7.3 Erzeugendenfunktionen

Für lineare und nicht lineare Rekursionsgleichungen erhält man oft eine Lösung, indem man die f_n als Koeffizienten einer Potenzreihe betrachtet und eine geschlossene Form der dadurch definierten Funktion sucht.

Definition 6 (Erzeugendenfunktion)

Sei die Folge $(f_n)_{n\geq 0}$ gegeben. Die zugehörige

- (gewöhnliche) Erzeugendenfunktion ist

$$F(z) := \sum_{n=0}^{\infty} f_n z^n; \qquad z \in \mathbb{C};$$

- exponentielle Erzeugendenfunktion ist

$$F(z) = \sum_{n\geq 0} \frac{f_n}{n!} z^n; \qquad z \in \mathbb{C}.$$

Beispiel 7

1. Die Erzeugendenfunktion der Folge $(1, 0, 0, \ldots)$ ist

$$F(z) = 1\,.$$

2. Die Erzeugendenfunktion der konstanten Folge $(1, 1, 1, \ldots)$ ist

$$F(z) = \frac{1}{1-z}\,.$$

Falls $F(z) = \sum_{n>0} f_n z^n$, bezeichnet

$$[z^n]F(z)$$

den n-ten Koeffizienten f_n.

Sei $F(z) = \sum_{n \geq 0} f_n z^n$ und $G(z) = \sum_{n \geq 0} g_n z^n$.

ErzFkt.	n-tes Folgenglied	Anmerkungen:
cF	cf_n	
$F + G$	$f_n + g_n$	
$F \cdot G$	$h_n := \sum_{i=0}^{n} f_i g_{n-i}$ (Konvolution)	$\left(\sum_{i \geq 0} f_i z^i\right)\left(\sum_{i \geq 0} g_i z^i\right) = \sum_{i \geq 0} h_i z^i$ (mit $h_n = \sum_{i=0}^{n} f_i g_{n-i}$)
$z^k F$	if $n < k$ then 0 else f_{n-k} fi	
$\frac{F(z)}{1-z}$	$\sum_{i=0}^{n} f_i$	$\frac{1}{1-z} = \sum_{n \geq 0} z^n$
$z\frac{dF(z)}{dz}$	nf_n	
$\int_0^x F(t)dt$	if $n = 0$ then 0 else $\frac{f_{n-1}}{n}$ fi	$f_n z^n$ geht über auf $f_n \frac{z^{n+1}}{n+1}$
$F(cz)$	$c^n f_n$	$F(cz) = \sum_{n \geq 0} f_n c^n z^n$

Beispiel 8

$$F(z) := \sum_{n \geq 0} 2^n z^n = \frac{1}{1-2z}$$

$$G(z) := \sum_{n \geq 0} n z^n = \frac{z}{(1-z)^2}$$

$$\Rightarrow F(z)G(z) = \frac{z}{(1-z)^2(1-2z)} = \sum_{n \geq 0} \sum_{i=0}^{n} (n-i) 2^i z^n$$

Partialbruchzerlegung:

$$\frac{z}{(1-z)^2(1-2z)} = \overbrace{\frac{-2}{(1-z)}}^{(1)} + \frac{-z}{(1-z)^2} + \overbrace{\frac{2}{1-2z}}^{(2)}$$
$$= \underbrace{\sum_{n\geq 0} 2^{n+1} z^n}_{(2)} - \sum_{n\geq 0} n z^n - \underbrace{2\sum_{n\geq 0} z^n}_{(1)} ;$$

Also:

$$\sum_{i=0}^{n} 2^i (n-i) = [z^n](FG)(z)$$
$$= [z^n]\left(\sum_{n\geq 0}(2^{n+1} - n - 2)z^n\right)$$
$$= 2^{n+1} - n - 2 \,.$$

7.4 Transformation des Definitions- bzw. Wertebereichs

Beispiel 9

$$f_0 = 1$$
$$f_1 = 2$$
$$f_n = f_{n-1} \cdot f_{n-2} \text{ für } n \geq 2 \, .$$

Setze

$$g_n := \log f_n \, .$$

Dann gilt

$$g_n = g_{n-1} + g_{n-2} \text{ für } n \geq 2$$
$$g_1 = \log 2 = 1, \; g_0 = 0 \text{ (für } \log = \log_2 \text{)}$$
$$g_n = F_n \text{ (n-te Fibonacci-Zahl)}$$
$$f_n = 2^{F_n}$$

Beispiel 10

$$f_1 = 1$$
$$f_n = 3f_{\frac{n}{2}} + n; \text{ für } n = 2^k \ ;$$

Setze

$$g_k := f_{2^k} \ .$$

Beispiel 10

Dann gilt:

$$g_0 = 1$$
$$g_k = 3g_{k-1} + 2^k,\ k \geq 1$$

Damit ergibt sich:

$$g_k = 3^{k+1} - 2^{k+1},\ \text{also}$$
$$f_n = 3 \cdot 3^k - 2 \cdot 2^k$$
$$= 3(2^{\log 3})^k - 2 \cdot 2^k$$
$$= 3(2^k)^{\log 3} - 2 \cdot 2^k$$
$$= 3n^{\log 3} - 2n\ .$$

Kapitel II Höhere Datenstrukturen

1. Grundlegende Operationen

Es sei U das Universum von Schlüsseln mit einer (totalen) Ordnung $\leq$. $S \subseteq U$ sei eine Teilmenge der Schlüssel. Gegeben seien eine Menge von Datensätzen $x_1, \cdots, x_n$, wobei jeder Datensatz x durch einen Schlüssel $k(x) \in S$ gekennzeichnet ist. Jeder Datensatz x besteht aus seinem Schlüssel $k(x)$ und seinem eigentlichen Wert $v(x)$.

IsElement(k, S):	ist $k \in S$, wenn ja, return $v(k)$
Insert(k, S):	$S := S \cup \{k\}$
Delete$(k; S)$:	$S := S \setminus \{k\}$
FindMin(S):	return $\min S$
FindMax(S):	return $\max S$
DeleteMin(S):	$S := S \setminus \min S$
ExtractMin(S):	return $\min S$, $S := S \setminus \min S$
DecreaseKey(k, Δ, S):	ersetze k durch $k - \Delta$
Union(S_1, S_2):	$S_1 := S_1 \cup S_2$
Find(k):	falls $k \in S$, so finde x mit $k = k(x)$
Merge(S_1, S_2):	$S_1 := S_1 \cup S_2$, falls $S_1 \cap S_2 = \emptyset$
Split(S_1, k, S_2):	$S_2 := \{k' \in S_1 \mid k' \geq k\}$ $S_1 = \{k' \in S_1 \mid k' < k\}$
Concatenate(S_1, S_2):	$S_1 := S_1 \cup S_2$; Vorauss.: *FindMax*$(S_1) \leq$ *FindMin*(S_2)

Datenstrukturklasse	mindestens angebotene Funktionen	realisiert in
Wörterbuch (Dictionary)	*IsElement()*, *Insert()*, *Delete()*	Hashtable, Suchbäume
Vorrangwarteschlange (Priority Queue)	*FindMin()*, *Insert()*, *Delete()*, *[IsElement()]*	balancierte, leftist Bäume, Binomial Queues
Mergeable heaps	*FindMin()*, *Insert()*, *Delete()*, *Merge()*	2-3-Bäume, Binomial Queues, Leftist-Bäume
Concatenable queues	*FindMin()*, *Insert()*, *Delete()*, *Concatenate()*	2-3-Bäume