Universität Siegen
Lehrstuhl Theoretische Informatik
Markus Lohrey

Berechenbarkeit und Logik
WS 2025/2026

Übungsblatt 6

Aufgabe 1. Welche der Eigenschaften sind entscheidbar, welche sind unentscheidbar? Geben Sie außerdem an, in welchen Fällen der Satz von Rice anwendbar ist.

(a) Eine gegebene Turingmaschine M terminiert immer.

(b) Eine gegebene Turingmaschine M terminiert immer in höchstens 13 Schritten.

(c) Eine gegebene Turingmaschine M terminiert für alle Eingaben der Länge 17.

(d) Eine gegebene Turingmaschine M gibt auf gegebener Eingabe w das Wort *chic* aus.

(e) Eine gegebene Turingmaschine M gibt auf keiner Eingabe das Wort *la-di-da* aus.

(f) Eine gegebene Turingmaschine M verwendet auf gegebener Eingabe w während der Berechnung höchstens $19 \cdot |w| + 23$ Bandzellen. (Hierbei gelte eine Bandzelle als verwendet, sobald der Schreib-Lese-Kopf der Maschine diese erreicht.)

Lösung.

(a) **Unentscheidbar.** Wir können hier den Satz von Rice anwenden, denn eine Turingmaschine M terminiert genau dann immer (d.h. auf jeder Eingabe), wenn die Funktion f_M die durch M berechnet wird eine totale Funktion ist.

(b) **Entscheidbar.** Um die Eigenschaft zu entscheiden genügt es, M auf allen Eingaben der Länge höchstens 13 für bis zu 13 Berechnungsschritte zu simulieren. Für längere Eingaben kann die Maschine die zusätzlichen Zeichen nämlich nur dann lesen, wenn sie innerhalb der ersten 13 Berechnungsschritte noch nicht terminiert.

Da diese Eigenschaft entscheidbar ist, ist der Satz von Rice nicht anwendbar.

(c) **Unentscheidbar.** Wir können wieder den Satz von Rice anwenden, denn M terminiert genau dann auf allen Eingben der Länge 17, wenn die Funktion f_M die durch M berechnet wird auf allen solchen Eingaben definiert ist.

(d) **Unentscheidbar.** Hier können wir den Satz von Rice *nicht* anwenden, da das Problem nicht nur von der Turingmaschine M (genauer, der Funktion f_M) abhängt sonder auch von der Eingabe w, denn es gibt eine Maschine M, die nur auf einer festen Eingabe (etwa $w = \varepsilon$) das Wort *chic* ausgibt. Dennoch ist das Problem unentscheidbar, wass sich beispielsweise durch eine Reduktion von H_0 auf das vorliegende Problem zeigen lässt: Der Kodierung u einer Turingmaschine M_u können wir die Turingmaschine M sowie die Eingabe $w = \varepsilon$ zuordnen, wobei M zunächst M_u auf ε simuliert und dann das Wort *chic* ausgibt. Dann haben M und w die zu prüfende Eigenschaft genau dann, wenn M_u auf ε hält, also wenn $u \in H_0$ gilt.

(e) **Unentscheidbar.** Die zu prüfende Eigenschaft liegt genau dann vor, wenn das Wort *la-di-da* nicht im Bildbereich der Funktion f_M liegt. Somit ist das Problem unentscheidbar nach dem Satz von Rice.

(f) **Entscheidbar.** Dass der Satz von Rice hier nicht anwendbar ist, lässt sich leicht wie in Teil (d) einsehen. Tatsächlich ist das Problem sogar entscheidbar:

Wir nennen eine Bandzelle *potenziell erreichbar*, wenn sie höchstens $19 \cdot |w| + 22$ Zellen von der Startzelle entfernt ist. Besitzt das Paar (M, w) die zu prüfende Eigenschaft, dann treten bei der Berechnung von M auf Eingabe w nur solche Konfigurationen auf, bei denen der Schreib-Lese-Kopf stets auf eine potenziell erreichbare Zelle gerichtet ist und bei denen diejenigen Zellen, die nicht potenziell erreichbar sind, stets das Blank-Symbol enthalten. Abhängig von M und w gibt es aber nur endlich viele solche Konfigurationen und die zu prüfende Eigenschaft kann auf dem entsprechend eingeschränkten Konfigurationsgraphen entscheiden werden.

Aufgabe 2. Zeigen Sie, dass das nachfolgende Problem entscheidbar ist. Reduzieren Sie das Problem dazu auf das Schnittproblem für reguläre Sprachen.

Auf Eingabe von Wortpaaren $(x_1, y_1), \ldots, (x_k, y_k)$ *soll entschieden werden, ob Indizes* $i_1, \ldots, i_m$ *und* $j_1, \ldots, j_n$ *mit* $m, n \geq 1$ *existieren, so dass* $x_{i_1} \cdots x_{i_m} = y_{j_1} \cdots y_{j_n}$.

