

## Föreläsning 6: Fortsättning på genererande funktioner · 1MA020

Vilhelm Agdur<sup>1</sup>

<sup>1</sup> vilhelm.agdur@math.uu.se

7 februari 2023

Vi fortsätter förra föreläsningens diskussion om genererande funktioner, och använder dem för att räkna lösningar till ekvationer. Sedan introducerar vi exponentiella genererande funktioner, och använder dessa för att lösa några problem.

### Antal lösningar till en ekvation, med begränsningar

I slutet på förra föreläsningen studerade vi antalet lösningar till ekvationen

$$x_1 + x_2 + x_3 + x_4 + x_5 = k$$

om vi kräver att alla  $x_i$  är icke negativa heltal. Det var ett första exempel på en mer generell kategori av problem med att räkna lösningar på ekvationer. Låt oss börja med ett lite mer invecklat problem:

**Exempel 1.** Hur många lösningar finns det till

$$x_1 + x_2 + x_3 + x_4 = k$$

om vi kräver att alla  $x_i$  är icke negativa heltal, men också kräver att  $x_2$  är jämnt, att  $x_3 \leq 10$ , och  $x_4$  är udda?

Låt, för varje  $k$ ,  $a_k$  vara antalet sådana lösningar. Låt sedan  $a_k^1$  vara antalet lösningar till  $x_1 = k$  i icke negativa heltal,  $a_k^2$  vara antalet lösningar till  $x_2 = k$  i icke negativa jämna heltal,  $a_k^3$  vara antalet lösningar till  $x_3 = k$  i heltal mellan 0 och 10, och  $a_k^4$  vara antalet lösningar till  $x_4 = k$  i udda heltal.

Precis som i förra exemplet studerar vi nu faltningen av dessa fyra följder, och ser att

$$(a^1 * a^2 * a^3 * a^4)_k = \sum_{\substack{k_1, k_2, k_3, k_4 \geq 0 \\ k_1 + k_2 + k_3 + k_4 = k}} a_{k_1}^1 a_{k_2}^2 a_{k_3}^3 a_{k_4}^4 = a_k$$

eftersom  $a_{k_1}^1 a_{k_2}^2 a_{k_3}^3 a_{k_4}^4$  är en produkt av ett och nollor – att  $k_1 + k_2 + k_3 + k_4 = k$  garanteras av definitionen av faltning, och sedan är varje term i produkten ett om värdet på  $k_i$  är tillåtet av våra begränsningar, och noll annars. Så produkten är ett om summan är korrekt och varje enskild begränsning är uppfylld.

Så precis som i förra exemplet kan vi få fram genererande funktionen för  $a_k$ , följden vi faktiskt är intresserade av, genom att plocka fram den genererande funktionen för de enklare följderna.

Vad genererande funktionen för  $a^1$  är vet vi sedan innan – den är bara en följd av ettor, så dess genererande funktion blir  $\frac{1}{1-x}$ . Likaledes vet vi sedan innan att följden av  $n$  stycken ettor och sedan nollor har genererande funktion  $\frac{1-x^{n+1}}{1-x}$ , så genererande funktionen för  $a^3$  blir  $\frac{1-x^{11}}{1-x}$ .

Däremot för  $a^2$  behöver vi räkna ut något nytt, nämligen den genererande funktionen för följden  $1, 0, 1, 0, 1, \dots$ , indikatorfunktionen av de jämna talen. Så vi får skriva att

$$\begin{aligned} F_{a^2}(x) &= \sum_{k=0}^{\infty} a_k^2 x^k \\ &= \sum_{\substack{k \geq 0 \\ k \in 2\mathbb{Z}}} x^k \\ &= \sum_{i=0}^{\infty} x^{2i} \\ &= \sum_{i=0}^{\infty} (x^2)^i \end{aligned}$$

och sista raden här kan vi känna igen som genererande funktionen av följden  $(1, 1, 1, 1, \dots)$ , utvärderad i  $x^2$ . Så detta är lika med  $\frac{1}{1-x^2}$ .

