# Theoretische Informatik Serie 8

Benjamin Simmonds Dario Näpfer Fabian Bösiger

## Aufgabe 22

### (a)

Wir beschreiben mit $C_1, C_2, ..., C_t$ die $t$ Konfiguration, die $M$ bei der Berechnung von $y$ in den $t$ Schritten hat. Offensichtlich sind diese Zustände immer die selben auf der selben Eingabe $\lambda$. Sei zudem $q_i$ der Zustand in der Konfiguration $C_i$ und $b_i$ der Buchstabe, der $M$ im $i$-ten Zeitschritt auf das Band schreibt für $i \in \{1, ..., t\}$.

Wir definieren die Turingmaschine $M'$ wie folgt. Die Turingmaschine $M'$ besteht aus den Zuständen $q_i$ und durchläuft bei der Berechnung von $y$ die Konfigurationen $C_i$. Das Alphabet $\Gamma_{M'}$ muss die maximal $t$ Buchstaben $b_i$ enthalten, sowie die Zeichen "¢" und "␣". Es gilt somit $\|\Gamma_{M'}\| \leq t + 2$. Da $M'$ die exakt selbe Berechnung wie $M$ macht, gilt, dass $M'$ auf $\lambda$ ebenfalls $y$ berechnet

### (b)

Offensichtlich existiert eine Turingmaschine $B$ mit $Kod(B) \in L_{träge}$, welche auf der Eingabe $\lambda$ in $t = |y|$ Zeitschritten $y$ auf das Band schreiben kann. Es gilt, dass $t = min\{Time_{M'}(\lambda) \mid M' \in L_{comp,\lambda} \text{ und } M'(\lambda) = B(\lambda)\} = Time_B(\lambda)$.

Wir beschreiben eine Turingmaschine $A$, die $L_{träge}$ erkennt. Diese arbeitet folgendermassen:

1. $A$ überprüft, ob die Eingabe eine Kodierung einer Turingmaschine $Kod(M)$ ist.
2. $A$ simuliert $M$ auf dem leeren Wort $\lambda$, dabei misst $A$ die Zeit $t_M$, die $M$ benötigt, um die Ausgabe zu schreiben.
3. Falls $M$ auf einem akzeptierenden Zustand terminiert, überprüft $A$, ob auf dem Band $y$ steht und wir fahren im nächsten Schritt fort. Wenn dies nicht der Fall ist, verwirft $A$.
4. Ansonsten überprüft $A$, ob $t_M \geq 2t$ ist. Falls ja, akzeptiert $A$, sonst verwirft $A$.

Für jedes Wort $w \in L_{träge}$ mit $Kod(C) = w$ und $Time_C(\lambda) \geq 2t$ gilt, dass $C$ auf $\lambda$ hält und $y$ auf das Band schreibt. Per Definition von $A$ überprüft $A$ als Nächstes, ob $Time_C(\lambda) \geq 2t$ ist, was offensichtlich der Fall ist. Somit akzeptiert $A$ alle $w \in L_{träge}$.

Für jedes Wort $w \notin L_{träge}$ gilt entweder $Kod(C) = w$ und $C$ schreibt nicht $y$ auf das Band oder $Time_C(\lambda) < 2t$, oder $Kod(C) \neq w$. Im ersten Fall hält $A$ entweder nicht oder verwirft, im zweiten Fall verwirft $A$ ebenfalls. Somit aktepiert $A$ nie bei allen $w \notin L_{träge}$.

Somit existiert eine Turingmaschine $A$ mit $L(A) = L_{träge}$ und es gilt $L_{träge} \in \mathcal{L}_{RE}$.

### (c)

Wir wissen, dass $L_U \notin \mathcal{L}_R$, und zeigen $L_{comp,\lambda} \leq_R L_U$. Sei $A$ eine TM, die $L_U$ entscheidet und somit immer hält. Wir bauen eine TM $B$, die $A$ als Teilprogramm enthält, und $L_{comp,\lambda}$ entscheidet. Für jedes Wort $w \in \{0,1\}^*$ konkatenieren wir die Eingabe $w$ zu $w' = w\#\lambda$. $A$ erhält $w'$ als Eingabe. Falls $A$ die Eingabe nicht akzeptiert, verwirft $B$. Sonst überprüfen wir, ob die Ausgabe $A(w') = y$. Falls dies der Fall ist, akzeptieren wir die Eingabe, sonst verwerfen wir. Es gilt, dass $L(B) = L_{comp,\lambda}$ und $B$ hält immer, da $A$ immer hält.

Formalismus: Sei $x \in L_{comp,\lambda}$. Also ist $x = Kod(M)$ für eine TM $M$, die $\lambda$ akzeptiert, und für Eingabe $\lambda$ $y$ auf das Band schreibt. A erhält die Eingabe $x' = Kod(M)\#\lambda$. Offensichtlich gilt, dass $x' \in L_U$. Ausserdem gilt per Definition, dass $y$ auf dem Band steht und $B$ somit immer akzeptiert.

Sei $x \notin L_{comp,\lambda}$. Somit gilt entweder, dass $x \neq Kod(M)$. Dann verwirft $A$ die Eingabe $x'$ und somit auch $B$. Oder es gilt, dass $x = Kod(M)$, aber entweder $M$ auf $\lambda$ nicht hält oder die Ausgabe nicht $y$ entspricht. Im ersten Fall verwirft $A$ und somit auch $B$. Im zweiten Fall verwirft $B$ bei der Überprüfung der Ausgabe von $A$.

### (d)

Aus der Teilaufgabe (c) wissen wir, dass $L_{comp,\lambda} \notin \mathcal{L}_R$. Wir zeigen, dass $L_{comp,\lambda} \leq_R L_{träge}$.

Sei $A$ eine TM, die $L_{träge}$ entscheidet. Wir bauen eine TM $B$, die $A$ als Teilprogramm enthält, und $L_{comp,\lambda}$ entscheidet. Jede Eingabe $w \in \{0,1\}^*$ in $B$ übernehmen wir als Eingabe für $A$. Wenn $A$ die Eingabe akzeptiert, akzeptiert auch $B$. Sonst überprüfen wir, ob $w = Kod(M)$. Wenn das nicht der Fall ist, verwirft $B$ die Eingabe. Sonst simuliert Teilprogramm $C$ $M$ auf $\lambda$ für $2t-1$ Zeitschritte, wobei wir annehmen, dass $t$ gegeben ist und $t = min\{Time_{M'}(\lambda) \mid M' \in L_{comp,\lambda} \text{ und } M'(\lambda) = y\}$. Falls M in dieser Zeit akzeptiert, überprüfen wir, ob die Ausgabe $y$ entspricht. Wenn das der Fall ist, akzeptiert $B$, sonst verwirft $B$. Es gilt, dass $L(B) = L_{comp,\lambda}$ und $B$ hält immer, da $A$ immer hält.

Formalismus: Sei $x \in L_{comp,\lambda}$. Es gilt, dass $x = Kod(M)$ und $M$ liefert $y$ als Ausgabe für Eingabe $\lambda$. Falls $Time_M(\lambda) \geq 2t$, akzeptiert $A$ und somit auch $B$. Sonst akzeptiert $A$ nicht und $C$ simuliert $M$ auf $\lambda$ für $2t-1$ Schritte. Da $x \in L_{comp,\lambda}$ und $Time_M(\lambda) < 2t$, muss $M$ in dieser Zeit $y$ auf das Band schreiben und somit akzeptiert $B$.

Sei $x \notin L_{comp,\lambda}$. Es gilt per Definition von $L_{träge}$, dass $x \notin L_{träge}$. Somit verwirft $A$ immer und dehalb verwirft auch $B$.

## Aufgabe 23