

Aufgabe 1

a)

Wenn die Sprache $\mathcal{L}$ entscheidbar ist, existiert eine entsprechende Turing Maschine

$$M \text{ mit } \mathcal{L}(M) = \mathcal{L}$$

die als Entscheider auf jeder Eingabe hält, sich also nach endlicher Zeit im Zustand q_{accept} oder q_{reject} befindet.

Mit diesem Wissen können wir jetzt also eine Turing Maschine

$$\overline{M} \text{ mit } \mathcal{L}(\overline{M}) = \overline{\mathcal{L}}$$

wie folgt konstruieren:

$$\overline{M} = \{Q_{\overline{M}}, \Sigma_{\overline{M}}, \Gamma_{\overline{M}}, \delta_{\overline{M}}, q_{init}\}$$

mit

$$\begin{aligned} Q_{\overline{M}} &= Q_M \\ \Sigma_{\overline{M}} &= \Sigma_M \\ \Gamma_{\overline{M}} &= \Gamma_M \\ \delta_{\overline{M}} &= \delta_M \end{aligned}$$

In der Zustandsmenge $Q_{\overline{M}}$ müssen nun noch die Zustände q_{accept} und q_{reject} vertauscht werden. Somit akzeptiert die Maschine $\overline{M}$ die Sprache

$$\overline{\mathcal{L}} = \Sigma^* \setminus \mathcal{L}$$

b)

Teil I

Wenn die Sprache $\mathcal{L}$ semi-entscheidbar ist existiert eine entsprechende Turing Maschine M , mit

$$\mathcal{L}(M) = \mathcal{L}$$

Die Maschine M ist damit ein Semi-Entscheider, was bedeutet, dass sie auf akzeptierende Eingaben auf jeden Fall hält, der Zustand q_{accept} also bei allen akzeptierten Eingaben erreicht wird.

Wenn die Sprache $\mathcal{L}$ co-semi-entscheidbar ist existiert eine Sprache

$$\overline{\mathcal{L}} = \Sigma^* \setminus \mathcal{L}$$

und eine entsprechende Turing Maschine $\overline{M}$, mit

$$\mathcal{L}(\overline{M}) = \overline{\mathcal{L}}$$

als Semi-Entscheider.

Teil II

Den Entscheider, der die Sprache $\mathcal{L}$ akzeptiert wird wie folgt konstruiert:
Sei

$$M_{Entscheider} = \{Q, \Sigma, \Gamma, \delta, q_{init}\}$$

eine nicht-deterministische Entscheider Turing Maschine, mit

$$\begin{aligned} Q &= Q_M \cup Q_{\overline{M}} \cup \{q_{init}, q_{accept}, q_{reject}\} \\ \Sigma &= \Sigma \\ \Gamma &= \Gamma_M \cup \Gamma_{\overline{M}} \\ \delta &= \delta_M \cup \delta_{\overline{M}} \end{aligned}$$

$$\mathcal{L}(M_{Entscheider}) = \mathcal{L}$$

Die Transitionsrelation muss noch wie folgt angepasst werden:

$$\begin{aligned} (q_{init}, \gamma) &\xrightarrow{\delta} \{(q_{init_M}, \gamma, S), (q_{init_{\overline{M}}}, \gamma, S)\} \\ (q_{accept_M}, \gamma) &\xrightarrow{\delta} \{(q_{accept}, \gamma, S)\} \\ (q_{accept_{\overline{M}}}, \gamma) &\xrightarrow{\delta} \{(q_{reject}, \gamma, S)\} \end{aligned}$$

mit $\gamma \in \Gamma$. Die Anpassungen an der Relationstransition bedeuten, dass aus dem Startzustand q_{init} auf jeden Fall eine Transition zum ehemaligen Startzustand beider Ursprungsmaschinen M und $\overline{M}$ passiert. Der ehemalige akzeptierende Zustand von M geht auf jeden Fall in der jetzt akzeptierenden Zustand q_{accept} über und der ehemalige akzeptierende Zustand von $\overline{M}$ geht in den jetzigen zurückweisenden Zustand q_{reject} über.

