

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.
 - b) Jeder determinierte Algorithmus ist deterministisch.
 - c) Jeder deterministische Algorithmus ist sequentiell.
 - d) Jeder nichtsequentielle Algorithmus ist nicht determiniert.
 - e) Ist jeder sequentielle Algorithmus determiniert?
2. Zum Problem der kombinatorischen Explosion bei möglichen Ablaufreihenfolgen:
 - a) Überprüfen Sie alle Angaben in den Zahlenbeispielen auf Seite 4 im Buch.
 - b) Nehmen wir an, wir könnten alle Möglichkeiten der Verzahnung von 10 Folgen aus je 10 Anweisungen wie im 3. Beispiel ausdrucken und das dazu benötigte Papier um die Sonne wickeln. Münchhausen behauptet, dass der Radius der entstehenden Papierkugel zwanzigmal größer wäre als der des gesamten Universums (s. Wikipedia). Lügt der Baron?

3. Wir betrachten das Programm

```
package main

func main() {
    var (
        asum, bsum, max int
        a = []int{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 }
        b = []int{ 2, 3, 5, 7, 11, 13, 17, 19 }
    )
    asum = 0                // (1)
    bsum = 0                // (2)
    for _, x := range a { asum += x } // (3)
    for _, x := range b { bsum += x } // (4)
    max = asum              // (5)
    if bsum > asum { max = bsum }    // (6)
    println (max)
}
```

zur Berechnung des Maximums der Summe von Feldelementen. Die sechs Anweisungen (1) ... (6) können in verschiedener Reihenfolge ausgeführt werden, ohne das Ergebnis zu beeinflussen; z. B. liefern die obige Reihenfolge (1); ... ; (6) und die Reihenfolge (2); (4); (1); (3); (5); (6) das gleiche Ergebnis.

Für das Folgende sei jede Programmzeile als eine Elementaranweisung betrachtet.

- a) Welche Ausführungsreihenfolgen liefern das korrekte Ergebnis?
 - b) Zu welchen Fehlern kann es bei beliebiger Ausführungsreihenfolge kommen?
4. Implementieren Sie das auf Seite 12 im Buch (am Ende des Abschnitts 1.5) angegebene Beispiel in Go.