Disclaimer

Aufgaben aus dieser Vorlage stammen aus der Vorlesung *Algorithmen, Sprachen und Komplexität* und wurden zu Übungszwecken verändert oder anders formuliert! Für die Korrektheit der Lösungen wird keine Gewähr gegeben.

1. Definitionen der Automatentheorie. Vervollständige die folgenden Definitionen:
   (a) Eine Regel $(l \to r)$ einer Grammatik $G = (V, \sum, P, S)$ heißt rechtslinear, falls ...

   Antwort: immer das an der am weitesten rechts stehende Nicht-Terminal in ein Terminal umgewandelt wird. Dazu muss $l \in V$ und $r \in \sum V \cup \epsilon$.

   (b) Die Menge $Reg(\sum)$ der regulären Ausdrücke über dem Alphabet ist...

   Antwort: ist die kleinste Menge mit folgenden Eigenschaften:

   - $\varnothing \in Reg(\sum), \lambda \in Reg(\sum), \sum \subseteq Reg(\sum)$
   - Wenn $\alpha, \beta \in Reg(\sum)$, dann auch $(\alpha * \beta), (\alpha + \beta), (\alpha^*) \in Reg(\sum)$

   (c) Ein NFA ist ein Tupel $M = (...)$

   Antwort: ein nichtdeterministischer endlicher Automat M ist ein 5-Tupel $M = (Z, \sum, S, \delta, E)$ mit

   - Z ist eine endliche Menge von Zuständen
   - $\sum$ ist das Eingabealphabet
   - $S \subseteq Z$ die Menge der Startzustände (können mehrere sein)
   - $\delta : Z \times \sum \to P(Z)$ ist die (Menge der) Überführungs/Übergangsfunktion
   - $E \subseteq Z$ die Menge der Endzustände

   (d) Die von einem NFA $M = (Z, \sum, S, \delta, E)$ akzeptierte Sprache ist $L(M) = ...$ (ohne Definition der Mehr-Schritt Übergangsfunktion δ)

   Antwort: $L(M) = \{w \in \sum^* | \hat{\delta}(S, w) \cap E \neq \emptyset\}$
   (Das Wort wird akzeptiert wenn es mindestens einen Pfad vom Anfangs in den Endzustand gibt)

   (e) Die von einem PDA $M = (Z, \sum, \Gamma, \delta, z_0, \#)$ akzeptierten Sprache ist $L(M) = ...$

   Antwort: $L(M) = \{x \in \sum^* \mid \text{es gibt } z \in Z \text{ mit } (z_0, x, \#)[...]^*(z, \epsilon, \epsilon)\}$

   (f) Sei L eine Sprache. Für $x, y \in \sum^*$ gilt xR_Ly genau dann, wenn ... (R_L ist die Myhill-Nerode-Äquivalenz zu L)

   Antwort: wenn $\forall z \in \sum^* : (xy \in L \leftrightarrow yz \in L)$ gilt

   (g) Sei $M = (Z, \sum, z_0, \delta, E)$ ein DFA. Die Zustände $z, z' \in Z$ heißen erkennungsäquivalent, wenn

   Antwort: Zwei Zustände $z, z' \in Z$ heißen erkennungsäquivalent ($z \equiv z'$) wenn für jedes Wort $w \in \sum^*$ gilt: $\hat{\sigma}(z, w) \in E \leftrightarrow \hat{\sigma}(z', w) \in E$.

2. Sätze und Lemmas aus der Automatentheorie. Vervollständige die folgenden Aussagen:
   (a) Sei $L \supseteq \sum^*$ eine Sprache. Dann sind äquivalent: 1) L ist regulär (d.h. wird von einem DFA akzeptiert), 2)..., 3)...

   Antwort:

   1. L ist regulär (d.h. von einem DFA akzeptiert)
   2. L wird von einem NFA akzeptiert
   3. L ist rechtslinear (d.h. von einer Typ-3 Grammatik erzeugt)

   (b) Die Klasse der regulären Sprachen ist unter anderem abgeschlossen unter folgenden Operationen:

   Antwort:

   - Vereinigung (L_1, L_2 regulär $\Rightarrow L_1 \cup L_2$ regulär)
   - Schnitt (L_1, L_2 regulär $\Rightarrow L_1 \cap L_2$ regulär)
   - Komplement (L regulär $\Rightarrow \sum^* \setminus L$ regulär)
   - Produkt/Konkatenation (L_1, L_2 regulär $\Rightarrow L_1 L_2$ regulär)
   - Abschluss/Stern-Operation (L regulär $\to L^*$ regulär)

(c) Sei $\sum$ ein Alphabet. Die Anzahl der Grammatiken über $\sum$ ist ... und die Anzahl der Sprachen über $\sum$ ist ... .

> **Antwort**: Für jedes Alphabet ist die Menge der Grammatiken abzählbar unendlich und die Anzahl der Sprachen überabzählbar.

