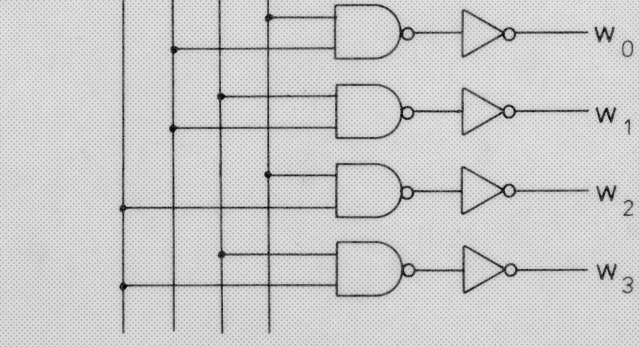
Jørgen Ebert

# BOOLESK ALGEBRA

## Digitalalgebra og talsystemer




SYSTIME

Jørgen Ebert

# BOOLESK ALGEBRA

## Digitalalgebra og talsystemer

SYSTIME

Boolesk Algebra



Omslag: Hans Møller

Tryk: Narayana Press, Gylling

I. oplag 1985

ISBN 87-7351-339-3



forlaget systime a/s
Klokkebakken 2o, Gjellerup
74oo Herning
(07) 11 9o 11

# Indhold

# Forord

Ved en overfladisk betragtning kan man få det indtryk, at de matematiske arbejdsmetoder har ændret sig voldsomt i de seneste 10-15 år. Den nye teknologi har latterliggjort gamle dyder som for eksempel omhyggelighed i forbindelse med kurvetegning. I dag har datamaten plottet funktionsgrafen allerede inden man har nået at spidse sin blyant. Selv billige lommeregnere kan udregne bestemte integraler og løse lineære ligninger med mange ubekendte. Dette kan give anledning til alvorlige frustrationer. Det er mit håb, at denne bog kan medvirke til at slå fast, at datateknologien hviler på menneskets evne til at ræsonnere, - og ikke omvendt.

I bogens første del udvikles den booleske algebra. Opbygningen sker på aksiomatisk grundlag og som en generalisation af logik- og mængdealgebraerne. Hovedresultaterne findes resumeret i en formelsamling bagest i bogen. Formelsamlingen kan være nyttig ved læsning af kapitlerne 3-9.

Bogens anden del introducerer digitalalgebraen som et specialtilfælde af den booleske algebra. Digitale funktioner behandles ved hjælp af udtryk, tabeller og diagrammer.

I tredie del undersøges talsystemernes matematiske baggrund, og med anvendelse af resultaterne fra digitalalgebraen konstrueres og analyseres digitale talbehandlingsnetværk.

Bogen kan bruges på gymnasiets matematisk-fysiske gren i forbindelse med den nye bekendtgørelses datalogiaspekt eller det valgfrie emne. Stoffet rummer en oplagt mulighed for et frugtbart samarbejde med faget fysik.

Jørgen Ebert,
Sønderborg,
juli 1985.

# 1. Logikalgebra

Den engelske filosof og matematiker, *George Boole*, der levede fra 1815 til 1864, arbejdede med udviklingen af et nyt logisk sprog, der skulle gøre det muligt at udtrykke selv meget komplicerede sammenstillinger af udsagn på en præcis og overskuelig måde.

Lad os begynde med at se på to matematiske udsagn:

p: $x+3=8$
q: $x^2=25$.

Om disse udsagn er sande eller falske afhænger helt af, hvilken værdi x tillægges. Hvis x tillægges værdien 5, er de begge sande. Hvis x tillægges værdien -5, er det kun q, der er sand. I alle andre tilfælde er de begge falske.

I *den matematiske analyse* beskæftiger man sig blandt andet med at undersøge, for hvilke værdier af x sådanne udsagn er sande.

I *den matematiske logik*, derimod, beskæftiger man sig med at undersøge sandhedsværdierne af kombinationer af udsagn, -uden at interessere sig for, hvad de enkelte udsagn egentlig står for.

De to nævnte udsagn kan kombineres på mangfoldige måder. For eksempel således:

"p og q er sande, eller p er ikke sand".

Her er benyttet ordene *og*, *eller* samt *ikke* til at kombinere udsagnene p og q. Disse ord er hentet fra det dagligt talte sprog, og dét kan give anledning til store forståelsesmæssige problemer. De to følgende sætninger viser, hvorledes ordet *og* eksempelvis kan bruges til daglig:

"Han tog jakken på *og* gik ud af døren".

"Han vejer 87 kg *og* har en højde på 1.75 meter".

I den første sætning udtrykker *og* en tidsmæssig rækkefølge, mens dette ikke er tilfældet i den anden sætning. Som et eksempel på forskellige anvendelser af ordet *eller* kan man nævne:

"Hun er enten 18 *eller* 19 år gammel".

"Hun fik en god matematikkarakter, så enten har hun været meget flittig, *eller* også er hun i besiddelse af et naturligt talent for matematik".

I den første af disse to sætninger udelukker de to muligheder hinanden, idet pigen jo ikke kan være både 18 og 19 år gammel. I den anden sætning er der derimod ikke noget i vejen for, at pigen både har været meget flittig og er i besiddelse af et naturligt talent for matematik.

Anvendelsen af almindelige ord som kobling mellem matematiske udsagn kan føre til en meget ringe præcision, fordi de matematiske udsagn ikke leverer nogen underforstået mening, således som tilfældet er med dagligdagens udsagn. For eksempel kan man ikke vide, hvilke værdier af x, der er tale om i den følgende kombination:

"Enten er $x+3=8$, *eller* også er $x^2=25$".

Hvis *eller* opfattes på samme måde som i eksemplet med pigens alder, er der kun én mulig værdi af x, nemlig -5. Men hvis *eller* opfattes som i eksemplet med pigens matematikkarakter, er der to mulige værdier af x, nemlig -5 og 5.

George Boole påviste, at alle sædvanlige koblingsmåder mellem logiske udsagn kan udtrykkes udelukkende ved anvendelse af ordene *og*, *eller* samt *ikke*. Hvis derfor disse tre ord gives en præcis betydning, som ikke er afhængig af den sammenhæng, hvori de indgår, er det muligt at udtrykke alle sammenstillinger af logiske udsagn på en entydig måde. Den måde, hvorpå de logiske udsagn kobles sammen, minder i mange henseender om den måde, hvorpå tal kan kobles sammen ved hjælp af forskellige operationer, som for eksempel *plus* og *gange*. Dette skal vi beskæftige os med i det følgende.

I sædvanlig talregning kan tallene opfattes som objekter, man kan manipulere med, således at der opstår nye objekter. For eksempel kan man koble tallene 3 og 4 sammen ved hjælp af *plus*, således at tallet 7 opstår. Operationen *plus* kræver to tal for at danne et tredie. Der findes også operationer, der kun kræver ét tal for at danne et andet. For eksempel er den naturlige eksponentialfunktion en sådan operation. Hvis den opererer på tallet 3 opstår tallet $e^3$.

