Kapitel 5

Rekursion og induktion

5.1 Eksempler på rekursivt definerede funktioner

I dette afsnit introducerer vi ideen om en rekursivt defineret funktion. Pointen i en *rekursion* i denne sammenhæng er simpelthen at definere en funktion eller et udtryk ved hjælp af selv samme funktion eller udtryk for andre inputværdier. Lad os lægge ud med et eksempel:

Eksempel 5.1.1

Fakultets-funktionen $\mathrm{fac} : \mathbb{N} \to \mathbb{N}$ er defineret ved $n \mapsto 1 \cdot 2 \cdots n$. Den afbilder n i produktet af de første n positive heltal. Man vil typisk også skrive $n!$ i stedet for $\mathrm{fac}(n)$. Vi har for eksempel $\mathrm{fac}(1) = 1$, $\mathrm{fac}(2) = 1 \cdot 2 = 2$, $\mathrm{fac}(3) = 1 \cdot 2 \cdot 3 = 6$, $\mathrm{fac}(4) = 1 \cdot 2 \cdot 3 \cdot 4 = 24$ og så videre. Bemærk nu, at hvis vi vil beregne den næste værdi, $\mathrm{fac}(5)$, kan vi udnytte, at vi allerede ved, hvad $\mathrm{fac}(4)$ er. Faktisk har vi:

$$\mathrm{fac}(5) = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = (1 \cdot 2 \cdot 3 \cdot 4) \cdot 5 = \mathrm{fac}(4) \cdot 5 = 24 \cdot 5 = 120.$$

Generelt, hvis vi for et valgt $n > 1$ allerede har udregnet $\mathrm{fac}(n-1)$, kan vi udregne værdien af $\mathrm{fac}(n)$ ved $\mathrm{fac}(n) = \mathrm{fac}(n-1) \cdot n$. Dette fører til følgende algoritmiske beskrivelse af fakultetsfunktionen:

Algoritme 8 $\mathrm{fac}(n)$

Input: $n \in \mathbb{Z}_{\geq 1}$.
if $n = 1$ **then**
    return 1
else
    return $\mathrm{fac}(n-1) \cdot n$.

Denne algoritme benytter sig selv til at udregne $\mathrm{fac}(n)$. Mere udpenslet, hvis $n = 1$, så returneres straks 1 som værdien for $\text{fac}(1)$ som angivet i linje 2 i algoritmen. Grafisk kan vi illustrere dette som følger:

$\text{fac}(1)$

$$\text{fac}(1) = 1$$

returnér 1

Hvis $n = 2$, så vil algoritmen hoppe til linje 4 og forsøge at returnere $\text{fac}(2-1) \cdot 2$. Men dette kræver, at værdien af $\text{fac}(1)$ først udregnes. Derfor vil algoritmen nu starte forfra, denne gang for værdien 1. Vi har allerede set, at algoritmen returnerer 1 i det tilfælde. Da algoritmen er kommet frem til den konklusion, at $\text{fac}(1) = 1$, kan den genbesøge linje 4 og udregne, at $\text{fac}(2) = \text{fac}(1) \cdot 2 = 1 \cdot 2 = 2$. Dermed kan algoritmen nu returnere 2. Grafisk er situationen som følger:

$\text{fac}(2)$

$$\text{fac}(2) = \text{fac}(1) \cdot 2$$

> $\text{fac}(1)$
>
> $$\text{fac}(1) = 1$$
>
> **returnér** 1

$$\text{fac}(2) = 1 \cdot 2 = 2$$

returnér 2

For større værdier af n vil flere "bokse inde i andre bokse" dukke op, da algoritmen nu skal benytte sig selv adskillige gange og udregne output for de lavere inputværdier $n-1, n-2, \ldots, 1$, før den kan returnere det endelige output. Følgende grafiske repræsentation viser, hvad der sker, når denne algoritme får inputværdien $n = 5$:

fac(5)

$$\mathrm{fac}(5) = \mathrm{fac}(4) \cdot 5$$

fac(4)

$$\mathrm{fac}(4) = \mathrm{fac}(3) \cdot 4$$

fac(3)

$$\mathrm{fac}(3) = \mathrm{fac}(2) \cdot 3$$

fac(2)

$$\mathrm{fac}(2) = \mathrm{fac}(1) \cdot 2$$

fac(1)

$$\mathrm{fac}(1) = 1$$

returnér 1

$$\mathrm{fac}(2) = 1 \cdot 2 = 2$$

returnér 2

$$\mathrm{fac}(3) = 2 \cdot 3 = 6$$

returnér 6

$$\mathrm{fac}(4) = 6 \cdot 4 = 24$$

returnér 24

$$\mathrm{fac}(5) = 24 \cdot 5 = 120$$

returnér 120

Da algoritmen benytter sig selv under gennemførslen (i algoritmiske termer siger man typisk, at algoritmen *kalder* sig selv), kaldes den en *rekursiv* algoritme. En rekursiv algoritme er simpelthen en algoritme, der kan kalde sig selv for visse inputværdier for at gennemføre udregningen af sin endelige outputværdi. Rekursioner forekommer også i matematikken. Vi har herunder angivet en rekursiv definition af fakultetsfunktionen fra vores eksempel:

$$\mathrm{fac}(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ \mathrm{fac}(n-1) \cdot n & \text{hvis } n \geq 2. \end{cases} \tag{5.1}$$

