

Name		Matrikel									SKZ				
------	--	----------	--	--	--	--	--	--	--	--	-----	--	--	--	--

Klausur 2

Berechenbarkeit und Komplexität

20. Januar 2012

Markieren Sie die jeweils richtige Antwort.

– Viel Erfolg!

Aufgabe 1 RecFun5

Die Funktionen $f : \mathbb{N} \rightarrow \mathbb{N}$ und $g : \mathbb{N} \rightarrow \mathbb{N}$ seien folgendermaßen definiert:

$$f(n) := \begin{cases} k & \text{falls } n = 2^k \text{ und } k \in \mathbb{N} \\ \text{undefiniert} & \text{sonst} \end{cases}$$

$$g(n) := \begin{cases} \lfloor \log_2 n \rfloor & \text{falls } n > 0 \\ 0 & \text{falls } n = 0 \end{cases}$$

Hinweis: Für eine reelle Zahl x bezeichnet $\lfloor x \rfloor$ die größte ganze Zahl, die kleiner oder gleich x ist.

1 ja nein

Ist f primitiv rekursiv?

f ist nicht total und kann somit nicht primitiv rekursiv sein.

2 ja nein

Ist f rekursiv?

$$f(n) = \min_k (n = 2^k)$$

3 ja nein

Ist f durch eine RASP berechenbar?

f ist rekursiv und damit Turing/RAM/RASP-berechenbar.

4 ja nein

Ist g primitiv rekursiv?

Weil $g(n) \leq n$ kann $g(n)$ durch eine beschränkte Suche berechnet werden.

5 ja nein

Ist g rekursiv?

g ist rekursiv, weil g sogar primitiv rekursiv ist.

Aufgabe 2 gcd

Über den natürlichen Zahlen $\mathbb{N} = \{0, 1, 2, \dots\}$ sind die Prädikate

$$S(t, a) = \exists n \in \mathbb{N} : nt = a \wedge n \leq a$$

$$T(t, a) = \exists n \in \mathbb{N} : nt = a$$

$$G(t, a, b) = T(t, a) \wedge T(t, b)$$

gegeben. Weiters sind über dem Alphabet $\Sigma = \{0, 1\}$ die Sprachen

$$L_1 = \{01^t 01^a 01^b 0 \mid t, a, b \in \mathbb{N} \wedge G(t, a, b)\}$$

und

$$L_2 = \{01^a 01^b 0 \mid a, b \in \mathbb{N} \wedge \exists t \in \mathbb{N} : G(t, a, b)\}$$

gegeben.

6	ja	<input type="checkbox"/>
---	----	--------------------------

Ist $S(t, a)$ primitiv rekursiv?

$S(t, a)$ ist in der Angabe durch eine beschränkte Suche implementiert.

7	ja	<input type="checkbox"/>
---	----	--------------------------

Ist $T(t, a)$ primitiv rekursiv?

Ja, weil $S(t, a) = T(t, a)$ und $S(t, a)$ primitiv rekursiv ist.

8	ja	<input type="checkbox"/>
---	----	--------------------------

Gilt stets $S(t, a) = T(t, a)$?

9	ja	<input type="checkbox"/>
---	----	--------------------------

Ist $G(t, a, b)$ primitiv rekursiv?

10	ja	<input type="checkbox"/>
----	----	--------------------------

Ist L_1 rekursiv?

11	ja	<input type="checkbox"/>
----	----	--------------------------

Ist L_2 eine kontextfreie Sprache?

Weil $G(1, a, b)$ immer wahr ist gilt $L_2 = L(01^*01^*0)$, es ist also regulär und somit insbesondere kontextfrei.

Aufgabe 3 Complexity8

Gelten folgende Aussagen allgemein für Funktionen von \mathbb{N} nach $\mathbb{R}_{>0}$?

12	<input type="checkbox"/>	nein
----	--------------------------	------

$f(n) = O(g(n)) \implies 1/f(n) = O(1/g(n))$

13	ja	<input type="checkbox"/>
----	----	--------------------------

$f(n) = O(g(n)) \implies 1/g(n) = O(1/f(n))$

14	ja	<input type="checkbox"/>
----	----	--------------------------

$f_1(n) = O(g_1(n))$ und $f_2(n) = O(g_2(n)) \implies f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$

Aufgabe 4 Complexity9

Sei $f(n) = n \log(n) + n^2$ und $g(n) = n \log(n) + n$.

15	ja	<input type="checkbox"/>
----	----	--------------------------

Gilt $f(n) + g(n) = O(f(n))$?

$f(n) + g(n) = \Theta(n^2) = O(f(n))$

16	<input type="checkbox"/>	nein
----	--------------------------	------

Gilt $f(n) + g(n) = O(g(n))$?