Generelt siger vi, at en *dyadisk operator* i en mængde er en forskrift, der til ethvert par af elementer i mængden tilordner et element i mængden.

Desuden definerer vi generelt, at en *monadisk operator* i en mængde er en forskrift, der til ethvert element i mængden tilordner et element i mængden.

Et system bestående af en mængde samt én eller flere dyadiske eller monadiske operatorer i mængden vil vi kalde en *algebraisk struktur*.

Systemet $(R,+,\cdot)$ er et simpelt eksempel på en algebraisk struktur bestående af mængden, R, af reelle tal samt de to dyadiske operatorer *plus* og *gange*. Denne algebraiske struktur har visse velkendte egenskaber som for eksempel:

$$x(y+z)=xy+xz,$$

der gælder for vilkårlige reelle tal, x, y og z.

Idet V betegner mængden af vektorer i en orienteret plan, er $(V,+,\hat{})$ et eksempel på en algebraisk struktur med én dyadisk og én monadisk operator. Den dyadiske operator er den sædvanlige vektoraddition, og den monadiske operator er den forskrift, der til enhver vektor tilordner vektorens tværvektor. Også denne algebraiske struktur har en række egenskaber, som man skal kende til for at beherske vektorregningen. For eksempel gælder der for vilkårlige vektorer $\vec{a}$ og $\vec{b}$ i V, at

$$\widehat{\vec{a}+\vec{b}} = \hat{\vec{a}}+\hat{\vec{b}}.$$

Vi skal nu definere en ny algebraisk struktur, og undersøge nogle af dens egenskaber. Den nye algebraiske struktur hedder *logikalgebraen*. Lidt løst kan man sige, at i logikalgebraen er objekterne samlingen af alle udsagn, og operatorerne er *eller*, *og* samt *ikke*.

Vi lader L betegne mængden af udsagn. Hvis p er et element i L, så er p altså et udsagn, der enten er sandt eller falsk. Hvad udsagnet p faktisk udtaler sig om, vil vi ikke interessere os for. Vi kan blot tænke på p som en *udsagnsvariabel*, på samme måde som man opfatter x som en talvariabel i udtrykket x+ln(x).

Den monadiske operator *ikke* vil vi fremover kalde *negation*. Vi definerer operatoren ved at fastlægge, hvorledes den påvirker et udsagns sandhedsværdi. Hvis p tilhører L, skal negationen af p være sand, præcis når p er falsk. Negationen af p vil vi skrive $\neg p$. Definitionen kan udtrykkes ved hjælp af en såkaldt *sandhedstabel*:

| p | $\neg$p |
|---|---|
| f | s |
| s | f |

Den dyadiske operator *eller* kalder vi *disjunktion*. Hvis p og q er elementer i L, vil vi lade $p \vee q$ betegne disjunktionen af udsagnsparret (p,q). Disjunktionens betydning fastlægger vi med kravet om, at $p \vee q$ er falsk, præcis når såvel p som q er falsk. Sandhedstabellen for disjunktionen er derfor:

| p | q | $p \vee q$ |
|---|---|---|
| f | f | f |
| f | s | s |
| s | f | s |
| s | s | s |

Den dyadiske operator *og* kalder vi *konjunktion*, og dens anvendelse på udsagnsparret (p,q) skriver vi $p \wedge q$. Konjunktionens betydning er fastlagt med kravet om, at $p \wedge q$ er sand, præcis når såvel p som q er sande. Sandhedstabellen for konjunktionen er altså:

| p | q | $p \wedge q$ |
|---|---|---|
| f | f | f |
| f | s | f |
| s | f | f |
| s | s | s |

Et *logisk udtryk* består af udsagnsvariable, der er koblet sammen ved hjælp af operatorer. For eksempel er

$$(p \vee (\neg q)) \wedge (\neg r)$$

et logisk udtryk med tre udsagnsvariable. Hvis man vil undersøge, hvorledes sandhedsværdien af et logisk udtryk afhænger af sandhedsværdierne af de enkelte udsagnsvariable, kan man konstruere sandhedstabellen for det logiske udtryk. Sandhedstabellen for ovennævnte logiske udtryk kan således konstrueres ved anvendelse af tre hjælpesøjler:

| p | q | r | ¬q | p∨(¬q) | ¬r | (p∨(¬q))∧(¬r) |
|---|---|---|---|---|---|---|
| f | f | f | s | s | s | s |
| f | f | s | s | s | f | f |
| f | s | f | f | f | s | f |
| f | s | s | f | f | f | f |
| s | f | f | s | s | s | s |
| s | f | s | s | s | f | f |
| s | s | f | f | s | s | s |
| s | s | s | f | s | f | f |

Hvis to logiske udtryk har de samme sandhedstabeller, vil vi sige, at de to logiske udtryk er *ækvivalente*, og vi vil opfatte dem som værende ens. For eksempel viser tabellen

| p | p | p∨p |
|---|---|---|
| f | f | f |
| s | s | s |

at udtrykket p∨p er ækvivalent med p. Dette vil vi skrive ved hjælp af *ækvivalenstegnet* ≡, således: p∨p≡p. Ækvivalenstegnet spiller samme rolle i logikken, som lighedstegnet spiller i sædvanlig talregning.

Hvis et logisk udtryk har en sandhedstabel bestående af lutter s'er, vil vi kalde udtrykket en *tautologi*. For eksempel er udtrykket q∨(p∨(¬p)) en tautologi, da udtrykket er sandt uafhængigt af sandhedsværdierne af udsagnene p og q. Sandhedstabellen for udtrykket er nemlig:

| p | q | ¬p | p∨(¬p) | q∨(p∨(¬p)) |
|---|---|---|---|---|
| f | f | s | s | s |
| f | s | s | s | s |
| s | f | f | s | s |
| s | s | f | s | s |

Af mellemregningerne i tabellen ses endvidere, at det logiske udtryk $p \vee (\neg p)$ også er en tautologi, og at

$$p \vee (\neg p) \equiv q \vee (p \vee (\neg p)).$$

Af definitionerne på ækvivalens og tautologi fås direkte, at *alle tautologier er indbyrdes ækvivalente*. Vi kan altså vælge *én tautologi*, *S*, som repræsentant for alle tautologierne. At $p \vee (\neg p)$ er en tautologi, kan herefter udtrykkes således:

$$p \vee (\neg p) \equiv S.$$

Hvis S optræder som udsagn i et logisk udtryk, skal man ved opstilling af sandhedstabellen for udtrykket være opmærksom på, at S kun kan antage sandhedsværdien s. S er altså i denne sammenhæng ikke en udsagnsvariabel men en *udsagnskonstant*, på samme måde som tallet 3 er en talkonstant i taludtrykket $3+\ln(x)$.

