Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Abteilung Algorithmik

Dr. Arne Schmidt

Sommersemester 2026

Klausur
(Muster)
Theoretische Informatik 2
13.07.2026

Nachname: .....

Vorname: .....

Matr.-Nr.: .....

Studiengang: .....

☐ Bachelor ☐ Master ☐ Andere: .....

Hinweise:

- Bitte das Deckblatt in Druckschrift vollständig ausfüllen.
- Die Klausur besteht aus 11 Blättern, bitte auf Vollständigkeit überprüfen.
- Die Bearbeitungszeit für die Klausur beträgt 120 Minuten.

- Erlaubte Hilfsmittel: Ein Blatt DIN-A4, beidseitig handschriftlich beschrieben.
- Eigenes Papier ist nicht erlaubt.
- Die Rückseiten der Blätter dürfen beschrieben werden.
- Am Ende der Klausur sind Schmierzettel beigefügt.
- Antworten, die *nicht* gewertet werden sollen, bitte deutlich durchstreichen.
- Kein Tippex verwenden.
- Werden mehrere Antworten gegeben, werten wir die mit der geringsten Punktzahl.
- Mit *Bleistift* oder in *Rot* geschriebene Klausurteile können nicht gewertet werden.
- Dies ist eine Musterklausur. Die aufgelisteten Aufgaben sind zur Übersicht der möglichen Fragen aufgelistet und sind nicht auf Zeit getestet. In der richtigen Klausur können auch andere Fragestellungen vorkommen.

Aufgabe	1	2	3	4	5	6	7	8	Σ
Max	24	10	10	10	10	10	16	10	100
Erreicht									

Aufgabe 1: Kurzfragen **(12*2 Punkte)**

a) Aussagen zu linear beschränkten Turing-Maschinen

	Wahr	Falsch
Ein LBA ist eine Turingmaschine mit beschränktem Band.	☐	☐
LBAs erkennen genau die kontextsensitiven Sprachen.	☐	☐
Ein LBA hat unbegrenzten Speicher.	☐	☐

b) Aussagen zu Turing-Maschinen

	Wahr	Falsch
Eine deterministische Turingmaschine hat in jedem Zustand höchstens eine mögliche Transition.	☐	☐
Deterministische endliche Automaten erkennen rekursiv-aufzählbare Sprachen, aber keine kontextfreien Sprachen.	☐	☐
Jede nichtdeterministische Turingmaschine kann durch eine deterministische simuliert werden.	☐	☐

c) Aussagen zu Turing-Maschinen.

	Wahr	Falsch
Turingmaschinen sind ein Modell für berechenbare Funktionen.	☐	☐
Jede Turingmaschine hält für jede Eingabe an.	☐	☐
Es existieren universelle Turingmaschinen.	☐	☐

d) Aussagen zur Entscheidbarkeit

	Wahr	Falsch
Eine Sprache $\mathcal{L}$ ist entscheidbar, wenn eine TM M mit $\mathcal{L}(M) = \mathcal{L}$ für jede Eingabe anhält.	☐	☐
Jede entscheidbare Sprache ist auch semi-entscheidbar.	☐	☐
Kein semi-entscheidbares Problem kann von einem Algorithmen entschieden werden.	☐	☐

e) Aussagen zu Sprachen.

	Wahr	Falsch
Das Halteproblem ist unentscheidbar.	☐	☐
Es existieren mehr Sprachen als Turingmaschinen.	☐	☐
Jede Sprache ist entscheidbar.	☐	☐
Der Satz von Rice gilt für Eigenschaften von Turingmaschinen-Sprachen.	☐	☐
Jede nichttriviale semantische Eigenschaft ist unentscheidbar.	☐	☐
Der Satz von Rice gilt für Eigenschaften beliebiger Sprachen.	☐	☐

f) Aussagen zu Reduktionen

	Wahr	Falsch
Reduktionen werden verwendet, um die Schwierigkeit von Problemen zu vergleichen.	☐	☐
Wenn A auf B reduziert werden kann und B entscheidbar ist, dann ist auch A entscheidbar.	☐	☐
Reduktionen können nicht verwendet werden, um Unentscheidbarkeit zu zeigen.	☐	☐

g) Aussagen zu P und NP.

