1. Klausur zur Vorlesung
Theoretische Grundlagen der Informatik
Wintersemester 2016/2017

Lösung!

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.
- Es sind keine Hilfsmittel zugelassen.

Aufgabe	Mögliche Punkte						Erreichte Punkte					
	a	b	c	d	e	Σ	a	b	c	d	e	Σ
1	2	2	2	–	–	6				–	–	
2	1	6	2	~~3~~	–	9					–	
3	2	4	2	–	–	8				–	–	
4	2	4	2	3	–	11					–	
5	1	3	2	3	–	9					–	
6	4	–	–	–	–	4		–	–	–	–	
7	10×1					10						
Σ						57						

Problem 1: Automaten $2 + 2 + 2 = 6$

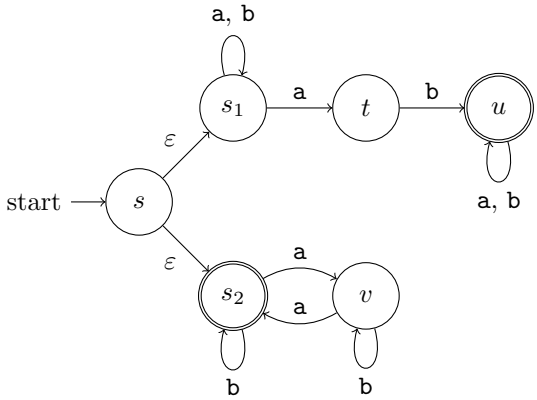
(a) Sei die Sprache

$$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^\star \mid w \text{ enthält } \mathtt{ab} \text{ als Teilwort} \vee |w|_{\mathtt{a}} \text{ ist gerade}\}$$

gegeben. Hierbei bezeichnet $|w|_a$ die Anzahl der Vorkommen von $\mathtt{a}$ in w. Geben Sie den Übergangsgraphen eines nichtdeterministischen endlichen Automaten, der L erkennt, an. Verwenden Sie höchstens sieben Zustände.

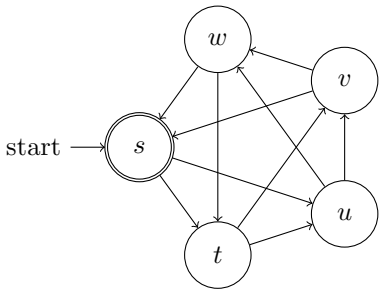
Hinweis: Einen Fehlerzustand dürfen Sie als implizit gegeben annehmen. Es wird angenommen, dass alle nicht explizit angegebenen Übergänge in diesen Fehlerzustand führen. Der Fehlerzustand zählt nicht zu den sieben Zuständen, die Sie verwenden dürfen.

Lösung:



Diskussion: Hier sollte man an alle Details denken, z.B. den Startzustand zu markieren, das Akzeptanzverhalten des leeren Worts richtig zu modellieren, ...

(b) Geben Sie die Sprache $L(\mathcal{A})$ über dem Alphabet $\{\mathtt{a}\}$ an, die der folgende nichtdeterministische endliche Automat $\mathcal{A}$ erkennt. Alle Übergänge sind für das Zeichen $\mathtt{a}$ angegeben.

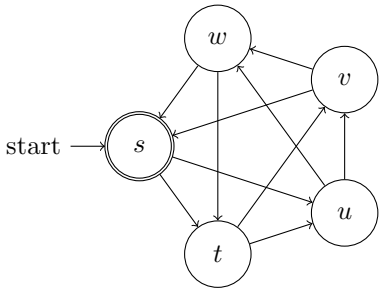


Lösung:

$$L(\mathcal{A}) = \{w \in \{\mathtt{a}\}^* \mid w = \varepsilon \vee |w| \geq 3\}$$

Diskussion: Auch hier nicht das leere Wort vergessen!

(c) Geben Sie einen zu dem Automaten $\mathcal{A}$ aus Teilaufgabe (b) äquivalenten vollständigen deterministischen endlichen Automaten tabellarisch an. Verwenden Sie hierfür die Potenzmengenkonstruktion, und geben Sie deren Schritte explizit an.



Zur Erinnerung: der Automat $\mathcal{A}$ aus Teilaufgabe (b).

