



Prüfung WS 2018/19

Studiengang: INF-B

Fach: Grundlagen der Informatik 2

Prüfer: Prof. Dr. J. Schmidt

Prüfung: 7.2.2019

90 Minuten. Hilfsmittel: alle Unterlagen, Taschenrechner, **kein** Laptop, Handy, u.ä.

Insgesamt sind 90 Punkte zu erreichen. Die Punktzahl gibt damit auch einen Anhaltspunkt für die Bearbeitungszeit.

Sollten Ihrer Meinung nach Angaben in der Aufgabenbeschreibung fehlen oder falsch sein, machen Sie sinnvolle Annahmen und dokumentieren Sie diese.

Der Lösungsweg muss ersichtlich/begründet sein.

Die Seiten dürfen nicht getrennt werden.

Konzeptpapier muss (mit Namen versehen) mit abgegeben werden.

Aufgabe	1	2	3	4	5	6	ges.
Punkte							

Note:

Name: _____

Matrikelnr.: _____

Aufgabe 1: Automaten/Grammatiken/Wortproblem (20 Punkte)

Gegeben sei das Alphabet $A = \{a, b, x, y\}$ und folgende Grammatik
(Großbuchstaben = Nichtterminale, Startsymbol S):

$S \rightarrow AB \mid ABC$

$A \rightarrow abA \mid ab$

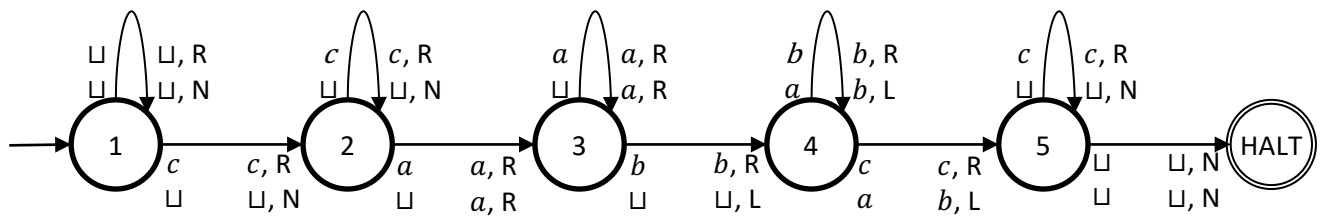
$B \rightarrow x \mid y$

$C \rightarrow xyC \mid y$

- Von welchem Typ der Chomsky-Hierarchie ist diese Grammatik?
- Bringen Sie die Grammatik in Chomsky-Normalform.
- Geben Sie einen äquivalenten regulären Ausdruck an.
- Von welchem Typ der Chomsky-Hierarchie ist die Sprache?
- Zeichnen Sie das Übergangsdiagramm eines äquivalenten Automaten. Fangzustände dürfen weggelassen werden.
- Prüfen Sie, ob das Wort $abyxyxy$ Teil der akzeptierten Sprache ist.

Aufgabe 2: Grammatiken/Turing-Maschinen (20 Punkte)

Gegeben sei das Übergangsdiagramm einer 2-Band Turingmaschine mit Eingabealphabet $T = \{a, b, c\}$ und dem Bandalphabet $B = T \cup \{\sqcup\}$:



Fangzustände wurden weggelassen.

Band 1 dient als Eingabeband, Band 2 zum Speichern von Daten während der Verarbeitung.

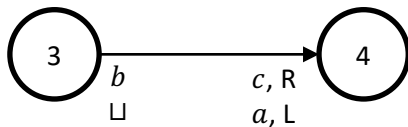
Band 1 enthält zu Beginn das Eingabewort, links und rechts davon ist das Band mit \sqcup gefüllt. Der Schreib-/Lesekopf von Band 1 befindet sich an beliebiger Stelle links des Eingabeworts.

Band 2 ist mit \sqcup gefüllt. Der Schreib-/Lesekopf von Band 2 befindet sich an beliebiger Stelle.

Für den Schreib-/Lesekopf sind die Bewegungsrichtungen links (L), neutral (N, d.h. keine Bewegung) und rechts (R) zulässig.

Die Notation der Zustandsübergänge im Diagramm ist entsprechend der in der Vorlesung definierten 1-Band Maschine gewählt, oben ist jeweils Band 1 angegeben, unten Band 2. Ein Übergang kann nur ausgeführt werden, wenn das aktuelle Zeichen auf beiden Bändern passt.

Beispiel:



Bedeutet: Der Übergang von Zustand 3 nach 4 ist möglich, wenn aktuell auf Band 1 ein b steht und auf Band 2 ein \sqcup .

Auf Band 1 wird ein c geschrieben, der Kopf bewegt sich nach rechts.

