FHNW : Studiengang Informatik

THEORETISCHE INFORMATIK: Prüfung 2A : Lösungen

D. Mall 5. Januar 2021

Aufgabe 1: Allgemeine Fragen zum ersten Teil

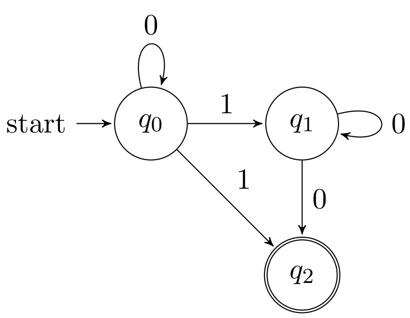
a) Gegeben sei die Grammatik $G = (\mathcal{N}, \mathcal{T}, R, S)$ mit $\mathcal{N} := \{S, A, B, C\}$ und $\mathcal{T} := \{a, b, c, d\}$
Bestimmen Sie für die Regeln der Grammatik G die höchste Klasse, in der die jeweilige Regel liegt.

S	$\longrightarrow$	AB	kontextfrei
$aAdBc$	$\longrightarrow$	$accdBc$	kontextsensitiv
Ac	$\longrightarrow$	AC	allgemein

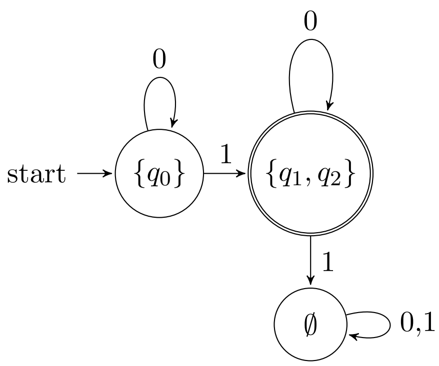
[2 Pkte]

Hinweis: Nur ein ganz richtiges Resultat gibt Punkte.

b) Geben Sie für den folgenden endlichen Automaten den Graphen eines äquivalenten, deterministischen Automaten an. Verwenden Sie das Verfahren von Rabin-Scott. **[3 Pkte]**



Lösung:



c) Geben Sie für die folgenden Sprachen die höchste Klasse in der Chomsky-Hierarchie an, in der die Sprache liegt.

$L_1 = \{\ (010)^k \mid k \in \mathbb{N}\}$	regulär
$L_2 = \{\ 0\ ,\ 1\ ,\ 01\ ,\ 011\ ,\ 0111\}$	regulär
$L_3 = \{\ 0^k10^k \mid k \in \mathbb{N}\}$	kontextfrei
$L_4 = \{\ 1110^k111 \mid k \in \mathbb{N}\}$	regulär

Bewertung: 3 richtig [**1 Pkt**], 4 richtig [**3 Pkte**],

d) Wir betrachten die Sprache $L = \{\ \epsilon\ ,\ 0\ ,\ 10\ ,\ 100\ \} \subset \Sigma^*$ mit $\Sigma = \{0,1\}$ und die Nerode-Relation von L auf Σ^*.

Geben Sie jeweils an, ob die folgenden Aussagen richtig oder falsch sind:

$[\epsilon]$	$=$	$\{\epsilon\}$	*richtig*
$[0]$	$=$	$[00]$	*falsch*
$[1]$	$=$	$[100]$	*falsch*
$[00]$	$=$	$\Sigma^* - \{\epsilon, 0, 1, 10, 100\}$	*richtig*
$[10]$	$=$	$\{10\}$	*richtig*

Bewertung: 3 richtig [**1 Pkt**], 4 richtig [**2 Pkte**], 5 richtig [**4 Pkte**].

Aufgabe 2

Es sei $L \subset \Sigma^*$ eine Sprache und $\bar{L}$ ihr Komplement: $\bar{L} = \Sigma^* - L$.

Entscheiden Sie, ob die folgenden Aussagen richtig oder falsch sind.

a) Wenn L rekursiv ist, dann ist $\bar{L}$ rekursiv aufzählbar, aber nicht notwendigerweise rekursiv.

b) Es gibt reguläre Sprachen, die zwar rekursiv aufzählbar sind, aber nicht rekursiv.

c) Wenn die Sprachen L und $\bar{L}$ rekursiv aufzählbar sind, dann ist L rekursiv.

d) Der Vereinigung zweier rekursiven Sprachen ist immer rekursiv.

e) Jede Sprache, für die es eine Grammatik gibt, ist rekursiv.

f) L_{univ} ist nicht rekursiv, aber es ist eine offene Frage, ob L_{univ} rekursiv aufzählbar ist.

