

ALP IV: Aufgabenblatt 1.
Tobias Lohse & Luisa Castano. Tutor: Max, Do.10-12

1. Welche der Aussagen a) bis d) sind wahr? Falls ja, begründen Sie das, falls nein, geben Sie ein Gegenbeispiel. Zur Frage e) sind Spekulationen erlaubt.

- a. *Jeder deterministische Algorithmus ist determiniert.* Wahr, wenn die Reihenfolge der Abarbeitung der Anweisungen fest ist, wird es immer bei der gleichen Eingabe, die gleiche Ausgabe geben. Die gleichen Schritte werden immer nacheinander angewendet, was kein Raum für ein unterschiedliches Resultat lässt.
- b. *Jeder determinierte Algorithmus ist deterministisch.* Falsch. Gegenbeispiel sei der Algorithmus beschrieben in Aufgabe 3, wobei dieser immer die gleiche Ausgabe liefert (determiniert), aber kann die Reihenfolgen (1-2-3-4-5-6) und (1-3-2-4-5-6) folgen, unter anderen.
- c. *Jeder deterministische Algorithmus ist sequentiell.* Wahr, die Sequenz eines deterministischen Algorithmus ist einfach die Folge der Schritte eines wohlbestimmten Ablaufs. [Zitate aus "Nichtsequentielle Programmierung mit Go...", Maurer, Springer, Seite 2]
- d. *Jeder nichtsequentielle Algorithmus ist nicht determiniert.* Falsch. Das Programm
 $x=x+5; \quad y=y+y$
 $x=x-2; \quad y=y*y$
ist nichtsequentiell, aber trotzdem determiniert. (Beispiel vom Tutorium).
- e. *Ist jeder sequentielle Algorithmus determiniert?* Falsch. Jeder sequentielle Algorithmus ist nicht unbedingt deterministisch, und ein nicht-deterministischer Algorithmus kann nicht determiniert sein.

2.

- a. *Überprüfen Sie alle Angaben in den Zahlenbeispielen auf Seite 4 im Buch.*
 - i. *Bei 2 Folgen aus je 3 Anweisungen sind das nur 20 Möglichkeiten:*
$$\frac{(3+3)!}{3! \cdot 3!} = \frac{6!}{36} = 20$$
 - ii. *Bei 4 Folgen aus je 4 Anweisungen über 60 Millionen:*
$$\frac{(4+4+4+4)!}{(4! \cdot 4! \cdot 4! \cdot 4!)} = \frac{16!}{24^4} = 63.063.000$$
 - iii. *Bei 6 Folgen aus je 5 Anweisungen bereits 90 Trillionen:*
$$\frac{(5+5+5+5+5+5)!}{(5! \cdot 5! \cdot 5! \cdot 5! \cdot 5! \cdot 5!)} = \frac{30!}{120^6} = 8.883e + 19$$
 - iv. *Bei 8 Folgen aus je 6 Anweisungen über 170 Sextillionen:*
$$\frac{(6+6+6+6+6+6+6+6)!}{(6! \cdot 6! \cdot 6! \cdot 6! \cdot 6! \cdot 6! \cdot 6! \cdot 6!)} = \frac{48!}{720^8} = 1.719e + 38$$
- b. *10 Folgen aus 10 Anweisungen ausdrucken und mit dem Papier die Sonne wickeln. Ist der Radius der entstehenden Papierkugel 20-Mal größer als der des gesamten Universums?*

Nach Wikipedia, Radius des Universums: >45 Mrd. Lj, wobei $1 \text{ Lj} = 9.461 \times 10^{15} \text{ m}$, also
Radius des Universums = $9.461 \times 10^{15} \cdot 45 > 4.257 \times 10^{17} \text{ m}$

Um die Erde 100-Mal zu einwickeln, muss eine Papierkugel 100-Mal so groß sein, wie das Volumen der Erde → Volumen der Erde: $1,083 \times 10^{12} \text{ km}^3$ (*100)

Also nach Beispiel 3, wissen wir, dass 90 Trillionen Anweisung eine Papierkugel mit Volumen $1,083 \times 10^{14} \text{ km}^3$ machen. ($1 \text{ km}^3 = 10^9 \text{ m}^3$)

Wenn 90 Trillionen Anweisungen, ein Volumen von $1,083 \times 10^{23} \text{ m}^3$ machen,

$$\frac{100!}{10!^{10}} = 2.3570746e+92 \text{ Anweisungen machen } \frac{2.357e+92 * 1,083 \times 10^{23}}{8.883e+19} = 2.874e+98 \text{ m}^3$$

$$\text{Volumen einer Kugel} = \frac{4}{3} \Pi r^3 \rightarrow r = \sqrt[3]{2.874e+98 * \frac{3}{4\Pi}} = 4.094e+32 \text{ m.}$$

Behauptung: $4.094e+32 * 20 > 4.257 \times 10^{17} \leftrightarrow \underline{8.188e+33 > 4.257 \times 10^{17}}$

⇒ **Der Baron lügt nicht**

3.

- a. *Welche Ausführungsreihenfolgen liefern das korrekte Ergebnis?*
(1-3-2-4-5-6), (2-1-3-4-5-6), (2-1-4-3-5-6), (3-4-5-6-1-2), (3-4-5-6-2-1), und die in der Aufgabe erwähnten Reihenfolgen liefern die gleiche Ausgabe.
- b. *Zu welchen Fehlern kann es bei beliebiger Ausführungsreihenfolgen kommen?*
Die Aufgabe kann 0 sein (3-4-1-2-5-6, u.a.), oder die Ausgabe kann die kleinste Summe sein (1,2,4,5,6,3, u.a.).

4.

```
package main
import "fmt"
type Tree struct {
    field []int
}
func compare(c1 chan int, c2 chan int) bool {
    for {
        a1, ok1 := <-c1
        a2, ok2 := <-c2
        if !ok1 && !ok2 {
            // both arrays done, all values were equal
            return true
        } else if !ok1 || !ok2 {
            // one array shorter than the other
            return false
        } else if a1 != a2 {
            // found values that don't match
            return false
        }
    }
    // fmt.Println(a1, a2)
}
```

```

}
func traverse(t Tree, c chan int) {
    for _, a := range t.field {
        c <- a
    }
    close(c)
}

func (t1 Tree) Equivalent(t2 Tree) bool {
    c1 := make(chan int)
    c2 := make(chan int)
    go traverse(t1, c1)
    go traverse(t2, c2)

    return (compare(c1, c2))
}

func main() {
    t1 := Tree{field: []int{1, 2, 3, 4}}
    t2 := Tree{field: []int{1, 2, 3}}
    t3 := Tree{field: []int{1, 2, 3, 4, 5}}
    t4 := Tree{field: []int{1, 2, 3, 5}}
    t5 := Tree{field: []int{1, 2, 3, 4}}

    fmt.Println(t1.Equivalent(t1))
    fmt.Println(t1.Equivalent(t2))
    fmt.Println(t1.Equivalent(t3))
    fmt.Println(t1.Equivalent(t4))
    fmt.Println(t1.Equivalent(t5))
}

```