Lösung:

Zustand	Übergang
$\underline{\{s\}}$	$\{t, u\}$
$\underline{\{t, u\}}$	$\{u, v, w\}$
$\{u, v, w\}$	$\{s, t, v, w\}$
$\underline{\{s, t, v, w\}}$	$\{s, t, u, v, w\}$
$\underline{\{s, t, u, v, w\}}$	$\{s, t, u, v, w\}$

Problem 2: Berechenbarkeit $1 + 6 + 2 + 3 = 12$

Eine Instanz I des Postschen Korrespondenzproblem (PKP) ist eine endliche Menge $I \subset \Sigma^\star \times \Sigma^\star$, wobei Σ ein endliches Alphabet ist. Eine Folge von Tupeln $t_1, t_2, \ldots, t_k$ mit $k \in \{1, 2, \ldots\}$ und $t_i = (x_i, y_i) \in I$ ist eine Lösung von I genau dann, wenn $x_1 \cdot x_2 \cdot \ldots \cdot x_k = y_1 \cdot y_2 \cdot \ldots \cdot y_k$ gilt.

(a) Geben Sie seine Lösung der folgenden PKP-Instanz an:

$$\left\{ \binom{a}{rant}, \binom{am}{m}, \binom{at}{tata}, \binom{par}{pa}, \binom{ntat}{a} \right\}$$

Lösung:

$$\binom{par}{pa}, \binom{a}{rant}, \binom{ntat}{a}, \binom{at}{tata}, \binom{am}{m}$$

Diskussion: Eine Lösung ist eine Folge von Tupeln. Stattdessen wurde hier gerne einfach das Wort *parantatatam* angegeben.

Sei $\mathcal{G}$ die Menge aller kontextfreien Grammatiken über einem festen endlichen Alphabet Σ, und sei $\langle G \rangle$ eine geeignete Kodierung einer kontextfreien Grammatik $G \in \mathcal{G}$. Dann ist die Sprache L definiert als

$$L = \left\{ \langle G \rangle \mid G \in \mathcal{G},\ \exists w \in L(G) : w = w^R \right\},$$

wobei w^R das Spiegelwort eines Wortes $w \in \Sigma^\star$ bezeichnet.

(b) Zeigen Sie, dass L nicht entscheidbar ist.

Hinweis: Konstuieren Sie als Teil Ihres Beweises zu einer PKP-Instanz I eine kontextfreie Grammatik G, so dass $\langle G \rangle$ genau dann in L enthalten ist, wenn I eine Lösung besitzt.

Lösung: Sei $I = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ eine PKP Instanz mit dem zugrunde liegenden Alphabet Σ. Wir konstruieren aus I folgende kontextfreie Grammatik $G_I = (\Sigma \cup \{\#\}, \{S\}, R\}$ mit $\# \notin \Sigma$ und

$$R = \{S \to x_i S y_i^R \mid x_i \# y_i^R \big| i = 1, 2, \ldots, n\}$$

.

Es bleibt zu zeigen, dass I genau dann lösbar ist, wenn $L(G_I)$ ein Palindrom enthält.

Zu einer Lösung $i(1), i(2), \ldots i(k) \in \{1, 2, \ldots n\}$ der Instanz I, wird durch Anwendung der Regeln in entsprechender Reihenfolge offenbar ein Palindrom erzeugt.

Sei w ein Palindrom in $L(G_I)$, dann stellt das Zeichen # sicher, dass das Palindrom nicht durch die Konkatenation zweier unterschiedlich langen Teilworte zustande kommt (sonst würde z.B. behauptet, dass die Instanz $\left\{ \binom{0}{00} \right\}$ eine Lösung hätte). Somit ist für $w = x\#y^R$, $x = y$ und die Ableitungsregeln liefern (z.B. über den CYK-Algorithmus) die Lösungssequenz $i(\cdot)$.

Angenommen, L ist entscheidbar. Dann existiert eine Turing Maschine $\mathcal{M}$ die L entscheidet. Wir konstruieren eine Turing Maschine $\mathcal{M}'$, die PKP entscheidet. Die Turing Maschine $\mathcal{M}'$ berechnet erst $\langle G_i \rangle$ und simuliert anschließend $\mathcal{M}$ auf dieser Eingabe. Die Turing Maschine $\mathcal{M}'$ akzeptiert eine Eingabe, wenn $\mathcal{M}$ akzeptiert. Andernfalls ist $\mathcal{M}'$ nicht akzeptierend. Da $\mathcal{M}$ nach endlich vielen Operationen hält, kann mit $\mathcal{M}'$ PKP entschieden werden. Dies ist ein Widerspruch zur Nichtentscheibarkeit von PKP.

Diskussion: Die Sprache L besteht aus Kodierungen kontextfreier Grammatiken, ist selber aber nicht kontextfrei!

