

# Lösung der 1. Übung – Informatik I für VIW

## Fakultät Verkehrswissenschaften

### Fachrichtung Verkehrsingenieurwesen

Zeitraum: 18.10. bis 29.10.2010 (WS 2010/11)

#### Aufgabe 1

**Algorithmus 1** Der Inhalt von Flasche 1 und Flasche 2 soll vertauscht werden. Dazu wird eine weitere Flasche zur Hilfe benötigt.

- 1 Fülle den Inhalt aus Flasche 1 in die Hilfsflasche
- 2 Fülle Flasche 2 in Flasche 1
- 3 Fülle die Hilfsflasche in Flasche 2

#### Algorithmus 2

##### VARIANTE MIT WAGENHEBER

- 1 Gehe zum ersten Rad
- 2 Führe folgende Tätigkeiten 4 mal aus
  - 2.1 Auto an dem aktuellen Rad mit Wagenheber anheben
  - 2.2 Führe folgende Tätigkeiten 5 mal aus
    - 2.2.1 Mutter lösen
    - 2.2.2 Mutter abnehmen
  - 2.3 Sommerrad abnehmen
  - 2.4 Winterrad ansetzen
  - 2.5 Führe folgende Tätigkeiten 5 mal aus
    - 2.5.1 Mutter ansetzen
    - 2.5.2 Mutter anziehen
  - 2.6 Wagenheber absinken lassen
  - 2.7 zum nächsten Rad weitergehen
- 3 zur Tankstelle fahren und Luftdruck prüfen

##### VARIANTE MIT HEBEBÜHNE

- 1 Auto auf Hebebühne fahren
- 2 Hebebühne anheben
- 3 Gehe zum ersten Rad

- 4 Führe folgende Tätigkeiten 4 mal aus
  - 4.1 Führe folgende Tätigkeiten 5 mal aus
    - 4.1.1 Mutter lösen
    - 4.1.2 Mutter abnehmen
  - 4.2 Sommerrad abnehmen
  - 4.3 Winterrad ansetzen
  - 4.4 Führe folgende Tätigkeiten 5 mal aus
    - 4.4.1 Mutter ansetzen
    - 4.4.2 Mutter anziehen
  - 4.5 zum nächsten Rad weitergehen
- 5 Hebebühne ablassen
- 6 zur Tankstelle fahren und Luftdruck prüfen

**Algorithmus 3** Hier kann man entweder die Seiten von vorn nach hinten durchsuchen, bis man auf den Eintrag trifft. Oder man nutzt die Eigenschaft, dass die Einträge im Telefonbuch alphabetisch sortiert wurden und wendet das Verfahren der binären Suche an:

- 1 linke Hand an den linken Buchrücken setzen
- 2 rechte Hand an den rechten Buchrücken setzen
- 3 führe folgende Tätigkeiten solange aus, bis zwischen linker und rechter Hand höchstens noch eine Seite ist
  - 3.1 schlage das Telefonbuch in der Mitte zwischen linker und rechter Hand auf
  - 3.2 Wurde das Telefonbuch zu weit hinten aufgeschlagen (also befindet sich der zu suchende Eintrag weiter vorn)?
    - JA: Setze die rechte Hand auf die gerade aufgeschlagene Seite
    - NEIN: Setze die linke Hand auf die gerade aufgeschlagene Seite
- 4 schaue auf der einen Seite zwischen linker und rechter Hand direkt nach, ob sich der zu suchende Eintrag dort befindet

**Algorithmus 4** Hier gibt es verschiedene Möglichkeiten, das Problem zu lösen.

**MÖGLICHKEIT 1** Man vergleicht alle Klausuren miteinander. Das kann man systematisch so machen, dass man sich die erste Klausur vom Stapel nimmt, diese auf einen Extraplatz legt und dann den gesamten Reststapel nach einer Übereinstimmung mit dieser ersten Klausur durchsucht. Wurde keine Übereinstimmung gefunden, nimmt man sich dann die zweite Klausur zur Hand und vergleicht diese dann mit all den restlichen Klausuren, und so weiter. In diesem Verfahren wird man im schlechtesten Fall ungefähr 500000 Vergleiche durchführen müssen, es ist also nicht praktikabel.