Så vad som återstår är alltså  $a^4$ , indikatorfunktionen för de udda talen. För att få fram dess genererande funktion kan vi använda vad vi just gjorde för de jämna talen:

$$\begin{aligned} F_{a^4}(x) &= \sum_{k=0}^{\infty} a_k x^k \\ &= \sum_{\substack{k \geq 1 \\ k \text{ udda}}} x^k \\ &= x \sum_{\substack{k \geq 1 \\ k \text{ udda}}} x^{k-1} \\ &= x \sum_{\substack{k \geq 0 \\ k \in 2\mathbb{Z}}} x^k \\ &= \frac{x}{1-x^2}. \end{aligned}$$

Så, om vi använder att genererande produkten av en faltning är produkten av de genererande funktionerna, ser vi att

$$\begin{aligned} F_a(x) &= \left( \frac{1}{1-x} \right) \left( \frac{1-x^{11}}{1-x} \right) \left( \frac{1}{1-x^2} \right) \left( \frac{x}{1-x^2} \right) \\ &= \frac{x(1-x^{11})}{(1-x)^2(1-x^2)^2} \end{aligned}$$

och ber vi vårt favorit-CAS<sup>2</sup> att Taylorutvidga detta uttryck så får vi att

$$F_a(x) = x + 2x^2 + 5x^3 + 8x^4 + 14x^5 + 20x^6 + 30x^7 + 40x^8 + \dots$$

<sup>2</sup> Computer Algebra System, alltså till exempel *WolframAlpha* eller något av dess öppna alternativ, såsom *Sage*.

så att följderna av antalet lösningar är

$$0, 1, 2, 5, 8, 14, 20, 30, 40, 55, 70, 91, 111, 138, 163, \dots$$

**Exempel 2.** Vi vill räkna antalet lösningar  $a_k$  till ekvationen

$$2x_1 + x_2 + x_3 = k$$

där alla  $x_i$  är heltal,  $x_2$  är en multipel av 6, och talet  $x_3$  kan vara antingen rött eller blått.<sup>3</sup>

Vi börjar med att göra variabelbytet  $y_1 = 2x_1$ , och vill alltså nu ha lösningar till  $y_1 + x_2 + x_3 = k$ , med begränsningen att  $y_1$  är jämnt. Det här förändrar så klart inte antalet lösningar, bara gör det lättare för oss att tillämpa vår metod.

Vi tillämpar samma metod som i förra exemplet, och låter  $a_k^1$  vara antalet lösningar till  $y_1 = k$  med  $y_1$  jämnt,  $a_k^2$  vara antalet lösningar till  $x_2 = k$  med  $x_2$  delbart med 6, och  $a_k^3$  vara antalet lösningar till  $x_3 = k$  med  $x_3$  färgat antingen rött eller blått. Faltningen blir då

$$(a^1 * a^2 * a^3)_k = \sum_{\substack{k_1, k_2, k_3 \geq 0 \\ k_1 + k_2 + k_3 = k}} a_{k_1}^1 a_{k_2}^2 a_{k_3}^3 = a_k.$$

Vi fortsätter precis som innan med att räkna ut den genererande funktionen för varje av våra följder. För  $a^1$  vet vi redan vad genererande funktionen för indikatorfunktionen av de jämna talen är, nämligen  $\frac{1}{1-x^2}$ .

För  $a^2$  kan vi använda samma metod som vi använde för de jämna talen för att se att

$$\begin{aligned} F_{a^2}(x) &= \sum_{k=0}^{\infty} a_k^2 x^k = \sum_{\substack{k \geq 0 \\ k \in 6\mathbb{Z}}} x^k \\ &= \sum_{i=0}^{\infty} x^{6i} = \sum_{i=0}^{\infty} (x^6)^i \end{aligned}$$

så att  $F_{a^2}(x) = \frac{1}{1-x^6}$ .

För  $a^3$  så blir denna helt enkelt en följd av bara tvåor, eftersom vi har två val för färg för varje tal, och kan välja vilket tal som helst. Så vi ser att

$$F_{a^3}(x) = \sum_{k=0}^{\infty} 2x^k = 2 \frac{1}{1-x}.$$

Sammantaget har vi alltså att

$$F_a(x) = \left( \frac{1}{1-x^2} \right) \left( \frac{1}{1-x^6} \right) \left( \frac{2}{1-x} \right) = \frac{2}{(1-x)(1-x^2)(1-x^6)}$$

vilket vi kan Taylorutvidga i vårt favoritprogram och få att<sup>4</sup>

$$\begin{aligned} F_a(x) &= 2 + 2x + 4x^2 + 4x^3 + 6x^4 + 6x^5 + 10x^6 + 10x^7 \\ &\quad + 14x^8 + 14x^9 + 18x^{10} + 18x^{11} + 24x^{12} + 24x^{13} \\ &\quad + 30x^{14} + 30x^{15} + 36x^{16} + 36x^{17} + 44x^{18} + \dots \end{aligned}$$