	Wahr	Falsch
P enthält Probleme, die in Polynomialzeit lösbar sind.	☐	☐
NP enthält Probleme, deren Lösungen in Polynomialzeit verifizierbar sind.	☐	☐
Alle Probleme in NP sind auch in P.	☐	☐
NP kann über nichtdeterministische Turingmaschinen definiert werden.	☐	☐
Jede Sprache in P ist auch in NP.	☐	☐
NP enthält nur Entscheidungsprobleme.	☐	☐

h) Aussagen zu 2SAT

	Wahr	Falsch
2-SAT ist in Polynomialzeit lösbar.	☐	☐
Jede Klausel enthält höchstens zwei Literale.	☐	☐
2-SAT ist NP-vollständig.	☐	☐

i) Aussagen zu PATH

	Wahr	Falsch
PATH fragt, ob ein Weg zwischen zwei Knoten existiert.	☐	☐
PATH $\in NL$.	☐	☐
PATH ist unentscheidbar.	☐	☐

j) Aussagen zu PSPACE.

	Wahr	Falsch
PSPACE enthält Probleme, die mit polynomialem Speicher lösbar sind.	☐	☐
P ist eine Teilmenge von PSPACE.	☐	☐
PSPACE enthält keine Probleme aus NP.	☐	☐

Aufgabe 2: TM-Analyse **(10 Punkte)**

Betrachte die DTM $M = (Q, \{0,1\}, \{0,1,\sqcup\}, \delta, q_0, \{q_F\})$, wobei Zustände und die Transitionsfunktion der folgenden Abbildung zu entnehmen sind. Nicht vorhandene Transitionen führen in einen Zustand, in welchem die Maschine stecken bleibt.

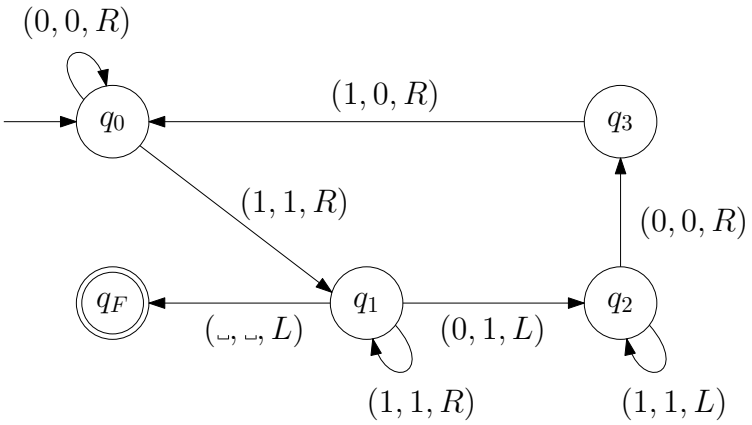

Abbildung 1: Eine DTM.

a) Führe M auf das Wort 010110010110 aus. Gib dabei die Konfiguration von M bei jedem betreten von q_0 (auch die initiale Konfiguration) und q_F an.

b) Leite eine Definition der von M berechneten partiellen Funktion $f : \{0,1\}^* \to_p \{0,1\}^*$ her.

Aufgabe 3: TM-Konstruktion **(10 Punkte)**

Betrachte die Funktion $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ mit $f(a,b) = a+b$, also eine Funktion, welche zwei Binärzahlen addiert. Auf leeren Wörten sei die Funktion nicht definiert, d.h. $f(\varepsilon, b) = \mathit{undef}.$ und $f(a, \varepsilon) = \mathit{undef}$. Die Binärstrings starten jeweils mit dem wertigsten Bit (bspw. entspricht die Zahl 4 dem String „100“).

a) Beschreibe die Funktionsweise einer DTM M mit 2 Bändern welche als Eingabe $a\#b$ erhält und $f(a,b)$ berechnet. Der Wert $f(a,b)$ darf am Ende auch auf Band 2 stehen.

b) Gib M formal inklusive dem Zustandsgraphen an.

Aufgabe 4: Entscheidbarkeit **(10 Punkte)**

Betrachte das folgendes Problem DOUBLE.

Gegeben Kodierung w einer DTM.

Entscheide Existiert ein $x \in \mathcal{L}(M_w)$, sodass auch $xx \in \mathcal{L}(M_w)$?

a) Zeige: DOUBLE ist semi-entscheidbar.

b) Zeige (ohne den Satz von Rice): DOUBLE ist unentscheidbar.

