

Berechenbarkeit und Komplexität

Beispielklausur

Wolfgang Schreiner
Wolfgang.Schreiner@risc.uni-linz.ac.at

Nachname:

Vorname:

Matrikelnummer:

Studienkennzahl:

Gesamt: 100 Punkte.

≥ 51 Punkte: GEN4

≥ 64 Punkte: BEF3

≥ 77 Punkte: GUT2

≥ 89 Punkte: SGT1

1. (20P) Welche der folgenden Sprachen über dem Alphabet $\{0, 1\}$ sind regulär? Begründen Sie Ihre Antwort.

(a) (5P) $L_a := \{0^m 1^n : m, n \in \mathbb{N} \wedge 2|m \wedge 3|n\}$

(b) (5P) $L_b := \{0^m 1^n : m, n \in \mathbb{N} \wedge m|n\}$

(c) (5P) $L_c := \{0^m 1^n : m, n \in \mathbb{N} \wedge m|n \wedge n < 1000\}$

(d) (5P) $L_d := \{0^m 1^n 0^p : m, n, p \in \mathbb{N} \wedge (m+n)|p \wedge p < 1000\}$

2. (20P) Sei L die Menge aller Wörter über dem Alphabet $\{0, 1\}$, in denen die Folge “110” mindestens zweimal, nämlich am Anfang und am Ende des Wortes, vorkommt.

Definieren Sie einen regulären Ausdruck und einen nichtdeterministischen endlichen Automaten

$$M = (Q, \Sigma, \delta, S, F)$$

mit Sprache L (jedes Bestimmungsstück von M einzeln, δ als Tabelle und durch einen Zustandsgraphen).

Kann man auch einen deterministischen Automaten mit Sprache L konstruieren? Warum (nicht)?

Kann man auch eine Turing-Maschine mit Sprache L konstruieren? Warum (nicht)?

3. (20P) Geben Sie für jede der folgenden Sprachen über dem Alphabet $\{0, 1\}$ an ob sie rekursiv aufzählbar und ob sie rekursiv ist. Begründen Sie ihre Antwort:

(a) (5P) $L_1 := \{w : \text{die Anzahl der 1en in } w \text{ ist eine Primzahl}\}$

(b) (5P) $L_2 := \{0^n 1^m 0^{(n^m)} : n, m \in \mathbb{N}\}$

(c) (5P) $L_3 := \{\langle M \rangle : 10101 \in L(M)\}$

(d) (5P) $L_4 := \{\langle M \rangle : 10101 \notin L(M)\}$

($M \dots$ eine Turing-Maschine, $\langle M \rangle \dots$ die Codierung von M).

4. (20P) Welche der folgenden Aussagen sind richtig? Begründen Sie Ihre Antwort ausführlich (Beweis anhand der Definition der \mathcal{O} -Notation).

(a) (8P) $n^2 + 8n + 64 = \mathcal{O}(n^3)$.

(b) (6P) $n \cdot 2^n = \mathcal{O}(2^n)$

- (c) (6P) $3^n = \mathcal{O}(2^n)$
5. (20P) Geben Sie für jedes der folgenden Probleme an, ob es semi-entscheidbar und ob es entscheidbar ist. Begründen Sie ihre Antwort.
- (a) (5P) Man entscheide für eine (durch ihren Code) gegebene Turing-Maschine M , und ein Eingabewort w , ob $w \notin L(M)$.
- (b) (5P) Man entscheide für eine (durch ihren Code) gegebene Turing-Maschine M , ob $L(M)$ rekursiv aufzählbar ist.
- (c) (5P) Man entscheide für zwei (durch ihre Codes) gegebene Turing-Maschinen M_1, M_2 , ob $L(M_1) \cap L(M_2) \neq \emptyset$.
- (d) (5P) Man entscheide mithilfe eines Orakels für das Akzeptanz-Problem, ob für zwei (durch ihre Codes) gegebene Turing-Maschinen M_1 und M_2 gilt, dass $L(M_1) \neq L(M_2)$ (hier ist nur die Frage der Semi-Entscheidbarkeit zu beantworten).
-

Zusätzliche Fragen:

6. (20P) Ist die Funktion $f(a, b) := \sum_{i=a}^b i$ primitiv rekursiv bzw. rekursiv? Und wie steht es mit der Funktion $g(a, b) := \min_i i \geq a \wedge (i = b \vee i \text{ is prime})$? (Antworten Sie im Detail, z.B. durch entsprechende Funktionsdefinitionen).
7. (20P) Beantworten Sie folgende Fragen (Begründung):
- (a) (5P) Ist es richtig, dass jedes Problem, das mit einer deterministischen Turing-Maschine in exponentieller Zeit lösbar ist, mit einer nicht-deterministischen Turing-Maschine in polynomialer Zeit lösbar ist?
- (b) (5P) Ist es richtig, dass wenn ein einziges Problem in \mathcal{NP} deterministisch in polynomialer Zeit lösbar ist, dann alle Probleme in \mathcal{NP} deterministisch in polynomialer Zeit lösbar sind?
- (c) (5P) Ist es richtig, dass wenn die Sprache L von einer nicht-deterministischen Turingmaschine in polynomialer Zeit erkannt werden kann, dann auch \bar{L} von einer nicht-deterministischen Turingmaschine in polynomialer Zeit erkannt werden kann?
- (d) (5P) Gibt es Probleme, die deterministisch in polynomialer Zeit, aber nicht mit polynomialer Speicher lösbar sind? Und wie steht es mit der umgekehrten Richtung?