<sup>3</sup> Vi ser alltså, för  $k = 6$ , alla dessa som godtagbara distinkta lösningar:

$$\begin{aligned} x_1 = 1, x_2 = 0, x_3 = 4, & \quad x_1 = 1, x_2 = 0, x_3 = 4, \\ x_1 = 2, x_2 = 0, x_3 = 2, & \quad x_1 = 0, x_2 = 6, x_3 = 0. \end{aligned}$$

<sup>4</sup> Vi ser ju ett tydligt mönster här av att  $a_{2k} = a_{2k+1}$ . Kan du förklara varför detta måste vara fallet, baserat på våra begränsningar av variablerna?

### Att omvandla ett problem till en rekursion

Vårt nästa exempel handlar inte om att räkna lösningar till en ekvation, utan om att översätta ett problem till en rekursion. Vi tar detta exemplet nu delvis eftersom vi kommer vilja ha en rekursion att lösa i ett senare exempel.

**Exempel 3.** Antag att vi har ett schackbräde med  $2 \times n$  rutor, och vi vill täcka det med brickor av formen  $2 \times 1$  eller  $1 \times 2$ . Hur många sätt kan vi göra detta på?

Låt  $t_n$  vara antalet sätt vi kan täcka vårt schackbräde. Vi vill hitta en rekursion för detta antal. Vi ser enkelt att det finns ett enda sätt att göra det för  $n = 1$  – bara en bricka får plats – så  $t_1 = 1$ . För  $n = 2$  finns det två sätt, antingen lägger vi dem horisontellt eller vertikalt, så  $t_2 = 2$ .

Om vi vill skapa oss en täckning av en  $2 \times n$ -bräda, för något  $n > 2$ , kan vi göra på två sätt:

- Vi börjar med en täckning av en  $2 \times (n - 1)$ -bräda, och lägger till en till bricka vertikalt.
- Vi börjar med en täckning av en  $2 \times (n - 2)$ -bräda, och lägger till två till brickor horisontellt.

Att varje täckning av en  $2 \times n$ -bräda kan skapas på detta vis är enkelt att se – antingen är den sista kolumnen täckt av en vertikal bricka, i vilket fall vi skapade täckningen på första viset, eller så är den täckt av två horisontella brickor, i vilket fall vi skapade den på det andra sättet.

Alltså har vi funnit följande rekursion för antalet täckningar

$$t_0 = 0, t_1 = 1, t_2 = 2, \quad t_n = t_{n-1} + t_{n-2} \quad \forall n > 2$$

som ju är extremt lik den vi har för Fibonaccitalen, så vi kan finna en genererande funktion och sluten form på precis samma vis som i det fallet.

### Exponentiella genererande funktioner

Ibland får vi problem med att hitta enkla uttryck för våra genererande funktioner, eftersom vår följd växer för snabbt. Hittills har vi bara studerat följder som växer långsamt nog, men om vi till exempel hade velat studera följden  $0!, 1!, 2!, \dots$  hade dess vanliga genererande funktion varit

$$\sum_{k=0}^{\infty} k! x^k$$

för vilken det inte finns något enkelt uttryck.<sup>5</sup>



Figur 1: Ett sätt att göra detta då  $n = 8$ . Figur tagen ur förra årets anteckningar.



Figur 2: De två sätten att göra det på då  $n = 2$ . Figur från förra årets föreläsninganteckningar.

<sup>5</sup> Om vi sätter på oss våra analytiker-glasögon kan vi dessutom se att detta uttryck inte konvergerar för något  $x > 0$ , så vad hade vi ens kunna ha för uttryck för en funktion som är oändlig överallt?

Ett annat exempel på detta är om vi räknar permutationer – dessa kommer vara ungefär  $k!$  stycken, i de flesta av våra exempel. Så vi hade haft samma problem. Det finns en anledning att vi hittills bara studerat binomialkoefficienter, som ju är betydligt mindre.

Så, låt oss definiera en variant på genererande funktioner som kan hantera dessa snabbväxande följder.

**Definition 4.** Om  $\{a_k\}_{k=0}^\infty$  är någon följd ges dess *exponentiella genererande funktion* av<sup>6</sup>

$$EG_a(x) = \sum_{k=0}^{\infty} a_k \frac{x^k}{k!}.$$

Så allt vi faktiskt har gjort är att skala ner vår följd så att den inte växer så kraftigt. Som tur är fungerar detta enkla trick väldigt väl – låt oss räkna några exempel för att se hur.