$f(n) + g(n) = \Theta(n^2)$, aber $g(n) = \Theta(n \log(n))$. Somit ist die Aussage falsch.

17	<input type="checkbox"/>	nein
----	--------------------------	------

Gilt $f(n)g(n) = O(n^2 \log(n)^2)$?

Denn $f(n)g(n) = \Theta(n^3 \log(n))$ und $3 > 2$.

18	ja	<input type="checkbox"/>
----	----	--------------------------

Gilt $f(n)g(n) = O(2^n)$?

Aufgabe 5 Accept101

Über dem Alphabet $\Sigma = \{0, 1\}$ ist die Sprache

$$C = \{ \langle M \rangle \mid M \text{ akzeptiert } 101 \}$$

der Codierungen aller Turingmaschinen (mit Eingabealphabet $\{0, 1\}$), die 101 akzeptieren, gegeben.

19 ja

Ist C rekursiv aufzählbar?

Simuliere M auf 101: Starte eine universelle Turingmaschine auf $(\langle M \rangle, 101)$.

20 nein

Ist C rekursiv?

Satz von Rice ist anwendbar: $101 \in L(M)$ ist eine nicht triviale Eigenschaft rekursiv aufzählbarer Sprachen.

21 nein

Ist das Komplement \bar{C} von C rekursiv aufzählbar?

Wäre neben C auch \bar{C} rekursiv aufzählbar, so wäre C rekursiv. Ist es aber nicht, siehe oben.

Aufgabe 6 RAMterm

Über dem Alphabet $\Sigma = \{0, 1\}$ ist für jede RAM R die Sprache

$$L_R = \{01^n 0 \mid n \in \mathbb{N} \text{ und } R \text{ terminiert auf der Eingabe } n\}$$

gegeben.

22 ja

Gibt es eine RAM R so dass L_R eine kontextfreie Sprache ist?

Ja. Grund 1: $L_R = L(01^*0)$ für eine RAM ohne Sprungbefehle.
Grund 2: $L_R = \{\}$ für eine RAM die nie terminiert.

23 nein

Ist L_R für jede RAM R rekursiv?

Nein. Nimm etwa eine RAM R , die eine universelle Turingmaschine U auf dem Tupel $(\langle M \rangle, x)$, als natürliche Zahl n kodiert, simuliert. (M ist die von U simulierte Maschine, und w die Eingabe von M .) Dann gilt $01^n 0 \in L_R \iff R$ terminiert auf $n \iff U$ terminiert auf $(\langle M \rangle, w) \iff M$ terminiert auf w . Das Problem ist also äquivalent zum Halteproblem und somit unentscheidbar.

Aufgabe 7 doubleloopcomplexity

Eine RAM R sei durch nebenstehenden Pseudocode gegeben. Analysieren Sie die Zeit- und Raumkomplexität von R in Abhängigkeit der vom Band gelesenen Zahl n sowohl nach dem uniformen als auch nach dem logarithmischen Kostenkriterium. Füllen Sie Ihre Antwort in der Form $\Theta(\dots)$ ein.

```

c = 0
n = read()
while n > 0
    k = n
    while k > 0
        c = c + 1
        k = k - 1
    n = n - 1
write(c)

```

1. (1 Punkt) asymptot. Zeitkomplexität (uniform): _____

2. (1 Punkt) asymptot. Raumkomplexität (logarithmisch): _____

(Sie brauchen Ihre Antworten nicht zu begründen.)

Es reichen 4 Register: Je eines für n , k , c und Register 0 als Akkumulator. Die innere Schleife wird $\frac{1}{2}n(n+1)$ Mal durchlaufen, was auch die Zeitkomplexität dominiert. Der größte Wert der je in einem Register steht ist $\frac{1}{2}n(n+1)$ in c , am Ende der Berechnung.

1. uniforme Kosten: Zeitkomplexität $\Theta(n^2)$, also $a = 2, b = 0$.
2. logarithmische Kosten: Raumkomplexität $\Theta(\log n)$. Dominant ist der Inhalt $\frac{1}{2}n(n+1)$ von c am Ende des Programms. Die Länge davon ist $\lfloor \log(\frac{1}{2}n(n+1)) \rfloor + 1 = \Theta(\log(\frac{1}{2}n(n+1))) = \Theta(\log(\frac{1}{2}) + \log(n) + \log(n+1)) = \Theta(\log(n))$. (Auch die maximalen Werte in n und k sind von der Ordnung $\Theta(n^2)$.)

Bepunktung: Jede richtige Antwort gibt einen Punkt, so können maximal 2 Punkte erreicht werden. Begründungen/Berechnungen sind nicht nötig und werden nicht gewertet.