

# Klausur Grundlagen der Informatik II

Semester: AI2	WS 02/03, 1.2.2002
Bearbeitungszeit: 70 Minuten	Hilfsmittel: A ohne prog. C

## Fächerübergreifende Prüfung zusammen mit Grundlagen der Informatik I.

### Aufgabe 1 (20 Punkte)

Gegeben sei der Kellerautomat  $K = (\{z_0\}, \{ (, ) \}, \{ [, ] \}, \delta, z_0, \#)$ , mit

$$\delta = \left\{ \begin{array}{l} z_0, (, \# \rightarrow z_0, [ \# \\ z_0, (, [ \rightarrow z_0, [ [ \\ z_0, ), [ \rightarrow z_0, \varepsilon \\ z_0, \varepsilon, \# \rightarrow z_0, \varepsilon \end{array} \right\}$$

a) Kreuzen Sie von den folgenden Worten genau die an, die von  $K$  erkannt werden.

- |  |                                   |                                  |                                   |  |                                      |
|--|-----------------------------------|----------------------------------|-----------------------------------|--|--------------------------------------|
| <input type="checkbox"/> ()              | <input type="checkbox"/> (()      | <input type="checkbox"/> ))((    | <input type="checkbox"/> ())(     | <input type="checkbox"/> ([[])         | <input type="checkbox"/> (((()()))   |
| <input type="checkbox"/> ()()()          | <input type="checkbox"/> [[][]]   | <input type="checkbox"/> [[]]    | <input type="checkbox"/> []       | <input type="checkbox"/> ]][[          | <input type="checkbox"/> []]         |
| <input type="checkbox"/> ([[()])]        | <input type="checkbox"/> [[[][]]] | <input type="checkbox"/> [][] [] | <input type="checkbox"/> (())()() | <input type="checkbox"/> $\varepsilon$ | <input type="checkbox"/> (((())()()) |
| <input type="checkbox"/> (((())((()()))) | <input type="checkbox"/> ()()()   | <input type="checkbox"/> )))(((  | <input type="checkbox"/> ()))     | <input type="checkbox"/> (((           | <input type="checkbox"/> [[[         |

b) Geben Sie eine Regelgrammatik an für  $L(K)$ .

$$G = (\{S\}, \{ (, ) \}, R, S), R = \{S \rightarrow (S) \mid S () \mid \varepsilon\}$$

c)  $L$  hat Chomsky Typ 2. Warum ist  $L$  keine Typ-1-Sprache?

*Typ 2, da von Kellerautomat erkannt.*

d) Warum ist  $L$  keine Typ-3-Sprache?

*Weil endliche Automaten nicht Klammern zählen können.*

Gegeben ist folgende Entfernungstabelle deutscher Städte:

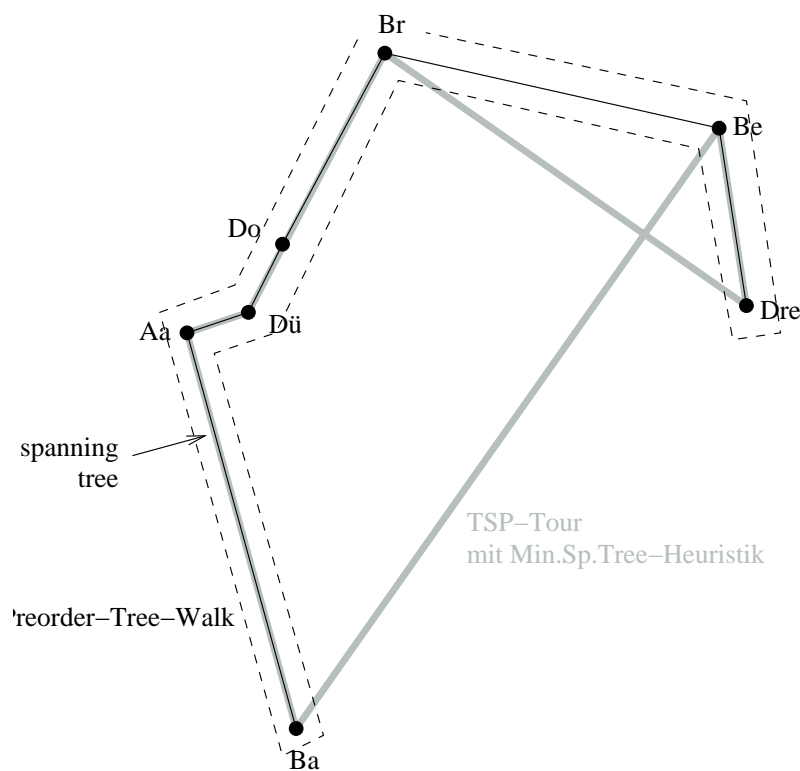
Aachen							
Basel	545						
Berlin	650	875					
Bremen	370	775	400				
Dortmund	155	555	495	235			
Dresden	645	745	200	490	515		
Düsseldorf	90	550	560	285	70	580	
	Aachen	Basel	Berlin	Bremen	Dortmund	Dresden	Düsseldorf

- a) Bestimmen Sie mit dem Greedy-Algorithmus eine Lösung für das TSP-Problem. Starten Sie in Düsseldorf. Geben Sie die Tour sowie deren Länge an.