I modsætning til en tautologi er en *kontradiktion* et logisk udtryk, hvis sandhedstabel består af lutter f'er. En kontradiktion er altså falsk uafhængigt af sandhedsværdierne af de udsagn, der indgår i udtrykket. På samme måde som tilfældet var med tautologierne, er det let at bevise, at *alle kontradiktioner er indbyrdes ækvivalente*. Vi kan derfor vælge *én kontradiktion*, *F*, som repræsentant for alle kontradiktionerne. F kan optræde som udsagnskonstant i et logisk udtryk, og kan kun antage sandhedsværdien f. Hvis man skriver

$$p \wedge (\neg p) \equiv F,$$

betyder det, at det logiske udtryk $p \wedge (\neg p)$ er en kontradiktion. Beviset for at dette er rigtigt, fremgår af sandhedstabellen for $p \wedge (\neg p)$:

| p | ¬p | p∧(¬p) |
|---|---|---|
| f | s | f |
| s | f | f |

Vi er nu klar til at give den endelige definition på den nye algebraiske struktur, logikalgebraen:

DEFINITION 1. *Logikalgebraen*, $(L,\vee,\wedge,\neg)$, består af mængden af alle udsagn, L, samt af de to dyadiske operatorer, disjunktion og konjunktion, og den monadiske operator, negation.

I det følgende skal vi bevise fire helt fundamentale sætninger angående nogle egenskaber ved logikalgebraen.

SÆTNING 1. (*De kommutative regler*).
For vilkårlige udsagn, p og q, gælder:

(1') $p\vee q \equiv q\vee p$

(1") $p\wedge q \equiv q\wedge p$

BEVIS: Vi opskriver sandhedstabellerne for henholdsvis venstre og højre side af ækvivalensen (1'):

| p | q | $p\vee q$ |
|---|---|---|
| f | f | f |
| f | s | s |
| s | f | s |
| s | s | s |

| p | q | p | $q\wedge p$ |
|---|---|---|---|
| f | f | f | f |
| f | s | f | s |
| s | f | s | s |
| s | s | s | s |

Da $p\vee q$ har samme sandhedstabel som $q\vee p$, er de to udtryk ækvivalente. Hermed er (1') bevist. Beviset for (1"), der forløber på næsten samme måde, overlades til læseren. □

SÆTNING 2. (*De distributive regler*).
For vilkårlige udsagn, p og q, gælder:

(2') $p\vee(q\wedge r) \equiv (p\vee q)\wedge(p\vee r)$

(2") $p\wedge(q\vee r) \equiv (p\wedge q)\vee(p\wedge r)$

BEVIS: Sandhedstabellerne for henholdsvis venstre og højre side af ækvivalensen (2') er:

| p | q | r | $q \wedge r$ | $p \vee (q \wedge r)$ |
|---|---|---|---|---|
| f | f | f | f | f |
| f | f | s | f | f |
| f | s | f | f | f |
| f | s | s | s | s |
| s | f | f | f | s |
| s | f | s | f | s |
| s | s | f | f | s |
| s | s | s | s | s |

| p | q | r | $p \vee q$ | $p \vee r$ | $(p \vee q) \wedge (p \vee r)$ |
|---|---|---|---|---|---|
| f | f | f | f | f | f |
| f | f | s | f | s | f |
| f | s | f | s | f | f |
| f | s | s | s | s | s |
| s | f | f | s | s | s |
| s | f | s | s | s | s |
| s | s | f | s | s | s |
| s | s | s | s | s | s |

Heraf ses, at venstre og højre side af (2') er ækvivalente, hvilket beviser gyldigheden af (2'). Beviset for (2") overlades til læseren. □

SÆTNING 3. (*Eksistens af neutrale elementer*).
For et vilkårligt udsagn, p, gælder:

(3') $p \vee F \equiv p$

(3") $p \wedge S \equiv p$

BEVIS: Sandhedstabellen for $p \vee F$ er:

| p | F | $p \vee F$ |
|---|---|---|
| f | f | f |
| s | f | s |

Heraf ses, at $p \vee F$ har de samme sandhedsværdier som p. Derfor er $p \vee F$ ækvivalent med p. Hermed er (3') bevist. Beviset for (3") overlades til læseren. □

Ækvivalensen (3') kan sprogligt formuleres således: En kontradiktion ændrer ikke sandhedsværdien af et logisk udsagn ved disjunktion. Man kan derfor i en vis forstand sige, at *en kontradiktion er neutral over for disjunktion.* På samme måde kan (3") udtrykkes ved: *En tautologi er neutral over for konjunktion.* Bemærk, at sætning 3 ikke udtaler sig om, hvorvidt der er andre neutrale elementer over for henholdsvis disjunktion og konjunktion.

> SÆTNING 4. (*Eksistens af komplementer*).
> For et vilkårligt udsagn, p, gælder:
>
> (4') $p \vee (\neg p) \equiv S$
>
> (4") $p \wedge (\neg p) \equiv F$

BEVIS: Sandhedstabellen for venstre side af ækvivalensen (4') ser således ud:

| p | ¬p | p∨(¬p) |
|---|---|---|
| f | s | s |
| s | f | s |

Heraf ses, at $p \vee (-p)$ er en tautologi, og dermed at (4') er gyldig. Beviset for (4") overlades til læseren. □

Et udsagn, q, med den egenskab, at

$$p \vee q \equiv S \quad \text{og} \quad p \wedge q \equiv F, \Rightarrow q = \neg p \text{ eller } \neg q = p$$

siges at være et *komplement til p*. Sætning 4 siger altså, at ¬p er et komplement til p. Sætningen siger intet om, hvorvidt p har andre komplementer.

Vi er nu færdige med at bevise de fire grundlæggende egenskaber ved $(L, \vee, \wedge, \neg)$. I kapitel 3 skal vi undersøge logikalgebraen nærmere, blot i en langt bredere sammenhæng.

# 2. Mængdealgebra

I det følgende betegner E en bestemt *grundmængde*, der kan være tom eller bestå af endelig eller uendelig mange elementer. Vi skal beskæftige os med delmængderne af denne grundmængde. Derfor lader vi M betegne *mængden af alle delmængder af grundmængden*. Hvis E eksempelvis består af tre elementer,

$$E = \{a,b,c\},$$

så har E præcis otte delmængder, nemlig:

$$\emptyset \quad \{a\} \quad \{b\} \quad \{c\} \quad \{a,b\} \quad \{a,c\} \quad \{b,c\} \quad \{a,b,c\}.$$

I dette tilfælde vil disse otte *delmængder* af E udgøre *elementerne* i M. Altså kan vi skrive:

$$M = \Big\{\emptyset,\{a\},\{b\},\{c\},\{a,b\},\{a,c\},\{b,c\},\{a,b,c\}\Big\}.$$

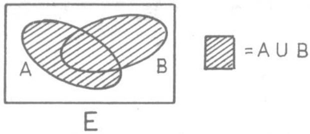
Bemærk, at M aldrig kan være tom, heller ikke hvis E er den tomme mængde. M vil nemlig altid indeholde elementerne $\emptyset$ og E, som eventuelt kan være sammenfaldende.

Elementerne i M skal fungere som objekter i en algebraisk struktur. Derfor skal vi nu definere tre operatorer i M.