(d) Unter anderem sind folgende (mind. drei) Probleme für kontextfreie Sprachen entscheidbar:

> **Antwort**: Wortproblem, Leerheitsproblem, Äquivalenzproblem

(e) Die Klasse der Kontextfreien Sprachen ist abgeschlossen unter den Operationen 1)... und 2)... . Sie ist aber nicht abgeschlossen unter 3)... und 4)... .

> **Antwort**: Abgeschlossen unter
>
> - Vereinigung ($L_1, L_2 \Rightarrow L_1 \cup L_2$)
> - Produkt/Konkatenation ($L_1, L_2 \Rightarrow L_1L_2$)
> - Stern-Operation ($L \rightarrow L^*$)
>
> Nicht abgschlossen unter
>
> - Schnitt ($L_1, L_2 \Rightarrow L_1 \cap L_2$)
> - Komplement ($L \Rightarrow \sum^* \setminus L$)
> - es gibt kontextfreie Sprachen, die nicht deterministisch kontextfrei sind

(f) Der Satz von Myhill-Nerode besagt,...

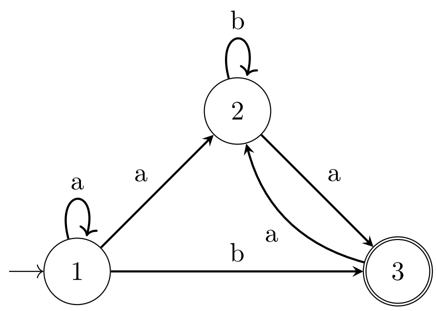
> **Antwort**: Sei L eine Sprache. L ist regulär $\Leftrightarrow index(R_L) < \infty$
> (d.h. nur wenn die Myhill-Nerode-Äquivalenz endliche Klassen hat).

(g) Das Pumping-Lemma für kontextfreie Sprachen ...

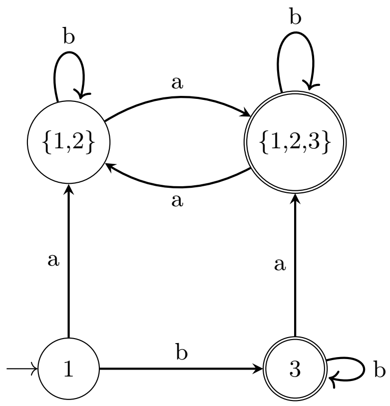
> **Antwort**: Man versucht auszunutzen, daß eine kontextfreie Sprache von einer Grammatik mit endlich vielen Nichtterminalen erzeugt werden muss. Das bedeutet auch: wenn ein Ableitungsbaum ausreichend tief ist, so gibt es einen Ast, der ein Nichtterminal mehrfach enthält. Die durch diese zwei Vorkommen bestimmten Teilbäume werden „gepumpt".
>
> Wenn L eine kontextfreie Sprache ist, dann gibt es $n >= 1$ derart, dass für alle z in L mit $|z| >= n$ gilt: es gibt Wörter u, v, w, x, y in SUM mit
>
> 1. $z = uvwxy$,
> 2. $|vwx| <= n$,
> 3. $|vx| >= 1$ und
> 4. $uv^iwx^iy \in L$ für alle $i >= 0$

3. Konstruktionen der Automatentheorie

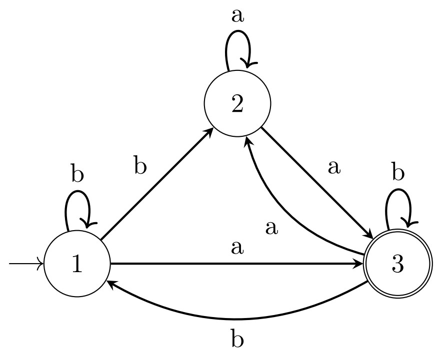
(a) Betrachte den folgenden NFA X. Berechne einen DFA Y mit $L(X) = L(Y)$.



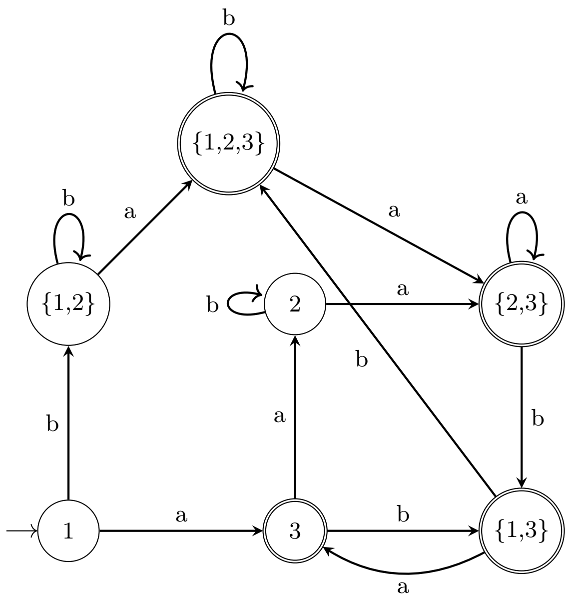
> **Antwort**:



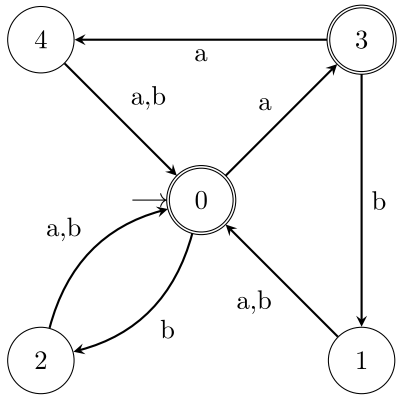
(b) Betrachte den folgenden NFA X. Berechne einen DFA Y mit $L(X) = L(Y)$.



Antwort:



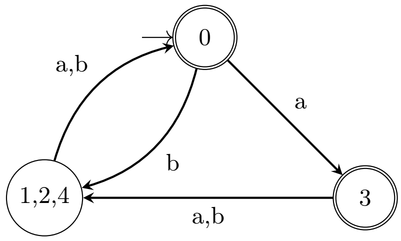
(c) Betrachte den folgenden DFA X. Berechne den minimalen DFA Y mit $L(X) = L(Y)$.



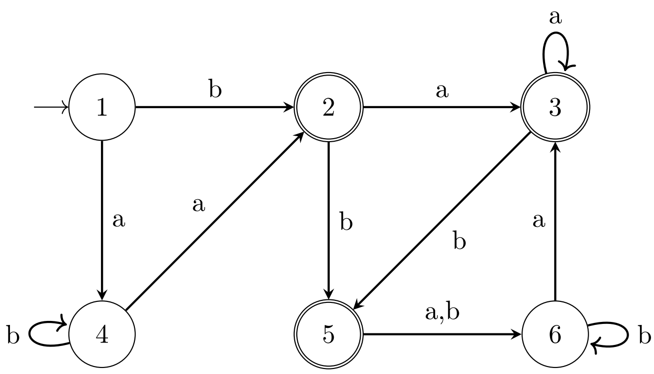
Antwort:

1. Stelle eine Tabelle aller ungeordneten Zustandspaare $\{z, z'\}$ mit $z \neq z'$ auf.
2. Markiere * alle Paare $\{z, z'\}$ mit $z \in E$ und $z' \notin E$.
3. Markiere (*) ein beliebiges unmarkiertes Paar $\{z, z'\}$, für das es ein $a \in \sum$ gibt, so dass $\{\delta(z, a), \delta(z', a)\}$ bereits markiert ist (falls dies möglich ist).
4. Wiederhole den vorherigen Schritt, bis sich keine Änderung in der Tabelle mehr ergibt.
5. Unmarkierte Paare werden verschmolzen

1	*			
2	*			
3	(*)	(*)	(*)	
4	*		*	*
	0	1	2	3

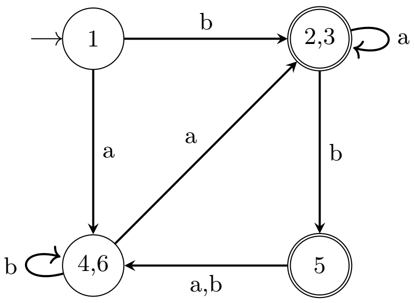


(d) Betrachte den folgenden DFA X. Berechne den minimalen DFA Y mit $L(X) = L(Y)$.

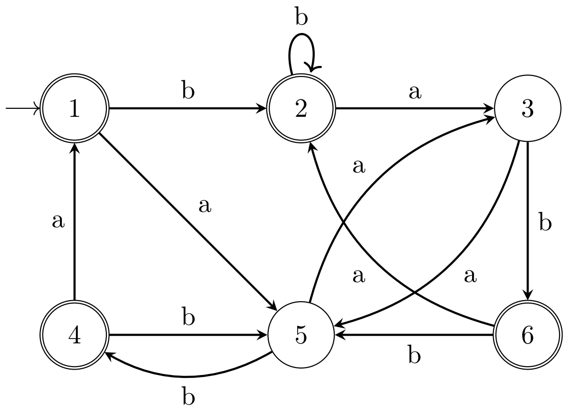


Antwort:

2	*				
3	*				
4	(*)	*	*		
5	*	(*)	(*)	*	
6	(*)	*	*		*
	1	2	3	4	5

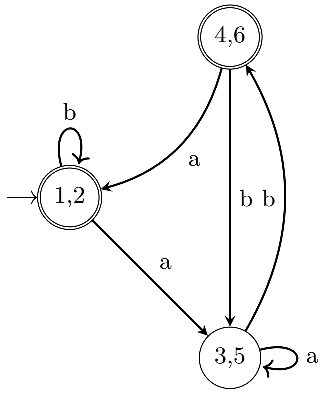


(e) Betrachte den folgenden DFA X. Berechne den minimalen DFA Y mit $L(X) = L(Y)$.