**Exempel 5.** Den exponentiella genererande funktionen för följden  $(1, 1, 1, 1, \dots)$  ges av

$$\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x.$$

**Exempel 6.** Den exponentiella genererande funktionen för följden  $(0!, 1!, 2!, 3!, \dots)$  ges av

$$\sum_{k=0}^{\infty} k! \frac{x^k}{k!} = \sum_{k=0}^{\infty} x^k = \frac{1}{1-x}.$$

**Exempel 7.** Fixera något heltal  $n$ , och låt  $a_k$  vara antalet permutationer av längd  $k$  ur ett alfabet med  $n$  bokstäver, så att  $a_k = \frac{n!}{(n-k)!}$ . Då ges den exponentiella genererande funktionen för  $a$  av

$$\begin{aligned} EG_a(x) &= \sum_{k=0}^{\infty} \frac{n!}{(n-k)!} \frac{x^k}{k!} \\ &= \sum_{k=0}^{\infty} \frac{n!}{k!(n-k)!} x^k \\ &= \sum_{k=0}^{\infty} \binom{n}{k} x^k = (1+x)^n \end{aligned}$$

där vi i sista ledet kände igen en *ordinär* genererande funktion för binomialkoefficienterna.

**Exempel 8.** Låt oss återvända till exemplet med schackbräderna. Vi hade en följd som gavs av rekursionen<sup>7</sup>

$$t_0 = 0, t_1 = 1, t_2 = 2, \quad t_{k+2} = t_{k+1} + t_k \quad \forall k \geq 1.$$

Vad är den exponentiella genererande funktionen för denna följden? Analogt med det ordinära fallet tar vi vår rekursion, multiplicerar den med  $\frac{x^k}{k!}$ , och summerar över alla  $k \geq 1$ , för att få att

$$\sum_{k=1}^{\infty} t_{k+2} \frac{x^k}{k!} = \sum_{k=1}^{\infty} t_{k+1} \frac{x^k}{k!} + \sum_{k=1}^{\infty} t_k \frac{x^k}{k!}.$$

<sup>6</sup> Om vi behöver förtydliga skillnaden kallar vi de icke-exponentiella genererande funktionerna vi studerade innan för *ordinära* genererande funktioner.

<sup>7</sup> Notera att vi har ändrat om i rekursionen så att vi har plustecken i index istället för minus – det är fortfarande precis samma rekursion, men vi får finare uttryck senare.

Nu kan vi komma ihåg att

$$\frac{d}{dx} \frac{x^k}{k!} = \frac{x^{k-1}}{(k-1)!}$$

så att

$$t_{k+1} \frac{x^k}{k!} = t_{k+1} \left( \frac{d}{dx} \frac{x^{k+1}}{(k+1)!} \right)$$

och

$$t_{k+2} \frac{x^k}{k!} = t_{k+2} \left( \frac{d^2}{dx^2} \frac{x^{k+2}}{(k+2)!} \right).$$

Om vi använder dessa likheter i vad vi hade innan, och bryter ut derivatorna ur summorna, får vi alltså att

$$\frac{d^2}{dx^2} \left( \sum_{k=1}^{\infty} t_{k+2} \frac{x^{k+2}}{(k+2)!} \right) = \frac{d}{dx} \left( \sum_{k=1}^{\infty} t_{k+1} \frac{x^{k+1}}{(k+1)!} \right) + \left( \sum_{k=1}^{\infty} t_k \frac{x^k}{k!} \right).$$

Precis som i vårt exempel med Fibonaccitalen ser vi nu att vi nästan har de exponentiella genererande funktionerna, förutom att det saknas några av de första termerna i summorna. Så om vi justerar för detta får vi att

$$\begin{aligned} \frac{d^2}{dx^2} \left( EG_t(x) - t_0 \frac{x^0}{0!} - t_1 \frac{x^1}{1!} - t_2 \frac{x^2}{2!} \right) &= \\ &= \frac{d}{dx} \left( EG_t(x) - t_0 \frac{x^0}{0!} - t_1 \frac{x^1}{1!} \right) + \left( EG_t(x) - t_0 \frac{x^0}{0!} \right) \end{aligned}$$

så om vi substituerar in  $t_0 = 0, t_1 = 1, t_2 = 2$  och flyttar in derivatorna i parenteserna så får vi att

$$EG_t''(x) - 2 = EG_t'(x) - 1 + EG_t(x)$$

vilket ju är en helt vanlig andra gradens linjär ordinär differentialekvation med konstanta koefficienter. Randvillkoren för den ges av  $EG_t(0) = t_0$  och  $EG_t'(0) = t_1$ .<sup>8</sup>

Så vi kan antingen lösa den för hand eller med hjälp av vårt favorit-CAS, och finna lösningen

$$\frac{1}{10} \left( \sqrt{5} e^{\left(\frac{\sqrt{5}}{2} + \frac{1}{2}\right)x} - \sqrt{5} e^{\left(\frac{1}{2} - \frac{\sqrt{5}}{2}\right)x} + 5 e^{\left(\frac{\sqrt{5}}{2} + \frac{1}{2}\right)x} + 5 e^{\left(\frac{1}{2} - \frac{\sqrt{5}}{2}\right)x} - 10 \right)$$

vilket vår CAS<sup>9</sup> låter oss se kan Taylorutvidgas till

$$\sum_{n=1}^{\infty} \frac{(5 - \sqrt{5}) \left(\frac{1-\sqrt{5}}{2}\right)^n + (5 + \sqrt{5}) \left(\frac{1+\sqrt{5}}{2}\right)^n}{10} \frac{x^n}{n!}$$

vilket alltså ger oss en direkt formel för varje term i följderna, i samma still som för Fibonaccitalen.

<sup>8</sup> Detta kommer ju alltid gälla för alla exponentiella genererande funktioner, av hur de är definierade.

<sup>9</sup> Eller vi, för hand, om vi inte är lata. Den enda funktionen som är involverad är ju  $e^x$ , så det är inget svårt att expandera, bara många termer att hålla ordning på.

Vad vi sett i detta exemplet är alltså att vi, när vi har en felmatchning mellan index i vår följd och index i  $\frac{x^k}{k!}$ , plockar på oss derivator – till skillnad från i fallet med ordinära genererande funktioner, då detta gav oss faktorer av  $x$ .

Vi har också sett att de exponentiella genererande funktioner vi hittar för linjära rekursioner kommer ha en bunt olika  $e^{Cx}$  för konstanter  $C$ ,<sup>10</sup> vilket är mycket enklare att Taylorutvidga. Så om man vill lösa linjära rekursioner i praktiken är exponentiella genererande funktioner ofta ett bättre val än ordinära.

### Binomial-faltningar och EGfer

Vi har lärt oss att motsvarigheten till en faltning på den kombinatoriska sidan är en produkt av genererande funktioner. Vad händer om vi i stället tar *exponentiella* genererande funktioner? Situationen blir lite mer komplicerad.<sup>11</sup>

**Definition 9.** Antag att  $\{a_k\}_{k=0}^\infty$  och  $\{b_k\}_{k=0}^\infty$  är två följder. Då ges deras *binomial-faltning*  $a \otimes b$  av

$$(a \otimes b)_k = \sum_{i=0}^k \binom{k}{i} a_i b_{k-i}.$$

**Lemma 10.** Antag att  $\{a_k\}_{k=0}^\infty$  och  $\{b_k\}_{k=0}^\infty$  är två följder, vars exponentiella genererande funktioner är  $EG_a(x)$  och  $EG_b(x)$ . Då ges den genererande funktionen för binomial-faltningen  $a \otimes b$  av

$$EG_{a \otimes b}(x) = EG_a(x)EG_b(x)$$

*Bevis.* Lägg märke till att  $EG_a$  inte bara är den exponentiella genererande funktionen för  $a$ , utan också är den ordinära genererande funktionen för  $A_k = \frac{a_k}{k!}$ , och likaledes för  $b$ . Alltså måste, enligt vad vi redan vet om ordinära genererande funktioner,

$$EG_a(x)EG_b(x) = G_A(x)G_B(x) = G_{A*B}(x).$$

Så vad vi studerar är den vanliga faltningen av  $A$  med  $B$  – vi ser att denna är

$$\begin{aligned} (A * B)_k &= \sum_{i=0}^k A_i B_{k-i} \\ &= \sum_{i=0}^k \frac{a_i}{i!} \frac{b_{k-i}}{(k-i)!} \\ &= \sum_{i=0}^k \frac{k!}{i!(k-i)!} \frac{a_i b_{k-i}}{k!} \\ &= \sum_{i=0}^k \binom{k}{i} a_i b_{k-i} \frac{1}{k!} = (a \otimes b)_k \frac{1}{k!} \end{aligned}$$

<sup>10</sup> Precis som vi såg för följden  $(1, 1, 1, \dots)$ , där  $C = 1$  – detta beror, i alla fall andligen, på att våra följder som ges av linjära rekursioner också kommer växa ungefär så långsamt, så vi får samma sorts funktion.

<sup>11</sup> Det verkar som att både kursboken och förra årets föreläsningssanteckningar sveper denna komplikation under mattan, på ett sätt som jag upplever som väldigt förvirrande. Det tog mig minst en timme att lista ut varför vad de gör inte är direkt felaktigt.



så alltså måste vi ha

$$\begin{aligned} G_{A*B}(x) &= \sum_{k=0}^{\infty} (A * B)_k x^k \\ &= \sum_{k=0}^{\infty} (a \otimes b)_k \frac{x^k}{k!} \\ &= EG_{a \otimes b}(x) \end{aligned}$$

och vårt lemma är bevisat.  $\square$

På ett fullkomligt analogt sätt, användandes resultatet för ordinära genererande funktioner, fast med fler index, kan vi bevisa följande resultat:

**Lemma 11.** Antag att  $\{a_k^1\}_{k=0}^{\infty}, \{a_k^2\}_{k=0}^{\infty}, \dots, \{a_k^d\}_{k=0}^{\infty}$  är följder. Då gäller det att<sup>12</sup>

$$(a^1 \otimes a^2 \otimes \dots \otimes a^d)_k = \sum_{\substack{k_1, k_2, \dots, k_d \\ k_1 + k_2 + \dots + k_d = k}} \left( \binom{k}{k_1, k_2, \dots, k_d} \prod_{j=1}^d a_{k_j}^j \right)$$

och

$$EG_{a^1 \otimes a^2 \otimes \dots \otimes a^d}(x) = \prod_{j=1}^d EG_{a^j}(x).$$

Som vårt sista exempel för denna föreläsning tar vi motsvarigheten för exponentiella genererande funktioner till våra problem med antal lösningar på en ekvation.

**Exempel 12.** Hur många strängar ur alfabetet  $\{a, b, c, d\}$  finns det av längd  $k$ , med åtminstone ett  $b$ , högst tre  $c$ , och ett jämnt antal  $d$ ?

Precis som i fallet med att räkna lösningar till ekvationer låter vi  $l_k$  vara antalet sådana strängar, och låter sedan

- $a_k$  vara antalet strängar ur alfabetet  $\{a\}$  av längd  $k$ ,
- $b_k$  vara antalet strängar ur alfabetet  $\{b\}$  av längd  $k$  med åtminstone ett  $b$ ,
- $c_k$  vara antalet strängar ur alfabetet  $\{c\}$  av längd  $k$  med högst tre  $c$ , och
- $d_k$  vara antalet strängar ur alfabetet  $\{d\}$  av längd  $k$  med ett jämnt antal  $d$ .

Om vi nu betraktar binomialfältningen av dessa fyra följder ser vi att

$$(a \otimes b \otimes c \otimes d)_k = \sum_{\substack{k_1, k_2, k_3, k_4 \\ k_1 + k_2 + k_3 + k_4 = k}} \binom{k}{k_1, k_2, k_3, k_4} a_{k_1} b_{k_2} c_{k_3} d_{k_4} = l_k$$

<sup>12</sup> Kom ihåg att notationen  $\binom{k}{k_1, k_2, \dots, k_d}$  betecknar en *multinomialkoefficient*.



eftersom vad vi räknar med multinomialkoefficienten  $\binom{k}{k_1, k_2, k_3, k_4}$  är precis antalet strängar av längd  $k$  med  $k_1$  stycken  $a$ ,  $k_2$  stycken  $b$ , och så vidare.

Vad som återstår är alltså att finna de exponentiella genererande funktionerna för våra fyra följder.

- Följden  $a$  är helt enkelt  $(1, 1, 1, \dots)$ , så dess exponentiella genererande funktion har vi redan sett är  $e^x$ .
- Följden  $b$  är  $(0, 1, 1, \dots)$ , så vi kan räkna att

$$EG_b(x) = \sum_{k=1}^{\infty} \frac{x^k}{k!} = \sum_{k=0}^{\infty} \frac{x^k}{k!} - 1 = e^x - 1.$$

- Följden  $c$  är  $(1, 1, 1, 1, 0, 0, \dots)$ , så

$$EG_c(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}.$$

- För följden  $d$  får vi använda oss av ett litet trick, och skriva att

$$\begin{aligned} EG_d(x) &= \sum_{k \in 2\mathbb{Z} \geq 0} \frac{x^k}{k!} \\ &= \frac{1}{2} \left( \sum_{k=0}^{\infty} \frac{x^k}{k!} - \sum_{k=0}^{\infty} \frac{(-x)^k}{k!} \right) \\ &= \frac{e^x - e^{-x}}{2}. \end{aligned}$$

Multiplikerar vi ihop dessa får vi att

$$EG_l(x) = e^x (e^x - 1) \left( 1 + x + \frac{x^2}{2} + \frac{x^3}{3} \right) \left( \frac{e^x - e^{-x}}{2} \right).$$

Om vi vill ha en explicit formel är ett bra nog CAS<sup>13</sup> kapabelt att ge en explicit formel för den  $n$ te termen i serieexpansionen av detta, men den är inte vacker.

<sup>13</sup> Mathematica kan det, men inte WolframAlpha.

## Övningar

**Övning 1.** Hur många heltalslösningar till ekvationen

$$x_1 + x_2 + x_3 = 578$$

finns det,<sup>14</sup> om vi kräver att

- $x_1 \geq -7$ ,<sup>15</sup>
- $x_2 \geq 0$  är ett jämnt tal,

<sup>14</sup> Om ni väl har hittat genererande funktionen för antalet lösningar när vi ersatt 578 med  $k$ , så kan ni enkelt få fram svaret med följande kod till WolframAlpha:

```
SeriesCoefficient[f, {x, 0, 578}]
```

där  $f$  då är genererande funktionen ni funnit. Att ange den genererande funktionen och säga att ni sedan plockade fram rätt koefficient ur den med hjälp av ett CAS är den förväntade metoden här.

<sup>15</sup> Ledtråd: Gör ett variabelbyte till  $y_1$  för att få den vanliga begränsningen att  $y_1 \geq 0$ .

- och  $x_3 \geq 0$  kan vara vilket tal som helst, men om det är jämnt kan det vara rött eller blått, och om det är udda kan det vara gult, grönt, eller lila.

**Övning 2.** Antag att vi vet att följden  $\{a_k\}_{k=0}^\infty$  har exponentiell genererande funktion  $F$ . Finn ett enkelt uttryck för den exponentiella genererande funktionen för följden  $b_k = ka_k$ , i termer av  $F$ .

**Övning 3.** I denna övning använder vi ett lite mer abstrakt språk. Låt  $\mathfrak{F}$  vara mängden av alla följder, och  $\mathfrak{G}$  vara mängden av alla genererande funktioner. Då kan vi alltså låta  $G : \mathfrak{F} \rightarrow \mathfrak{G}$  vara funktionen som skickar en följd till dess ordinära genererande funktion, och  $EG : \mathfrak{F} \rightarrow \mathfrak{G}$  funktionen som skickar en följd till dess exponentiella genererande funktion.

Vi kan också definiera operatoren (funktionen)  $A : \mathfrak{F} \rightarrow \mathfrak{F}$  som att  $(Aa)_k = a_{k+1}$ . Vi förskjuter alltså index ett steg, och slänger bort första termen i vår följd.<sup>16</sup> Likaledes kan vi definiera operatoren  $D : \mathfrak{G} \rightarrow \mathfrak{G}$  som funktionen som skickar en genererande funktion på dess derivata.<sup>17</sup>

Slutligen kan vi, för varje heltal  $n$ , definiera operatoren  $T_n : \mathfrak{F} \rightarrow \mathbb{R}$  som att  $Ta = a_n$ . Den plockar alltså helt enkelt ut den  $n$ te termen i följden.

Låt oss nu plocka ner all denna abstraktionen på en lite mer konkret nivå. Övertyga er själva, genom att skriva ut konkret vad påståendena betyder, om att vi redan visat i föreläsningarna att

- $G(A(a)) = \frac{1}{x} (G(a) - T_0(a))$  och
- $EG(A(a)) = D(EG(a))$ . Vill man vara oerhört förfinad och abstrakt kan man formulera detta som att följande diagram kommuterar:

$$\begin{array}{ccc}
 \mathfrak{F} & \xrightarrow{A} & \mathfrak{F} \\
 EG \downarrow & & \downarrow EG \\
 \mathfrak{G} & \xrightarrow{D} & \mathfrak{G}
 \end{array}$$

**Övning 4.** Låt oss nu se varför detta abstrakta språk faktiskt är användbart ibland.<sup>18</sup> Låt  $a = (1, 0, 0, 1, 0, 0, 1, \dots)$ , och låt  $\vec{1} = (1, 1, 1, \dots)$ .

Visa att

$$a = \vec{1} - Aa - AAa,$$

<sup>16</sup> Så till exempel har vi att  $A(1, 0, 1, 0, 1, 0, \dots) = (0, 1, 0, 1, 0, 1, \dots)$ .

<sup>17</sup> Vi har ju egentligen inte *definierat* vad vi menar med derivatan av en genererande funktion, utan bara låtsats att vi får behandla dem som vilken funktion som helst, och ignorerat alla konvergensfrågor och allt krångel om att summorna har oändligt med termer. Vi fortsätter att göra så.

<sup>18</sup> I det här fallet ger det oss ett mycket koncisare sätt att uttrycka räkningen för att hitta genererande funktionen för indikatorföljden för talen delbara med tre. Dessutom blir det tydligt hur man hade gjort om vi hade sju istället för tre.

Kanske att man till och med kan göra räkningen för godtyckliga  $n\mathbb{Z}$ , men det kräver nog mer differentialekvationsteori än jag kan, eller kan förvänta mig att ni skall kunna. (Jag försökte i ungefär en minut med Laplacetransformer, men det blev svårt och kändes invecklat.)

använd vad vi just gjorde i förra övningen för att omvandla detta till en differentialekvation för  $EG(a)$ , och lös differentialekvationen<sup>19</sup> för att få fram ett uttryck för exponentiella genererande funktionen för  $a$ .

**Övning 5.** Låt  $\ell_k$  vara antalet strängar ur alfabetet  $\{a, b, c\}$  av längd  $k$ , sådana att

- det omedelbart efter varje  $a$  följer ett  $b$ ,<sup>20</sup>
- det finns ett jämnt antal  $c$ ,
- antalet  $a$  är delbart med tre,
- och varje  $b$  föregås av ett  $a$ ?

Finn genererande funktionen för följden  $\ell_k$ .

**Övning 6.** För varje  $n$  ges det  $n$ te Bell-talet av

$$B_n = \sum_{k=1}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\},$$

så det räknar alltså det totala antalet mängdpartitioner av en mängd av  $n$  element, oavsett antal delar. Man kan visa att<sup>21</sup>

$$B_{n+1} = \sum_{k=0}^n \binom{n}{k} B_k. \quad (1)$$

Låt  $EG_B(x)$  vara den exponentiella genererande funktionen för Bell-talen, och visa att

$$\frac{d}{dx} EG_B(x) = \sum_{n=0}^{\infty} B_{n+1} \frac{x^n}{n!}. \quad (2)$$

Substituera sedan in vår rekursion (1) i (2), och observera att det ni får ser väldigt mycket ut som exponentiella genererande funktionen för binomialfältningen mellan Bell-talen och någon annan följd.

Använd detta för att få fram en differentialekvation för  $EG_B$  – lös den för att få fram ett uttryck för exponentiella genererande funktionen för Bell-talen.<sup>22,23</sup>

<sup>19</sup> Det här steget vill jag inte ha några räkningar på – bara ange vad lösningen blir. Ni får använda vilket verktyg ni vill för att faktiskt lösa den.

<sup>20</sup> Ledtråd: För att hantera denna begränsning behöver vi göra ett "variabelbyte". Vad kan det tänkas betyda i denna kontext?

<sup>21</sup> Detta är en av övningarna i vår samling med extra övningar.

<sup>22</sup> Som vanligt behöver ni inte faktiskt ange hur ni löser den – bara skriv vad lösningen är.

<sup>23</sup> Ni har härmed lyckats plocka fram en EGF för en följd där vi faktiskt inte har något enkelt uttryck för vad varje enskild term blir – det finns inget fint uttryck för  $B_n$  i termer av  $n$ . Hade vi kunnat mer komplexanalys hade vi kunnat använda detta för att få fram skattningar för hur snabbt Belltalen växer, till exempel. (Svaret är: "väldigt väldigt snabbt", men vi hade kunnat vara precisare än så.)