# Diskrete Modellierung

Wintersemester 2017/18

Mario Holldack, M. Sc.
Prof. Dr. Georg Schnitger
Hannes Seiwert, M. Sc.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

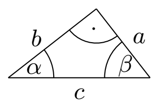
# Übungsblatt 6

Ausgabe: 23.11.17
Abgabe: 30.11.17

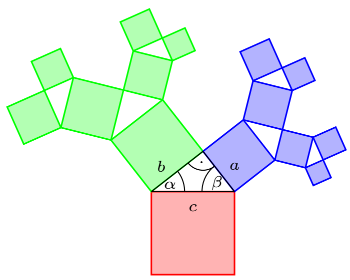
**Aufgabe 6.1** *Pythagoras-Bäume* (12 + 13 = 25 Punkte)

Gegeben sei ein Winkel $\alpha$ zwischen $1°$ und $89°$.

Für ein $n \in \mathbb{N}$ wird der $n$-te *Pythagoras-Baum* $P_n$ folgendermaßen konstruiert:




- $P_0$ ist ein Quadrat mit Kantenlänge 1.
- Um $P_{n+1}$ zu erhalten, gehe wie folgt vor (vgl. die Skizze rechts).
  - Zeichne ein Dreieck mit den Winkeln $90°$, $\alpha$ und $\beta := 90° - \alpha$ sowie den Kantenlängen $c = 1$, $b = \cos(\alpha)$ und $a = \sin(\alpha)$.
  - Setze an die Kante $c$ ein Quadrat mit Seitenlänge 1.
  - Setze an die Kante $b$ einen Pythagoras-Baum $P_n$, dessen Kantenlängen um den Faktor $\cos(\alpha)$ verkleinert sind.
  - Setze an die Kante $a$ einen Pythagoras-Baum $P_n$, dessen Kantenlängen um den Faktor $\sin(\alpha)$ verkleinert sind.



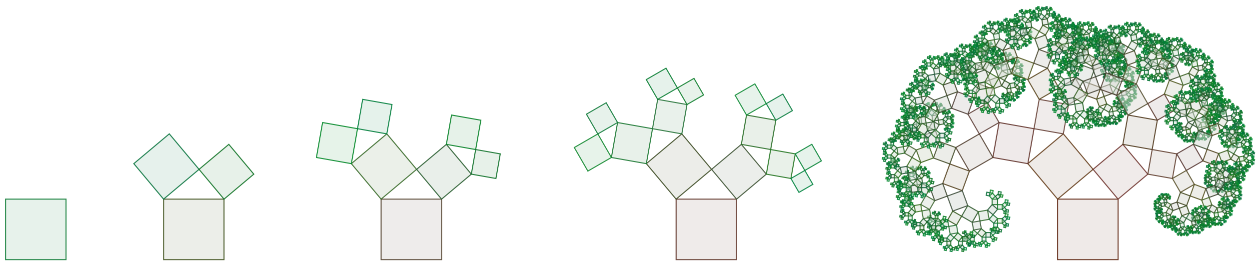



**Abbildung 1:** von links nach rechts: $P_0, P_1, P_2, P_3$ und $P_{11}$, jeweils mit $\alpha = 40°$.

a) Sei $E_n$ die Anzahl der Ecken des $n$-ten Pythagoras-Baums. (Z.B. gilt $E_0{=}4, E_1{=}9, E_2{=}19$.)
   i) Geben Sie eine Rekursionsgleichung für $E_n$ an.
   ii) Lösen Sie die Rekursionsgleichung aus Teil i), d. h. geben Sie einen geschlossenen (nicht-rekursiven) Ausdruck für $E_n$ an.
   *Hinweis*: Überprüfen Sie die Korrektheit Ihrer Lösung für kleine $n$. Die geometrische Reihe könnte sich als hilfreich erweisen.

b) Sei $A_n$ der Flächeninhalt des $n$-ten Pythagoras-Baumes, wobei wir nur die Flächen in den Quadraten, nicht aber in den Dreiecken zählen.
   i) Geben Sie eine Rekursionsgleichung für $A_n$ an.
   ii) Lösen Sie die Rekursionsgleichung aus Teil i).
   *Hinweis*: Für beliebige Winkel $\alpha$ gilt der Satz des Pythagoras: $(\cos(\alpha))^2 + (\sin(\alpha))^2 = 1$.

*Hinweis*: In allen Aufgaben sollen (kurze) Begründungen angegeben werden. Ein formaler Beweis mit Induktion wird nicht verlangt.

Für große $n$ kann es dazu kommen, dass sich Ecken oder Flächen der Bäume überschneiden. Dieser Umstand soll in allen Teilaufgaben ignoriert werden.