Antwort:

2					
3	*	*			
4	(*)	(*)	*		
5	*	*		*	
6	(*)	(*)	*		*
	1	2	3	4	5



4. Algorithmen für reguläre Sprachen. Sei $\sum = \{a, b, c\}$. Gebe einen Algorithmus an, der bei Eingabe eines NFA X entscheidet, ob alle Wörter $\omega \in L(X)$ ungerade Länge besitzen und *abc* als Infix enthalten.

Antwort:

5. Kontextfreie Sprachen: Sei $\sum = \{a, b, c\}$. Betrachte die Sprache $K = \{a^k b^l c^m | k \leq l \text{ oder } k \leq m\}$.
   (a) Zeige, dass K eine kontextfreie Sprache ist.

   Antwort:

   (b) Zeige, dass $L = \sum^* \backslash K$ (Komplement von L) nicht kontextfrei ist.

   Antwort:

   (c) Begründe warum K deterministisch kontextfrei ist oder warum nicht.

   Antwort:

6. Kontextfreie Grammatiken: Sei $\sum = \{a, b, c, \}$
   (a) Sei G die kontextfreie Grammatik mit Startsymbol S und der Regelmenge $S \to AB$, $A \to aBS|a$ und $B \to bBa|b|\epsilon$. Überführe G in eine äquivalente Grammatik in Chomsky Normalform.

   Antwort: Chomsky Normalform hat auf rechter Ableitungsseite nur ein Terminal oder zwei Nicht-Terminale

   1. Startzustand
      $S \to AB$, $A \to aBS|a$, $B \to bBa|b|\epsilon$
   2. ϵ-Regel: Menge $M = \{B\}$ der *epsilon* Terminal-Überführungen kompensieren;
      $S \to AB|A$, $A \to aBS|a|aS$, $B \to bBa|b|ba$
   3. Kettenregel: Menge $M = \{(S, A), (S, S), (A, A), (B, B)\}$ von Ketten (Ableitungen auf ein Nicht-Terminal)
      $S \to AB|aBS|a|aS$, $A \to aBS|a|aS$, $B \to bBa|b|ba$

4. Terminale und Nicht-Terminal trennen:
   $S \to AB|CBS|C|CS$, $A \to CBS|C|CS$, $B \to DBC|b|DC$, $C \to a$, $D \to b$
5. Längen verkürzen:
   $S \to AB|CX|C|CS$, $A \to CX|C|CS$, $B \to DY|b|DC$, $C \to a$, $D \to b$, $X \to BS$, $Y \to BC$

(b) Sei G' die kontextfreie Grammatik mit Startsymbol S und der Regelmenge

$$S \to AB,\ A \to CD|CF,\ F \to AD,\ B \to c|EB,\ C \to a,\ D \to b,\ E \to c$$

Entscheide mit dem CYK-Algorithmus, ob die Wörter $w_1 = aaabbbcc$ oder $w_2 = aaabbccc$ von G' erzeugt werden.

Antwort:

$w_1 = aaabbbcc$

a	a	a	b	b	b	c	c
C	C	C	D	D	D	B,E	B,E
		A				B	
		F					
	A						
	F						
A							
S							
S							