*Dü, Do, Aa, Br, Ber, Dre, Bas, Dü, Gesamtentfernung: 2490*

- b) Bestimmen Sie einen Minimum Spanning Tree mit dem Kruskal-Algorithmus. Zeichnen Sie diesen in die Landkarte (unten) ein.
- c) Verwenden Sie die Minimum-Spanning-Tree-Heuristik mit Wurzelknoten Bremen zur Bestimmung einer Lösung des TSP-Problems. Zeichnen Sie die Lösung in die Karte ein und geben Sie Tour sowie Länge an.

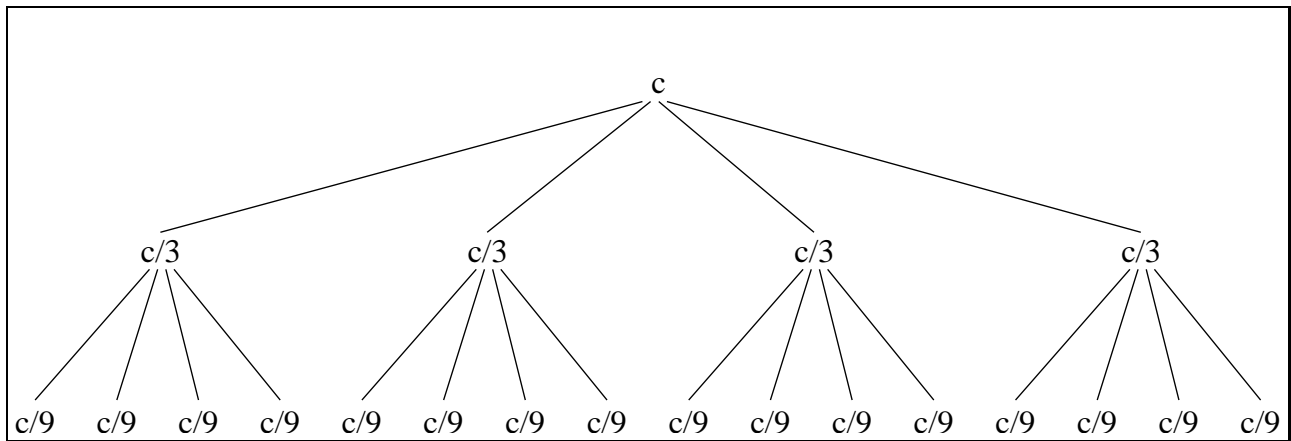
*Dü, Do, Br, Dre, Ber, Bas, Aa, Dü, Gesamtentfernung: 2505*



Gegeben sei die C-Funktion:

```
void f(n)
int n;
{ if( n > 1 )
  { printf("A"); f(n/3); f(n/3); f(n/3); f(n/3); }
  else
  { printf("X"); }
}
```

a) Zeichnen Sie den Rekursionsbaum der Funktion f für  $n > 1$ .



b) Welche Ausgabe erzeugt der Aufruf  $f(9)$ ? (Herleitung/Begründung!)

*AAAXXXAAXXXAAXXXAAXXX*  
Jeder rekursive Aufruf druckt A, also  $1 + 4$  mal A, bei der Terminierung wird 16 mal X gedruckt.

c) Berechnen Sie die Tiefe des Rekursionsbaumes von f in Abhängigkeit vom Wert des Übergabeparameters n.

$$d = \lfloor \frac{\log n}{\log 3} \rfloor$$

d) Berechnen Sie die Komplexität von f mit dem Mastertheorem.

$$n^{\log_b a} = n^{\log_3 4} = n^{\frac{\log 4}{\log 3}} \approx n^{1.262}$$
$$\Rightarrow T(n) = \Theta(n^{\frac{\log 4}{\log 3}}) \approx \Theta(n^{1.262}).$$

$$T(n) = c \sum_{k=1}^d (4/3)^k = \frac{4}{3} \cdot \frac{1 - (\frac{4}{3})^d}{1 - \frac{4}{3}} = 4 \left( (\frac{4}{3})^d - 1 \right) = 4n \left[ \left( \frac{4}{3} \right)^{\lfloor \frac{\log n}{\log 3} \rfloor} - 1 \right]$$

$$\Rightarrow T(n) = O\left(n \left( \frac{4}{3} \right)^{\lfloor \frac{\log n}{\log 3} \rfloor}\right)$$

f) Zeigen Sie, dass  $n^{\frac{\log 4}{\log 3}} = 4^{\frac{\log n}{\log 3}}$ .

$$\log n^{\frac{\log 4}{\log 3}} = \frac{\log 4}{\log 3} \log n = \frac{\log n}{\log 3} \log 4 = \log 4^{\frac{\log n}{\log 3}} \Rightarrow n^{\frac{\log 4}{\log 3}} = 4^{\frac{\log n}{\log 3}}$$

#### Aufgabe 4 (20 Punkte)

Geben Sie ein FLEX-Programm mit insgesamt 5 Regeln (7 Zeilen) an, das eine Eingabedatei der Form

```
7^-1
3^-1
25^-1
abcde23
Log[23^56]
Log[14^39]
Log[2^6]
4^1/2
5^1/2
15^1/2
```

zeilenweise transformiert in eine Datei der Form

```
1/7
1/3

56 * Log[23]
39 * Log[14]

Sqrt[4]
Sqrt[5]
```

```
%option noyywrap
%%
^[0-9]\^-1      printf("1/%c", yytext[0]);
^[0-9]\^1\1/2  printf("Sqrt[%c]", yytext[0]);
Log\[ [0-9]{2}\^[0-9]{2}\] printf("%c%c * Log[%c%c]", yytext[7],yytext[8],yytext[4],yytext[5])
\n              printf("\n");
```