Hinweis: Als Antwort wird nur *richtig* oder *falsch* erwartet. Kreuzen Sie Ihre Antworten in der folgenden Tabelle an.

	richtig	falsch
a)		X
b)		X
c)	X	
d)	X	
e)		X
f)		X

Bewertung: 4 richtig [**1 Pkt**], 5 richtig [**3 Pkte**], 6 richtig [**6 Pkte**],

Aufgabe 3

Wir haben die Sprachen LOOP und WHILE eingeführt und eine operative Semantik für diese Sprachen definiert.

a) Erklären Sie den Unterschied zwischen den Sprachkonstrukten **loop** und **while**.

[2 Pkte]

Lösung: Beide Konstrukte stellen Schleifen dar. Bei loop x do ... od wird der Wert x bei Eintritt der Schleife ausgelesen und die Schleife so oft ausgeführt, wie dieser Wert angibt. Bei while x do ... od wird bei jedem Schleifendurchgang kontrolliert, ob der Wert x noch ungleich 0 ist.

b) Die beiden folgenden Programme berechnen je eine Funktion

$$f: \mathbb{N}^2 \longrightarrow \mathbb{N}^2$$

Falls das jeweilige Programm terminiert, geben Sie den Wert der Variablen x_1 und x_2 nach Terminierung des Programms für den Input $x_1 := 2, x_2 := 2$ an, sonst geben Sie an, dass das Programm nicht terminiert.

```
while x1 do
        x2 := P(x1);
        loop x1 do
                x1 := P(x1);
                x2 := S(x2)
        od
od
```

```
loop x1 do
        x2 := P(x1);
        while x1 do
                x1 := P(x1);
                x2 := S(x2)
        od
od
```

Resultat des 1. Programms $(x_1, x_2) = (0, 3)$	**[2 Pkte]**

Resultat des 2. Programms $(x_1, x_2) = (0, 0)$	**[2 Pkte]**

Aufgabe 4

Zeigen Sie, dass die Funktion

$$\begin{array}{rccc} f: & \mathbb{N}\times\mathbb{N} & \longrightarrow & \mathbb{N} \\ & (x,y) & \longmapsto & (x+1)^y \end{array}$$

primitiv rekursiv ist. **[5 Pkte]**

Hinweis: Sie dürfen annehmen, dass die Grundfunktionen und die Funktionen

$$\begin{array}{|cccc|} \hline \mathbf{1}: & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & x & \longmapsto & 1 \\ \hline add: & \mathbb{N}\times\mathbb{N} & \longrightarrow & \mathbb{N} \\ & (x,y) & \longmapsto & x+y \\ \hline mult: & \mathbb{N}\times\mathbb{N} & \longrightarrow & \mathbb{N} \\ & (x,y) & \longmapsto & xy \\ \hline \end{array}$$

primitiv rekursiv sind. Erklären Sie genau, was Sie tun müssen.

Lösung: Wir wenden das primitive Rekursionsschema an:

$$\begin{array}{rcl} f(x,0) & = & g(x) \\ f(x,y+1) & = & h(x,y,f(x,y)) \end{array}$$

Wir müssen also eine 1-stellige Funktion $g(x)$ und eine 3-stellige Funktion $h(x,y,z)$, die primitiv rekursiv sind und die Bedingungen

$$\begin{array}{rcl} f(x,0) & = & (x+1)^0 = \mathbf{1}(x) = g(x) \\ f(x,y+1) & = & (x+1)^{y+1} = (x+1)\cdot(x+1)^y = S(x)(x+1)^y = \\ & = & S(x)f(x,y) = mult(S(x),f(x,y)) = h(x,y,f(x,y)) \end{array}$$

erfüllen, angeben. Wir haben also

1. $g(x) = \mathbf{1}(x)$ und
2. $h(x,y,z) = S(x)\cdot z = mult(S(x),z)$ oder $h = mult(S(U_1^3),U_3^3)$.

Beide Funktionen sind nach Voraussetzung primitiv rekursiv. (Wir verwenden bei S und *mult* das Kompositionsschema.)

Aufgabe 5 : Gödelnummern und universelle Turing-Maschinen

Sei M eine Turing-Maschine vom Typ $(Q, \Sigma, \Gamma, \delta, q_0, F)$ mit $Q = \{q_0, q_1\}$, $\Sigma = \{0, 1\}$, $\Gamma = \Sigma \cup \{b\}$, $F = \{q_1\}$ und

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, H, R\}.$$

M sei durch die Gödelnummer

$$< M >= 111010101001000110100100101011001001001000101100100010101000111$$

gegeben, wobei die Codierung durch

q_i	x_j	q_k	x_l	D_m	$\longrightarrow 0^{i+1}\ 1\ 0^j\ 1\ 0^{k+1}\ 1\ 0^l\ 1\ 0^m$

mit $x_1 = 0, x_2 = 1, x_3 = b$ und $D_1 = L, D_2 = H, D_3 = R$ gegeben ist.

