

## Aufgabe 1: Betriebssysteme

Nennen Sie zwei Komponenten eines Betriebssystems und beschreiben Sie deren jeweilige Funktion mit nicht mehr als 50 Worten.

Komponente 1: **Prozess-Scheduler**

Ein Prozess-Scheduler ist eine Arbitrationslogik, die die zeitliche Ausführung mehrerer Prozesse in Betriebssystemen regelt. Prozess-Scheduler kann man grob in unterbrechende (preemptive) und nicht unterbrechende (non preemptive, auch kooperativ genannt) aufteilen [Wikipedia].

Komponente 2: **Dateisystem (Filesystem)**

Das Dateisystem ist die Ablageorganisation auf einem Datenträger eines Computers. Dateien müssen gelesen, gespeichert oder gelöscht werden. Für den Nutzer müssen Dateiname und computerinterne Dateiadressen in Einklang gebracht werden. Das leichte Wiederfinden und das sichere Abspeichern sind wesentlich. Das Ordnungs- und Zugriffssystem berücksichtigt die Geräteeigenschaften [Wikipedia].

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 2: Grammatiken

Gegeben sei eine Grammatik wie folgt ( $\varepsilon$  ist Leersymbol):

$G$  =  $\langle \text{Startsymbol, Terminale, Nicht-Terminale, Regeln} \rangle$  mit

Startsymbol =  $\text{Start}$

Terminale =  $\{a, b, c\}$

Nicht-Terminale =  $\{X, Y\}$

Regeln =  $\{ \text{Start} \rightarrow Xc, \\ X \rightarrow aYc, \\ X \rightarrow aXc, \\ Y \rightarrow b, \\ Y \rightarrow bY \}$

Entscheiden Sie, ob folgende Wörter in der durch die Grammatik definierten Sprache enthalten sind:

- |                  |                      |
|------------------|----------------------|
| 1. $\varepsilon$ | <del>ja</del> / nein |
| 2. ab            | <del>ja</del> / nein |
| 3. abc           | <del>ja</del> / nein |
| 4. abcc          | ja / <del>nein</del> |
| 5. aabbbbccc     | ja / <del>nein</del> |
| 6. c             | <del>ja</del> / nein |
| 7. aaabcccc      | ja / <del>nein</del> |
| 8. aabbcc        | <del>ja</del> / nein |

Bestimmen Sie die Wörter, die durch die Grammatik akzeptiert werden in der Art  $a^n b^n$ :

$a^m b^n c^{m+1}$  mit  $m, n > 0$

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 3: Suche in Binärbäumen

Gegeben sei ein Binärbaum, realisiert als verkettete Struktur wie unten, aufgebaut mit Datenstrukturen der Art (Pseudocode):

```
ListElem { int data, pointer left, pointer right }
```

als

```
ListElem X1, X2, X3, X4, X5, X6, X7;
```

```
Top = adr(X1);
```

```
X1.left = adr(X2); X1.right = adr(X3); X1.data = 0010;
```

```
X2.left = adr(X4); X2.right = adr(X5); X2.data = 0008;
```

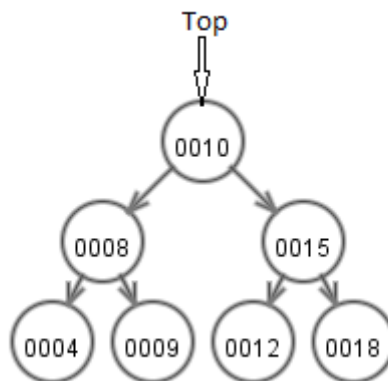
```
X4.left = null; X4.right = null; X4.data = 0004;
```

```
X5.left = null; X5.right = null; X5.data = 0009;
```

```
X3.left = adr(X6); X3.right = adr(X7); X3.data = 0015;
```

```
X6.left = null; X6.right = null; X6.data = 0012;
```

```
X7.left = null; X7.right = null; X7.data = 0018;
```



Skizzieren Sie eine Boolean Funktion (Pseudocode), welche überprüft, ob ein Wert Val im Baum vorhanden ist und die nur von Top und Val abhängt:

Boolean Search(Val, Top)

Begin

if (Top = null) return false

else if (Top.data = Val) return true

else if (Top.data > Val) return Search(Val, Top.left)

else return Search(Val, Top.right)

End.