(c) Zeigen Sie, dass L semi-entscheidbar ist.

Lösung: Es können alle Palindrome w ihrer Länge nach rekursiv aufgezählt werden. Das Wortproblem für kontextfreie Sprachen ist entscheidbar: $w \in L(G)$ kann beispielsweise mit dem CYK-Algorithmus entschieden werden.

(d) Beweisen Sie mit Hilfe der Aussagen von Teilaufgabe (b) und (c), dass die Menge der kontextfreien Sprachen unter Komplementbildung nicht abgeschlossen ist.

Lösung: Die Aufgabe wurde nicht gewertet.

Diskussion: Kontextfreie Sprachen sind unter Komplementbildung nicht abgeschlossen. Dies folgt allerdings nicht direkt aus Teilaufgabe (b) und (c).

Problem 3: $\mathcal{NP}$-Vollständigkeit $2 + 4 + 2 = 8$

Sei für $k \in \{1, 2, \ldots\}$ folgende Familie von Problemen gegeben.

Problem $\text{SAT}_{\geq k}$

Gegeben: Menge U von n Variablen, Menge C von m Klauseln über U, Parameter k.
Frage: Existieren mindestens k unterschiedliche wahrheitserfüllende Belegungen von C?

(a) Geben Sie für ein beliebiges aber festes k eine polynomiale Transformation $\text{SAT} \propto \text{SAT}_k$ an.

Lösung: Sei $I = (C, U)$ eine SAT Instanz. Wir erweitern U um $k' = \lceil \log_2 k \rceil + 1$ Variablen $U' = U \dot\cup \{x_1, x_2, \ldots, x_{k'}\}, x_i \notin U$.

Diskussion: C kann um eine neue Klausel $C' = C \cup \{x_1, x_2, \ldots, x_{k'}\}$ ergänzt werden.

Häufige Fehler:

- Die Klauseln in C wurden k mal kopiert. Dies führt nicht automatisch zu mehr erfüllenden Belegungen.
- Das Problem wurde auf SAT zurückgeführt indem $k = 1$ gesetzt wurde. In der Aufgabenstellung ist k allerdings eine feste Konstante!

(b) Zeigen Sie mit Ihrer polynomialen Transformation $\propto$ aus Teilaufgabe (a), dass das Problem $\text{SAT}_{\geq k}$ für ein beliebiges aber festes k $\mathcal{NP}$-vollständig ist.

Lösung:

$\text{SAT}_{\geq k} \in \mathcal{NP}$: Für eine erfüllende Belegung B einer SAT Instanz I lässt sich in Zeit $O(|C| \cdot |U|)$ die Korrektheit von B verifiziert werden.

Da k konstant ist, lässt sich eine erfüllende Belegung B' einer $\text{SAT}_{\geq k}$-Instanz I' in der gleichen Zeit verifizieren.

Ob die erfüllenden Belegungen B' verschieden sind lässt sich in $O(k^2|U|)$ Zeit verifizieren.

$\propto$ ist polynomial: Es werden nur eine konstante Menge an Variablen und Klauseln zu der SAT Instanz hinzugefügt. Die Transformation ist somit polynomial (linear) in der Größe der SAT Instanz.

$\Rightarrow$ Sei I eine erfüllbar SAT Instanz. Dann ist auch die transformierte Instanz I' erfüllbar. Die neuen Variablen $x_1, \ldots, x_{k'}$ kommen in keiner Klausur vor und können beliebig belegt werden. Die Instanz hat damit I' mindestens $2^{k'+1} \geq k$ erfüllende Belegungen. Damit ist auch die $\text{SAT}_{\geq k}$ Instanz I' erfüllt.

$\Leftarrow$ Sei I' eine Instanz mit mindestens k erfüllenden Belegungen. Die neuen Variablen haben keinen Einfluss auf die Klausen C. Somit ist auch I erfüllbar.

(c) Für $k \in \{0, 1, 2, \ldots\}$ fragt die Familie von Problemen $\text{SAT}_{=k}$ nach *genau* k erfüllenden Belegungen. Geben Sie von den aus Vorlesung und Übung bekannten Komplexitätsklassen möglichst scharf diejenige an, in der das Problem $\text{SAT}_{=0}$ liegt. Definieren Sie diese Komplexitätsklasse außerdem formal.

Lösung: Es wird nach einer Instanz gefragt, die keine erfüllende Belegung hat. Dies ist das Problem co-SAT. Somit ist das Problem in co-$\mathcal{NP}$.

