

2. Klausur 12/IDauer: 4 SchulstundenName: www.r-krell.deHilfsmittel: normaler Taschenrechner,* *Achte auf sorgfältige Darstellung mit vollständigem, nachvollziehbarem Lösungsweg!* ** *Kommentiere deine Programme!* ***1 Automaten mit Ausgabe**

a) Es sollen unäre Zahlen (Bierdeckelnotation) verarbeitet werden:

a1) Zeichne einen Automaten, der durch 2 teilt (d.h. aus `////` oder `/////` die Ausgabe `//` macht).a2) Jetzt soll das Eingabealphabet $X = \{ /, + \}$ sein und der Automat 'addieren', d.h. z.B. aus der Eingabe `/// + //` die Ausgabe `/////` machen. Zeichne den Graph und stelle die Tafel auf!

b) Ein Elefant soll durch einen endlichen Automaten mit Ausgabe simuliert werden: Wenn man sich dem Elefanten gegenüber normal/neutral verhält (Eingabe = n), tut er nichts (Ausgabe = N). Füttert man den Elefanten (f), so wird man mit fröhlichem Trompeten (T) belohnt. Wirft man hingegen kleine Äste und Zweige auf den Elefanten (w), so passiert die ersten beiden Male nichts (N). Aber ab dem dritten Mal – egal, welche anderen Verhaltensweisen dazwischen liegen – straft der Elefant den Angreifer, in dem er ihn bei jedem Werfen (w) mit Wasser besprüht (S). (Beispiel s. Kasten).

```
eingabe = "nwfnwfnfw"
ausgabe = "NNTNNTSNTS"
```

b1) Zeichne den Automatengraph

b2) Notiere die formale Beschreibung $A=(X,Y,Z,\delta,\lambda,z_0)$ dieses Automatenb3) Schreibe eine Java-Methode `public void simuliereElefant (String eingabe)`, die den Automaten mit Zuständen schematisch nachbildet und schließlich die Ausgabe auf die Konsole schreibt!b4) Gegeben ist die Methode `elefant2`:

(1) Gibt sie bei der Eingabe von "nwfnwfnfw" am Ende "NNTNNTSNTS" aus?

(2) Nenne die Ausgabe von `elefant2 ("wwwffwn ")`(3) Erläutere, wieso `elefant2` ohne Zustände auskommt. Welche Beschränkungen des Automaten wurden aufgehoben?

```
public void elefant2 (String eingabe)
{
    String ausgabe=""; int zähler = 0;
    for (int i=0; i<eingabe.length(); i++)
    {
        switch (eingabe.charAt(i))
        {
            case 'n': ausgabe = ausgabe+'N'; break;
            case 'f': ausgabe = ausgabe+'T'; break;
            case 'w': zähler++;
                if (zähler<3)
                    {ausgabe = ausgabe+'N';}
                else
                    {ausgabe = ausgabe+'S';}
        }
    }
    System.out.println (ausgabe);
}
```

2 Erkennende Automaten

a) Ein Automat soll den Funkverkehr abhören.

Eingabezeichen sind die Kleinbuchstaben der gefunkten Nachricht, das Leerzeichen und das Komma. Wörter werden durch Leerzeichen () oder Kommas (,) oder beides getrennt. Werden die ganzen Wörter hilfe oder help erkannt, soll der Automat in einen Endzustand übergehen, der auch bei allen weiteren Eingaben nicht mehr verlassen wird. Um Fehlalarme zu vermeiden, sollen aber z.B. die Wörter schilf, helpdesk oder kochelpass nicht in einen Endzustand führen! Zeichne den Automatengraph!b) Gegeben ist die Automatentafel auf der nächsten Seite / Rückseite. Das Eingabealphabet besteht nur aus den Ziffern 0 oder 1, Startzustand ist Z_0 , Endzustände sind Z_3 und Z_4 .

b1) Zeichne den Automatengraph

- b2) Welchen Zustand erreicht der Automat jeweils am Zahlen-ende, wenn (angefangen beim Startzustand Z_0) nacheinander die Ziffern folgender Zahlen eingegeben werden: 1) 100 2) 101 3) 110 4) 111 5) 0 6) 1 7) 10 8) 11 9) 1010010