*Foreningsmængdeoperatoren*, $\cup$, er en dyadisk operator i M. Den er defineret ved, at for vilkårlige elementer A og B i M (det vil sige for vilkårlige delmængder A og B af E), er

$$A \cup B = \{x \in E \mid x \in A \vee x \in B\}.$$

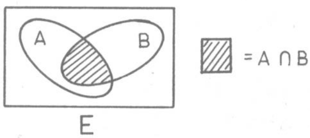
Da $A \cup B$ ses at være en delmængde af E, altså et element i M, er der virkelig tale om, at foreningsmængdeoperatoren er en dyadisk operator i M. Dens virkemåde illustreres ofte ved hjælp af et såkaldt Venn-diagram:




*Fællesmængdeoperatoren*, $\cap$, er en dyadisk operator i M. Den er defineret ved, at for vilkårlige elementer A og B i M, er

$$A \cap B = \{x \in E \mid x \in A \wedge x \in B\}.$$

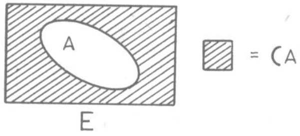
Igen ses det, at der virkelig er tale om en operator i M, thi $A \cap B$ er en delmængde af E, og dermed et element i M. Venn-diagrammet for fællesmængdeoperatoren ser således ud:




*Komplementærmængdeoperatoren*, $\complement$, er en monadisk operator i M. Den er defineret ved, at for et vilkårligt element, A, i M er

$$\complement A = \{x \in E \mid \neg(x \in A)\}.$$

Da $\complement A$ ses at være en delmængde af E, er $\complement A$ et element i M. Derfor er komplementærmængdeoperatoren virkelig en monadisk operator i M. Venn-diagrammet ser således ud:




Da vi nu har defineret de tre operatorer i M, er vi rede til definitionen af mængdealgebraen:

DEFINITION 1. *Mængdealgebraen*, $(M,\cup,\cap,\complement)$, består af mængden, M, af alle delmængder af grundmængden, E, samt af de to dyadiske operatorer, foreningsmængdeoperatoren og fællesmængdeoperatoren, og den monadiske operator, komplementærmængdeoperatoren.

Ligesom tilfældet var med logikalgebraen, skal vi bevise fire fundamentale sætninger om mængdealgebraens egenskaber.

SÆTNING 1. (*De kommutative regler*).
For vilkårlige delmængder, A og B, af E gælder:

(1') $A\cup B = B\cup A$

(1") $A\cap B = B\cap A$

BEVIS: Da disjunktionen er kommutativ, fås:

$$\begin{aligned} A\cup B &= \{x\in E \mid x\in A \vee x\in B\} \\ &= \{x\in E \mid x\in B \vee x\in A\} \\ &= B\cup A. \end{aligned}$$

Hermed er (1') bevist. Beviset for (1") overlades til læseren. □

SÆTNING 2. (*De distributive regler*).
For vilkårlige delmængder, A, B og C, af E gælder:

(2') $A\cup(B\cap C) = (A\cup B)\cap(A\cup C)$

(2") $A\cap(B\cup C) = (A\cap B)\cup(A\cap C)$

BEVIS: Da disjunktionen er distributiv med hensyn til konjunktionen, fås:

$$\begin{aligned} A\cup(B\cap C) &= \{x\in E \mid x\in A \lor x\in B\cap C\} \\ &= \{x\in E \mid x\in A \lor (x\in B \land x\in C)\} \\ &= \{x\in E \mid (x\in A \lor x\in B) \land (x\in A \lor x\in C)\} \\ &= \{x\in E \mid x\in A\cup B \land x\in A\cup C\} \\ &= (A\cup B)\cap(A\cup C). \end{aligned}$$

Hermed er (2') bevist. Beviset for (2") forløber på næsten samme måde. Dette overlades til læseren. □

**SÆTNING 3.** (*Eksistens af neutrale elementer*).
For en vilkårlig delmængde, A, af E gælder:

(3') $A\cup\emptyset = A \Rightarrow \emptyset \subseteq A$

(3") $A\cap E = A \Rightarrow A \subseteq E$

BEVIS: Da udsagnet $x\in\emptyset$ er falsk for alle elementer, x, i E, er dette udsagn en kontradiktion. Da en kontradiktion er neutral over for disjunktion, gælder:

$$\begin{aligned} A\cup\emptyset &= \{x\in E \mid x\in A \lor x\in\emptyset\} \\ &= \{x\in E \mid (x\in A) \lor F\} \\ &= \{x\in E \mid x\in A\} \\ &= A. \end{aligned}$$

Dette beviser gyldigheden af (3'). Beviset for (3") anvender, at udsagnet $x\in E$ er en tautologi, når det på forhånd vides, at vi befinder os inden for grundmængden, E. Dette bevis overlades til læseren. □

Af sætning 3 ses, at $\emptyset$ er neutral over for foreningsmængdeoperatoren, og at E er neutral over for fællesmængdeoperatoren. Mængdealgebraen har altså ligesom logikalgebraen to neutrale elementer over for henholdsvis den ene og den anden af de to dyadiske operatorer.

**SÆTNING 4.** (*Eksistens af komplementer*).
For en vilkårlig delmængde, A, af E gælder:

$$(4') \quad A \cup (\complement A) = E$$
$$(4'') \quad A \cap (\complement A) = \emptyset$$

BEVIS: Da udsagnet $(x \in A) \vee \neg (x \in A)$ er en tautologi ifølge sætning 4 i kapitel 1, er dette udsagn sandt for ethvert element, x, i E. Altså er:

$$\begin{aligned} A \cup (\complement A) &= \{x \in E \mid (x \in A) \vee \neg (x \in A)\} \\ &= \{x \in E \mid x \in E\} \\ &= E. \end{aligned}$$

Hermed er rigtigheden af (4') bevist. Beviset for (4") overlades til læseren. □

I overensstemmelse med terminologien i kapitel 1 vil vi sige, at en delmængde, B, af E, som opfylder

$$A \cup B = E \qquad \text{og} \qquad A \cap B = \emptyset, \implies B = \complement A \text{ eller } A = \complement B$$

er et komplement til A. Sætning 4 siger altså, at $\complement A$ er et komplement til A.

Hermed har vi bevist de fire grundlæggende sætninger om mængdealgebraen, $(M, \cup, \cap, \complement)$. Beviserne har ret nøje fulgt de tilsvarende sætninger i kapitel 1. Det slægtsskab, som derfor må formodes at være mellem logikalgebraen og mængdealgebraen, er emnet for det næste kapitel.

# 3. Boolesk algebra

I de to foregående kapitler har vi beskæftiget os med henholdsvis logikalgebraen og mængdealgebraen. Disse to algebraiske strukturer har så mange lighedspunkter, at den videre analyse af deres egenskaber med fordel kan foretages på én gang.

Dette kan vi kun opnå, hvis vi definerer en mere generel algebraisk struktur, der rummer både logikalgebraen og mængdealgebraen som specialtilfælde. De matematiske sætninger, vi kan bevise om denne generaliserede algebraiske struktur, vil også være gyldige for logikalgebraen og for mængdealgebraen, samt for *alle* andre specialtilfælde, for eksempel digitalalgebraen, som vi senere skal beskæftige os med.