Die Klasse co-$\mathcal{NP}$ ist die Klasse aller Sprachen $\Sigma^\star \setminus L$ für $L \subset \Sigma^\star$ und $L \in \mathcal{NP}$.

Diskussion: co-SAT ist sogar co-$\mathcal{NP}$-vollständig.

Problem 4: Approximationsalgorithmen $2 + 4 + 2 + 3 = 11$

Der ebenso selbstbewusste wie fürsorgliche Superbösewicht Doktor Meta ist erzürnt. Die mediale Berichterstattung über ihn ist schrecklich unehrlich! Um die Bewohner des Metaversiums vor solchen Nachrichten zu beschützen, möchte Doktor Meta alle Straßen mit Störsendern ausstatten.

Eine gerade Straße besteht aus insgesamt $n \in \{1, 2, \ldots\}$ Zellen. In manchen Zellen befindet sich ein Haus, in anderen nicht. Die Störsender können ausschließlich auf Häusern installiert werden. Ein Störsender hat eine Reichweite von $k \in \{0, 1, 2, \ldots\}$. Ein Haus mit Position i liegt genau dann in Reichweite eines Störsenders mit Position j, wenn $|i - j| \leq k$ gilt. Jedes Haus soll innerhalb der Reichweite mindestens eines Störsenders liegen. Insgesamt sollen möglichst wenige Störsender installiert werden.

Dazu wird folgender Algorithmus $\mathcal{A}$ verwendet. Speichere die Position i des Hauses mit der kleinsten Position, das noch nicht abgedeckt ist. Finde dann das Haus mit der größten Position j im Intervall $\{i, i+1, \ldots, i+k\}$ und platziere dort den nächsten Störsender. Wiederhole diesen Prozess solange, bis alle Häuser innerhalb der Reichweite eines Störsenders liegen.

(a) Führen Sie diesen Algorithmus für $k = 2$ auf folgendem Beispiel einer geraden Straße aus, indem Sie genau die Positionen markieren, an denen Störsender installiert werden.

Hinweis: Markieren Sie eine Position, indem Sie das Kästchen über dem entsprechenden Haus ankreuzen.

1	2	3	4	5	6	7	8	9	10	11

Lösung:

			X							X
1	2	3	4	5	6	7	8	9	10	11

(b) Zeigen Sie, dass der Algorithmus für allgemeine k immer eine optimale Lösung liefert.

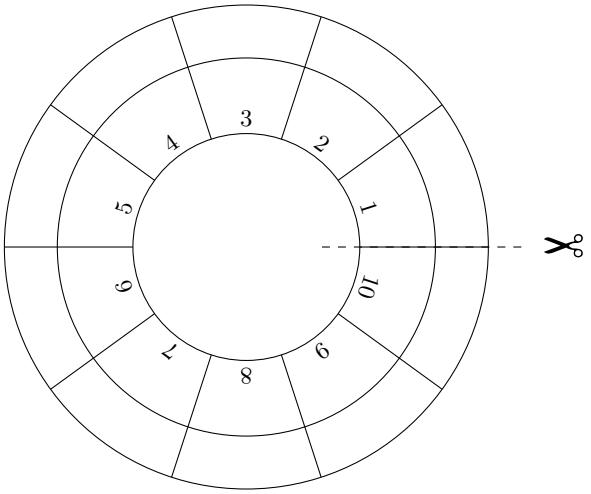
Hinweis: Vergleichen Sie die Lösung, die der Algorithmus $\mathcal{A}$ berechnet, mit einer optimalen Lösung.

Lösung: Sei $(a_1, a_2, \ldots, a_t)$ die von $\mathcal{A}$ berechnete aufsteigend sortierte Folge der Positionen der Störsender. Sei ferner $(o_1, o_2, \ldots, o_{t'})$ eine beliebige optimale Lösung. Aufgrund der Optimalität gilt $t' \leq t$. Zeige nun, dass außerdem $t \leq t'$ gilt, woraus dann folgt, dass $\mathcal{A}$ auch eine optimale Lösung liefert. Das Haus mit der kleinsten Position muss innerhalb der Reichweite von o_1 liegen. Da $\mathcal{A}$ den ersten Radiosender an der größtmöglichen Position platziert, an der das erste Haus noch abgedeckt wird, gilt $o_1 \geq a_1$. Betrachte nun die Bereiche der beiden Lösungen, die nicht vom ersten Störsender abgedeckt werden, und wiederhole dieselbe Argumentation. Dadurch ergibt sich, dass $\mathcal{A}$ nicht mehr Störsender platzieren wird, als eine optimale Lösung, also $t \leq t'$.

