Nein, da es auch NP-schwere Sprachen gibt, die nicht in NP liegen.
- Ja, dies gilt, und auch unabhängig davon, ob P = NP oder nicht, da $\mathrm{P} \subseteq \mathrm{NP}$ und Sat NP-schwer ist. Laut Definition von NP-Schwere lässt sich jede Sprache in NP polynomiell auf Sat reduzieren.
- Ja, dies gilt. Unter der Annahme P = NP liegen alle Sprachen $L \in \mathrm{NP}$ bereits in P, also existiert ein Polynomzeitalgorithmus auf einer deterministischen Turingmaschine, der L entscheidet. Ein Algorithmus zur Polynomzeitreduktion von L auf A könnte mittels dieses Algorithmus zunächst entscheiden, ob die Eingabe in der Sprache L liegt, und anschließend ein Wort $w \in A$ ausgeben, falls die Eingabe in L liegt, ansonsten ein Wort $w' \notin A$. Da A endlich und nicht leer ist, existieren solche zwei festen Wörter finden.

(c) Die Aussage ist nicht korrekt. Ist zum Beispiel die Sprache A die leere Menge, so ist unabhängig von der Sprache B der Schnitt wieder leer und somit auch kontextfrei.

(d) Jede reguläre Sprache kann durch einen regulären Ausdruck beschrieben werden. Da alle regulären Sprachen auch Typ-1 Sprachen sind und es offensichtlich unendlich viele reguläre Sprachen gibt (z.B. $\{a\}, \{aa\}, \{aaa\}, \ldots$), ist die Aussage korrekt.