En *generalisering* vil således reducere den arbejdsbyrde, der er forbundet med undersøgelsen af beslægtede algebraiske strukturer. For mange videnskabsfolk rummer en generalisering desuden en kilde til æstetisk nydelse. Den fornemmelse af velvære, der ligger i at kunne forklare mange forskelligartede fænomener ud fra én simpel, overordnet *model*, må formodes at have været -og stadig er- en af de allerstørste tilskyndelser til videnskabelig forskning. Desværre har en generalisering den ulempe, at vi kommer til at arbejde mere abstrakt og dermed mindre intuitivt. Dette forhold stiller store krav til præcision i bevisførelser for sætningerne.

Når vi skal definere den generaliserede algebraiske struktur, må vi lade os vejlede af, hvad logikalgebraen og mængdealgebraen har fælles. De består begge af fire ting:

- En mængde.
- En dyadisk operator i mængden.
- En anden dyadisk operator i mængden.
- En monadisk operator i mængden.

Desuden gælder der fire regler i hver af de to algebraiske strukturer. Disse fire regler kan vi lidt upræcist resumére således:

- Hver af de to dyadiske operatorer er kommutative.
- De to dyadiske operatorer er distributive med hensyn til hinanden.
- Der findes et neutralt element for hver af de to dyadiske operatorer.
- Når den monadiske operator virker på et element i mængden, så opstår der et komplement til elementet.

Enhver algebraisk struktur med disse egenskaber vil vi kalde en *boolesk algebra*. De fire regler vil vi opfatte som *aksiomer*, det vil sige regler ud fra hvilke alle andre regler skal bevises.

For at præcisere definitionen på en boolesk algebra, må vi indføre *betegnelser* for mængden, de tre operatorer samt for de neutrale elementer. Det letteste ville være at adoptere betegnelserne fra logikalgebraen eller fra mængdealgebraen, men så kan der let opstå tvivl om, hvorvidt vi i en given situation arbejder med den generaliserede algebraiske struktur eller med en af de "gamle" algebraiske strukturer. Derfor vælger vi at indføre nye betegnelser. Den følgende oversigt viser betegnelserne i logikalgebraen og i mængdealgebraen samt de tilsvarende betegnelser i den generelle booleske algebra:

| | Logikalgebra | Mængdealgebra | Boolesk algebra |
|---|---|---|---|
| Mængde | L | M | B |
| Objekter | Udsagn | Delmgd. af E | ? |
| Identitet | $\equiv$ | $=$ | $=$ |
| Dyadisk operator (1) | $\vee$ | $\cup$ | $+$ |
| Dyadisk operator (2) | $\wedge$ | $\cap$ | $\cdot$ |
| Monadisk operator | $\neg$ | $\complement$ | $^{-}$ |
| Neutralt element (1) | F | $\emptyset$ | 0 |
| Neutralt element (2) | S | E | 1 |

Herefter er vi klar til at give en præcis definition på en boolesk algebra:

DEFINITION 1. En *boolesk algebra*, $(B,+,\cdot,^{-})$, er en algebraisk struktur bestående af en mængde, B, med to dyadiske operatorer, + og $\cdot$, samt én monadisk operator, $^{-}$. Mængden, B, skal indeholde et element, 0, og et element, 1. Desuden skal de følgende fire aksiomer være opfyldt.

AKSIOM 1. (*De kommutative regler*).
For vilkårlige elementer, x og y, i B gælder:

(1') $x+y = y+x$
(1") $x \cdot y = y \cdot x$

AKSIOM 2. (*De distributive regler*).
For vilkårlige elementer, x og y, i B gælder:

(2') $x+(y \cdot z) = (x+y) \cdot (x+z)$
(2") $x \cdot (y+z) = (x \cdot y)+(x \cdot z)$

AKSIOM 3. (*Eksistens af neutrale elementer*).
For et vilkårligt element, x, i B gælder:

(3') $x+0 = x$
(3") $x \cdot 1 = x$

AKSIOM 4. (*Eksistens af komplementer*).
For et vilkårligt element, x, i B gælder:

(4') $x+\bar{x} = 1$
(4") $x \cdot \bar{x} = 0$

Denne definition er blot en oversættelse af definitionen på $(L, \vee, \wedge, \neg)$ samt af de fire grundlæggende sætninger om logikalgebraen. Derfor er logikalgebraen en boolesk algebra. Præcis de samme bemærkninger gør sig gældende for mængdealgebraen. Herefter er det klart, at enhver egenskab ved den generelle booleske algebra kan fortolkes som en egenskab ved logikalgebraen og ved mængdealgebraen, men *ikke* omvendt.

I valget af betegnelserne, +, ·, 0 og 1, er der *ikke* indbygget nogen antagelse om, at de sædvanlige talregneregler gælder. Alligevel vælger vi at kalde operatorerne, + og ·, for henholdsvis *addition* og *multiplikation*.

Det første aksiom siger, at såvel additionen som multiplikationen er kommutative operatorer i B.

Det andet aksiom siger, at additionen er distributiv med hensyn til multiplikationen, og at multiplikationen er distributiv med hensyn til additionen. Den ene (hvilken ?) af disse distributive regler er helt ny i forhold til "vanen" fra almindelig talregning. at + er distributiv med.

Det tredie aksiom siger, at 0 og 1 er neutrale elementer over for henholdsvis additionen og multiplikationen. På forhånd kan vi ikke udelukke muligheden for, at additionen eller multiplikationen har andre neutrale elementer. Dette er ét af de problemer, vi skal undersøge i det følgende.

Det fjerde aksiom er det eneste af de fire aksiomer, der inddrager den monadiske operator. Aksiomet siger, at $\bar{x}$ er et komplement til x. Af denne grund kalder vi operatoren, $^{-}$, for *komplementoperatoren*. Spørgsmålet om, hvorvidt et element, x, kan have andre komplementer end $\bar{x}$, er endnu uafklaret.

Om selve mængden, B, kan vi ikke sige ret meget, udover at den i hvert fald ikke er tom, da den jo indeholder 0 og 1. Muligheden for, at 0 og 1 er ens, er *ikke* udelukket, og det vides heller ikke, hvor mange øvrige elementer B indeholder.

Ovenstående betragtninger viser, at vi ikke skal *tro* på nogen egenskab ved den booleske algebra, med mindre vi har ført et præcist *bevis* for egenskaben. En sådan egenskab er kun gyldig, såfremt den kan føres tilbage til de fire aksiomer. Når vi i det følgende skal bevise rigtigheden af en lang række egenskaber ved den booleske algebra, skal beviserne altså føres alene på grundlag af de fire aksiomer eller på grundlag af resultater, der tidligere er udledt af de fire aksiomer. Man siger generelt, at en matematisk teori har en *aksiomatisk opbygning*, hvis alle teoriens sætninger kan bevises udelukkende ved anvendelse af et givet sæt af aksiomer. I resten af dette kapitel skal vi foretage en aksiomatisk opbygning af teorien for den booleske algebra. Vi skal først bevise, at 0 og 1 faktisk *er* de eneste elementer, der er neutrale over for henholdsvis addition og multiplikation.