a) Geben Sie eine Turing-Tabelle von M an. **[3 Pkte]**

Lösung:

q_0	0	q_0	1	R
q_0	1	q_1	0	L
q_1	1	q_1	b	L
q_1	b	q_0	0	R

b) Erklären Sie, wie L_{univ} definiert ist. **[1 Pkt]**

Lösung: L_{univ} besteht aus den Wörtern der Form $< M > \omega$, M eine Gödelnummer und ω ein beliebiges Wort über dem Alphabet Σ, für die gilt, dass die Turing-Maschine M das Wort ω akzeptiert.

c) Das folgende Wort wird einer universellen Turing-Maschine als Input gegeben:

$$< M > 0001$$

Geben Sie die Berechnung, die die universelle Turing-Maschine auf diesem Input ausführt, als Folge von Konfigurationen an. **[3 Pkte]**

Lösung:

$q_0 0001 \vdash 1 q_0 001 \vdash 11 q_0 01 \vdash 111 q_0 1 \vdash 11 q_1 10 \vdash 1 q_1 1b0 \vdash q_1 1bb0 \vdash q_1 bbbb0 \vdash 0 q_0 bbb0$

d) Gibt es ein Wort $< M > \omega$, $\omega \in \Sigma^*$, mit $|\omega| = 3$, das von der universellen Turing-Maschine akzeptiert wird?

Falls ja, geben Sie eines an, falls nein, erklären Sie warum nicht. **[3 Pkte]**

Lösung: Nein, es gibt kein solches Wort.

Fall 1: $\omega = 000$: Wir erhalten $q_0 000 \vdash \cdots \vdash 111 q_0 b$

Fall 2: ω enthält eine 1: $\omega = 0 \cdots 01 \cdots$. Wir erhalten

$q_0 0 \cdots 01 \cdots \vdash \cdots \vdash 1 \cdots 1 q_0 1 \cdots \vdash 1 \cdots 1 q_1 10 \cdots \vdash \cdots q_1 1b \cdots b0 \cdots \vdash q_1 bb \cdots b0 \cdots \vdash 0 q_0 b \cdots b0\,.$

In beiden Fällen stoppt die Turing-Maschine im Zustand q_0, d.h. sie akzeptiert nicht.

Aufgabe 6 : Komplexitätsklassen

a) Betrachten Sie die folgende Sprache

$$L = \{\ 1110^{2n}11110^{n} \mid n \in \mathbb{N}\ \}$$

über dem Alphabet $\Sigma := \{0, 1\}$.

Spezifizieren Sie eine Komplexitätsklasse, in der die Sprache L liegt. **[2 Pkte]**

Hinweis: Sie erinnern sich, dass eine Komplexitätsklasse durch vier Parameter spezifiziert ist: Berechnungsmodell, Berechnungsmodus, Berechnungsressource, Berechnungsschranke.

Sie können diese Parameter selber wählen, sofern sie für dieses Beispiel sinnvoll sind, aber die Sprache L muss in der dadurch definierten Klasse liegen.

Lösung: Z.B.

1. Berechnungsmodell: 1-Band-Turing-Maschine,
2. Berechnungsmodus: deterministisch,
3. Berechnungsressource: Zeit,
4. Berechnungsschranke: n^2

b) Geben Sie eine **Sprache** an, die in **P** liegt. **[1 Pkt]**

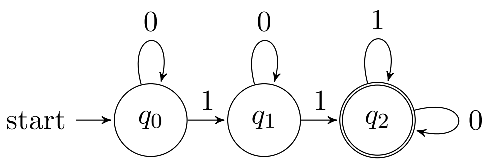
Lösung: Z.B. $L = \{10, 11\}$

c) Erklären Sie kurz, was ein **NP**-Problem charakterisiert. **[2 Pkte]**

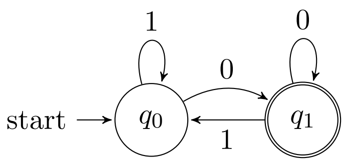
Lösung: Es ist kein polynomialer Algorithmus bekannt, der eine Lösung des Problems liefert, aber man kann eine Lösung in polynomialer Zeit überprüfen.

