Gelöste Aufgaben:

26	27	28	29	30

Name:

Matrikel-Nr.:

Aufgabe 26. Geben Sie informell eine Turingmaschine an, die das Minimum gegebener positiver natürlicher Zahlen $p_1, ..., p_n$, $(n > 0)$ berechnet. Dazu nehmen wir an, dass für jedes $1 \leq i \leq n$ die Zahl p_i auf dem Eingabeband durch p_i Einsen repräsentiert ist und die Eingabezahlen durch eine Null getrennt sind. Nach erfolgreicher Berechnung möge auf dem Ausgabeband $\min\{p_1, ..., p_n\}$ stehen. Hinweis: Sie dürfen mehrere Bänder verwenden.

Aufgabe 27. Vorbemerkung: Sei T eine Turingmaschine mit einem Eingabe- und einem Ausgabeband, die eine Sprache $G(T)$ über dem Alphabet $\Sigma = \{0, 1\}$ erzeugt. Nach Definition darf der L/S-Kopf des Ausgabebandes niemals nach links bewegt werden. Das Eingabeband kann für beliebige Rechnungen benutzt werden. Die *akzeptierte Sprache* ist für jede Turingmaschine definiert. Also kann man auch $L(T)$ betrachten.
Begründung Ihrer Antworten zu folgenden Fragen. Wir betrachten hier immer Sprachen über dem Alphabet $\Sigma = \{0, 1\}$.

1. Gibt es eine nicht-leere Sprache L und eine Turingmaschine M, so dass $L(M) = G(M)$?
2. Gibt es eine unendliche Sprache L und eine Turingmaschine M, so dass $L(M) = G(M)$?
3. Gibt es für jede endliche Sprache L eine Turingmaschine M, so dass $L(M) = G(M)$?

Aufgabe 28. Eine Turingmaschine wurde definiert als ein 6-tupel

$$M = (Q, \Sigma, \Gamma, q_0, F, \delta)$$

wobei $\Sigma \subseteq \Gamma$, $q_0 \in Q$, $F \subseteq Q$ und δ eine partielle Funktion $Q \times \Gamma \to Q \times \Gamma \times \{L, S, R\}$ ist. Ein Produkt zweier Turingmaschinen

$$M_1 = (Q_1, \Sigma_1, \Gamma_1, q_{1,0}, F_1, \delta_1)$$

und

$$M_2 = (Q_2, \Sigma_2, \Gamma_2, q_{2,0}, F_2, \delta_2)$$

ist ein 6-tupel

$$M_1 \times M_2 = (Q_1 \times Q_2, \Sigma_1 \times \Sigma_2, \Gamma_1 \times \Gamma_2, (q_{1,0}, q_{2,0}), F, \delta_1 \times \delta_2)$$

mit $F \subseteq Q_1 \times Q_2$. Für Paare $(q_1, \gamma_1) \in Q_1 \times \Gamma_1$ und $(q_2, \gamma_2) \in Q_2 \times \Gamma_2$ im Definitionsbereich von δ_1 bzw δ_2 mit $\delta_i(q_i, \gamma_i) = (p_i, g_i, m_i)$ $(i = 1, 2)$ ist

$$\delta_1 \times \delta_2(q_1, q_2, \gamma_1, \gamma_2) = (p_1, p_2, g_1, g_2, m_1, m_2).$$

$M_1 \times M_2$ hat eine natürliche Interpretation als Turingmaschine mit zwei Bändern. Im Falle, dass $\Sigma_1 = \Sigma_2$, kann durch passende Wahl der Menge F der zu

akzeptierenden Zustände (und eventuell durch Modifikation von M_1 und M_2), $M_1 \times M_2$ zum Erkennen boolescher Kombinationen von $L(M_1)$ und $L(M_2)$ verwendet werden.
In den folgenden Tabellen sind zwei Turingmaschinen M_1 und M_2 mit $\Sigma_1 = \Sigma_2 = \{0, 1\}$ gegeben.

M_1	0	1	X	Y	$\sqcup$
p_0	p_1XR	$-$	$-$	p_3YR	$-$
p_1	p_10R	p_2YL	$-$	p_1YR	$-$
p_2	p_20L	$-$	p_0XR	p_2YL	$-$
p_3	$-$	$-$	$-$	p_3YR	$p_4 \sqcup R$
p_4	$-$	$-$	$-$	$-$	$-$

M_2	0	1	$\sqcup$
q_0	q_10R	q_21R	$-$
q_1	$-$	q_21R	$q_3 \sqcup S$
q_2	q_10R	$-$	$q_3 \sqcup S$
q_3	$-$	$-$	$-$

Die Endzustände von M_1 bzw. M_2 sind p_4 bzw q_3.

1. Beschreiben Sie die Sprachen $L(M_1)$ und $L(M_2)$.
2. Wie ist F für $M_1 \times M_2$ festzulegen (und ggf. M_1 und M_2 zu modifizieren), so dass $L(M_1' \times M_2') = L(M_1) \cap L(M_2)$ gilt.
3. Wie ist F für $M_1 \times M_2$ festzulegen (und ggf. M_1 und M_2 zu modifizieren), so dass $L(M_1' \times M_2') = L(M_1) \cup L(M_2)$ gilt.

Aufgabe 29. Sei $M = (Q, \Sigma, \Gamma, q_0, F, \delta)$ eine Turing-Maschine. Eine Berechnung von M endet nicht notwendigerweise, wenn in der Folge der Konfigurationen ein Zustand $q \in F$ erreicht wird. Sei $A(M)$ die Menge der Wörter von Σ^* bei der die Berechnung von M wenigstens einmal einen Zustand aus F durchläuft. Ist $A(M)$ eine rekursiv aufzählbare Sprache? Falls Sie mit ja antworten, geben Sie eine Turingmaschine M' an, so dass $L(M') = A(M)$ ist. Falls Sie mit nein antworten, zeigen Sie, dass $A(M)$ von keiner Turingmaschine akzeptiert wird.

Aufgabe 30. Seien M_1, M_2 zwei Turingmaschinen und seien $L_1 = L(M_1)$ und $L_2 = L(M_2)$ die zugehörigen akzeptierten Sprachen. Man könnte versuchen, eine Prozedur (im intuitiven Sinn) zum Akzeptieren von $L_1 \cup L_2$ wie folgt zu beschreiben:

> Man gebe ein gegebenes Inputwort x zunächst als Input für M_1; falls x von M_1 akzeptiert wird, wird x durch die Prozedur akzeptiert. Anderenfalls gebe man x als Input für M_2 und akzeptiere x falls M_2 das Wort x akzeptiert.

1. Was stimmt an diesem Argument nicht?
2. Man gebe eine verfeinerte Prozedur an, welche $L_1 \cup L_2$ tatsächlich akzeptiert.
3. Man zeige, dass auch $L_1 \cap L_2$ von einer geeigneten Prozedur (im intuitiven Sinn) akzeptiert wird.

Anmerkung: Die Prozeduren aus den letzten beiden Punkten lassen sich ohne weiteres auf Turingmaschinen implementieren; $L_1 \cup L_2$ und $L_1 \cap L_2$ werden also ebenfalls von geeigneten Turingmaschinen akzeptiert.