Aus Teil I ist bekannt, dass bei gleichem Eingabewort eine der beiden Maschinen akzeptierend halten muss, wenn das Wort Teil der Sprache $\mathcal{L}$ ist hält die Maschine M , wenn das Eingabewort nicht Teil der Sprache $\mathcal{L}$ ist hält die Maschine $\overline{M}$. Genau den akzeptierenden Zustand von $\overline{M}$ ist jetzt unser abweisender Zustand, somit ist $M_{Entscheider}$ tatsächlich ein Entscheider für die Sprache $\mathcal{L}$.

c)

$$\begin{aligned}\mathcal{L}_{\text{entscheidbar}} &\Leftrightarrow \mathcal{L}_{\text{semi-entscheidbar}} \wedge \mathcal{L}_{\text{co-semi-entscheidbar}} \\ \mathcal{L}_{\text{semi-entscheidbar}} \wedge \mathcal{L}_{\text{co-semi-entscheidbar}} &\Rightarrow \mathcal{L}_{\text{entscheidbar}} \\ \mathcal{L}_{\text{entscheidbar}} &\Rightarrow \mathcal{L}_{\text{semi-entscheidbar}} \wedge \mathcal{L}_{\text{co-semi-entscheidbar}}\end{aligned}$$

$M_{\text{Entscheider}}$ mit $\mathcal{L}(M_{\text{Entscheider}}) = \mathcal{L}_{\text{entscheidbar}}$
 $M_{\text{Entscheider}}$ ist auch ein Semi-Entscheider fuer $\mathcal{L}$

Konstruiere einen Semi-Entscheider fuer $\bar{\mathcal{L}} = \Sigma^* \setminus \mathcal{L}$

Sei $\bar{M} = (Q, \Sigma, \Lambda, \delta, q_{\text{init}})$

$$Q = Q_M$$

$$\Sigma = \Sigma_M$$

$$\Lambda = \Lambda_M$$

$$\delta = \delta_M$$

Vertausche q_{accept} und q_{reject}

Aufgabe 2

a)

Da die beiden Sprachen $\mathcal{L}_1$ und $\mathcal{L}_2$ entscheidbar sind existieren auch die beiden entsprechenden Turing Maschinen

$$M_1 \text{ mit } \mathcal{L}(M_1) = \mathcal{L}_1 \qquad M_2 \text{ mit } \mathcal{L}(M_2) = \mathcal{L}_2$$

Zum Entscheiden der Sprache $\mathcal{L}_1 \cup \mathcal{L}_2$ konstruieren wir folgende nicht-deterministische Turing Maschine.

$$M = (Q, \Sigma, \Gamma, \delta, q_{init})$$

mit

$$\begin{aligned} Q &= Q_{M_1} \cup Q_{M_2} \cup \{q_{init}, q_{accept}, q_{reject}\} \\ \Sigma &= \Sigma_{M_1} \cup \Sigma_{M_2} \\ \Gamma &= \Gamma_{M_1} \cup \Gamma_{M_2} \end{aligned}$$

Die Transitionsrelation δ besteht aus den Transitionen aus δ_{M_1} und δ_{M_2} , und muss noch wie folgt erweitert werden.

$$\begin{aligned} (q_{init}, \gamma) &\xrightarrow{\delta} (q_{init_{M_1}}, \gamma, S) \\ (q_{init}, \gamma) &\xrightarrow{\delta} (q_{init_{M_2}}, \gamma, S) \\ (q_{accept_{M_1}}, \gamma) &\xrightarrow{\delta} (q_{accept}, \gamma, S) \\ (q_{accept_{M_2}}, \gamma) &\xrightarrow{\delta} (q_{accept}, \gamma, S) \\ (q_{reject_{M_1}}, \gamma) &\xrightarrow{\delta} (q_{reject}, \gamma, S) \\ (q_{reject_{M_2}}, \gamma) &\xrightarrow{\delta} (q_{reject}, \gamma, S) \end{aligned}$$

