Mathematatik für Informatiker
Kombinatorik, Stochastik und Statistik
Übungsblatt 4

Tom Paßberg , Iain Dorsch

Aufgabe 1

a)

Der Professor versucht die Menge der Partitionen einer 5 elementigen Menge in 3 Teilen aufzuschreiben. Er hat bereits die Menge der Partitionen einer 4 elementigen Menge in 3 Teilen notiert.

c)

Es gilt $S(3,1) = S(2,1) = S(2,2) = S(3,3) = 1$.

$$\begin{aligned}
S(3,2) &= S(2,1) + 2 \cdot S(2,2) = 1 + 2 \cdot 1 = 3 \\
S(4,2) &= S(3,1) + 2 \cdot S(3,2) = 1 + 2 \cdot 3 = 7 \\
S(4,3) &= S(3,2) + 3 \cdot S(3,3) = 3 + 3 \cdot 1 = 6 \\
S(5,3) &= S(4,2) + 3 \cdot S(4,3) = 7 + 3 \cdot 6 = 25
\end{aligned}$$

Aufgabe 3

a)

Die Anzahl der Möglichen Endergenbnisse ist $3^3 = 27$. Die Anzahl der möglichen Totalordnungen ist $3! = 6$. Die Anzahl der Ergebisse bei denen der Spieler verliert ist $3! + 2 = 8$ da der Spieler auch verliert wenn alle Kanten des Dreiecks einen Spielstein der selber Ausrichtung enthalten. Die Wahrscheinlichkeit das der Spieler verliert ist also $\frac{8}{27} = 29,6\%$.
Gewinnwarscheinlichkeit: $\frac{19}{27} = 70,4\%$.

b)

Halbordnungen auf $\{1,2,3\}$

$$\begin{aligned}
R_1 &= \{(1,1),(2,2),(3,3)\}\\
R_2 &= \{(1,1),(2,2),(3,3),(1,2)\}\\
R_3 &= \{(1,1),(2,2),(3,3),(1,3)\}\\
R_4 &= \{(1,1),(2,2),(3,3),(2,3)\}\\
R_5 &= \{(1,1),(2,2),(3,3),(2,1)\}\\
R_6 &= \{(1,1),(2,2),(3,3),(3,1)\}\\
R_7 &= \{(1,1),(2,2),(3,3),(3,2)\}\\
R_8 &= \{(1,1),(2,2),(3,3),(1,2),(2,3),(1,3)\}\\
R_9 &= \{(1,1),(2,2),(3,3),(3,1),(1,2),(3,2)\}\\
R_{10} &= \{(1,1),(2,2),(3,3),(1,3),(3,2),(1,2)\}\\
R_{11} &= \{(1,1),(2,2),(3,3),(2,1),(1,3),(2,3)\}\\
R_{12} &= \{(1,1),(2,2),(3,3),(2,3),(3,1),(2,1)\}\\
R_{13} &= \{(1,1),(2,2),(3,3),(3,2),(2,1),(3,1)\}
\end{aligned}$$

R_8 bis R_{13} sind Totalordnungen.

Aufgabe 4

a)

$$R = \{(1,1),(2,2),(1,2)\}R = \{(1,1),(2,2)\}$$

c)

Die Anzahl der Totalordnungen auf einer n elementigen Menge ist $n!$. Das ergibt sich daraus, dass es $n!$ mögliche Reihenfolgen gibt eine n elementige Menge anzuordnen. Eine Totalordnung weißt allen Elementen eine eindeutige Reihenfolgen zu.

Aufgabe 5

Code:

```
fn partitions(n: u8, m: u8) -> Vec<Vec<Vec<u8>>> {
    if m == 0 || n == 0 {
        return Vec::new();
    }
```

```rust
    if m == 1 {
        return vec![vec![(1..=n).collect()]];
    }

    let mut result = Vec::new();
    for p in partitions(n - 1, m - 1).iter_mut() {
        p.push(vec![n]);
        result.push(p.clone());
    }
    for p in partitions(n-1, m) {
        for i in 0..p.len() {
            let mut p = p.clone();
            p[i].push(n);
            result.push(p);
        }
    }
    result
}
```

Funktionsaufruf:

```rust
let parts = partitions(5, 3);

for (i, part) in parts.iter().enumerate() {
    println!("{:3}: {:?}", i, part);
}
```

Ausgabe:

```
  0: [[1, 2, 3], [4], [5]]
  1: [[1, 2, 4], [3], [5]]
  2: [[1, 2], [3, 4], [5]]
  3: [[1, 3, 4], [2], [5]]
  4: [[1, 3], [2, 4], [5]]
  5: [[1, 4], [2, 3], [5]]
  6: [[1], [2, 3, 4], [5]]
  7: [[1, 2, 5], [3], [4]]
  8: [[1, 2], [3, 5], [4]]
  9: [[1, 2], [3], [4, 5]]
 10: [[1, 3, 5], [2], [4]]
 11: [[1, 3], [2, 5], [4]]
 12: [[1, 3], [2], [4, 5]]
 13: [[1, 5], [2, 3], [4]]
 14: [[1], [2, 3, 5], [4]]
 15: [[1], [2, 3], [4, 5]]
 16: [[1, 4, 5], [2], [3]]
```

```
17:  [[1,  4],  [2,  5],  [3]]
18:  [[1,  4],  [2],  [3,  5]]
19:  [[1,  5],  [2,  4],  [3]]
20:  [[1],  [2,  4,  5],  [3]]
21:  [[1],  [2,  4],  [3,  5]]
22:  [[1,  5],  [2],  [3,  4]]
23:  [[1],  [2,  5],  [3,  4]]
24:  [[1],  [2],  [3,  4,  5]]
```