Dette eksempel illustrerer princippet i en rekursiv definition: at definere de værdier, en funktion tager, ved brug af funktionen selv. Bemærk i øvrigt, at det også er sædvane at definere $0! = 1$, men det er en anden sag. Her er et andet eksempel på en rekursivt defineret funktion: lad $z \in \mathbb{C}$ være et komplekst tal, og definér $f : \mathbb{N} \to \mathbb{C}$ rekursivt som:

$$f(n) = \begin{cases} z & \text{hvis } n = 1, \\ f(n-1) \cdot z & \text{hvis } n \geq 2. \end{cases} \tag{5.2}$$

Da er $f(1) = z$, da dette svarer til tilfældet $n = 1$ i den rekursive definition. Som det næste er $f(2) = f(1) \cdot z$, da dette er, hvad den rekursive definition angiver for $n = 2$. Ved at udnytte, at vi allerede har bestemt, at $f(1) = z$, kan vi konkludere, at $f(2) = f(1) \cdot z = z \cdot z$. Endelig, ved at udnytte at $z \cdot z = z^2$, ser vi, at $f(2) = f(1) \cdot z = z \cdot z = z^2$. Ligeledes er $f(3) = z^3$. Derfor er det helt rimeligt at *definere* udtrykket z^n for ethvert naturligt tal n rekursivt som $f(n)$. I tidligere kapitler har vi benyttet n'te potenser af komplekse tal flere gange. Nu har vi en mere formel definition af det. I denne forbindelse er det også almindeligt at definere $z^0 = 1$ og $z^{-n} = 1/z^n$ for ethvert naturligt tal n. Vi har hermed præcist defineret, hvad z^n betyder for ethvert heltal $n \in \mathbb{Z}$.

Når man forsøger at definere en funktion rekursivt, skal man efterfølgende sikre sig, at en sådan rekursiv beskrivelse definerer funktionen for alle værdier fra dens definitionsmængde. For de funktioner, der er defineret i Ligningerne (5.1) og (5.2), kan du finde en redegørelse herfor i Eksempel 5.4.2, men vær velkommen til at springe eksemplet over under første gennemlæsning. For nu vil vi vise et eksempel på et forsøg på en rekursiv beskrivelse, der ikke fungerer. Lad $g : \mathbb{N} \to \mathbb{C}$ være en funktion, og antag at

$$g(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ g(n+1) & \text{hvis } n \geq 2. \end{cases}$$

Jævnfør definitionen ser vi, at $g(1) = 1$, men vi har ikke nok information til at bestemme, hvad $g(2)$ er. Hvis vi anvender den rekursive definition, ville vi bare nå frem til, at $g(2) = g(3)$. Derefter vil vi forsøge at beregne $g(3)$, men rekursionen giver os blot $g(3) = g(4)$. Fortsætter vi sådan, får vi blot, at $g(2) = g(3) = g(4) = g(5) = \cdots$, og vi finder aldrig ud af, hvad $g(2)$ rent faktisk er.

Som et sidste eksempel på en rekursiv definition vil vi se på de berømte Fibonacci-tal.

Eksempel 5.1.2

Lad os tage fat på en rekursiv definition, der ser en smule anderledes ud. Vi skal definere en funktion $F : \mathbb{N} \to \mathbb{N}$ rekursivt, hvis værdier $F(1), F(2), F(3), F(4), \ldots$ kaldes *Fibonacci-tal*:

$$F(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ 1 & \text{hvis } n = 2, \\ F(n-1) + F(n-2) & \text{hvis } n \geq 3. \end{cases} \tag{5.3}$$

Lad os se, hvordan denne definition fungerer i praksis ved at udregne de første par Fibonacci-tal. Først og fremmest er $F(1) = 1$, da vi med $n = 1$ skal se på første linje i Ligning (5.3). Hvis $n = 2$, gælder den anden linje i Ligning (5.3), så $F(2) = 1$. For $n = 3$ gælder den tredje linje i Ligning (5.3), og vi får, at $F(3) = F(2) + F(1) = 1 + 1 = 2$. Ligeledes for $n = 4$ får vi, at $F(4) = F(3) + F(2) = 2 + 1 = 3$ ved at udnytte, at vi allerede har udregnet, at $F(3) = 2$.

Når man arbejder med en sekvens af tal, som for eksempel Fibonacci-tallene, vil man typisk ændre en smule i notationen: i stedet for at skrive $F(n)$, skriver man ofte F_n. I denne notation har vi $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, $F_4 = 3$ og så videre. Det viser sig, at det er muligt at udlede et lukket formeludtryk for Fibonacci-tallene:

$$F_n = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n. \tag{5.4}$$

Vi vil senere i dette kapitel få en forklaring på, hvordan vi er kommet til dette udtryk.