$\Rightarrow w_1$ wird von G' erzeugt

$w_2 = aaabbccc$

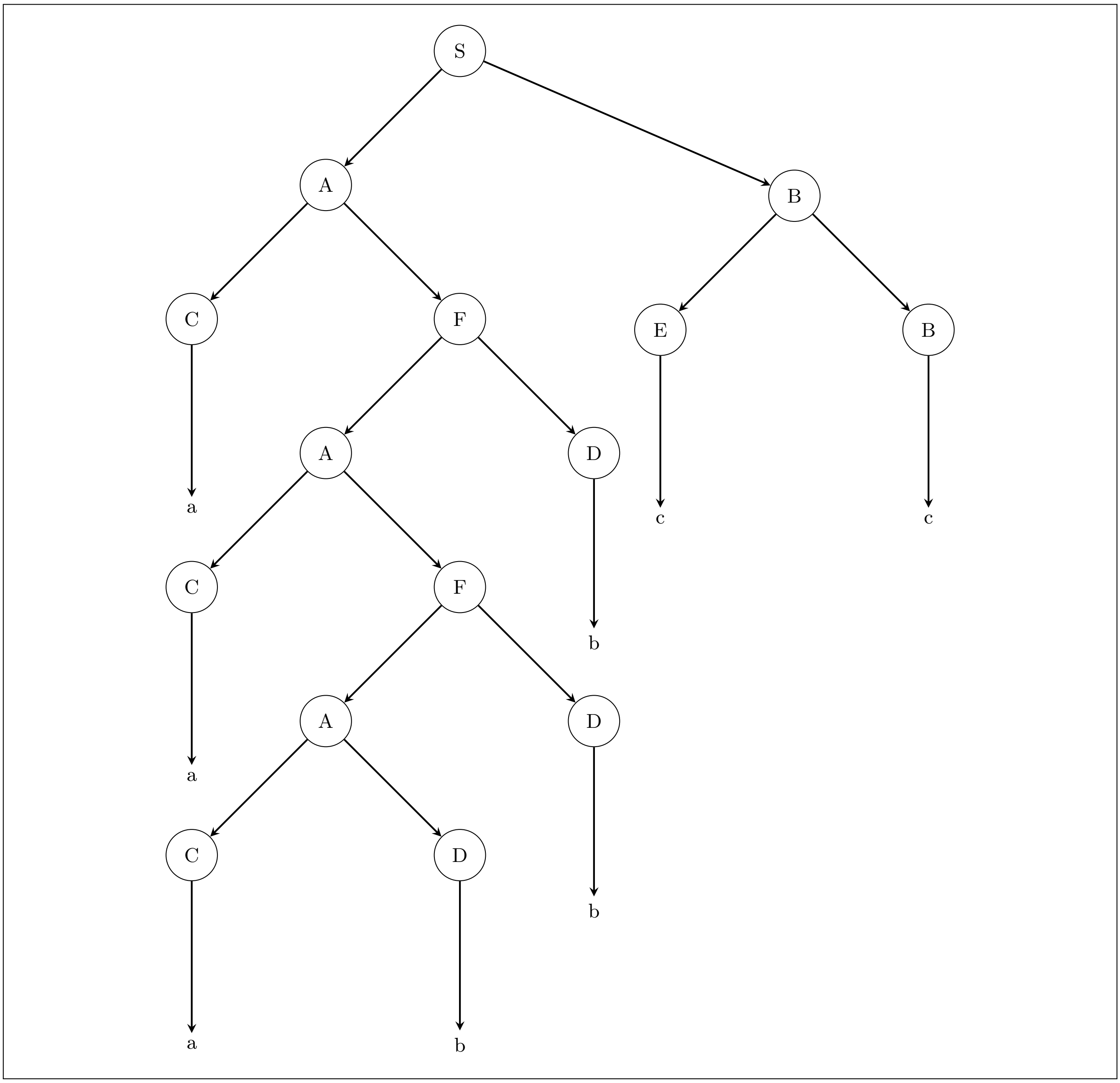
a	a	a	b	b	c	c	c
C	C	C	D	D	B,E	B,E	B,E
		A			B	B	
		F			B		
	A						
	F						
A							
S							

$\Rightarrow w_2$ wird nicht von G' erzeugt

(c) Gebe für die Wörter aus b), die von G' erzeugt werden, den Ableitungsbaum an.

Antwort:

$w_1 = aaabbbcc$



7. Definitionen der Berechnbarkeitstheorie. Verfollständige die Definitionen
   (a) Ein While Programm ist von der Form...

Antwort:

- $x_i = c, x_i = x_j + c, x_i = x_j - c$ mit $c \in \{0, 1\}$ und $i, j \geq 1$ (Wertzuweisung) oder
- P_1, P_2, wobei P_1 und P_2 bereits While Programme sind (sequentielle Komposition) oder
- while $x_i \neq 0$ do P end, wobei P ein While Programm ist und $i \geq 1$.

   (b) Ein Loop-Programm ist von der Form

Antwort:

- $x_i := c, x_i := x_j + c, x_i := x_j \div c$ mit $c \in \{0, 1\}$ und i, j (Wertzuweisung) oder
- $P_1; P_2$, wobei P_1 und P_2 Loop-Programme sind (sequentielle Komposition) oder
- loop x_i do P end, wobei P ein Loop-Programm ist und i_1.

   (c) Eine Turingmaschine ist ein 7-Tupel $M = (Z, \sum, \Gamma, \delta, z_0, \Box, E)$, wobei...

Antwort:

- 7-Tupel $M = (Z, \sum, \Gamma, \delta, z_o, \Box, E)$
- $\sum$ das Eingabealphabet
- Γ mit $\Gamma \supseteq \sum$ und $\Gamma \cap Z \neq 0$ das Arbeits- oder Bandalphabet,

- $z_0 \in Z$ der Startzustand,
- $\delta : Z \times \Gamma \rightarrow (Z \times \Gamma \times \{L, N, R\})$ die Überführungsfunktion
- $\square \in \Gamma / \sum$ das Leerzeichen oder Blank und
- $E \subseteq Z$ die Menge der Endzustände ist

(d) Die von einer Turingmaschine M akzeptierte Sprache ist $L(M) = ...$

Antwort: $L(M) = \{w \in \sum^* |$es gibt akzeptierte Haltekonfiguration mit $z_0 w\square \vdash_M^* k\}$.