Aufgabe 7: Produkt-Automat

Gegeben seien die beiden endlichen Automaten A



und B:



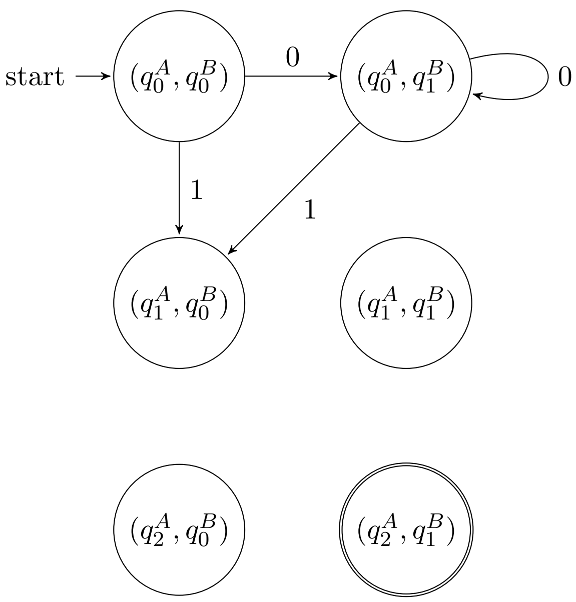
a) Bestimmen Sie die Sprache $L(A)$. **[2 Pkte]**

Lösung: $L(A) = \{\ 0^k 1 0^l 1 \omega \mid k, l \in \mathbb{N} \text{ und } \omega \in \Sigma^*\ \}$

b) Unten sehen Sie die Knoten des Graphen des Produktautomaten $A \times B$, der die Sprache $L(A) \cap L(B)$ akzeptiert.

Zeichnen Sie den Startknoten und die akzeptierenden Knoten ein. **[1 Pkt]**

Zeichnen Sie alle Pfeile ein die von den Knoten (q_0^A, q_0^B) und (q_0^A, q_1^B) abgehen. **[3 Pkte]**



Aufgabe 8: Diagonalisierung

Das Alphabet sei $\Sigma := \{0, 1\}$ Wir wollen zeigen, dass es Sprachen über Σ gibt, die nicht einmal rekursiv aufzählbar sind.

a) Geben Sie die Definition einer *rekursiv aufzählbaren* (semi-entscheidbaren) Sprache über Σ an. **[2 Pkte]**

Lösung: Eine Sprache L ist rekursiv aufzählbar, wenn eine deterministische Turing-Maschine existiert, die genau die Wörter der Sprache L akzeptiert.

b) Erklären Sie, warum die Menge der rekursiv aufzählbaren Sprachen über Σ abzählbar ist. **[3 Pkte]**

Lösung: Gemäss der Definition gibt es zu jeder rekursiv aufzählbaren Sprache mindestens eine Turing-Maschine, die genau die Wörter der Sprache L akzeptiert. Wir ordnen jeder rekursiv aufzählbaren Sprache die Turing-Maschine mit der kleinsten Gödelnummer zu, die L akzeptiert. Wir nummerieren die Gödelnummern, die Bild dieser Abildung sind nach ihrer Grösse durch. Damit haben wir eine Aufzählung der rekursiv aufzählbaren Sprachen konstruiert.

c) Die Menge aller Sprachen über Σ ist *überabzählbar*.

Zeigen Sie das mit dem **Diagnonalisierungsverfahren**. **[4 Pkte]**

Hinweis: Ordnen Sie die Wörter von Σ^*, wie wir das im Kurs gemacht haben und betrachten Sie die charakteristischen Funktionen der Sprachen (Teilmengen) von Σ^*.

Erklären Sie, was Sie tun!

Lösung: Wir nehmen an, dass die Menge aller Sprachen über Σ abzählbar ist, d.h. dass jede Sprache bzw. die entsprechende charakteristische Funktion eine Nummer trägt:

$$f_0, f_1, f_2 \cdots .$$

Wir werden zeigen, dass diese Annahme auf einen Widerspruch führt.

Wir schreiben eine unendliche Tabelle wie folgt an:

$L \setminus \Sigma^*$	ϵ	0	1	00	01	10	11	000	$\cdots$
f_0									
f_1									
f_2									
f_3									
f_4									
f_5									
$\vdots$									

Die Einträge der Tabelle sind jeweils eine Eins oder eine Null, je nach dem, ob das Wort in der Spalte in der Sprache, die in der Zeile steht, vorkommt oder nicht.

Wir konstruieren nun eine charakteristische Funktion f (eine Sprache), die in keiner Zeile vorkommt:

$$\begin{array}{ccc}
f(\epsilon) & = & \neg f_0(\epsilon) \\
f(0) & = & \neg f_1(0) \\
f(1) & = & \neg f_2(1) \\
f(00) & = & \neg f_3(00) \\
f(01) & = & \neg f_4(01) \\
\vdots & \vdots & \vdots
\end{array}$$

Damit haben wir gezeigt, dass wir bei unserer Aufzählung mindestens eine Funktion (Sprache) vergessen haben: Widerspruch.