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Aufgabe 4: Prolog Unifikation

Gegeben seien jeweils 2 Prolog-Ausdrücke. Sie sollen entscheiden, ob diese beiden Ausdrücke unifizierbar sind und, falls dieses der Fall ist, den durch Substitution erzeugten Ausdruck angeben. Sollte die Unifikation misslingen, geben Sie den Grund an. Als Beispiele sind gegeben (Konstanten fangen mit kleinem, Variablen mit großem Buchstaben an):

- $\text{test}(X, \text{foo}, Y) = \text{test}(M, N, M)$  ☺ mit  $\text{test}(X, \text{foo}, X)$ .
- $\text{test}(\text{kai}, X) = \text{test1}(\text{kai}, \text{udo})$  ☹ weil  $\text{test} \neq \text{test1}$ .

Hier nun die Ausdrücke:

- $\text{frage}(a, B, c, D) = \text{muster}(A, b, C, d)$ .

☹  $\text{frage} \neq \text{muster}$ .

- $\text{muster}(a, B, c, D) = \text{muster}(A, b, C, d)$ .

☺ mit  $A = a, B = b, C = c, D = d$ .

- $\text{verwandt}(\text{bruder}(\text{kai}), \text{Hans}), = \text{verwandt}(\text{Udo}, \text{bruder}(\text{kai}))$ .

☺ mit  $\text{Udo} = \text{bruder}(\text{kai}), \text{Hans} = \text{bruder}(\text{kai})$ .

- $\text{schwester}(\text{paul}, \text{paula}) = \text{bruder}(\text{paula}, \text{paul})$ .

☹  $\text{schwester} \neq \text{bruder}$ .

- $\text{berechne}([\text{plus}, 1, 2, Z]) = \text{berechne}([\text{plus}, X, Y, 4])$ .

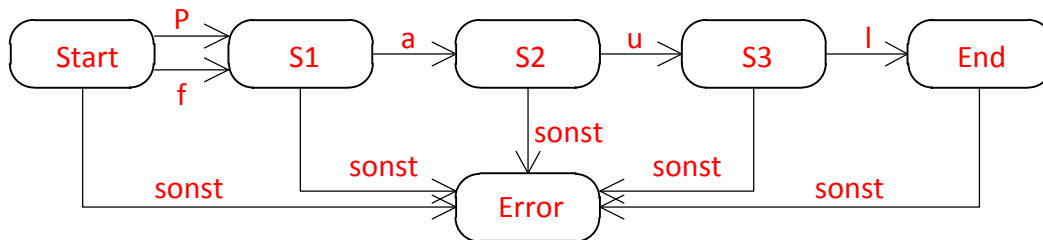
☺ mit  $X = 1, Y = 2, Z = 4$ .

## Aufgabe 5: Endliche Automaten

Gegeben sei ein Worterkennungsautomat, der die folgende Sprache erkennt:

$$L = \{\text{Paul, faul}\}$$

Geben Sie **einen** endlichen Automaten an, der die Sprache L akzeptiert und bei allen anderen Zeichenfolgen in einen Fehlerzustand übergeht.



Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_

## Bewertung der Aufgaben

|                                        |                                                          |   |            |
|----------------------------------------|----------------------------------------------------------|---|------------|
| <b>Aufgabe 1:</b>                      | irgendwie zutreffende Beschreibung<br>(maximal 2 Punkte) | = | 1 Punkt    |
| <b>Aufgabe 2:</b>                      | pro richtige Wortentscheidung                            | = | 0.5 Punkte |
|                                        | richtige regulärer Ausdruck<br>(maximal 5 Punkte)        | = | 1 Punkt    |
| <b>Aufgabe 3:</b>                      | richtige Codeskizze                                      | = | 6 Punkte   |
|                                        | fast richtige Codeskizze                                 | = | 4 Punkte   |
|                                        | Idee Traversierung Baum                                  | = | 3 Punkte   |
|                                        | fast richtige Code Idee<br>(maximal 6 Punkte)            | = | 2 Punkte   |
| <b>Aufgabe 4:</b>                      | jeweils richtiges Ergebnis<br>(maximal 5 Punkte)         | = | 1 Punkt    |
| <b>Aufgabe 5:</b>                      | richtiger Automat                                        | = | 4 Punkte   |
|                                        | fast richtiger Automat                                   | = | 3 Punkt    |
|                                        | richtiger Ansatz                                         | = | 2 Punkt    |
|                                        | fast richtiger Ansatz<br>(maximal 4 Punkte)              | = | 1 Punkt    |
| <b>Maximale Punkte aller Aufgaben:</b> |                                                          | = | 22 Punkte  |

**Notengebung:**  $\text{Maximum}(\text{Minimum}(\text{Aufrunden}(\text{Summe aller Punkte} - 5), 15), 0)$

Name: \_\_\_\_\_

Matr.-Nr: \_\_\_\_\_