(e) Gödels Vermutung lautet,...

Antwort: Eine partielle Funktion $\mathbb{N}^k \rightarrow \mathbb{N}$ ist genau dann intuitiv berechenbar, wenn sie μ-rekursiv ist.

(f) Wann ist eine Sprache semi-entscheidbar?

Antwort: Eine Sprache ist genau dann semi-entscheidbar, wenn sie von einer nichtdeterministischen Turingmaschine akzeptiert wird.

(g) Seien $A \subseteq \sum^*$ und B$\subseteq \Gamma^*$. Eine Reduktion von A auf B ist ...

Antwort: Eine Reduktion von A auf B ist eine totale und berechenbare Funktion $f : \sum^* \rightarrow \Gamma^*$, so dass für alle $w \in \sum^*$ gilt: $w \in A \leftrightarrow f(x) \in B$. A heißt auf B reduzierbar (in Zeichen $A \leq B$), falls es eine Reduktion von A auf B gibt.

(h) Eine Sprache L heißt rekursiv aufzählbar, falls ...

Antwort:

- L ist semi-entscheidbar
- L wird von einer Turing-Maschine akzeptiert
- L ist vom Typ 0 (d.h. von Grammatik erzeugt)
- L ist Bild berechenbarer partiellen Funktion $\sum^* \rightarrow \sum^*$
- L ist Bild berechenbarer totalen Funktion $\sum^* \rightarrow \sum^*$
- L ist Definitionsbereich einer berechenbaren partiellen Funktion $\sum^* \rightarrow \sum^*$

(i) Sei $f : N \rightarrow N$ eine monotone Funktion. Die Klasse $TIME(f)$ besteht aus allen Sprachen L, für die es eine Turingmaschine M gibt mit ...

Antwort:

- M berechnet die charakteristische Funktion von L.
- Für jede Eingabe $w \in \sum^*$ erreicht M von der Startkonfiguration $z_0 w\square$ aus nach höchstens $f(|w|)$ Rechenschritten eine akzeptierende Haltekonfiguration (und gibt 0 oder 1 aus, je nachdem ob $w \notin L$ oder $w \in L$ gilt).

8. Sätze der Berechnbarkeitstheorie: Vervollständige die folgenden Aussagen

(a) Zu jeder Mehrband-Turingmaschine M gibt es ...

Antwort: eine Turingmaschine M' die diesselbe Funktion löst

- Simulation mittels Einband-Turingmaschine durch Erweiterung des Alphabets: Wir fassen die übereinanderliegenden Bandeinträge zu einem Feld zusammen und markieren die Kopfpositionen auf jedem Band durch $*$.
- Alphabetsymbol der Form $(a, *, b, \diamond, c, *, ...) \in (\Gamma \times \{*, \diamond\})^k$ bedeutet: 1. und 3. Kopf anwesend ($*$ Kopf anwesend, $\diamond$ Kopf nicht anwesend)

(b) Sei $f : N^k \rightarrow \mathbb{N}$ eine Funktion für ein $k \in \mathbb{N}$. Die folgenden Aussagen sind äquivalent: 1) f ist Turing-berechenbar, 2)..., 3)..., 4)...

Antwort:

1. f ist Turing berechenbar
2. f ist μ rekursiv
3. f ist rekursiv aufzählbar
4. f ist von Menschen berechenbar

(c) Sei $L \subseteq \sum^*$ eine Sprache. Sind L und $\sum^* \setminus L$ semi-entscheidbar, dann...

Antwort:

(d) Der Satz von Rice lautet...

Antwort: dass es unmöglich ist, eine beliebige nicht-triviale Eigenschaft der erzeugten Funktion einer Turing-Maschine (oder eines Algorithmus in einem anderen Berechenbarkeitsmodell) algorithmisch zu entscheiden.

Es sei $\mathcal{P}$ die Menge aller partiellen Turing-berechenbaren Funktionen und $\mathcal{S} \subsetneq \mathcal{P}$ eine nicht-leere, echte Teilmenge davon. Außerdem sei eine effektive Nummerierung vorausgesetzt, die einer natürlichen Zahl $n \in \mathbb{N}$ die dadurch codierte Turing-Maschine M_n zuordnet. Dann ist die Menge $\mathcal{C}(\mathcal{S}) = \{n \mid \text{die von } M_n \text{ berechnete Funktion liegt in } \mathcal{S}\}$ nicht entscheidbar.

„Sei U eine nicht-triviale Eigenschaft der partiellen berechenbaren Funktionen, dann ist die Sprache $L_U = \{<M> \mid \text{M berechnet } f \in U\}$ nicht entscheidbar."