Diskussion: Es ist bei diesem Beweis essentiell, die Lösung von $\mathcal{A}$ mit einer optimalen Lösung zu vergleichen. Stattdessen wurde sehr häufig versucht, direkt zu argumentieren, dass $\mathcal{A}$ optimal ist, weil „$\mathcal{A}$ ja nie etwas falsches tut“. Diese Argumentation funktioniert aber bei jedem Greedy-Algorithmus, auch bei solchen, die keine optimalen Lösungen liefern. Herauskamen deshalb oft keine Beweise, sondern nur wiederholte Beschreibungen des Vorgehens von $\mathcal{A}$.

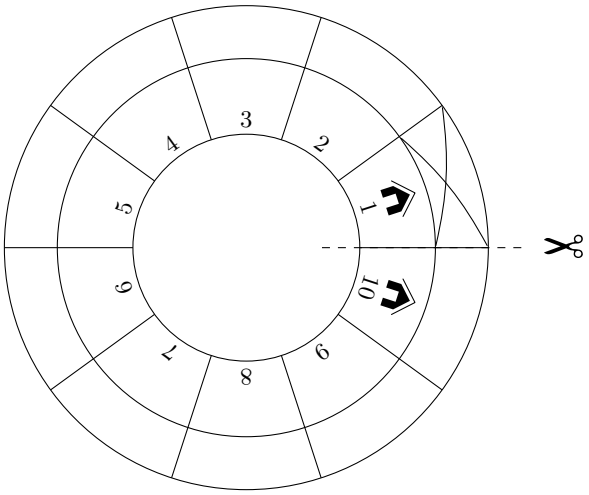
Der Algorithmus soll nun erweitert werden, um nicht nur gerade Straßen mit Störsendern ausstatten zu können, sondern auch ringförmige Straßen. Dazu wird eine ringförmige Straße zwischen zwei beliebigen benachbarten Zellen „aufgeschnitten“ und dann so bearbeitet, wie eine gerade Straße.

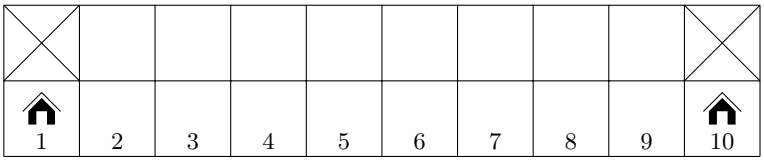
(c) Zeigen Sie, dass dieser Ansatz dazu führen kann, dass zu viele Störsender installiert werden. Vervollständigen Sie dazu folgende Instanzen mit Häusern und markieren Sie die optimalen Lösungen der ringförmigen und der aufgeschnittenen geraden Straßen. Die Reichweite der Störsender ist $k = 2$.



1	2	3	4	5	6	7	8	9	10

Lösung:





(d) Bei dem Algorithmus $\mathcal{A}$ handelt es sich für ringförmige Straßen um einen Approximationsalgorithmus. Bestimmen Sie die Gütegarantie des Algorithmus. Geben Sie dazu möglichst scharf eine Funktion f an, so dass für alle Instanzen I gilt $\mathcal{A}(I) \leq f(\mathrm{OPT}(I))$. Beweisen Sie die Gütegarantie.

Lösung: In Teilaufgabe (c) wurde gezeigt, dass mindestens ein Störsender zu viel platziert werden kann. Zeige jetzt, dass umgekehrt immer höchstens ein Radiosender zu viel installiert wird. Der Algorithmus hat damit eine absolute Gütegarantie von Eins, also $f : m \mapsto m + 1$. Betrachte dazu eine optimale Lösung der ringförmigen Straße und schneide diese auf, wie der Algorithmus. Dann sind alle Häuser mit Positionen in $[k + 1, k + 2, \ldots, n - k]$ sicher von einem Sender abgedeckt.

Angenommen, es sind nun alle Häuser abgedeckt. Dann findet $\mathcal{A}$ nach Teilaufgabe (b) eine ebenso gute Lösung, und es wird kein überflüssiger Störsender platziert.

Angenommen, alle nicht abgedeckten Häuser befinden sich im Bereich $[1, 2, \ldots, k]$. Dann reicht ein weiterer Störsender aus, um all diese Häuser abzudecken. Nach (b) findet $\mathcal{A}$ wieder eine ebenso gute Lösung und platziert also höchstens einen überflüssigen Störsender. Der Fall, dass sich alle nicht abgedeckten Häuser im Bereich $[n - k + 1, n - k + 2, \ldots, n]$ befinden verläuft analog.