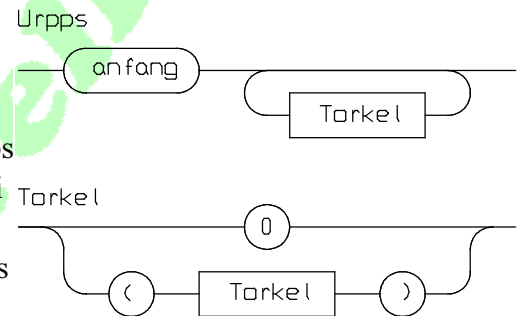
δ	0	1
Z_0	Z_1	Z_2
Z_1	Z_1	Z_3
Z_2	Z_4	Z_2
Z_3	Z_4	Z_2
Z_4	Z_1	Z_3

- b3) Dualzahlen, die in einem der beiden Zustände Z_3 oder Z_4 enden, sollen als „gute“ Zahlen betrachtet werden; Zahlen die in einem der Zustände Z_0 , Z_1 oder Z_2 enden, gelten als „schlecht“. Gib selbst noch zwei gute und zwei schlechte Zahlen an und versuche in Worten zu beschreiben, was eine gute und was eine schlechte Zahl ausmacht.
- b4) Schreibe eine Typ-3-Grammatik $G=(N,T,P,S)$ für den Automaten!
- b5) Alternativ wird die Grammatik $G'=(N',T,P',S)$ vorgeschlagen mit $N'=\{S,A,B\}$, $T=\{0,1\}$, $P'=\{S \rightarrow 1A, S \rightarrow 0B, A \rightarrow 0, A \rightarrow 0B, A \rightarrow 1A, B \rightarrow 1, B \rightarrow 0B, B \rightarrow 1A\}$. Prüfe, ob G' dieselbe Sprache beschreibt wie b4) bzw. der Automat b1). Begründe oder finde Wörter, die nur zu einer der beiden Sprachen $L(G)$ oder $L(G')$ gehören.

3 Syntax und Compiler

Gegeben ist das abgebildete Syntaxdiagramm.

- a) Entscheide begründet, ob I. „anfang“, II. „anfang0((0))“ und III. „anfang((0)0)“ erlaubte Urpps sind, und erfinde selbst noch je zwei gültige und je zwei ungültige Urpps!
- b) Wie in den Beispielen in a) sollen als Eingabe höchstens Kleinbuchstaben, die beiden runden Klammern und die Ziffer 0 auftreten -- keine Leerzeichen. Ein einfacher Scanner soll als Token normale Strings liefern: entweder Kleinbuchstabenwörter (nicht nur "anfang", sondern jede Kombination von Kleinbuchstaben bis zum nächsten anderen Zeichen), oder Klammern "(", ") " oder den String "0" mit der Ziffer Null. Eine eigene Klasse Token mit Typ oder Art ist daher nicht nötig. Schreibe die Methode `public String nächstesToken()`.



```
public class Scanner
{
    quelltext="anfang0((0))"; //als Beispiel
    int position = 0;

    public String nächstesToken()

```

- c) Schreibe in Java die Klasse `Parser` mit den Methoden `parseUrpps()` und `parseTorkel()`, die genau dann `true` liefern sollen, wenn der Quelltext dem Syntaxdiagramm entspricht. Verwende dabei den Scanner, der bereits den Quelltext enthalten soll (und der funktionieren soll, auch wenn du b) noch nicht gelöst hast)..
- d) Für eine ähnliche Sprache ist die Grammatik $G=(N,T,P,S)$ gegeben durch $N=\{MUH, MÄH, HILF\}$, $T=\{1,2,3,4, mitte\}$, $P=\{MUH \rightarrow 1 HILF, HILF \rightarrow MÄH 4, HILF \rightarrow HILF MÄH 4, MÄH \rightarrow 2 MÄH 3, MÄH \rightarrow mitte \}$ und $S=MUH$. Zeichne das Syntaxdiagramm und gib je zwei richtige und zwei falsche MUH-Wörter an!
- e) Gib zum ursprünglich gegebenen Syntaxdiagramm a) eine Grammatik an. Begründe außerdem, ob die Grammatik regulär (d.h. vom Typ 3 und damit durch endliche Automaten erkennbar) sein kann!