9. Berechnungsmodelle

(a) Gebe ein Loop-Programm an, das die Funktion $n \rightarrow n^2 - n$ berechnet

Antwort:

```
h= 1
for(i= 0; i < 2; i++) do {
  h= h * n
}
h= h - 1;
return h
```

(b) Gebe ein Loop Programm an, das die Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$ mit $f(n_1, n_2) = 2n_1n_2$ berechnet. Verwende nur elementare Anweisungen und keine Abkürzungen.

Antwort:

```
h= 1
for (i=0; i<2; i++) do {
  h = h * n_i
}
h = 2 * h
return h
```

(c) Gebe ein GoTo Programm an, das die Funktion $g : \mathbb{N} \rightarrow \mathbb{N}$ mit $g(n_1, n_2) = |n_1 - n_2|$ berechnet. Verwende nur elementare Anweisungen und keine Abkürzungen.

Antwort:

```
start:
  h = n_1 - n_2
  if h > 0:
    goto end
  h = -h
end:
  return h
```

(d) Gebe eine deterministische Turingmaschine M für das Eingabealphabet $\{0, 1\}$ an, das folgende Funktion berechnet: Für Eingabe $a_1a_2...a_{n-1}a_n$ berechnet M die Ausgabe $a_na_1...a_{n-1}$ (letzte Symbol der Eingabe an erste Stelle).

Antwort: $\sum = \{0, 1\}$

z_0 Zahlenende finden: $\delta(z_0, 0) = (z_0, 0, R), \delta(z_0, 1) = (z_0, 1, R), \delta(z_0, \square) = (z_1, \square, L)$

z_1 letzte Zahl löschen: $\delta(z_1, 0) = (z_2, \square, L), \delta(z_1, 1) = (z_3, \square, L), \delta(z_1, \square) = (z_2, \square, N)$

z_2 zurück zum Anfang bei $a_n = 0$: $\delta(z_2, 0) = (z_2, 0, L), \delta(z_2, 1) = (z_2, 1, L), \delta(z_2, \square) = (z_4, \square, R)$

z_3 zurück zum Anfang bei $a_n = 1$: $\delta(z_3, 0) = (z_3, 0, L), \delta(z_3, 1) = (z_3, 1, L), \delta(z_3, \square) = (z_5, \square, N)$

z_4 $a_n = 0$ an Anfang schreiben: $\delta(z_4, \square) = (z_e, 0, N)$

z_5 $a_n = 1$ an Anfang schreiben: $\delta(z_5, \square) = (z_e, 1, N)$

z_e Endzustand: $\delta(z_e, 0) = (z_e, 0, N), \delta(z_e, 1) = (z_e, 1, N), \delta(z_e, \square) = (z_e, \square, N)$

10. Reduktionen

(a) Seien $A, L \subseteq \sum^*$ nichtleere Sprachen und A entscheidbar. Gebe eine Reduktion von $L \cup A$ auf L an.

Antwort:

(b) Gebe eine Bedingung für A an, sodass $L \cup A \leq_p L$ für alle nichtleeren Sprachen $L \subseteq \sum^*$ gilt. Begründe.

Antwort:

11. Komplexitätsklassen. Ergänze zu den Paaren von Komplexitätsklassen das Relationssymbol zur Teilmengenbeziehung.

(a) EXPSPACE ? EXPTIME

Antwort: EXPSPACE $\geq$ EXPTIME

(b) NP ? P

Antwort: NP $\geq$ P

(c) NP ? NPSPACE

Antwort: NP $\leq$ NPSPACE

(d) NPSPACE ? PSPACE

Antwort: NPSPACE = PSPACE

12. Komplexitätsklassen. Bringe in die richtige Reihenfolge:

(a) EXPSPACE, PSPACE, 2EXPTIME, EXPTIME, P

Antwort: $P \subseteq PSPACE \subseteq EXPTIME \subseteq EXPSPACE \subseteq 2EXPTIME$

(b) PSPACE, EXPSPACE,2EXPSPACE, NEXPTIME, 2NEXPTIME, NP

Antwort: $NP \subseteq PSPACE, NEXPTIME \subseteq EXPSPACE, 2NEXPTIME \subseteq 2EXPSPACE$

(c) NP, P, EXPTIME, NEXPTIME, PSPACE, NPSPACE, NEXPSPACE, EXPSPACE

Antwort: $P \leq NP \leq PSPACE, NPSPACE \leq EXPTIME \leq NEXPTIME \leq EXPSPACE, NEXPSPACE$

13. Unentscheidbare Probleme:

(a) Gebe entscheidbare Probleme an (als Menge oder als Eingabe-Frage-Paar)

Antwort:

- Wortproblem: Gilt $w \in L(M)$ für eine gegebene Sprache L und $w \in \sum^*$
- Leerheitsproblem: Gilt $L(M) = \varnothing$ für eine gegebene Sprache L
- Endlichkeitsproblem: Ist eine gegebene Sprache endlich?
- Schnittproblem: Gilt $L_1 \cap L_2 = \varnothing$ für gegebene L_1, L_2?
- Inklusionsproblem: Gilt $L_1 \subseteq L_2$ für gegebene L_1, L_2?
- Äquivalenzproblem: Gilt $L_1 = L_2$ für gegebene L_1, L_2?