Schließlich können sich nicht in *beiden* Randbereichen nicht abgedeckte Häuser befinden, denn solch ein Haus wäre auch in der ringförmigen Straße nicht abgedeckt.

Diskussion: Auch bei diesem Beweis ist es wichtig, von einer optimalen Lösung der ringförmigen Straße auszugehen. Man kann nicht ohne Weiteres davon ausgehen, dass das „Zusammenkleben“ einer optimalen Lösung für gerade Straßen zu einer 1-Approximation im ringförmigen Fall führt. Außerdem wurde häufig nur argumentiert, dass es „Probleme im Randbereich“ gibt, die mit einem zusätzlichen Störsender gelöst werden können. Ohne das „wie“ und „wieso“ ist so etwas allerdings kein Beweis.

Problem 5: Maschinenmodelle $1 + 3 + 2 + 3 = 9$

Es sei der durch leeren Stack erkennende Kellerautomat $\mathcal{A} = (Q, \Sigma, \Gamma, s, Z_0, \delta, F = \emptyset)$ gegeben mit der Zustandsmenge $Q = \{s, q_0, q_1, s', q_0', q_1'\}$, dem Eingabealphabet $\Sigma = \{\mathtt{0}, \mathtt{1}, \mathtt{\#}\}$, dem Stackalphabet $\Gamma = \{Z_0, A_0, A_1, A_2, A_3\}$, Startzustand s und Stackinitialisierung Z_0. Die Übergangsfunktion δ ist folgendermaßen definiert:

$\forall Z \in \Gamma:$	$\delta(s, \mathtt{0}, Z) = \{(q_0, Z)\}$	$\forall Z \in \Gamma:$	$\delta(s', \mathtt{0}, Z) = \{(q_0', Z)\}$
$\forall Z \in \Gamma:$	$\delta(s, \mathtt{1}, Z) = \{(q_1, Z)\}$	$\forall Z \in \Gamma:$	$\delta(s', \mathtt{1}, Z) = \{(q_1', Z)\}$
$\forall Z \in \Gamma:$	$\delta(s, \mathtt{\#}, Z) = \{(s', Z)\}$		$\delta(s', \varepsilon, Z_0) = \{(s', \varepsilon)\}$
$\forall Z \in \Gamma:$	$\delta(q_0, \mathtt{0}, Z) = \{(s, ZA_0)\}$		$\delta(q_0', \mathtt{0}, A_0) = \{(s', \varepsilon)\}$
$\forall Z \in \Gamma:$	$\delta(q_0, \mathtt{1}, Z) = \{(s, ZA_1)\}$		$\delta(q_0', \mathtt{1}, A_1) = \{(s', \varepsilon)\}$
$\forall Z \in \Gamma:$	$\delta(q_1, \mathtt{0}, Z) = \{(s, ZA_2)\}$		$\delta(q_1', \mathtt{0}, A_2) = \{(s', \varepsilon)\}$
$\forall Z \in \Gamma:$	$\delta(q_1, \mathtt{1}, Z) = \{(s, ZA_3)\}$		$\delta(q_1', \mathtt{1}, A_3) = \{(s', \varepsilon)\}$

(a) Liegt das Wort `01#10` in der Sprache $L(\mathcal{A})$? Liegt das Wort `0010#1000` in der Sprache $L(\mathcal{A})$?

Lösung: Nein. Ja.

(b) Geben Sie die Sprache $L(\mathcal{A})$ an, die $\mathcal{A}$ erkennt.

Lösung: $L(\mathcal{A}) = \left\{w_1 w_2 \dots w_n \mathtt{\#} w_n w_{n-1} \dots w_1 \mid w_i \in \{\mathtt{0}, \mathtt{1}\}^2 \text{ für } 1 \leq i \leq n\right\}$

Diskussion: Man kann hier auch anders vorgehen, z.B. indem man eine Grammatik G mit $L(G) = L(\mathcal{A})$ mit folgenden Ableitungsregeln angibt:

$$S \to \mathtt{00}\,S\,\mathtt{00} \mid \mathtt{01}\,S\,\mathtt{01} \mid \mathtt{10}\,S\,\mathtt{10} \mid \mathtt{11}\,S\,\mathtt{11} \mid \mathtt{\#}$$

Sei nun $k \in \{0, 1, 2, \ldots\}$ eine beliebige aber feste Konstante. Das Berechnungsmodell Kellerautomat wird nun so eingeschränkt, dass auf dem Stack zu jedem Zeitpunkt nur noch höchstens k Symbole liegen können. Für jede Konfiguration (q, w, α) gilt also $\alpha \in \Gamma^k$.