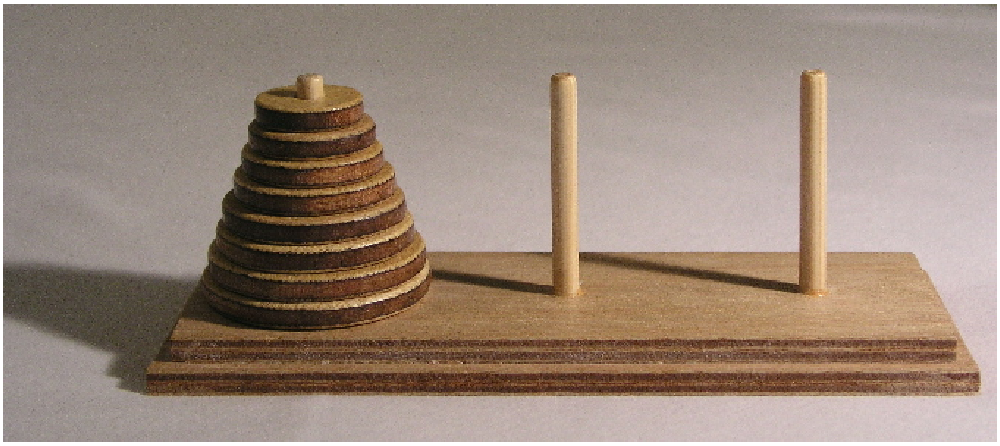
5.2 Hanois tårne

I dette afsnit fortsætter vi med at illustrere nyttigheden af en rekursiv tankegang ved at analysere en gåde kaldet *Hanois tårne.* Hanois tårne er en gåde, der foregår på et bræt påmonteret tre lodrette pinde med samme længder og størrelser. Man har et antal cirkulære skiver til rådighed, alle med forskellig diameter og med et hul i midten, så de kan placeres på en pind. Gådens startopstilling er, at alle skiver er stablet på den første pind. Skiven med den største diameter er nederst i stakken, og de andre skiver følger derpå i faldende diameterstørrelse opefter. Antallet af skiver kan variere. For et eksempel med otte skiver, se Figur 5.1. Billedet i denne figur er fra Wikipedia; se https://commons.wikimedia.org/wiki/File:Tower_of_Hanoi.jpeg for flere detaljer.

Målet med gåden er at flytte stakken af skiver fra den første til den tredje pind, hvor de skal være stablet på samme måde, altså fra stor til lille set fra bunden. Udfordringen er dog, at man skal følge tre regler, når man flytter skiver:

- Kun én skive må flyttes ad gangen.
- Kun en skive øverst på en stak må flyttes.

Figur 5.1: Hanois tårne med otte skiver.



- En skive må kun placeres på en større skive.

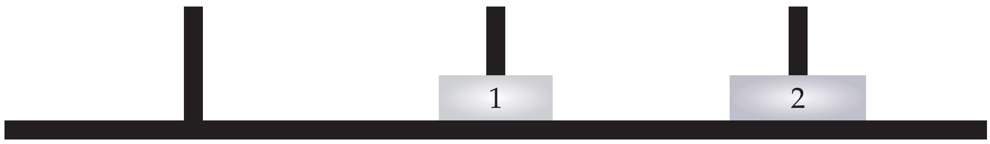
Hvis der kun er meget få skiver, er det ikke svært at løse opgaven. Hvis der er mange skiver, bliver spillet mere kompliceret, og det er a priori ikke engang klart, om der altid findes en løsning. For at komme i gang kan vi se på nogle eksempler med kun ganske få skiver. Hvis der kun er én skive, kan vi løse opgaven i ét træk:

Flyt skive 1 fra pind 1 til pind 3

Hvis der er to skiver, kan gåden løses med tre flytninger:

Flyt skive 1 fra pind 1 til pind 2

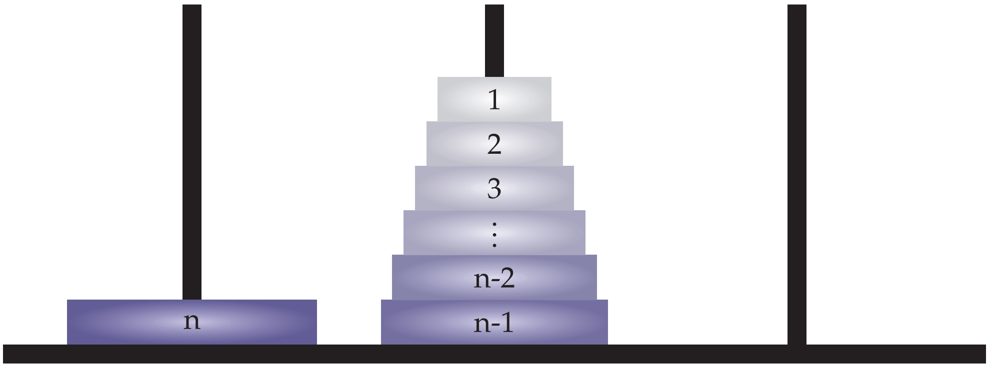
Flyt skive 2 fra pind 1 til pind 3



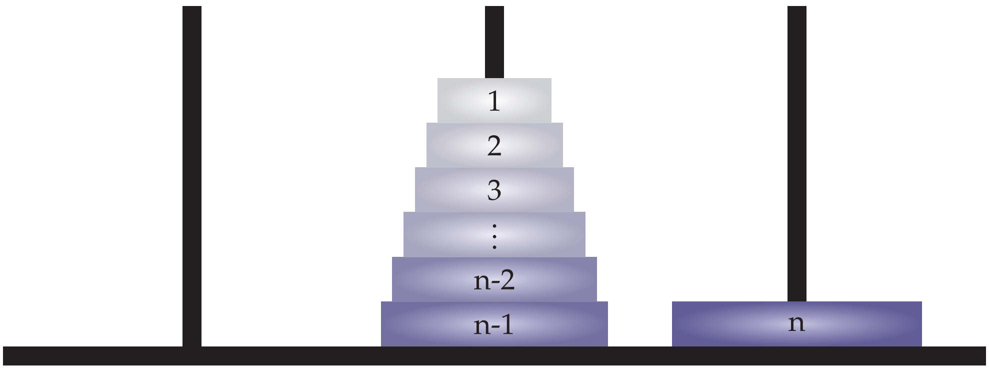
Flyt skive 1 fra pind 2 til pind 3

Med tre skiver er det stadig ikke så svært at løse gåden ved at prøve sig frem, men hvad nu hvis der er ti skiver, eller hundrede? Lad os prøve at tænke på en rekursiv måde for at finde frem til en løsningsmetode. Vi ved allerede, hvordan vi løser opgaven, hvis der kun er én skive (og også hvis der er to skiver). Måske kan vi, som med fakultetsfunktionen, finde ud af, hvad vi skal gøre i tilfælde af et større antal skiver, lad os sige n skiver, hvis vi allerede ved, hvad vi skal gøre, hvis vi har færre end n skiver. Antag derfor, at $n \geq 2$ er et naturligt tal, og at vi allerede ved, hvordan vi løser opgaven, hvis der er $n-1$ skiver. Dette betyder, at vi ved, hvordan vi flytter en stak af $n-1$ skiver fra én pind til en anden. I så fald virker følgende strategi til at flytte n skiver:

Da vi ved, hvordan vi flytter $n-1$ skiver fra én pind til en anden, flytter vi nu alle skiver fra skive nummer 1 til nummer $n-1$ i stakken fra pind 1 til pind 2.



Dernæst flytter vi skive n fra pind 1 til pind 3. Dette tager kun ét træk.



Igen anvender vi vores givne viden om, hvordan man flytter $n-1$ skiver, til at flytte stakken af skiver indeholdende skive nummer 1 til nummer $n-1$ fra pind 2 til pind 3.

Dette viser, at gåden kan løses rekursivt! Og vi har endda vist, at der findes en løsning uanset antallet af skiver.

5.3 Summationssymbolet

Hvis n er et naturligt tal, og $z_1, \ldots, z_n$ er komplekse tal, kan man angive deres sum med et udtryk som $z_1 + z_2 + \cdots + z_n$ eller $z_1 + \cdots + z_n$. Dog er det ofte nemmere med den mere kompakte notation for dette: $\sum_{k=1}^{n} z_k$. Ved at bruge en rekursiv definition kan vi være meget præcise:

$$\sum_{k=1}^{n} z_k = \begin{cases} z_1 & \text{hvis } n = 1, \\ \left(\sum_{k=1}^{n-1} z_k\right) + z_n & \text{hvis } n > 1. \end{cases} \tag{5.5}$$

Ved at bruge denne rekursive definition opnår vi præcis det, vi ønsker. Man kan nemt benytte definitionen og verificere, at man ganske rigtigt for små værdier af n opnår:

n	$\sum_{k=1}^{n} z_k$
1	z_1
2	$z_1 + z_2$
3	$z_1 + z_2 + z_3$
4	$z_1 + z_2 + z_3 + z_4$

(5.6)

Hvis $f : \mathbb{N} \to \mathbb{C}$ er en funktion, kan man på lignende vis erstatte summen $f(1) + f(2) + \cdots + f(n)$ med det mere kompakte udtryk $\sum_{k=1}^{n} f(k)$. Overvej for eksempel udtrykket $\sum_{k=1}^{n} k$, det vil sige summen af de første n naturlige tal. Som i Tabel 5.6 får vi følgende:

n	$\sum_{k=1}^{n} k$
1	1
2	$1 + 2 = 3$
3	$1 + 2 + 3 = 6$
4	$1 + 2 + 3 + 4 = 10$

(5.7)

Denne notation er praktisk at have ved hånden i flere senere kapitler, og den benyttes flittigt indenfor flere områder i matematikken og naturvidenskabben.

Variablen k i et udtryk som $\sum_{k=1}^{n} z_k$ kaldes sumindekset. Der er som sådan ingen grund til at vælge symbolet k, og det er helt fint at benytte et andet symbol. For eksempel har man $\sum_{k=1}^{n} z_k = \sum_{j=1}^{n} z_j$, da begge summer svarer til at lægge tallene $z_1, \ldots, z_n$ sammen. Man kan også indeksere de tal, der skal lægges sammen, på en anden måde. Hvis vi vil lægge tallene $z_2, \ldots, z_{10}$ sammen, kan man simpelthen skrive $\sum_{k=2}^{10} z_k$.

5.4 Induktion

I det foregående afsnit sluttede vi af med at løse gåden om Hanois tårne fuldstændigt med en rekursiv tilgang. Antallet af træk, vores løsning kræver, kan beskrives rekursivt. Hvis vi lader $T(n)$ betegne antallet af træk, som vores strategi vil kræve, når gåden stilles med n skiver, så ved vi, at $T(1) = 1$ (gåden med kun én skive kan løses i ét træk), men også at $T(n) = T(n-1) + 1 + T(n-1) = 2T(n-1) + 1$ for $n \geq 2$ (vores strategi lod os flytte en stak af $n-1$ skiver to gange, mens den n'te skive skulle flyttes én gang). Med andre ord har vi

$$T(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ 2 \cdot T(n-1) + 1 & \text{hvis } n \geq 2. \end{cases} \tag{5.8}$$

For eksempel er $T(2) = 2 \cdot 1 + 1 = 3$, $T(3) = 2 \cdot 3 + 1 = 7$ og $T(4) = 2 \cdot 7 + 1 = 15$. Det slår os nu, at værdien af $T(n)$ for disse små værdier af n ser ud til altid at være én mindre end 2^n. Vi vil derfor "gætte" på, at $T(n) = 2^n - 1$ for alle naturlige tal n. Lad os teste denne formodning, et ord vi hellere benytter frem for "gætte", ved at beregne $T(5)$. Vi har $T(5) = 2 \cdot T(4) + 1 = 2 \cdot 15 + 1 = 31$. Dette bekræfter vores formodning om, at $T(n) = 2^n - 1$ for $n = 5$. Ulempen er, vi nu kun ved, at formodningen er sand for alle n i mængden $\{1, 2, 3, 4, 5\}$. Vi kunne selvfølgelig fortsætte med at bekræfte vores formodning for flere værdier af n ved at beregne $T(6), T(7)$ og så videre, men da der er uendeligt mange naturlige tal, er det umuligt for os at verificere formlen $T(n) = 2^n - 1$ for *alle* naturlige tal n på denne måde. Heldigvis har de naturlige tal en ganske intuitiv egenskab, der kan hjælpe os, og som vi fremsætter uden bevis:

Sætning 5.4.1 Induktionsprincippet

Lad S være en delmængde af de naturlige tal, og antag, at S har følgende to egenskaber:

1. $1 \in S$,
2. hvis $n-1 \in S$ for et vilkårligt naturligt tal $n \geq 2$, da er også $n \in S$.

I så fald har vi $S = \mathbb{N}$.

Udsagnet i denne sætning kaldes ofte *induktionsprincippet* eller simpelthen *induktion*. Krav 1. ($1 \in S$) kaldes *basistrinnet for induktionen*, mens krav 2. (hvis $n-1 \in S$ for et naturligt tal n, så er også $n \in S$) kaldes *induktionstrinnet*. Grunden til, at det naturlige tal n i krav 2. skal være mindst to, er, at ellers er $n-1$ måske ikke et naturligt tal. Hvis $n = 1$, så er $n - 1 = 0$, men 0 tilhører ikke $\mathbb{N}$. Krav 2. kan omformuleres ved brug af udsagnslogikken som følger.

2. for alle $n \in \mathbb{N}_{\geq 2}: \quad n-1 \in S \quad \Rightarrow \quad n \in S$.

Verificering af krav 2., det vil sige verificering af induktionstrinnet, gøres typisk ved at vise, at $n \in S$ er sandt, hvis vi antager, at $n-1 \in S$. Når man verificerer induktionstrinnet $n-1 \in S \quad \Rightarrow \quad n \in S$, kaldes antagelsen $n-1 \in S$ for *induktionshypotesen*. Processen, der verificerer de to krav, kaldes typisk *bevis ved induktion* eller, hvis variablen n's rolle skal understreges, et *bevis ved induktion på n*.

Induktionsprincippet er nøglen til at forstå mange udsagn i matematikken og er også centralt i datalogi, da det dér kan bruges til at vise korrektheden af forskellige algoritmer, rekursive definitioner og computerprogrammer.

I matematikken er det bekvemt at benytte en omformulering af induktionsprincippet, så arbejde med en delmængde $S \subseteq \mathbb{N}$ undgås. Det kan nemlig undgås ved udnyttelse af en pæn konsekvens af Sætning 5.4.1. Sådanne konsekvenser kaldes typisk for "korollarer" i matematiske tekster, en terminologi vi vil benytte her.

Korollar 5.4.2

Lad $P(n)$ være et logisk udsagn for ethvert naturligt tal n. Antag, at følgende to udsagn er sande:

1. $P(1)$,
2. for alle $n \in \mathbb{N}_{\geq 2}: \quad P(n-1) \Rightarrow P(n)$.

Da er $P(n)$ sandt for alle $n \in \mathbb{N}$.

Bevis. For at kunne benytte Sætning 5.4.1 gør vi brug af et trick ved at definere $S = \{n \in \mathbb{N} \mid P(n)\}$. Med andre ord, $n \in S$, netop hvis $P(n)$ er sandt. For at kunne konkludere, at $P(n)$ er sandt for alle naturlige tal n, er det nok at vise, at $S = \mathbb{N}$. Hvis der eksisterede et naturligt tal m, således at $P(m)$ er falsk, så ville vi ved definitionen af S få, at $m \notin S$ og derfor, at $S \neq \mathbb{N}$.

Nu benytter vi Sætning 5.4.1 til at vise, at $S = \mathbb{N}$. Antagelsen om, at $P(1)$ er sandt, betyder blot, at $1 \in S$. Antagelsen om, at for alle $n \in \mathbb{N}_{\geq 2}: \quad P(n-1) \Rightarrow P(n)$, betyder, at når $n-1 \in S$, er også $n \in S$. Derfor er de to krav fra Sætning 5.4.1 opfyldt. Dermed kan vi ved brug af Sætning 5.4.1 konkludere, at $S = \mathbb{N}$. Dette betyder blot, som allerede bemærket, at $P(n)$ er sandt for alle naturlige tal n. $\square$

Som i Sætning 5.4.1 kaldes det at tjekke, at $P(1)$ er opfyldt, for basistrinnet for induktionen, mens det at tjekke, at for alle $n \in \mathbb{N}_{\geq 2}: \quad P(n-1) \Rightarrow P(n)$, kaldes for induktionstrinnet. Mens man udfører induktionstrinnet, kaldes det logiske udsagn $P(n-1)$ for induktionshypotesen ligesom før. Også udsagnet i Korollar 5.4.2 kaldes som helhed stadig for induktionsprincippet. Vi kan derfor bevise en påstand af formen "$P(n)$ er sandt for alle naturlige tal n" ved at følge følgende strategi:

(i) Informér læseren om, at du nu vil bevise påstanden, at "$P(n)$ er sandt for alle naturlige tal n," ved brug af induktion på n.

(ii) **Basistrin:** Tjek, at $P(1)$ er opfyldt.

