

# **Mathematik für Informatiker**

## **Kombinatorik, Stochastik und Statistik**

### **Übungsblatt 6**

Tom Paßberg , Iain Dorsch

## Aufgabe 1

Die Gewinnwahrscheinlichkeit lässt sich durch  $(\frac{1}{2})^n$  beschreiben. Der erwartete Gewinn kann durch die Summe der gewichteten Gewinne beschrieben werden. Die Summe ist aufgeteilt in den Teil in dem der Gewinn variabel ist (hat noch nicht  $2^{47}$  erreicht) und den Teil in dem der Gewinn aufgrund der Obergrenze konstant ist.

$$\sum_{n=1}^{46} \frac{2^n}{2^n} + \sum_{n=47}^{\infty} \frac{2^{47}}{2^n} \quad (1)$$

$$= \sum_{n=1}^{46} 1 + \sum_{n=0}^{\infty} \frac{2^{47}}{2^{n+47}} \quad (2)$$

$$= 46 + \sum_{n=0}^{\infty} \frac{1}{2^n} \quad (3)$$

$$= 46 + \frac{1}{1 - \frac{1}{2}} \quad (4)$$

$$= 48 \quad (5)$$

(3) geometrische Reihe  $\sum_{k=0}^{\infty} q^n = \frac{1}{1-q}$

Der erwartete Gewinn beträgt also 48.

## Aufgabe 2

a)

D1: 6, D2: 2, Sum: 8

D1: 5, D2: 1, Sum: 6

D1: 5, D2: 3, Sum: 8

Gewonnen: true, Gewinn: 1

D1: 2, D2: 1, Sum: 3

Gewonnen: false, Gewinn: 0

D1: 4, D2: 2, Sum: 6

D1: 3, D2: 4, Sum: 7

Gewonnen: false, Gewinn: -1

D1: 2, D2: 2, Sum: 4

D1: 1, D2: 2, Sum: 3

D1: 5, D2: 1, Sum: 6

D1: 5, D2: 5, Sum: 10

D1: 1, D2: 6, Sum: 7

Gewonnen: false, Gewinn: -2  
D1: 2, D2: 3, Sum: 5  
D1: 1, D2: 6, Sum: 7  
Gewonnen: false, Gewinn: -3  
D1: 2, D2: 2, Sum: 4  
D1: 6, D2: 4, Sum: 10  
D1: 1, D2: 2, Sum: 3  
D1: 2, D2: 5, Sum: 7  
Gewonnen: false, Gewinn: -4  
D1: 3, D2: 3, Sum: 6  
D1: 2, D2: 6, Sum: 8  
D1: 3, D2: 2, Sum: 5  
D1: 3, D2: 4, Sum: 7  
Gewonnen: false, Gewinn: -5  
D1: 6, D2: 3, Sum: 9  
D1: 3, D2: 4, Sum: 7  
Gewonnen: false, Gewinn: -6  
D1: 2, D2: 1, Sum: 3  
Gewonnen: false, Gewinn: -7  
D1: 2, D2: 5, Sum: 7  
Gewonnen: true, Gewinn: -6

Mittlerer Gewinn:  $\frac{-6}{10} = -0.6$

b)

Programm in Rust:

```
1 fn main() {
2     let sample_size = 1000;
3
4     let money: i64 = (0..sample_size)
5         .map(|_| match roll_dice(2) {
6             7 | 11 => 1,
7             2 | 3 | 12 => -1,
8             s => loop {
9                 match roll_dice(2) {
10                    7 => return -1,
11                    n => if n == s { return 1; }
12                }
13            }
14        })
15        .sum();
16
17     println!("Mittlerer Gewinn: {:.2}", money as f64 / sample_size as f64)
18 }
19
20 fn roll_dice(number: usize) -> u8 {
21     (0..number).map(|_| rand::random::<u8>() % 6 + 1).sum()
22 }
```

Output:

Mittlerer Gewinn: -0.02

### Aufgabe 3

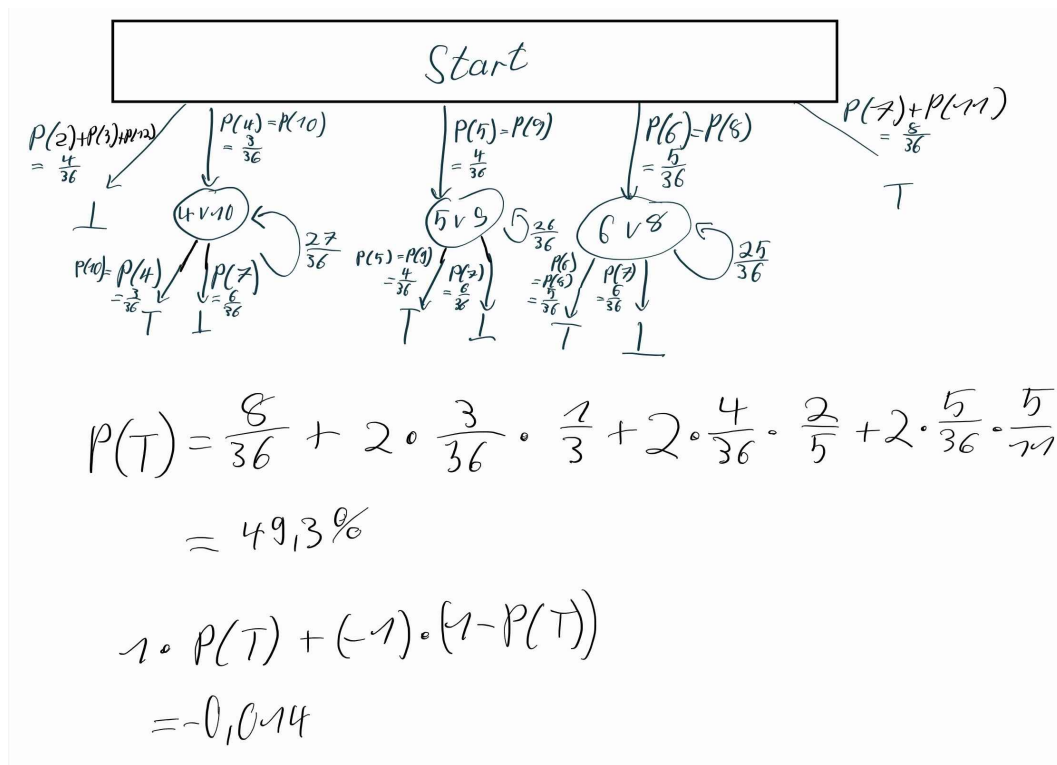


Abbildung 1: Wahrscheinlichkeitsbaum