

- Vom Prüfungsteilnehmer vollständig auszufüllen -

Winter / Sommer – Semester 2020

Studiengang : _____

Prüfungsfach: Formale Sprachen _____

Matrikelnummer: <table border="1" style="display: inline-table; width: 150px; height: 20px; vertical-align: middle;"><tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px;"></td></tr></table>											
Platznummer _____	Raumnummer/ Turnhalle _____										
		Vom Prüfer/in bzw. 1. Korrektor/in auszufüllen									
		(Note)									
Prüfungstag: _____											
Arbeitszeit: _____		Unterschrift des Erstprüfer/s/in									
Beginn der Bearbeitung: _____		Vom 2. Korrektor/in auszufüllen									
		(Note)									
Ende der Bearbeitung: _____											
		Unterschrift des Zweitprüfer/s/in									
Erlaubte Hilfsmittel: _____ _____ _____											
Bemerkungen (z.B. Unterbrechung der Bearbeitung, wird vom Aufsichtführenden ausgefüllt) <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> von _____ Uhr bis _____ Uhr </div> <div style="display: flex; justify-content: space-between; margin-bottom: 5px;"> von _____ Uhr bis _____ Uhr </div> <div style="display: flex; justify-content: space-between;"> von _____ Uhr bis _____ Uhr </div>											

1. Aufgabe **Maschinenmodelle****10**

1. Auf dem Eingabeband einer RAM steht eine Sequenz, die aus den Zeichen ,a', ,b' und ,c' besteht. Die Zeichenkette beginnt mit einer Sequenz des Zeichens ,a' wobei mindestens ein ,a' vorkommen muss. Danach folgt eine Sequenz mit den Zeichen ,b' und ,c' mit mindestens einem dieser Zeichen. Entwickeln Sie ein Programm für die RAM, das die Eingabe auf diese Struktur überprüft und ein ,t' ausgibt, falls sie dieser Struktur entspricht. Im anderen Fall soll ,f' ausgegeben werden.

aaabccb# → t

bbcc# → f

aaa# → f

ab# → t

6

1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	
16.	
17.	
18.	
19.	
20.	
21.	
22.	
23.	
24.	

2. Aufgabe **Komplexität**

10

1. Für einen gegebenen Geldbetrag sollen Aktien verschiedener Unternehmen gekauft werden. Die Aktien der Unternehmen können stückweise entsprechend des aktuellen Aktienkurses gekauft werden. Von jeder Aktie eines Unternehmens steht nur eine begrenzte Anzahl zur Verfügung. Der Geldbetrag soll so angelegt werden, dass die Summe der letzten Dividende für alle gekauften Aktien maximiert wird. Mit welchem Zeitaufwand müssen Sie hier in Abhängigkeit von der Anzahl der verfügbaren Aktien rechnen? Begründen Sie Ihre Antwort!

4

2. Gegeben sind drei Entscheidungsprobleme P, Q und R. Sie finden Transformationen, mit denen Sie Eingaben für P nach Q, für Q nach R sowie R nach P transformieren können. Die Transformationen haben einen linearen Zeitaufwand. Außerdem haben Sie für P einen Algorithmus gefunden, der diese Aufgabenstellung mit kubischem Zeitaufwand lösen kann. Schließlich gibt es für R einen Algorithmus, der diese Aufgabenstellung mit exponentiellem Zeitaufwand löst. Welche Schlussfolgerungen können Sie daraus für die Zeitkomplexitäten dieser drei Entscheidungsprobleme ziehen?

3

3. Für die Lösung einer Aufgabenstellung existiert ein Programm auf einer deterministischen Turingmaschine, das einen exponentiellen Zeitaufwand hat. Wäre es denkbar, dass eine solche Aufgabenstellung auch mit einem endlichen Automaten gelöst werden kann?

3

3. Aufgabe **Grammatiken****12**

1. Geben Sie eine Grammatik für Zeichenketten mit den Zeichen 'a', 'b', 'c' und 'd' an. Die Zeichenkette soll mit einer Sequenz aus den Zeichen 'a' und 'b' beginnen (mindestens eines dieser Zeichen). Danach folgt eine weitere Sequenz aus den Zeichen 'c' und 'd', die ebenfalls mindestens eines dieser Zeichen enthalten muss.

3

Beispiele:

- abccd
- bcdcd
- aaababc
- ac

2. Für einen Grammatiktyp existieren die folgenden Einschränkungen. Auf der linken Seite darf nur ein Nichtterminalsymbol stehen. Auf der rechten Seite stehen nur entweder ein Terminalsymbol oder zwei Nichtterminalsymbole. Zu welcher Sprachklasse gehören die mit diesen Grammatiken erzeugten Sprachen? Geben Sie die spezifischste Sprachklasse an (mit der höchsten Nummer).

3

3. Analysieren Sie die folgende Grammatik und geben Sie drei Wörter der zugehörigen Sprache an. Beschreiben Sie danach möglichst genau die gesamte Menge der Wörter, die für diese Grammatik erzeugt werden. Die Nichtterminalsymbole sind hier 'q' und 's' wobei 's' das Startsymbol ist. Die Terminalsymbole sind 'a', 'b' und 'c'.