Auf Band 2 wird ein a geschrieben, der Kopf bewegt sich nach links.

a) Zeigen Sie: $caabbc$ ist Teil der akzeptierten Sprache:

- Wie lautet die Zustandsfolge, wenn der Schreib-/Lesekopf auf Band 1 direkt links neben dem ersten Zeichen steht?
- Was steht am Ende auf den beiden Bändern?

b) Neben $caabbc$ sind z.B. auch $ccccccaaabbbc$ und $caaaabbbbccccc$ Teil der akzeptierten Sprache. Geben Sie die Sprache in mengenschreibweise an. Von welchem Typ ist die Sprache? Ist die Sprache μ -rekursiv?

c) Geben Sie eine äquivalente Grammatik an. Von welchem Typ ist Ihre Grammatik? Für den Fall, dass Sie die Teilaufgabe (b) nicht bearbeiten konnten, geben Sie bitte eine Grammatik für folgende Sprache an: $L = \{a^u c^v b^u c^w \mid u, v, w \in \mathbb{N}\}$.

Aufgabe 3: Verschiedenes (20 Punkte)

Welche der folgenden Aussagen sind richtig bzw. falsch? Kreuzen Sie das entsprechende Feld an. Für jede korrekte Antwort gibt es zwei Punkte. Ein Punktabzug für falsche Antworten erfolgt **nicht**.

Aussage	richtig	falsch
$2n^2 + \log_2 4n = O(n^2 \log n)$		
Algorithmen mit Zeitkomplexität $O(\log n)$ liegen in der Klasse P		
Das Halteproblem hat exponentielle Zeitkomplexität		
Algorithmen mit Zeitkomplexität $O(n^2)$ liegen in der Klasse EXPSPACE		
Das Travelling Salesman Problem ist WHILE-berechenbar		
Der Fermat-Primzahltest liefert für die Zahl 59 „ist prim“		
Mit Hilfe des Pumping Theorems für reguläre Sprachen lässt sich zeigen, dass die Sprache $L = \{ a^i bc \mid i \in \mathbb{N} \}$ regulär ist		
Jede reguläre Grammatik lässt sich in Chomsky-Normalform angeben		
Das Wortproblem für kontextsensitive Sprachen ist NP-schwer, liegt aber nicht in der Klasse NP		
Mit einem deterministischen Kellerautomaten lässt sich das Wortproblem für Typ 3 Sprachen lösen		

Aufgabe 4: Karatsuba-Multiplikation (16 Punkte)

Multiplizieren Sie die beiden Zahlen 1011 und 1112 (dezimal) mit dem Karatsuba-Verfahren. Die Rekursion soll bei einstelligen Zahlen enden, wo dann „normal“ multipliziert wird. Ersichtlich sein muss:

- Welche rekursiven Aufrufe in welcher Reihenfolge mit welchen Parametern gibt es?
- Wie werden die Zahlen auf jeder Rekursionsebene jeweils zerlegt?
- Wie werden die Teilergebnisse kombiniert?

Aufgabe 5: Pseudo-Zufallszahlen (6 Punkte)

Mit der Formel $x_{n+1} = (a x_n + c) \bmod m$ lassen sich ganzzahlige Zufallszahlen erzeugen.

Es sei nun: $a = 17$, $c = 3$, $m = 42$

- Berechnen Sie die ersten drei Zufallszahlen x_1 , x_2 und x_3 beginnend mit Startwert $x_0 = 4$
- Ist garantiert, dass sich mit dieser Parameterwahl die maximal mögliche Periodenlänge ergibt?

Aufgabe 6: Komplexität (8 Punkte)

Die folgende Funktion berechnet rekursiv das arithmetische Mittel der Zahlen, die im Array `a` gespeichert sind. Die Parameter `start` und `end` geben jeweils die Indizes des zu betrachtenden Array-Teils an.

Hat man z.B. 10 Zahlen in einem Array „feld“ gespeichert, dann wird die Funktion wie folgt aufgerufen (`x` enthält dann den Mittelwert):

```
float x = avg(feld, 0, 9);
```

Bestimmen Sie die Zeitkomplexität der Funktion in O-Notation.

Hinweis: Bedenken Sie für die Analyse:

- wie viele Teile erhält man bei der Rekursion? Wie
- wie groß sind diese?
- wie groß ist der Aufwand für das Zusammenfügen der Teile?

```
float avg(float a[], int start, int end) {
    int n = end - start + 1;

    if(n == 1) return a[start];
    else
    {
        int mitte = n/2 - 1;
        float m1 = avg(a, start, start + mitte);
        float m2 = avg(a, start + mitte + 1, end);

        float m = (m1 * (mitte + 1) +
                   m2 * (end - (start + mitte + 1) + 1)) / n;

        return m;
    }
}
```