(b) Gebe unentscheidbare Probleme an (als Menge oder als Eingabe-Frage-Paar)

Antwort:

- allgemeine Halteproblem: Das Halteproblem ist die Menge aller Paare (M, x), wobei M eine TM ist und $x \in \{0,1\}^*$, so dass M bei Eingabe von x hält. $H = \{w\#w \mid w \in L_{TM}, x \in \{0,1\}^*, M_w \text{ angesetzt auf x hält}\}$
- spezielle Halteproblem: $K = \{w \in L_{TM} \mid M_w \text{ angesetzt auf w hält}\}$
- Halteproblem auf leerem Band: $H_0 = \{w \in L_{TM} \mid M_w \text{ hält angesetzt auf ein leeres Band}\}$
- Posts Korrespondenzproblem: PCP ist die Menge der Korrespondenzsysteme (endliche Folge von Paaren), die eine Lösung besitzen
- Schnittproblem: $\{(G_1, G_2) \mid G_1, G_2 \text{ kontextfreie Grammatiken }, L(G_1) \cap L(G_2) = \varnothing\}$
- Regularitätsproblem für PDA: $Reg_{PDA} = \{P \mid P\ PDA \text{ mit } L(P) \text{ regulär}\}$
- Inklusionsproblem DPDA: $\{(P_1, P_2) \mid P_1, P_2 \text{ DPDAs mit } L(P_1) \subseteq L(P_2)$
- Universalitätsproblem: $\{P\ PDA \mid L(P) = \sum^*\}$
- Äquivalenzproblem PDA: $\{(P_1, P_2) \mid P_1, P_2 \text{ PDAs mit } L(P_1) = L(P_2)$

14. NP-Vollständigkeit

(a) Eine Sprache B ist NP-vollständig, falls ...

Antwort: Eine Sprache ist NP-vollständig, falls sie zu NP gehört und NP-hart ist.
Eine Sprache B ist NP-hart, falls für alle $A \in NP$ gilt: $A \leq_P B$ (A ist mindestens so schwer wie jedes Problem in NP).
Wenn B NP-vollständig ist, dann gilt: $P = NP \Leftrightarrow B \in P$.

(b) Gebe NP-vollständige Probleme an (als Menge oder Eingabe-Frage-Paar).

Antwort:
Gerichteter Hamiltonkreis?

- Eingabe: gerichteter Graph $G = (V, E)$ mit Knotenmenge V und Kantenmenge $E \subseteq V \times V$
- Frage: Besitzt der Graph G einen Hamiltonkreis, d.h. kann man den Graphen so durchlaufen, daß jeder Knoten genau einmal besucht wird?

Ungerichteter Hamiltonkreis

- Eingabe: ungerichteter Graph $G = (V, E)$ mit Knotenmenge V und Kantenmenge $E \subseteq \binom{V}{2} = \{X \subseteq V \mid |X| = 2\}$.
- Frage: Kann ein ungerichteter Graph so durchlaufen werden, dass jeder Knoten genau ein mal besucht wird?

3-Färbbarkeit

- Eingabe: ungerichteter Graph(V,E)
- Frage: Gibt es einen ungerichteten Graphen, deren Knoten sich mit drei Farben färben lassen, so dass benachbarte Knoten unterschiedliche Farben haben
- Frage (alternativ): Gibt es Zuordnung von k verschiedenen Farben zu Knoten in V, so dass keine zwei benachbarten Knoten v_1, v_2 dieselbe Farbe haben?

3-SAT

- Ist eine aussagenlogische Formel in konjunktiver Normalform mit ≥ 3 Literalen pro Klausel erfüllbar?
- Eingabe: eine aussagenlogische Formel φ in konjunktiver Normalform mit höchstens drei Literalen pro Klausel.
- Frage: Hat φ eine erfüllende Belegung?

Travelling Salesman Problem

- eine $n \times n$-Matrix $M = (M_{i,j})$ von Entfernungen zwischen n Städten und eine Zahl d.
- FRAGE: Gibt es eine Tour durch alle Städte, die maximal die Länge d hat?

15. Polynomialzeitreduktion: Betrachte das Problem 4C, also die Menge der ungerichteten Graphen die sich mit vier Farben färben lassen.

(a) Gebe eine Polynomialzeitreduktion von 3C auf 4C an.

Antwort:

(b) Zeige, dass wenn $4C \in P$, dann gilt $P = NP$.

Antwort: