

Aufgabe 1

a)

Wenn gilt

$$w \in \overline{\mathcal{L}_{\text{Universality}}}$$

dann gilt auch

$$\forall x \in \{0, 1\}^* | x \notin \mathcal{L}(M_w)$$

Es ist nicht semi-entscheidbar, ob ein Wort nicht in $\mathcal{L}(M_w)$ liegt, somit ist $\overline{\mathcal{L}_{\text{Universality}}}$ auch nicht semi-entscheidbar. Somit ist $\mathcal{L}_{\text{Universality}}$ nicht co-semi-entscheidbar.

b)

Die Sprache $\mathcal{L}_{\text{Totality}}$ ist wie folgt definiert:

$$\mathcal{L}_{\text{Totality}} = \{w \in \{0, 1\}^* | M_w \text{ hält auf alle Eingaben } x \in \{0, 1\}^*\}$$

Da die semi-entscheidbaren Sprachen keine eindeutige Zuordnung als Nein-Instanz zulassen, da eine Berechnung ja in der Theorie ewig laufen kann ohne zu halten, müsste der Beweis auf jeden Fall konkretisiert werden. Zudem fehlt eine klare formale Ausführung der zugrundeliegenden Reduktion.

Aufgabe 2

a)

Zuerst prüfen wir, ob es sich bei der Eingabe, um eine syntaktisch richtige Kodierung einer Konfigurationsmenge handelt, wenn nicht, dann weißt die Maschine ab. In einer Konfiguration ist der aktuelle Zustand, der aktuelle Bandinhalt, sowie die aktuelle Kopfposition gespeichert. Also müssen wir zuerst prüfen, ob der Bandinhalt des Eingabebandes in der Konfiguration c_0 dem leeren Wort ε entspricht. Ist dies nicht der Fall weißt die Maschine ab. Nun nehmen wir an, dass der abweisende und der akzeptierende Zustand in der Kodierung der Konfigurationen unterschiedlich aber immer gleich Benannt sind, somit bei jeder gegebenen Kodierung klar ist, wann es sich um einen

akzeptierenden Zustand handelt und wann um einen abweisenden. Mit dieser Annahme als Grundlage springen wir zur letzten Konfiguration c_k und prüfen ob der dort kodierte Zustand der akzeptierende Zustand ist, ist dies der Fall akzeptiert die Maschine, sonst weiß sie ab.

b)

Eine Turing-Maschine, die kein Entscheider ist (was hier zumindest nicht explizit definiert ist) kann eine Eingabe nach endlicher Zeit akzeptieren, nach endlicher Zeit abweisen oder unendlich lange rechnen.

Eben weil die Möglichkeit besteht, dass die Maschine unendlich lange rechnet können wir zu keinen Zeitpunkt sicher sagen, ob die Eingabe nun AKzeptiert wird oder nicht.

Somit ist M nicht semi-entscheidbar.

c)

Aufgabe 3

a)

$$\begin{aligned}
 & \{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_{\mathcal{O}} \in \mathbb{R}_{>0} : \forall n \geq n_0 : f(n) \leq c_{\mathcal{O}}g(n)\} \cap \{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_{\Omega} \in \mathbb{R}_{>0} : \forall n \geq n_0 : c_{\Omega}g(n) \leq f(n)\} \\
 &= \\
 & \{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_{\mathcal{O}}, c_{\Omega} \in \mathbb{R}_{>0} : \forall n \geq n_0 : c_{\Omega}g(n) \leq f(n) \wedge f(n) \leq c_{\mathcal{O}}g(n)\} \\
 &= \\
 & \{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_{\mathcal{O}}, c_{\Omega} \in \mathbb{R}_{>0} : \forall n \geq n_0 : c_{\Omega}g(n) \leq f(n) \leq c_{\mathcal{O}}g(n)\} \\
 &= \\
 & \Theta
 \end{aligned}$$

b)

“ $\Rightarrow$ ”

$$\{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_0, c_1 \in \mathbb{R}_{>0} : \forall n \geq n_0 : c_0g(n) \leq f(n) \leq c_1g(n)\}$$

Wenn die Funktion $g(n)$ mittels der Multiplikation so verändert werden kann, dass sie größer bzw. kleiner als $f(n)$ wird, dann können wir die Funktion $f(n)$ mittels des Kehrwertes von c_0 und c_1 auch so manipulieren, dass sie kleiner bzw. größer als $g(n)$ ist.

$$\{f : \mathbb{N} \rightarrow \mathbb{N} \mid \exists n_0 \in \mathbb{N}, c_0, c_1 \in \mathbb{R}_{>0} : \forall n \geq n_0 : (1/c_1)f(n) \leq g(n) \leq (1/c_0)f(n)\}$$

“ $\Leftarrow$ ” Analog dazu.

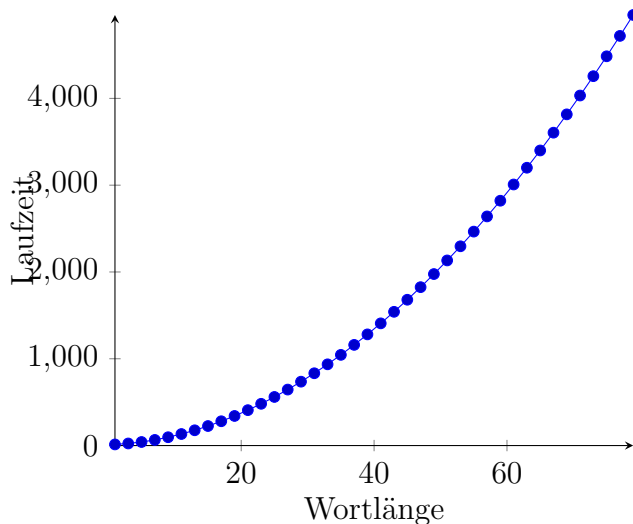
c)

$$f(x) = 3^x$$

$$g(x) = 2^{2^x}$$

Aufgabe 4

a)



Quadratische Regression:

$$f(x) = 5.571968940879403 + 3.4122384996261763 * x + 0.7508970891111456 * x^2$$

b)

Wir kopieren den ersten Teil des Wortes, also bis # auf das 2. Band, überspringen das # und vergleichen beim iterieren durch den 2. Wortteil das erst und das 2. Band.

c)