## **Aufgabe 6.2** *Rekursionsgleichungen und Induktion* $(10 + 10 + 10 = 30$ Punkte)

a) Lösen Sie die folgenden Rekursionsgleichungen, d. h. finden Sie jeweils einen (möglichst einfachen) geschlossenen Ausdruck für $a_n$ und geben Sie jeweils eine kurze Begründung an. Sie müssen Ihre Lösungen nicht durch vollständige Induktion beweisen.

*Beispiel.* Es gelte $a_1 := 1$ und $a_{n+1} := a_n + 1$ f.a. $n \in \mathbb{N}_{>0}$. Dann lautet die *Lösung* der Rekursionsgleichung:

$$a_n = n$$

Begründung: $a_n = a_{n-1} + 1 = a_{n-2} + 1 + 1 = \cdots = a_{n-(n-1)} \underbrace{+1 + 1 + \cdots + 1}_{(n-1)\text{-mal}} = \underbrace{a_1}_{=1} + n - 1 = n$

i) $a_1 := 1$ und $a_{n+1} := 2 + a_n$ f.a. $n \in \mathbb{N}_{>0}$

ii) $a_0 := 0$, $a_1 := 2$ und $a_{n+1} := 2 \cdot a_{n-1}$ f.a. $n \in \mathbb{N}_{>0}$

iii) $a_2 := 1$ und $a_{n+1} := a_n \cdot \frac{n+1}{n-1}$ f.a. $n \in \mathbb{N}_{>0}$ mit $n \geq 2$

b) Zeigen Sie mit vollständiger Induktion nach $n$:

Für alle $n \in \mathbb{N}$ und $x \in \mathbb{R}$ mit $x \geq -1$ gilt $(1 + x)^n \geq 1 + nx$.

c) Für alle $n \in \mathbb{N}_{>0}$ definieren wir die aussagenlogische Formel $\varphi_n$ rekursiv.

REKURSIONSANFANG: $\varphi_1 := V_1$

REKURSIONSSCHRITT: $\varphi_{n+1} := (\varphi_n \leftrightarrow V_{n+1})$ für alle $n \in \mathbb{N}_{>0}$

Zeigen Sie mit vollständiger Induktion nach $n$:

Für alle $n \in \mathbb{N}_{>0}$ gilt

$$\varphi_n \equiv \begin{cases} \bigoplus_{i=1}^{n} V_i, & \text{falls } n \text{ ungerade} \\ \neg \bigoplus_{i=1}^{n} V_i, & \text{falls } n \text{ gerade.} \end{cases}$$

## **Aufgabe 6.3** *Korrektheit rekursiver Programme beweisen* (20 Punkte)

Wir betrachten die *Russische Bauernmultiplikation*[1] zum Berechnen eines Produktes $x \cdot k$, wobei $x \in \mathbb{R}$ und $k \in \mathbb{N}$.

```
def prod(x,k):
    if k == 0:
        return 0
    elif k % 2 == 0:                # k ist gerade und groesser 0
        return prod(2*x, k/2)
    else:                           # k ist ungerade
        return prod(2*x, k//2) + x  # k//2 entspricht der
                                    # ganzzahligen Division (k-1)/2
```

Hierbei wird ausgenutzt, dass eine *Verdopplung* von $x$ (bzw. eine Halbierung von $k$) bei binärer Darstellung relativ einfach ist: ein Bitshift genügt. Wir wollen uns nun von der Korrektheit des Verfahrens überzeugen.

Zeigen Sie mit vollständiger Induktion nach $k$:

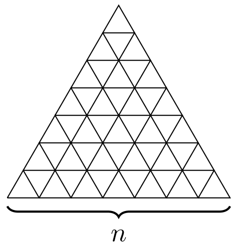
Für alle $x \in \mathbb{R}$ und alle $k \in \mathbb{N}$ gilt: $x \cdot k =$ `prod(x,k)`

**Bitte wenden!**

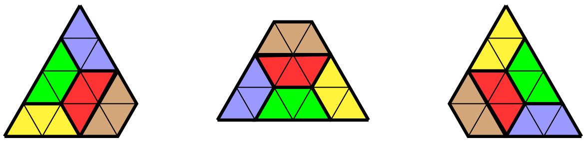
[1] Diese Methode war sogar bereits vor über 3500 Jahren im Alten Ägypten bekannt: https://en.wikipedia.org/wiki/Rhind_Mathematical_Papyrus

**Aufgabe 6.4** *Dreiecke kacheln* (10 + 15 = 25 Punkte)

Gegeben sei ein gleichseitiges Dreieck $D_n$ mit Kantenlänge $n$, das wiederum aus $n^2$ gleichseitigen Dreiecken $D_1$ mit Kantenlänge 1 zusammengesetzt ist. Wir nennen ein Dreieck $D_1$ kurz *Feld*.
Uns stehen Kacheln der Form  zur Verfügung. Wir wollen alle Felder des Dreiecks $D_n$ – bis auf eines – mit Kacheln überdecken. In einer *legalen Kachelung* dürfen Kacheln gedreht werden, aber sich nicht überlappen.