SÆTNING 5. (*Entydighed af neutrale elementer*).

(5') 0 er det eneste element i B, der er neutralt over for addition.

(5") 1 er det eneste element i B, der er neutralt over for multiplikation.

BEVIS: Antag, at z er et element i B, der er neutralt over for addition. Det er så vores opgave at bevise, at z er lig med 0. Dette beviser vi således:

| | | |
|---|---|---|
| $z$ | $= z+0$ | ifølge (3') |
| | $= 0+z$ | ifølge (1') |
| | $= 0$ | ifølge antagelsen |

Dette beviser, at hvis z er neutral over for addition, så er z lig med 0. Altså er 0 det eneste element i B, der er neutralt over for addition. Hermed er (5') bevist. Antag dernæst, at u er et element i B, der er neutralt over for multiplikation. Så gælder:

| | | |
|---|---|---|
| $u$ | $= u\cdot 1$ | ifølge (3") |
| | $= 1\cdot u$ | ifølge (1") |
| | $= 1$ | ifølge antagelsen |

Hvis u er neutralt over for multiplikation, er således u lig med 1. Derfor er der ikke andre elementer end 1, der er neutrale over for multiplikation. Dette beviser (5"). Hermed er sætning 5 bevist. □

Sætningen tillader os at tale om *det* neutrale element over for addition, og om *det* neutrale element over for multiplikation. Sætningens resultat kan direkte oversættes til en egenskab ved logikalgebraen, thi da sætningen gælder i enhver boolesk algebra, så gælder den også for logikalgebraen. Oversat til logikalgebraen lyder sætningen:

- Kontradiktionerne er de eneste udsagn, der er neutrale over for disjunktion.
- Tautologierne er de eneste udsagn, der er neutrale over for konjunktion.

ØVELSE 1. Oversæt sætning 5 til mængdealgebraen.

Den følgende sætning udelukker muligheden for, at et element i B kan have flere forskellige komplementer. Dermed bortfalder endnu en af de usikkerheder, der blev omtalt tidligere.

SÆTNING 6. (*Entydighed af komplementer*).
For et vilkårligt element, x, i B gælder, at $\bar{x}$ er det eneste komplement til x.

BEVIS: Antag, at c er et komplement til x. Vi skal så bevise, at c er lig med $\bar{x}$. Da c er antaget at være et komplement til x, véd vi, at der gælder:

$$x+c = 1 \quad \wedge \quad x\cdot c = 0.$$

Dette benyttes nedenfor:

| | |
|---|---|
| $c = c\cdot 1$ | ifølge (3") |
| $= c\cdot(x+\bar{x})$ | ifølge (4') |
| $= (c\cdot x)+(c\cdot\bar{x})$ | ifølge (2") |
| $= (x\cdot c)+(c\cdot\bar{x})$ | ifølge (1") |
| $= 0+(c\cdot\bar{x})$ | ifølge antagelsen |
| $= (x\cdot\bar{x})+(c\cdot\bar{x})$ | ifølge (4") |
| $= (\bar{x}\cdot x)+(\bar{x}\cdot c)$ | ifølge (1") |
| $= \bar{x}\cdot(x+c)$ | ifølge (2") |
| $= \bar{x}\cdot 1$ | ifølge antagelsen |
| $= \bar{x}$ | ifølge (3") |

Dette beviser, at hvis c er et komplement til x, så er c lig med $\bar{x}$. Derfor er $\bar{x}$ det eneste komplement til x. Hermed er sætning 6 bevist. □

Sætningen tillader os at tale om komplement*et* til et element i B. Entydigheden af komplementet til et element, vil senere vise sig at være et værdifuldt matematisk værktøj, idet beviserne for to af de allervigtigste booleske sætninger (sætning 10 og 16) benytter netop denne egenskab.

ØVELSE 2. To mængder siges at være *disjunkte*, såfremt de ikke indeholder noget fælles element. Bevis, ved anvendelse af sætning 6, at hvis foreningsmængden af to disjunkte delmængder af grundmængden, E, udgør hele E, så er den ene delmængde lig med komplementærmængden til den anden.

ØVELSE 3. Gør, ved hjælp af sætning 6, rede for, at hvis $p \vee q$ er en tautologi, og hvis $p \wedge q$ er en kontradiktion, så er q ækvivalent med $\neg p$.

ØVELSE 4. Bevis, at i en boolesk algebra gælder formlerne:

$$\bar{0} = 1 \quad \text{og} \quad \bar{1} = 0.$$

SÆTNING 7. (*Dualitetsprincippet*).
Enhver formel, som kan udledes af den booleske algebras aksiomer, forbliver gyldig, såfremt man overalt i formlen erstatter addition med multiplikation, multiplikation med addition, 0 med 1 og 1 med 0.

BEVIS: I hver af de fire aksiomer for den booleske algebra optræder der to formler. Den ene af disse fremkommer af den anden ved anvendelse af den beskrevne procedure. Da det aksiomatiske grundlag for den booleske algebra således overholder den angivne symmetri, vil alle formler, der udledes heraf også optræde parvis (som såkaldte *duale formler*). Hermed er sætning 7 bevist. □

Dualitetsprincippet indebærer den store fordel, at hver gang vi beviser en boolesk formel, får vi "foræret" den duale formel. (Under visse omstændigheder kan de to duale formler være identiske, dette gælder for eksempel for sætning 10). Dualitetsprincippet er altså med til at reducere bevisbyrden. Desuden giver dualitetsprincippet en meget attraktiv struktur i opbygningen af den booleske teori.

EKSEMPEL 1. Når man anvender dualitetsprincippet på formlen $x+0 = x$, så fremkommer formlen $x\cdot 1 = x$.

ØVELSE 5. Anvend dualitetsprincippet på formlen $p\vee(\neg p) \equiv S$.
$\to p\wedge(\neg p) \equiv F$

ØVELSE 6. Anvend dualitetsprincippet på den følgende mængdeformel: $A\cup(B\cap C) = (A\cup B)\cap(A\cup C)$. $\to A\cap(B\cup C) = (A\cap B)\cup(A\cap C)$

| |
|---|
| SÆTNING 8. For et vilkårligt element, x, i B gælder:<br><br>(8') $x+1 = 1$<br>(8") $x\cdot 0 = 0$ |

BEVIS: Ved anvendelse af aksiomerne fås:

$$
\begin{aligned}
x+1 &= (x+1)\cdot 1 && \text{iflg. } (3") \\
&= (x+1)\cdot(x+\bar{x}) && \text{iflg. } (4') \\
&= x+(1\cdot\bar{x}) && \text{iflg. } (2') \\
&= x+(\bar{x}\cdot 1) && \text{iflg. } (1") \\
&= x+\bar{x} && \text{iflg. } (3") \\
&= 1 && \text{iflg. } (4')
\end{aligned}
$$

Hermed er (8') bevist. Ved anvendelse af dualitetsprincippet fremkommer (8"). Dette beviser sætning 8. □

$(x) + 1 = 1$

ØVELSE 7. Reducér udtrykket $((0\cdot x)+(1\cdot x))+((0+x)+(1+x))$.