Aufgabe 5: NL-Vollständigkeit **(10 Punkte)**

Zeige NL-Vollständigkeit bzgl. Logspace-Reduktionen des folgenden Entscheidungsproblems Odd-Walk.

Gegeben: Ein gerichteter Graph $G = (V, E)$ und $s, t \in V$.

Frage: Existiert eine s-t-Kantenfolge in G mit einer ungeraden Anzahl an Kanten?

(Hinweis: In einer Kantenfolge dürfen Knoten und Kanten mehrfach vorkommen.)

Aufgabe 6: NP-Vollständigkeit **(10 Punkte)**

Zeige NP-Vollständigkeit bzgl. Logspace-Reduktionen des Entscheidungsproblems HITTING-SET.

Gegeben: Eine Menge $U = \{u_1, \ldots, u_n\}$ von n Objekten, eine Menge $S \subseteq \mathcal{P}(U)$ von Teilmengen von U, und eine Zahl $k \in \mathbb{N}$.

Frage: Existiert $U' \subseteq U$ mit $U' \cap s \neq \varnothing$ für alle $s \in S$ und $|U'| \leq k$?

Aufgabe 7: Fragen **(16 Punkte)**

a) Untersuche die folgende Sprache mit Hilfe des Satzes von Rice:

- Lässt sich der Satz anwenden?
- Falls anwendbar, was folgt daraus für die Entscheidbarkeit der Spracheigenschaft?
- Falls unentscheidbar, ist die Spracheigenschaft nicht-monoton?

Begründe deine Antworten jeweils.

$L = \{w \in \{0,1\}^* \mid \exists x \in \mathcal{L}(M_w) : \forall k \in \mathbb{N}_{>0} : x^k \in \mathcal{L}(M_w)\}$

b) Betrachte folgende 2SAT-Instanz.

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_3)$$

Bestimme eine Lösung der Instanz mit dem Algorithmus aus der Vorlesung. Kommen zu einem Zeitpunkt mehrere Variablen für die nächste Iteration in Frage, wähle diejenige mit dem kleinsten Index.

c) Betrachte die Grammatik $G = (\{S, A, B, Y\}, \{0, 1\}, P, S)$, wobei die Produktionsregeln folgendermaßen definiert sind.

$$\begin{aligned} S &\to Y0AB0 \\ A &\to 0AB0 \mid 0B0 \\ YB &\to 11 \\ 0B &\to B0 \\ 1B &\to 11 \end{aligned}$$

Leite formal die Sprache $\mathcal{L}(G)$ her.

d) Zeige: $\mathbb{B} = \{\{0, 1\}^{\mathbb{N}}\}$, die Menge aller abzählbar unendlich langen Binärstrings, ist überabzählbar unendlich.

Aufgabe 8: Orakelmaschinen **(10 Punkte)**

Eine Orakel-Turingmaschine ist eine gewöhnliche Turingmaschine, die zusätzlich Zugriff auf ein Orakel hat. Das Orakel ist ein „magisches“ Hilfsmittel, das eine bestimmte Entscheidungsfrage in einem einzigen Rechenschritt beantworten kann. Man kann sich das Orakel wie eine Black-Box vorstellen: Die Maschine stellt eine Ja/Nein-Frage (also ein Entscheidungsproblem) an das Orakel und erhält sofort die korrekte Antwort.

Die Klasse von Entscheidungsproblemen, die durch einen Algorithmus aus Klasse C mit einem Orakel für eine Sprache L lösen lassen, wird mit C^L bezeichnet. Beispielsweise ist P^{CVP} die Menge von Entscheidungsproblemen, welche mit einer deterministischen Turing Maschine in polynomieller Zeit gelöst werden können, wenn ein Orakel benutzt werden darf, welches Instanzen des Circuit Value Problem (CVP) entscheidet.

a) Betrachte das Problem 1-QBF, d.h. QBF mit einem Quantorenwechseln. Für eine boolesche Formel F fragen wir also:

$$\exists x_1, \ldots, x_k : \forall x_{k+1}, \ldots, x_n : F(x_1, \ldots, x_n) \text{ ist erfüllt}$$

Zeige: 1-QBF mit einem Quantorenwechsel ist in $NP^{\text{TAUTOLOGY}}$.

b) Zeige: $P^{\text{CVP}} = P$

c) Zeige oder widerlege: $NP \subseteq P^{3SAT}$

Viel Erfolg ☺