(iii) **Induktionstrin:** Antag for et vilkårligt naturligt tal $n \geq 2$, at $P(n-1)$ er sandt, og benyt denne antagelse (induktionshypotesen) til at vise, at så er også $P(n)$ sandt. Udfordringen her er nogle gange at finde ud af, hvordan man bruger induktionshypotesen $P(n-1)$ til sin fordel.

(iv) Informér læseren, når de foregående punkter er udført, om, at man ved induktionsprincippet nu kan konkludere, at $P(n)$ gælder for alle naturlige tal n.

Lad os nu benytte denne strategi til at bevise vores formodning om, at $T(n) = 2^n - 1$. Med andre ord, lad os bevise følgende:

Påstand: Lad $T : \mathbb{N} \to \mathbb{N}$ opfylde rekursionen

$$T(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ 2 \cdot T(n-1) + 1 & \text{hvis } n \geq 2. \end{cases}$$

Så har vi for alle $n \in \mathbb{N}$, at $T(n) = 2^n - 1$.

Bevis. Lad $P(n)$ være udsagnet $T(n) = 2^n - 1$. Vi vil bevise påstanden ved induktion på n.

Basistrin: Vi har $T(1) = 1$. Da $2^1 - 1 = 1$, ser vi, at $T(1) = 2^1 - 1$. Derfor er $P(1)$ opfyldt.

Induktionstrin: Lad $n \geq 2$ være et vilkårligt naturligt tal. Induktionshypotesen er $P(n-1)$, som i vores tilfælde blot betyder ligningen $T(n-1) = 2^{n-1} - 1$. Hvis vi antager dette, skal vi nu udlede, at $P(n)$ er gyldigt. Med andre ord, hvis vi antager, at $T(n-1) = 2^{n-1} - 1$, skal vi udlede, at $T(n) = 2^n - 1$. Fra den rekursive definition af $T(n)$ for $n \geq 2$ ved vi, at $T(n) = 2 \cdot T(n-1) + 1$. Ved at kombinere dette med induktionshypotesen, ser vi, at

$$T(n) = 2 \cdot T(n-1) + 1 = 2 \cdot (2^{n-1} - 1) + 1 = 2 \cdot 2^{n-1} - 2 \cdot 1 + 1 = 2^n - 1.$$

Dette er præcis, hvad vi skulle vise.

Nu hvor vi har vist basistrinnet for induktionen samt induktionstrinnet, kan vi konkludere ved induktionsprincippet, at udsagnet $T(n) = 2^n - 1$ er sandt for alle naturlige tal n. □

Man kan faktisk vise, at den strategi, vi fandt i Afsnit 5.2, er den bedst mulige. Med andre ord vil enhver løsning af gåden med n skiver tage mindst $T(n)$ træk. Vi ser ved udregning, at løsningen af en ti-skive-version af Hanois tårne ville kræve $2^{10} - 1 = 1023$ træk.

Den bedste måde at lære induktionsbeviser på er at gennemgå adskillige eksempler og derefter prøve at udføre et induktionsbevis selv. Lad os derfor finde nogle flere eksempler frem. Her er et berømt et:

Eksempel 5.4.1

Lad os betegne ved $S(n)$ summen af de første n naturlige tal. Uformelt skriver man ofte $1 + 2 + \cdots + n$ for denne sum, men vi kan benytte os af summationstegnet og skrive $S(n) = \sum_{k=1}^{n} k$. Som vi så i Tabel 5.7, har vi for eksempel $S(1) = 1$, $S(2) = 1 + 2 = 3$, $S(3) = 1 + 2 + 3 = 6$ og $S(4) = 1 + 2 + 3 + 4 = 10$. Påstanden er nu, at følgende formel gælder for ethvert naturligt tal n:

$$S(n) = \frac{n \cdot (n+1)}{2}.$$

Bemærk, at $S(n)$ opfylder følgende rekursion:

$$S(n) = \begin{cases} 1 & \text{hvis } n = 1, \\ S(n-1) + n & \text{hvis } n \geq 2. \end{cases}$$

Faktisk har vi allerede observeret, at $S(1) = 1$. I tilfældet $n \geq 2$ får vi ved brug af Ligning (5.5), at

$$S(n) = 1 + \cdots + n = \sum_{k=1}^{n} k = \left(\sum_{k=1}^{n-1} k \right) + n = S(n-1) + n.$$

Lad os nu bevise følgende påstand.

Påstand: Lad $S(n) = 1 + \cdots + n$ for $n \in \mathbb{N}$, altså summen af de første n naturlige tal. Så er $S(n) = \frac{n \cdot (n+1)}{2}$.

Bevis. Vi beviser påstanden ved induktion på n.

Basistrin: Hvis $n = 1$, så er $S(1) = 1$, mens $\frac{1 \cdot (1+1)}{2} = 1$. Derfor er formlen $S(n) = \frac{n \cdot (n+1)}{2}$ gyldig for $n = 1$.

Induktionstrin: Lad $n \geq 2$ være et vilkårligt naturligt tal, og antag som induktionshypotese, at $S(n-1) = \frac{(n-1) \cdot (n-1+1)}{2}$. Vi kan reducere induktionshypotesen en smule, da der gælder, at

$S(n-1) = \frac{(n-1)\cdot n}{2}$. Hvis vi antager induktionshypotesen og benytter, at $S(n) = S(n-1) + n$, kan vi konkludere, at

$$\begin{aligned} S(n) &= S(n-1) + n \\ &= \frac{(n-1)\cdot n}{2} + n \\ &= \frac{(n-1)\cdot n}{2} + \frac{2\cdot n}{2} \\ &= \frac{n^2 - n}{2} + \frac{2\cdot n}{2} \\ &= \frac{n^2 - n + 2\cdot n}{2} \\ &= \frac{n^2 + n}{2} \\ &= \frac{n\cdot(n+1)}{2}. \end{aligned}$$

Dette er præcis, hvad vi skulle vise, hvilket fuldfører induktionstrinnet.