Wobei gilt $\gamma \in \Gamma$. Die Erweiterungen der Transitionrelation bedeuten, dass es aus dem Startzustand sowohl einen Übergang in den Startzustand von M_1 als auch eine Überleitung in den Startzustand von M_2 gibt. Des weiteren gehen die ehemaligen akzeptierenden Zustände von M_1 und M_2 bei jedem gelesenen Symbol in den Akzeptierenden Zustand q_{accept} , analog dazu die abweisenden Zustände.

b)

Da die beiden Sprachen $\mathcal{L}_1$ und $\mathcal{L}_2$ entscheidbar sind existieren auch die beiden entsprechenden Turing Maschinen

$$M_1 \text{ mit } \mathcal{L}(M_1) = \mathcal{L}_1 \quad M_2 \text{ mit } \mathcal{L}(M_2) = \mathcal{L}_2$$

Zum Entscheiden der Sprache $\mathcal{L}_1 \cap \mathcal{L}_2$ konstruieren wir folgende deterministische Turing Maschine.

$$M = (Q, \Sigma, \Gamma, \delta, q_{init})$$

mit

$$\begin{aligned} Q &= Q_{M_1} \cup Q_{M_2} \cup \{q_{accept}, q_{reject}\} \\ \Sigma &= \Sigma_{M_1} \cup \Sigma_{M_2} \\ \Gamma &= \Gamma_{M_1} \cup \Gamma_{M_2} \\ q_{init} &= q_{init_{M_1}} \end{aligned}$$

Die Transitionsfunktion δ besteht aus den Transitionen aus δ_{M_1} und δ_{M_2} , und muss noch wie folgt erweitert werden.

$$\begin{aligned} (q_{accept_{M_1}}, \sigma) &\rightarrow (q_{accept_{M_1}}, \sigma, L) \\ (q_{accept_{M_1}}, \$) &\rightarrow (q_{init_{M_2}}, \sigma, S) \\ (q_{accept_{M_2}}, \gamma) &\rightarrow (q_{accept}, \gamma, S) \\ (q_{reject_{M_1}}, \gamma) &\rightarrow (q_{reject}, \gamma, S) \\ (q_{reject_{M_2}}, \gamma) &\rightarrow (q_{reject}, \gamma, S) \end{aligned}$$

Wobei gilt $\gamma \in \Gamma, \sigma \in \Sigma$. Die Erweiterungen der Transitionsfunktion bedeuten, dass der Schreib-/Lesekopf im akzeptierenden Zustand von M_1 wieder auf den Startmarker des Bandes zurück gesetzt wird und anschließend der Zustand zum initialen Zustand von M_2 wechselt. Zudem wird der akzeptierende Zustand von M_2 und die nicht akzeptierenden Zustände wie bei Aufgabenteil a) in den neuen akzeptierenden bzw. nicht akzeptierenden Zustand überführt.

c)

Da die beiden Sprachen $\mathcal{L}_1$ und $\mathcal{L}_2$ entscheidbar sind existieren auch die beiden entsprechenden Turing Maschinen

$$M_1 \text{ mit } \mathcal{L}(M_1) = \mathcal{L}_1 \qquad M_2 \text{ mit } \mathcal{L}(M_2) = \mathcal{L}_2$$

Zum Entscheiden der Sprache $\mathcal{L}_1 \cdot \mathcal{L}_2$ konstruieren wir folgende deterministische Turing Maschine.

$$M = (Q, \Sigma, \Gamma, \delta, q_{init})$$

mit

$$Q = Q_{M_1} \cup Q_{M_2} \cup \{q_{accept}, q_{reject}\}$$

$$\Sigma = \Sigma_{M_1} \cup \Sigma_{M_2}$$

$$\Gamma = \Gamma_{M_1} \cup \Gamma_{M_2}$$

$$q_{init} = q_{init_{M_1}}$$

Die Transitionsfunktion δ besteht aus den Transitionen aus δ_{M_1} und δ_{M_2} , und muss noch wie folgt erweitert werden.