Hier der Algorithmus:

- 1 führe folgende Tätigkeiten so lange aus, bis der Stapel leer ist
  - 1.1 nimm die oberste Klausur von dem Stapel und lege sie auf auf einen anderen Platz; wir bezeichnen diese Klausur im Folgenden als „ausgewählte Klausur“
  - 1.2 führe folgende Tätigkeiten für jede Klausur im Stapel aus
    - 1.2.1 haben die ausgewählte Klausur und die aktuelle Klausur im Stapel den gleichen Vornamen?  
JA: Breche den Algorithmus ab, eine Übereinstimmung wurde gefunden
- 2 es wurde keine Übereinstimmung gefunden

MÖGLICHKEIT 2 Man sortiert die Klausuren und findet währenddessen mögliche Übereinstimmungen. Der Sortiervorgang kann ähnlich wie das Einsortieren von Spielkarten auf die Hand ablaufen. Dieser Algorithmus arbeitet sehr viel schneller als der Algorithmus aus Möglichkeit 1.

- 1 Führe folgende Tätigkeiten so lange aus, bis der Stapel leer ist
  - 1.1 Nimm die oberste Klausur vom Stapel
  - 1.2 sortiere sie in die bereits vom Stapel entnommenen Klausuren an die richtige Stelle ein (Sortierung nach Vornamen)  
da die vom Stapel entnommenen Klausuren immer bereits sortiert sind, kann die richtige Position zum Einsortieren durch das Verfahren der binären Suche ermittelt werden (siehe vorheriger Algorithmus)
  - 1.3 befindet sich an der einsortierten Stelle bereits eine Klausur mit dem gleichen Vornamen?  
JA: Breche den Algorithmus ab, eine Übereinstimmung wurde gefunden
- 2 es wurde keine Übereinstimmung gefunden

## Aufgabe 2

Zur Definition der Begriffe Objektsprache, Metasprache, Alphabet, Wort, Konkatenation, formale Sprache, Komplexprodukt und  $L^*$  sei auf die Folien der Vorlesung verwiesen.

BEISPIELWÖRTER:  $12, abb1, 1ab2, 1111, \varepsilon, b112112$

BEISPIELSPRACHEN:  $\{12, abb1, 1ab2\}, \{abbba, 1111\}, \{\varepsilon, a, aa, aaa, aaaa, \dots\}, \emptyset, \{\varepsilon\}$

## Aufgabe 3

TEIL 1

- $L_1 \cup L_2 \cup L_3 = \{b, bc, a, ca, a\},$
- $L_2^* = \{\varepsilon, a, aa, aaa, aaaa, \dots\} = \{a^n \mid n \geq 0\},$

- $L_1 \cdot L_3 = \{bca, ba, bcca\}$   
Achtung: das Wort  $bca$  kommt nur einmal in der Sprache vor; Wörter können in Sprachen nicht mehrfach, sondern nur ein- oder nullmal vorkommen,
- $L_2^* \cdot L_3 = \{ca, a, aca, aa, aaca, aaa, aaaca, aaaaa, aaaaaca, aaaaaa, \dots\}$ ,
- $(L_1 \cup L_2) \cdot L_3 = \{bca, bcca, aca, ba, aa\}$
- $(L_1 \cup L_3)^* = \{\varepsilon, b, bc, ca, a, bcabcaa, cacacaaaca, bcbabaca, bbb, \dots\}$

## TEIL 2

- $\{aaa, aaca\} = L_2 \cdot L_2 \cdot L_3$ ,
- $\{baca, baa, bcaca, bcaa\} = L_1 \cdot L_2 \cdot L_3$ ,
- $L_3^*$ .

### Aufgabe 4 (Zusatz)

- Ja, die Sprachen sind unterschiedlich. Zwei Mengen (also auch Sprachen) sind immer dann unterschiedlich, wenn es mindestens ein Element gibt, welches in der einen Menge enthalten ist, aber nicht in der anderen.  
In diesem Fall enthält die Sprache  $\{\varepsilon\}$  das leere Wort. Dieses ist in der leeren Sprache nicht enthalten.
- $L \cdot \emptyset = \emptyset = \emptyset \cdot L$  und  $L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L$ .
- Die Sprache  $L \cdot L'$  enthält maximal  $n \cdot m$  Wörter. Es müssen aber nicht so viele sein, wie das Beispiel  $L_1 \cdot L_3$  aus Aufgabe 2 zeigt. Das passiert immer dann, wenn sich ein Wort aus  $L \cdot L'$  aus verschiedenen Kombinationen von Wörtern in  $L$  und  $L'$  bilden lässt.