Ved induktionsprincippet kan vi dermed konkludere, at formlen $S(n) = \frac{n\cdot(n+1)}{2}$ er gyldig for alle naturlige tal n. $\square$

Eksempel 5.4.2

Dette eksempel er af en mere teoretisk karakter og kan springes over ved første læsning. Vi vil sikre os, at den rekursive definition, vi tidligere gav af fakultetsfunktionen $\mathrm{fac} : \mathbb{N} \to \mathbb{N}$ i Ligning (5.1), rent faktisk var korrekt i matematisk forstand. Problemet er, at vi aldrig viste, at fac er defineret ved sin rekursive beskrivelse for *ethvert* naturligt tal n. Med andre ord, når vi skriver $\mathrm{fac} : \mathbb{N} \to \mathbb{N}$, siger vi implicit, at funktionens definitionsmængde er $\mathbb{N}$, men hvordan ved vi det? Det, vi skal gøre, er at vise, at for ethvert naturligt tal n vil den rekursive beskrivelse i Ligning (5.1) give outputværdien $\mathrm{fac}(n)$ efter endeligt mange trin.

Lad derfor $P(n)$ være udsagnet, at $\mathrm{fac}(n)$ kan beregnes i endeligt mange trin ved hjælp af Ligning (5.1) for ethvert naturligt tal n. Vi vil vise, at dette udsagn $P(n)$ er sandt for alle naturlige tal. Basistrinnet for induktionen er hurtigt vist opfyldt, da Ligning (5.1) straks medfører, at $\mathrm{fac}(1) = 1$. Lad nu $n \geq 2$ være et vilkårligt naturligt tal, og antag som induktionshypotese, at $\mathrm{fac}(n-1)$ kan beregnes i endeligt mange trin ved hjælp af Ligning (5.1). Da $n \geq 2$, medfører Ligning (5.1), at $\mathrm{fac}(n) = \mathrm{fac}(n-1)\cdot n$. Givet $\mathrm{fac}(n-1)$ har vi derfor kun brug for en multiplikation med n for at udregne $\mathrm{fac}(n)$. Altså kan $\mathrm{fac}(n)$ udregnes i endeligt mange trin, hvis $\mathrm{fac}(n-1)$ kan. Dette fuldfører induktionstrinnet.

Mere generelt kan en funktion $f : \mathbb{N} \to B$ fra de naturlige tal $\mathbb{N}$ til en givet mængde B defineres rekursivt, så længe $f(1)$ er bestemt, og så længe værdien $f(n)$ kan udregnes ud fra $f(n-1)$ for ethvert $n \geq 2$. Det skyldes, at der i sådanne tilfælde gælder et lignende ræsonnement som det, vi netop har udført for fakultetsfunktionen. For eksempel definerer Ligning (5.2) z^n for ethvert naturligt tal n.

5.5 En variant af induktion

Der findes mange varianter af induktion. I dette afsnit vil vi nævne en af dem: induktion, der starter med et andet basistrin. Indtil videre har basistrinnet for vores induktionsbeviser været for $n = 1$, hvorefter vi har haft at gøre med større naturlige tal n. I nogle tilfælde giver et logisk udsagn dog også mening for andre værdier af n. Betragt for eksempel udsagnet:

Et polynomium $p(Z) \in \mathbb{C}[Z]$ af grad n har højst n rødder i $\mathbb{C}$.

Dette udsagn giver også mening for $n = 0$. Faktisk er udsagnet for $n = 0$ ret let at verificere: et polynomium $p(Z)$ af grad nul er simpelthen en konstant p_0, som ikke er nul. Konstanten p_0 er forskellig fra nul, netop fordi den ledende koefficient i et d'te-gradspolynomium generelt er forskellig fra nul ifølge Definition 4.1.1. Men så er $p(z) = p_0 \neq 0$ for alle $z \in \mathbb{C}$, hvilket medfører, at polynomiet ikke har nogen rødder.

Omvendt er der udsagn, der kun bliver sande for store nok værdier af n. Betragt for eksempel udsagnet:

Der eksisterer n punkter i planen $\mathbb{R}^2$, som ikke ligger på linje.

Hvis $n = 1$, er dette forkert, da der findes mange linjer igennem ethvert givet punkt. Også hvis $n = 2$, er dette forkert, da linjen, der forbinder de to punkter, vil indeholde disse punkter. Men for $n \geq 3$ er udsagnet sandt. Hvis $n \geq 3$, kan vi for eksempel vælge tre af punkterne som hjørner i en ligesidet trekant og de resterende $n - 3$ punkter vilkårligt.

På grund af denne slags eksempler er det praktisk at have en lidt mere fleksibel variant af induktion. For et givet heltal $a \in \mathbb{Z}$ har vi betegnelsen $\mathbb{Z}_{\geq a} = \{n \in \mathbb{Z} \mid n \geq a\}$. For eksempel er $\mathbb{Z}_{\geq -1} = \{-1, 0, 1, 2, \dots\}$. Med denne notation på plads kan vi formulere følgende variant af induktion, kaldet *induktion med basistrin b*:

Sætning 5.5.1

Lad $b \in \mathbb{Z}$ være et heltal, og lad $P(n)$ være et logisk udsagn for ethvert heltal $n \geq b$. Antag, at følgende to udsagn er sande:

1. $P(b)$,
2. for alle $n \in \mathbb{Z}_{\geq b+1}: \quad P(n-1) \Rightarrow P(n)$.