Beispiel: $D_n$ für $n = 7$.



Legale Kachelungen des Dreiecks $D_4$, bei dem jeweils ein Feld entfernt worden ist.

a) Zeigen Sie: Das Dreieck $D_6$ (mit Kantenlänge $n$=6) besitzt keine legale Kachelung, egal welches Feld entfernt wird.

*Hinweis*: Das Dreieck $D_n$ ist aus $n^2$ Feldern zusammengesetzt.

b) Zeigen Sie mit vollständiger Induktion:

Für jedes $k \in \mathbb{N}_{>0}$ besitzt das Dreieck $D_n$ mit $n = 2^k$ eine legale Kachelung, wenn ein beliebiges Feld an einer Ecke entfernt wird.

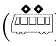
*Hinweis*: Beschreiben Sie im Induktionsschritt, wie Sie die Kacheln platzieren. Wohin legen Sie Ihre erste Kachel?

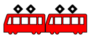
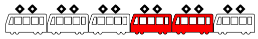
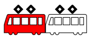
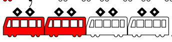
**Bitte wenden!**

## **Aufgabe 6.5*** *Bonusaufgabe* ((8* + 2*) + 10* = 20* Extrapunkte)

*Diese Aufgabe ist eine Bonusaufgabe, in der Sie Extrapunkte erwerben können.*

a) Im Rahmen einer Werbeaktion lässt die Verwaltung der Stadt Frankfurt (passend zu den Wappenfarben) rot-weiße Straßenbahnen durch Frankfurt fahren. Dabei gilt:

*Rote* Wagons () dürfen nur in *gerader* Anzahl direkt hintereinander in einer Straßenbahn vorkommen, *weiße* Wagons () nur in *ungerader* Anzahl.

*Beispiel*: Die Straßenbahnen , ,  und  sind erlaubt, nicht jedoch  oder .

Sei $n \in \mathbb{N}_{>0}$. Mit $r_n$ bzw. $w_n$ bezeichnen wir die Anzahl der möglichen unterschiedlichen Straßenbahnen, die mit einem roten bzw. weißen Wagon enden und aus genau $n$ Wagons bestehen.

i) Stellen Sie Rekursionsgleichungen für $r_n$ und $w_n$ auf.

ii) Wie viele verschiedene rot-weiße Straßenbahnen können mit $1, 2, 3, 4$ bzw. $5$ Wagons zusammengestellt werden, wenn die obigen Regeln beachtet werden?

*Hinweis*: Auch einfarbige Straßenbahnen sind zu berücksichtigen.

b) Sei $n \geq 2$ eine Zweierpotenz. Gegeben seien $n$ ganze Zahlen `A[1]`, ..., `A[n]`. Wir betrachten die Funktion `min_and_max`, die sowohl die kleinste als auch die größte der Zahlen zurückgibt:

```
1      def min_and_max(left, right):
2          if right-left <= 1:
3              if A[right] < A[left]:                 # ein Vergleich
4                  minimum = A[right]
5                  maximum = A[left]
6              else:
7                  minimum = A[left]
8                  maximum = A[right]
9          else:
10             middle = (left+right-1) // 2
11
12             (min_left,  max_left)  = min_and_max(left, middle)
13             (min_right, max_right) = min_and_max(middle+1, right)
14
15             minimum = min(min_left, min_right)     # ein Vergleich
16             maximum = max(max_left, max_right)     # ein Vergleich
17
18         return (minimum, maximum)
19
```

Sei $V_n$ die Anzahl der von `min_and_max(1,n)` in den Zeilen 3, 15 und 16 durchgeführten Vergleiche zwischen den Zahlen `A[1]`, ..., `A[n]`. (Die Funktionen `min` und `max` führen jeweils einen Vergleich aus.)

i) Bestimmen Sie eine Rekursionsgleichung für $V_n$.

ii) Zeigen Sie: Für jedes $k \in \mathbb{N}_{>0}$ mit $n = 2^k$ gilt $V_n = \frac{3}{2} \cdot n - 2$.