$$(q_{accept_{M_1}}, \gamma) \rightarrow (q_{init_{M_2}}, \gamma, S)$$

$$(q_{accept_{M_2}}, \gamma) \rightarrow (q_{accept}, \gamma, S)$$

$$(q_{reject_{M_1}}, \gamma) \rightarrow (q_{reject}, \gamma, S)$$

$$(q_{reject_{M_2}}, \gamma) \rightarrow (q_{reject}, \gamma, S)$$

Wobei gilt $\gamma \in \Gamma$. Die Erweiterungen der Transitionsfunktion bedeuten, dass nach dem Durchlaufen des Machinenteils M_1 direkt der Maschinenteil aus M_2 gestartet wird. Der Zustand $q_{accept_{M_2}}$ wird in jedem Falle in den Zustand q_{accept} überführt und die beiden abweisenden Zustände der Ursprungsmaschinen $q_{reject_{M_1}}$ und $q_{reject_{M_2}}$ werden in jedem Falle in den Zustand q_{reject} überführt.

Aufgabe 3

a)

Die Turing Maschine, die einen **binären Inkrement** implementiert ist wie folgt definiert:

Sei

$$M = (\{q_{count}, q_{copy}, q_{halt}\}, \{0, 1\}, \{0, 1, \$, _ \}, \delta, q_{count})$$

eine 2-Band Berechnungs Turing Maschine, mit:

$\delta :$

$$\begin{aligned} (q_{count}, \$) &\rightarrow (q_{count}, \$, \$), (R, R) \\ (q_{count}, _) &\rightarrow (q_{halt}, _, 1), (S, S) \\ (q_{count}, 1) &\rightarrow (q_{count}, 1, 0), (R, R) \\ (q_{count}, 0) &\rightarrow (q_{copy}, 0, 1), (R, R) \\ (q_{copy}, _) &\rightarrow (q_{halt}, _, _), (S, S) \\ (q_{copy}, 1) &\rightarrow (q_{copy}, 1, 1), (R, R) \\ (q_{copy}, 0) &\rightarrow (q_{copy}, 0, 0), (R, R) \end{aligned}$$

Alle hier nicht definierten Transitionen ändern nichts am Bandinhalt und bleiben stehen.

Funktionsweise:

Die hier angegebene 2-Band Turing Maschine M liest nur von ersten Band und schreibt nur auf das letzte Band, somit ist sie eine berechnende Turing Maschine.

Im initialen Zustand q_{count} führt M die eigentliche Berechnung durch und hält hier insbesondere die Information, dass noch ein Übertrag beachtet werden muss. Im Falle der binären inkrementation ist es so, dass sobald eine 0 gegen eine 1 getauscht wurde, der Übertrag aufgelöst wird und der Rest der Zahl nur noch kopiert werden muss, genau dies tut der Zustand q_{copy} . Sieht q_{copy} das Ende des Eingabewortes wechselt M in den Zustand q_{halt} und die Berechnung ist beendet. Sieht hingegen q_{count} das Ende des Eingabewortes und ist somit noch ein übertrag ausstehend, so wird das Ausgabewort um eine 1 verlängert und auch hier wechselt M in den Zustand q_{halt} und die Berechnung ist beendet.

b)

Sei

$$M = (\{q_0, q_{count}, q_{back}, q_{halt}\}, \{1\}, \{0, 1, \$, _ \}, \delta, q_0)$$

eine 3 Band Berechnungs Turing Maschine, mit:

$\delta :$

$$\begin{aligned} (q_0, (\$, \$)) &\rightarrow (q_0, (\$, \$, \$), (R, S, S)) \\ (q_0, (1, \$)) &\rightarrow (q_{count}, (1, \$, \$), (R, R, S)) \\ (q_0, (_, \$)) &\rightarrow (q_{copy}, (_, \$, \$), (S, S, S)) \\ (q_{count}, (\gamma, _)) &\rightarrow (q_{back}, (\gamma, 1, \$), (S, L, S)) \gamma \in \Gamma \\ (q_{count}, (\gamma, 0)) &\rightarrow (q_{back}, (\gamma, 1, \$), (S, L, S)) \gamma \in \Gamma \\ (q_{count}, (\gamma, 1)) &\rightarrow (q_{count}, (\gamma, 0, \$), (S, R, S)) \gamma \in \Gamma \\ (q_{back}, (\gamma_1, \gamma_2)) &\rightarrow (q_{back}, (\gamma_1, \gamma_2, \$), (S, L, S)) \gamma_1 \in \Gamma, \gamma_2 \in \{0, 1\} \\ (q_{back}, (\gamma, \$)) &\rightarrow (q_0, (\gamma, \$, \$), (S, S, S)) \gamma \in \Gamma \\ (q_{copy}, (_, \$)) &\rightarrow (q_{copy}, (_, \$, \$), (S, R, R)) \\ (q_{copy}, (_, 0)) &\rightarrow (q_{copy}, (_, 0, 0), (S, R, R)) \\ (q_{copy}, (_, 1)) &\rightarrow (q_{copy}, (_, 1, 1), (S, R, R)) \\ (q_{copy}, (_, _)) &\rightarrow (q_{halt}, (_, _, _), (S, S, S)) \end{aligned}$$

Alle hier nicht definierten Transitionen ändern nichts am Bandinhalt und bleiben stehen.

Funktionsweise:

Die hier angegebene 3-Band Turing Maschine M hat ein read-only erstes band und ein write-only letztes band, somit ist sie eine berechnende Turing Maschine.

Im initialen Zustand q_0 wird durch das erste Band iteriert, falls eine 1 gelesen wurde wird auf dem zweiten Band die lsb-Darstellung einer binären Zahl inkrementiert (q_{count}). Falls ein $_$ gelesen wird kopiert M das zweite Band auf das letzte Band (q_{copy}).

Die Inkrementation funktioniert hier, wie in Aufgabenteil a), lediglich der kopierende Zustand für den gleichbleibenden Rest des Wortes wurde eingespart, und statt dessen ein Zustand q_{back} eingeführt, der den Kopf auf den zweiten Band wieder zurück auf den Startmarker $\$$ setzt.

Aufgabe 4

a)

Die akzeptierte Sprache $\mathcal{L}(M)$ ist

$$\mathcal{L}(M) = \{\mathcal{P}(\{a, b\}^*) \mid |a| = |b|\} \cup \{\emptyset\}$$

Die Turing Maschine akzeptiert also die Eingabe eines beliebigen Wortes aus $\{a, b\}^*$ mit gleicher Anzahl a und b im Wort und das leere Wort.

Funktionsweise:

Im Startzustand (q_{init}) schreibt die Maschine entweder beliebig viele a auf das zweite Band und genau so viele b auf das dritte Band oder springt in den Zustand q_{run} . Beim ersten Eintritt in den Zustand q_{run} steht der Lese-/Schreibkopf des ersten Bandes auf dem Ersten Buchstaben des Eingabewortes und die Lese-/Schreibköpfe des zweiten und dritten Bandes auf dem letzten Buchstaben des jeweiligen Bandes.

im Zustand (q_{run}) iteriert die Maschine jetzt durch das Eingabewort und setzt für jedes gelesene a den Lese-/Schreibkopf des zweiten Bandes und für jedes gelesene b den Lese-/undSchreibkopf des dritten Bandes um eine Position nach links. Wenn im ersten Band nun ein $\sqcup$ gelesen wird geht die Maschine in den Zustand q_{accept} , wenn auf den beiden anderen Bändern jeweils $\$$ gelesen wird. Wird auf den beiden anderen Bändern etwas anderes gelesen, wechselt die Maschine in den Zustand q_{reject} .

b)

c)