Så er $P(n)$ sandt for alle $n \in \mathbb{Z}_{\geq b}$.

Bevis. Lad os definere det logiske udsagn $Q(n)$ til at være $P(n + b - 1)$. Så er $Q(n)$ defineret for ethvert naturligt tal n. Vi ser nemlig, at hvis $n \geq 1$, så er $n + b - 1 \geq b$. Nu anvender vi Korollar 5.4.2 på det logiske udsagn $Q(n)$. Det første krav fra Korollar 5.4.2 er, at $Q(1)$ skal være gyldigt. Dette er opfyldt, da $Q(1) = P(b)$, og det er givet, at $P(b)$ er gyldigt. Det andet krav fra Korollar 5.4.2 bliver, at for alle $n \in \mathbb{N}_{\geq 2}: \quad Q(n-1) \Rightarrow Q(n)$. Men da $n \geq 2$, har vi $n + b - 1 \geq b + 1$ og derfor $n + b - 1 \in \mathbb{Z}_{\geq b+1}$. Da $Q(n-1) = P(n + b - 2)$

og $Q(n) = P(n+b-1)$, og da implikationen $P(n+b-2) \Rightarrow P(n+b-1)$ er gyldig (vi ved, at $n+b-1 \in \mathbb{Z}_{\geq b+1}$), ser vi, at implikationen $Q(n-1) \Rightarrow Q(n)$ er gyldig. Derfor er det andet krav til det logiske udsagn $Q(n)$, ved anvendelse af Korollar 5.4.2, også opfyldt. Derfor medfører korollaret, at $Q(n)$ er gyldigt for alle naturlige tal n. Da $Q(n) = P(n+b-1)$, betyder dette, at $P(n+b-1)$ er gyldigt for alle naturlige tal n. For eksempel er $P(1+b-1) = P(b)$ gyldigt, $P(2+b-1) = P(b+1)$ er gyldigt og så videre. Dette svarer til udsagnet, at $P(n)$ er gyldigt for alle heltal $n \geq b$, hvilket er, hvad vi ønskede at vise. □

Bemærk, at hvis vi vælger $b = 1$, genskaber vi Korollar 5.4.2. Den overordnede struktur af et bevis ved induktion med basistrin b er den samme som for den sædvanlige induktion. Man har stadig et basistrin og et induktionstrin. Lad os gennemgå et eksempel på et bevis ved induktion af denne type.

Eksempel 5.5.1

Betragt uligheden $n + 10 \leq n^2 - n$. Da et polynomium af anden grad, så som $n^2 - n$, vokser hurtigere end et polynomium af første grad, så som $n + 10$, bør man kunne forvente, at hvis n bliver stor nok, er den givne ulighed sand. Lad os nu ved $P(n)$ betegne udsagnet, at $n + 10 \leq n^2 - n$. I dette tilfælde kan vi definere $P(n)$ for ethvert heltal n. Udsagnet $P(4)$ er for eksempel uligheden $4 + 10 \leq 4^2 - 4$. Dette er dog falsk, da $14 = 4 + 10 > 4^2 - 4 = 12$. På den anden side er udsagnet $P(5)$ sandt, da $15 = 5 + 10 \leq 5^2 - 5 = 20$. Vi hævder derfor, at $P(n)$ er sandt for ethvert $n \in \mathbb{Z}_{\geq 5}$ og giver et bevis ved induktion ved hjælp af Sætning 5.5.1 med $b = 5$:

Basistrin: Vi har allerede verificeret, at $P(5)$ er gyldigt, så basistilfældet er klaret.

Induktionstrin: Lad $n \geq 6$ være et vilkårligt naturligt tal, og antag som induktionshypotese, at $P(n-1)$ er gyldigt. Dette betyder, at vi må antage, at $(n-1) + 10 \leq (n-1)^2 - (n-1)$. Ved hjælp af denne antagelse skal vi udlede, at $P(n)$ er gyldigt. Lad os først omskrive induktionshypotesen til en mere bekvem form. Vi har $(n-1) + 10 = n + 9$, mens $(n-1)^2 - (n-1) = n^2 - 2n + 1 - n + 1 = n^2 - 3n + 2$. Derfor svarer induktionshypotesen til antagelsen, at uligheden $n + 9 \leq n^2 - 3n + 2$ er gyldig. Men så kan vi udlede:

$$\begin{aligned} n + 10 &= (n+9) + 1 \\ &\leq (n^2 - 3n + 2) + 1 \\ &= n^2 - 3n + 3 \\ &= n^2 - n - 2n + 3 \\ &\leq n^2 - n. \end{aligned}$$

Den endelige ulighed holder, da $-2n + 3 \leq 0$ for ethvert $n \geq 6$ (faktisk endda for ethvert $n \geq 2$). Vi konkluderer, at hvis $P(n-1)$ er sandt, så er $n + 10 \leq n^2 - n$, det vil sige $P(n)$, også sandt. Dette er præcis, hvad vi skulle vise, hvilket fuldfører induktionstrinnet.

Ved at bruge induktion med basistrin 5 kan vi konkludere, at uligheden $n + 10 \leq n^2 - n$ er gyldig for alle $n \in \mathbb{Z}_{\geq 5}$.