(c) Argumentieren Sie, dass der Kellerautomat $\mathcal{A}$ dieser Einschränkung nicht genügt, indem Sie ein Wort angeben, bei dessen Abarbeitung durch $\mathcal{A}$ zu einem Zeitpunkt mindestens $k+1$ Symbole auf dem Stack liegen.

Lösung: Setze $w = \mathtt{0}^{2k}$, dann legt $\mathcal{A}$ insgesamt k-Mal das Zeichen A_0 auf den Stack. Zusammen mit dem Initialisierungssymbol Z_0 ergibt das $k+1$ Zeichen.

Diskussion: k ist hier allgemein, es reicht also nicht, einfach $k = 1$ festzulegen und dann `00` als Wort anzugeben!

(d) Können derart eingeschränkte nichtdeterministische Kellerautomaten genau die regulären Sprachen erkennen? Beweisen Sie Ihre Behauptung!

Lösung: Auf diese Art eingeschränkte Kellerautomaten können genau die regulären Sprachen erkennen.

Ein derart eingeschränkter Kellerautomat kann natürlich einfach einen nichtdeterministischen endlichen Automaten simulieren, ohne den Stack zu nutzen. Mit einem endlichen Automaten kann ein eingeschränkter Kellerautomat mit Zustandsmenge Q wie folgt simuliert werden. Dadurch, dass das Stackalphabet Γ und der Stack konstante Größe haben, lassen sich nämlich alle möglichen Kombinationen aus Zustand und Konfiguration des Stacks durch $|Q| \cdot |\Gamma|^k$, also konstant viele Zustände beschreiben. Der konstant große Stack lässt sich also einfach durch weitere Zustände ersetzen.

Schließlich ist aus der Vorlesung bekannt, dass nichtdeterministische endliche Automaten genau die regulären Sprachen erkennen können.

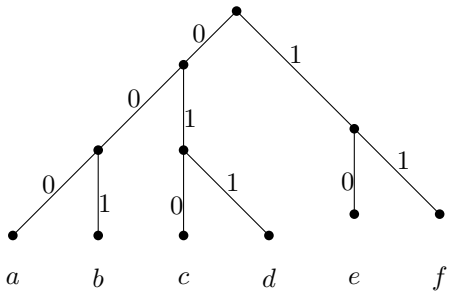
Diskussion: Hier wurde manchmal argumentiert, dass ein derart eingeschränkter Kellerautomat nicht „richtig zählen“ kann. Das kann eine Intuition geben, ist aber kein formal korrekter Beweis.

Problem 6: Informationstheorie 4

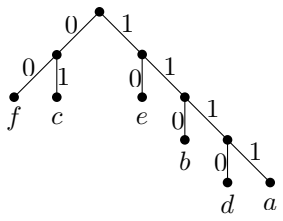
Die Häufigkeitsverteilung H der Symbole $\Sigma = \{a, b, c, d, e, f\}$ ist durch folgende Tabelle gegeben.

Zeichen	a	b	c	d	e	f
Wahrscheinlichkeit	$\frac{2}{47}$	$\frac{8}{47}$	$\frac{11}{47}$	$\frac{4}{47}$	$\frac{9}{47}$	$\frac{13}{47}$

Gegeben ist folgender Codierungsbaum T zur Häufigkeitsverteilung H. Zeigen oder widerlegen Sie, dass T eine minimale mittlere Codewortlänge hat.



Lösung: Das Huffmann Verfahren liefert folgenden Kodierungsbaum T' mit geringere mittlerer Wortlänge.



Kodierung von T

Zeichen	a	b	c	d	e	f
Wahrscheinlichkeit	$\frac{2}{47}$	$\frac{8}{47}$	$\frac{11}{47}$	$\frac{4}{47}$	$\frac{9}{47}$	$\frac{13}{47}$
Kodierung	000	001	010	011	10	11

$$\frac{6 + 24 + 33 + 12 + 18 + 26}{47} = \frac{119}{47}$$

Kodierung von T'

Zeichen	a	b	c	d	e	f
Wahrscheinlichkeit	$\frac{2}{47}$	$\frac{8}{47}$	$\frac{11}{47}$	$\frac{4}{47}$	$\frac{9}{47}$	$\frac{13}{47}$
Kodierung	1111	110	01	1110	10	00

$$\frac{2 \cdot 4 + 3 \cdot 8 + 2 \cdot 11 + 4 \cdot 4 + 2 \cdot 9 + 2 \cdot 13}{47} = \frac{114}{47}$$

T' hat also eine geringere mittlere Wortlänge.