ØVELSE 8. Bevis ved hjælp af sætning 8, at disjunktionen af en tautologi og et andet udsagn i sig selv er en tautologi. Giv derefter et sprogligt eksempel på denne regel.

ØVELSE 9. Oversæt sætning 8 til en regel, der gælder i mængdealgebraen.

$E = E \cup A \, , \; \emptyset = \emptyset \cap A$

ØVELSE 10. Bevis (8") *uden* at anvende dualitetsprincippet på (8').

SÆTNING 9. For et vilkårligt element, x, i B gælder:

$$(9') \quad x+x = x$$

$$(9") \quad x \cdot x = x$$

BEVIS: En lidt trick'et udregning giver:

| | |
|---|---|
| $x = x+0$ | ifølge (3') |
| $= x+(x\cdot\bar{x})$ | ifølge (4") |
| $= (x+x)\cdot(x+\bar{x})$ | ifølge (2') |
| $= (x+x)\cdot 1$ | ifølge (4') |
| $= x+x$ | ifølge (3") |

Dette beviser rigtigheden af (9'). Ved at anvende dualitetsprincippet på (9') fremkommer (9"). Hermed er sætning 9 bevist. □

ØVELSE 11. Reducér det booleske udtryk $x+(x\cdot\overline{(x\cdot\bar{x})})$ så meget som muligt.

ØVELSE 12. Gør ved hjælp af sætning 9 rede for, at formlen

$$\{t \in R \mid |t|<1 \wedge -1<t<1\} = ]-1,1[$$

er gyldig.

ØVELSE 13. Oversæt sætning 9 til betegnelserne fra mængdealgebraen og til betegnelserne fra logikalgebraen.

SÆTNING 10. For et vilkårligt element, x, i B gælder:

$$(10) \quad \bar{\bar{x}} = x$$

BEVIS: Vi indfører y som forkortelse for $\bar{x}$. Så gælder:

$$(*) \quad y = \bar{x}.$$

Heraf fås:

$$\begin{aligned} y+x &= \bar{x}+x && \text{ifølge } (*) \\ &= x+\bar{x} && \text{ifølge } (1') \\ &= 1 && \text{ifølge } (4') \end{aligned}$$

Desuden er:

$$\begin{aligned} y \cdot x &= \bar{x} \cdot x && \text{ifølge } (*) \\ &= x \cdot \bar{x} && \text{ifølge } (1'') \\ &= 0 && \text{ifølge } (4'') \end{aligned}$$

Altså gælder:

$$y+x = 1 \quad \wedge \quad y \cdot x = 0.$$

Men dette betyder jo, at x er et komplement til y. Ifølge sætning 6 er $\bar{y}$ det eneste komplement til y. Altså er $\bar{y}$ lig med x. Når man heri indsætter (*), fås, at $\bar{\bar{x}}$ er lig med x. Dette afslutter beviset for sætning 10. □

ØVELSE 14. Oversæt sætning 10 til betegnelserne fra mængdealgebraen.

ØVELSE 15. Leo Mathiesen (1906-1969) var en af pionererne inden for dansk jazz. I en af sine tekster siger han:

"- but I aint got no cigar."

Hvad betyder det?

ØVELSE 16. Løs den booleske ligning $(\bar{\bar{x}}+x) \cdot \bar{\bar{x}} = 1$.

ØVELSE 17. Find den duale formel til (10).

> **SÆTNING 11.** For vilkårlige elementer, x og y, i B gælder:
>
> $$(11') \quad (x+y)\cdot(x+\bar{y}) = x$$
> $$(11'') \quad (x\cdot y)+(x\cdot\bar{y}) = x$$

BEVIS: Ved hjælp af aksiomerne fås:

$$\begin{aligned} (x+y)\cdot(x+\bar{y}) &= x+(y\cdot\bar{y}) && \text{ifølge } (2') \\ &= x+0 && \text{ifølge } (4'') \\ &= x && \text{ifølge } (3') \end{aligned}$$

Hermed er (11') bevist. Ved anvendelse af dualitetsprincippet fremkommer (11"). □

ØVELSE 18. Anvend (11") til en fortolkning af teksten:

"Enten er det mandag og solen skinner, eller også er det mandag og solen skinner ikke."

ØVELSE 19. Bevis (11") direkte ud fra de fire aksiomer.

> **SÆTNING 12.** For vilkårlige elementer, x og y, i B gælder:
>
> $$(12') \quad x+(x\cdot y) = x$$
> $$(12'') \quad x\cdot(x+y) = x$$

BEVIS: Ved anvendelse af aksiomerne samt sætning 8 fås:

$$\begin{aligned} x+(x\cdot y) &= (x\cdot 1)+(x\cdot y) && \text{ifølge } (3'') \\ &= x\cdot(1+y) && \text{ifølge } (2'') \\ &= x\cdot(y+1) && \text{ifølge } (1') \\ &= x\cdot 1 && \text{ifølge } (8') \\ &= x && \text{ifølge } (3'') \end{aligned}$$

Hermed er (12') bevist. Når dualitetsprincippet anvendes på (12') fremkommer (12"). □

ØVELSE 20. Oversæt sætning 12 til betegnelserne fra mængdealgebraen, og illustrér såvel (12') som (12") ved hjælp af Venn-diagrammer.

ØVELSE 21. Bevis formlen $\overline{y\cdot(x+y)}+y = 1$.

ØVELSE 22. Reducér udsagnet: $t\in Q \wedge (t\in R \vee t\in Q)$, hvor Q og R som sædvanlig betegner henholdsvis mængden af rationale tal og mængden af reelle tal.

SÆTNING 13. For vilkårlige elementer, x og y, i B gælder:

$$(13')\quad x+(\bar{x}\cdot y) = x+y$$
$$(13'')\quad x\cdot(\bar{x}+y) = x\cdot y$$

BEVIS: Der gælder:

$$\begin{aligned} x+(\bar{x}\cdot y) &= (x+\bar{x})\cdot(x+y) && \text{ifølge } (2') \\ &= 1\cdot(x+y) && \text{ifølge } (4') \\ &= (x+y)\cdot 1 && \text{ifølge } (1'') \\ &= x+y && \text{ifølge } (3'') \end{aligned}$$

Hermed er (13') bevist. Dualitetsprincippet viser så, at også (13") er gyldig. □

ØVELSE 23. Anvend (13") til at reducere udsagnet:

$$t<3 \wedge (t\geqq 3 \vee t>2).$$

ØVELSE 24. Anvend (13') til at reducere mængdeudtrykket:

$$B\cup(D\cap(\complement B)),$$

hvor B og D betegner delmængder af grundmængden, E.

ØVELSE 25. Oversæt (13') og (13") til betegnelserne fra logikalgebraen. Illustrér derefter resultaterne ved hjælp af sandhedstabeller.