Lösung. Wir weisen der Eingabeinstanz die beiden regulären Ausdrücke

$$\xi = (x_1 \mid \cdots \mid x_k)^+ \text{ und } \eta = (y_1 \mid \cdots \mid y_k)^+$$

zu. (Hierbei haben wir die Kurzschreibweise $\alpha^+ := \alpha\alpha^*$ verwendet.) Ein Wort w ist genau dann in $L(\xi)$, wenn es sich als $w = x_{i_1} \cdots x_{i_m}$ für Indizes $i_1, \ldots, i_m$ mit $m \geq 1$ schreiben lässt. Analog gilt $w \in L(\eta)$ genau dann, wenn $w = y_{j_1} \cdots y_{j_n}$ für geeignete Indizes $j_1, \ldots, j_n$ mit $n \geq 1$. Also ist $L(\xi) \cap L(\eta) \neq \emptyset$ genau dann, wenn es sich um eine positive Instanz handelt. Die Zuordnung $((x_1, y_1), \ldots, (x_k, y_k)) \mapsto (\xi, \eta)$ definiert eine Reduktion von obigem Problem auf das Schnittproblem für reguläre Sprachen (welche in diesem Fall durch reguläre Ausdrücke gegeben sind). Da letzteres entscheidbar ist, ist auch das Problem der Aufgabe entscheidbar.

Notation. Für das *Postsche Korrespondenzproblem* (PCP) geben wir im Folgenden die Wortpaare, die als Eingabe des Problems auftreten, in Matrixschreibweise an. So stellt

$$\begin{pmatrix} ab & b & aaa & b \\ ba & bb & b & aba \end{pmatrix}$$

beispielsweise die PCP-Instanz $((ab, ba), (b, bb), (aaa, b), (b, aba))$ dar.

Aufgabe 3. Entscheiden Sie die folgenden PCP-Instanzen. Geben Sie im positiven Fall eine Lösung an und beweisen Sie im negativen Fall, dass keine solche existiert.

(a) $\begin{pmatrix} aaaa & aa \\ aaa & aaaa \end{pmatrix}$ (b) $\begin{pmatrix} ab & bb & aa \\ aba & abb & bab \end{pmatrix}$ (c) $\begin{pmatrix} a & ba & abb & bab \\ ab & ab & bb & abb \end{pmatrix}$

Lösung.

(a) Eine Lösung der PCP-Instanz ist etwa gegeben durch

$$(aaaa)(aaaa)(aa) = (aaa)(aaa)(aaaa).$$

(b) Die PCP-Instanz besitzt keine Lösung: Für jedes Wortpaar (x_i, y_i) gilt $|x_i| < |y_i|$ und somit gilt $|x_{i_1} \cdots x_{i_n}| < |y_{i_1} \cdots y_{i_n}|$ für jede Indexfolge $i_1, \ldots, i_n$ mit $n \geq 1$.

(c) Die PCP-Instanz besitzt keine Lösung: Eine solche Lösung müsste mit dem Wortpaar $(x_1, y_1) = (a, ab)$ beginnen, da sich alle anderen Wortpaare bereits im Anfangsbuchstaben unterscheiden. Es gilt aber $|x_i|_b \leq |y_i|_b$ für alle Wortpaare (x_i, y_i), sowie $|x_1|_b < |y_1|_b$. Daraus schließen wir $|x_{i_1} \cdots x_{i_n}|_b < |y_{i_1} \cdots y_{i_n}|_b$ für alle noch relevanten Indexfolgen $i_1, \ldots, i_n$, d.h. für solche mit $n \geq 1$ und $i_1 = 1$.

Aufgabe 4. Zeigen Sie, dass PCP eingeschränkt auf unäre Alphabete entscheidbar ist. Gilt eine entsprechende Aussage auch für die Einschränkung auf binäre Alphabete?

Lösung. Gilt $|x| < |y|$ für alle Wortpaare (x, y) einer PCP-Instanz, dann besitzt diese keine Lösung. Genau so verhält es sich auch, wenn $|x| > |y|$ für alle Wortpaare (x, y) gilt. Wir wollen im Folgenden zeigen, dass alle *anderen* PCP-Instanzen über einem unären Alphabet lösbar sind. Da das obige Kriterium (sehr leicht) algorithimisch überprüfbar ist, folgt daraus, dass PCP für unäre Alphabete entscheidbar ist.

Besitzt eine PCP-Instanz über einem unären Alphabet ein Wortpaar (x, y) mit $|x| = |y|$, dann ist diese durch $x = y$ lösbar. Eine PCP-Instanz, die weder diese, noch obige Bedingungen erfüllt, muss sowohl ein Wortpaar (x_1, y_1) mit $|x_1| < |y_1|$, als auch ein Wortpaar (x_2, y_2) mit $|x_2| > |y_2|$ besitzen. Eine solche Instanz über einem unären Alphabet hat dann etwa die Lösung

$$(x_1)^{n_2}(x_2)^{n_1} = (y_1)^{n_2}(y_2)^{n_1}$$

mit $n_1 = |y_1| - |x_1|$ und $n_2 = |x_2| - |y_2|$, denn beiden Seiten haben die gleiche Länge:

$$|x_1| \cdot n_2 + |x_2| \cdot n_1 = |x_2||y_1| - |x_1||y_2| = |y_1| \cdot n_2 + |y_2| \cdot n_1.$$

Eine entsprechende Aussage für die Einschränkung von PCP auf binäre Alphabete, hier mit PCP_2 bezeichnet, ist falsch. Wir können nämlich jeder PCP-Instanz I über einem beliebigen Alphabet $\Sigma_I = \{a_1, \ldots, a_n\}$ eine lösbarkeitsäquivalente PCP-Instanz über dem Alphabet $\{0, 1\}$ zuordnen (etwa durch den injektiven Homomorphismus $h\colon \Sigma_I^* \to \{0,1\}^*$, der durch $h(a_i) = 0^{i-1}10^{n-i}$ für alle $1 \leq i \leq n$ definiert ist). Es gilt also $\mathrm{PCP} \leq \mathrm{PCP}_2$ und hieraus folgt dann, da PCP unentscheidbar ist, dass auch PCP_2 unentscheidbar ist.