Diskussion: Alternativ führt das Vertauschen der der Zeichen c und e in T zu einem Baum der eine kleinere mittlere Wortlänge haben muss. Das Zeichen c hat einer höhere Wahrscheinlichkeit aber eine längere Kodierung im Vergleich zu e.

Problem 7: Gemischtes 10

Sind die folgenden Aussagen korrekt? Begründen Sie jeweils kurz.

(a) Sei ein deterministischer endlicher Automat $\mathcal{A} = (Q, \Sigma, \delta, s, F)$ gegeben. Dann akzeptiert der Automat $(Q, \Sigma, \delta, s, Q \setminus F)$ die Sprache $L(\mathcal{A})^c$.

Lösung: Wahr. Da $\mathcal{A}$ deterministisch ist, befindet sich $\mathcal{A}$ zu jedem Zeitpunkt der Abarbeitung einer Eingabe in genau einem Zustand. Durch invertieren des Akzeptanzverhaltens dieses Zustands wird genau die Komplementsprache erkannt.

(b) Sei $L \subseteq \Sigma^*$ eine Sprache über einem endlichen Alphabet Σ. Existiert für jedes $w \in L$ ein endlicher Automat, der w erkennt, so ist L regulär.

Lösung: Falsch. Die Aussage gilt für jede formale Sprache, aber nicht jede formale Sprache ist regulär.

(c) Jede Sprache über einem endlichen Alphabet ist rekursiv aufzählbar.

Lösung: Falsch. Die Diagonalsprache ist nicht rekursiv aufzählbar.

(d) Ein 2-Approximationsalgorithmus liefert mit Wahrscheinlichkeit $\frac{1}{2}$ eine optimale Lösung.

Lösung: Falsch. Es kann z.B. auch 2-Approximationsalgorithmen geben, die immer oder nie eine optimale Lösung liefern.

(e) Eine Grammatik, die mindestens eine Ableitungsregel enthält, erzeugt mindestens ein Wort.

Lösung: Falsch, z.B. $S \to S$.

Diskussion: Das leere Wort ϵ wurde häufig als „kein Wort“ interpretiert, was falsch ist.

(f) Das Pumping-Lemma ist für die Sprache

$$L = \{a^i b^j \mid i, j > 0, i \text{ gerade}, j \text{ prim}\}$$

erfüllt.

Lösung: Wahr, betrachte folgende Zerlegung:

$$w = a^i b^j \in L_2$$

$$n = 3, u = \varepsilon, v = aa, x = a^{i-2} b^j$$

Diskussion: Das Pumping-Lemma gilt sogar immer, sonst wäre es ja kein korrektes Lemma! Über manche Sprachen macht es schlicht keine Aussage.

(g) Es existiert eine Sprache in $\mathcal{NP}$, die in Polynomialzeit entschieden werden kann.

Lösung: Wahr, denn $\mathcal{P} \subseteq \mathcal{NP}$.

(h) Es existiert ein Polynom p, sodass die Anzahl n der Äquivalenzklassen eines minimalen nichtdeterministischen Automaten mit m Zuständen durch $p(m)$ beschränkt ist.

Lösung: Falsch, aus der Übung ist bekannt jeder DEA der die $L = \{w \in \{0,1\} \mid$ das n letzte Zeichen von w ist eine $1\}$ erkennt 2^n Zustände hat. Es existiert ein NEA mit n Zuständen der L erkennt.

(i) Der Schnitt einer kontextfreien Sprache mit einer regulären Sprache ist regulär.

Lösung: Falsch. Sei L eine kontextfreie, nicht-reguläre Sprache. Dann ist $L \cap \Sigma^\star = L$ nicht-regulär, obwohl $\Sigma^\star$ regulär ist.

(j) Sei $K \in \{1, 2, \ldots\}$ eine beliebige aber feste Konstante. Das Problem KNAPSACK mit in K beschränkten Gewichten und Kosten liegt in $\mathcal{P}$.

Lösung: Ja, da Knapsack in diesem Fall pseudopolynomiell ist; siehe Vorlesung.

Diskussion: Aus der Einschränkung folgt nicht, dass alle Instanzen konstante Größe haben – es kann nach wie vor sehr viele Gegenstände geben.