6

$s \rightarrow abc$
 $s \rightarrow abcq$
 $q \rightarrow qq$
 $cq \rightarrow qc$
 $bqc \rightarrow qbcc$
 $bqb \rightarrow qbb$
 $aqb \rightarrow aabb$

4. Aufgabe *Rekursiv absteigender Parser*

12

Mit den booleschen Operatoren `&`, `|` und `~` lassen sich in Java ganze Zahlen bitweise verknüpfen. So würde zum Beispiel der Operator `&` angewendet auf die Zahlen 0101 und 0011 in Binärdarstellung das Ergebnis 0001 liefern. Der Operator `|` ist die Oderoperation und der Operator `~` das logische „Nicht“. Mit Hilfe dieser Operatoren soll jetzt ein Parser entwickelt werden, mit dem boolesche Ausdrücke berechnet werden können. Diese Ausdrücke haben die folgende Form:

```

<bool> ::= <and> { '|' <and> }
<and> ::= <literal> { '&' <literal> }
<literal> ::= [ '~' ] <bin> | ( '(' <bool> ')' )
<bin> ::= 0 | 1 { 0 | 1 }

```

Ein Beispiel entsprechend dieser Grammatik könnte wie folgt aussehen:

!1010&1111&(1000|0100)|0101&1010 -> 0100

Ergänzen Sie den folgenden Code an den markierten Stellen so, dass der Parser einen solchen Ausdruck einliest und das Ergebnis berechnet. Dazu sollen insbesondere die Funktionen „parseAnd“ und „parseLiteral“ implementiert werden. Diese und die weiteren Funktionen geben jeweils das Ergebnis der einzelnen Operationen als ganze Zahl zurück. Für das obige Beispiel würde das Ergebnis 0100 lauten (die Ausgabe in der Funktion „main“ unterdrückt allerdings die führende Null).

```
public class BoolParser {
    int c;
    FileReader fr;
    Set<Integer> personen = new HashSet<Integer>();

    public BoolParser (String fileName) throws IOException {
        fr = new FileReader(fileName);
        read();
    }

    public void read() throws IOException {
        c = fr.read();
        System.out.print((char) c);
    }

    // <bool> ::= <and> { '|' <and> }
    int parseBool() throws Exception {
        int result = parseAnd();
        while (c == '|') {
            read();
            result = result | parseAnd();
        }
        return result;
    }

    // <and> ::= <literal> { '&' <literal> }
    int parseAnd() throws Exception {
```

[illegible]

result = 0;
}
read();
while (c == '1' c == '0') {
result *= 2;
if (c == '1') {
result += 1;
}
read();
}
return result;
}
public static void main(String[] args) throws Exception {
BoolParser sp = new BoolParser(args[0]);
System.out.println(Integer.toBinaryString(sp.parseBool()));
}
}

5. Aufgabe XML-Parser

8

Für die Planungen in einer Region wird die Bevölkerungsverteilung mit Hilfe einer XML-Datei angegeben. Die Datei gibt die Anzahl von Personen in verschiedenen Altersgruppen für die Zellen eines Rasters an. Die Zellen werden mit der X-Koordinate und der Y-Koordinate über die entsprechenden Attribute „X“ und „Y“ referenziert. Die Attribute „Altersgruppe“ und „Anzahl“ geben die Anzahl der Personen der Altersgruppe in der Zelle an. Ein Beispiel für eine solche Datei könnte wie folgt aussehen:

```
<Personenverteilung>
  <Zelle X="1" Y="1" Altersgruppe="Senioren" Anzahl="7"/>
  <Zelle X="1" Y="1" Altersgruppe="Erwachsene" Anzahl="15"/>
  <Zelle X="1" Y="1" Altersgruppe="Junioren" Anzahl="4"/>
  <Zelle X="1" Y="2" Altersgruppe="Senioren" Anzahl="12"/>
  <Zelle X="1" Y="2" Altersgruppe="Erwachsene" Anzahl="31"/>
  <Zelle X="1" Y="2" Altersgruppe="Junioren" Anzahl="10"/>
  <Zelle X="2" Y="1" Altersgruppe="Senioren" Anzahl="6"/>
  <Zelle X="2" Y="1" Altersgruppe="Erwachsene" Anzahl="3"/>
  <Zelle X="2" Y="1" Altersgruppe="Jugend" Anzahl="0"/>
  <Zelle X="2" Y="2" Altersgruppe="Senioren" Anzahl="11"/>
  <Zelle X="2" Y="2" Altersgruppe="Erwachsene" Anzahl="8"/>
  <Zelle X="2" Y="2" Altersgruppe="Junioren" Anzahl="1"/>
</Personenverteilung>
```

Entwickeln Sie ein Programm, das für eine solche Datei die Summe aller Personen in der Zelle mit den Koordinaten <2,2> (X="2" Y="2") aufsummiert und ausgibt. Für das obige Beispiel wäre das Ergebnis 20 (11+8+1).

[illegible]

}
public static void main(String[] args) {
PersonenParser parser;
try {
parser = new PersonenParser(new File(args[0]));
parser.parseFile();
} catch (ParserConfigurationException SAXException IOException e) {
e.printStackTrace();
}
}
}
}