SÆTNING 14. For vilkårlige elementer, x og y, i B gælder:

(14') $\bar{x}+(x\cdot y) = \bar{x}+y$

(14") $\bar{x}\cdot(x+y) = \bar{x}\cdot y$

BEVIS: Der gælder:

$$\begin{aligned} \bar{x}+(x\cdot y) &= (\bar{x}+x)\cdot(\bar{x}+y) && \text{ifølge (2')} \\ &= (x+\bar{x})\cdot(\bar{x}+y) && \text{ifølge (1')} \\ &= 1\cdot(\bar{x}+y) && \text{ifølge (4')} \\ &= (\bar{x}+y)\cdot 1 && \text{ifølge (1")} \\ &= \bar{x}+y && \text{ifølge (3")} \end{aligned}$$

Hermed er (14') bevist. Ved anvendelse af dualitetsprincippet fremkommer (14"). □

ØVELSE 26. Sætning 14 kan bevises på en meget elegant måde ved at anvende sætning 10 og sætning 13. Prøv dette.

ØVELSE 27. Oversæt (14") til betegnelserne fra logikalgebraen, og opstil dernæst sandhedstabellerne for henholdsvis den venstre og den højre side af den oversatte formel.

SÆTNING 15. (*De associative regler*).
For vilkårlige elementer, x, y og z, i B gælder:

(15') $x+(y+z) = (x+y)+z$

(15") $x\cdot(y\cdot z) = (x\cdot y)\cdot z$

BEVIS: Vi indfører forkortelser for henholdsvis venstre og højre side af (15'):

(i) $v := x+(y+z)$

(ii) $h := (x+y)+z$

Beviset for (15') udføres i tre trin, idet vi beviser:

(iii) $v\cdot x = h\cdot x$

(iv) $v\cdot\bar{x} = h\cdot\bar{x}$

(v) $v = h$

Det ses, at (15') vil være bevist, når vi har bevist (v).

Bevis for (iii):

$$\begin{aligned} v\cdot x &= x\cdot v && \text{ifølge (1")} \\ &= x\cdot(x+(y+z)) && \text{ifølge (i)} \\ &= x && \text{ifølge (12")} \\ h\cdot x &= x\cdot h && \text{ifølge (1")} \\ &= x\cdot((x+y)+z) && \text{ifølge (ii)} \\ &= (x\cdot(x+y))+(x\cdot z) && \text{ifølge (2")} \\ &= x+(x\cdot z) && \text{ifølge (12")} \\ &= x && \text{ifølge (12')} \end{aligned}$$

Disse to udregninger viser, at (iii) er sand.

Bevis for (iv):

$$\begin{aligned} v\cdot\bar{x} &= \bar{x}\cdot v && \text{ifølge (1")} \\ &= \bar{x}\cdot(x+(y+z)) && \text{ifølge (i)} \\ &= \bar{x}\cdot(y+z) && \text{ifølge (14")} \\ h\cdot\bar{x} &= \bar{x}\cdot h && \text{ifølge (1")} \\ &= \bar{x}\cdot((x+y)+z) && \text{ifølge (ii)} \\ &= (\bar{x}\cdot(x+y))+(\bar{x}\cdot z) && \text{ifølge (2")} \\ &= (\bar{x}\cdot y)+(\bar{x}\cdot z) && \text{ifølge (14")} \\ &= \bar{x}\cdot(y+z) && \text{ifølge (2")} \end{aligned}$$

Heraf ses, at (iv) er sand.

Bevis for (v):

$$\begin{aligned} v &= v\cdot 1 && \text{ifølge (3")} \\ &= v\cdot(x+\bar{x}) && \text{ifølge (4')} \\ &= (v\cdot x)+(v\cdot\bar{x}) && \text{ifølge (2")} \\ &= (h\cdot x)+(h\cdot\bar{x}) && \text{ifølge (iii) og (iv)} \\ &= h\cdot(x+\bar{x}) && \text{ifølge (2")} \\ &= h\cdot 1 && \text{ifølge (4')} \\ &= h && \text{ifølge (3")} \end{aligned}$$

Hermed er (v) bevist. Som nævnt afslutter dette beviset for (15'). Ved at anvende dualitetsprincippet på (15') fremkommer (15"). Hermed er sætning 15 bevist. □

De associative regler har stor praktisk betydning, idet vi uden fare for misforståelser kan tillade os at skrive for eksempel x+y+z. Der er to mulige fortolkninger af dette udtryk, nemlig henholdsvis x+(y+z) og (x+y)+z. Ifølge den associative regel for addition er disse fortolkninger identiske. Det ses let, at man kan udvide skrivemåden til at omfatte et vilkårligt antal elementer, for eksempel x+y+z+w. De samme bemærkninger gør sig gældende for den associative regel for multiplikation.

ØVELSE 28. Udtrykket x+y+z+w har mange mulige fortolkninger, for eksempel følgende:

$$((x+y)+z)+w$$
$$(x+y)+(z+w)$$
$$x+((y+z)+w)$$

Gør rede for, at disse tre fortolkninger er identiske.

ØVELSE 29. Undersøg, om man uden fare for misforståelse kan tillade sig at skrive x+y·z *uden* angivelse af parenteser. Prøv for eksempel at fortolke det booleske udtryk 1+1·0 på forskellige måder.

ØVELSE 30. Når man i almindelig talregning skriver 3+7·2, så giver det *ikke* anledning til misforståelser. Sammenlign dette med resultatet i øvelse 29, og find en forklaring.

ØVELSE 31. Oversæt sætning 15 til betegnelserne fra mængdealgebraen, og illustrér resultaterne ved hjælp af Venn-diagrammer.

ØVELSE 32. Oversæt sætning 15 til betegnelserne fra logikalgebraen, og illustrér resultaterne ved hjælp af sandhedstabeller.

SÆTNING 16. (*De Morgans regler*).
For vilkårlige elementer, x og y, i B gælder:

$$(16') \quad \overline{x+y} = \bar{x}\cdot\bar{y}$$

$$(16'') \quad \overline{x\cdot y} = \bar{x}+\bar{y}$$

BEVIS: Først beviser vi (16') ved hjælp af to mellemregninger:

0 + 0
0 + 1
1 + 0
1 + 1

$$\begin{aligned}
(x+y)+(\bar{x}\cdot\bar{y}) &= x+(y+(\bar{x}\cdot\bar{y})) && \text{ifølge } (15')\\
&= x+(y+(\bar{y}\cdot\bar{x})) && \text{ifølge } (1'')\\
&= x+(y+\bar{x}) && \text{ifølge } (13')\\
&= x+(\bar{x}+y) && \text{ifølge } (1')\\
&= (x+\bar{x})+y && \text{ifølge } (15')\\
&= 1+y && \text{ifølge } (4')\\
&= y+1 && \text{ifølge } (1')\\
&= 1 && \text{ifølge } (8')
\end{aligned}$$

mindst en af værdierne = 1

$$\begin{aligned}
(x+y)\cdot(\bar{x}\cdot\bar{y}) &= ((x+y)\cdot\bar{x})\cdot\bar{y} && \text{ifølge } (15'')\\
&= (\bar{x}\cdot(x+y))\cdot\bar{y} && \text{ifølge } (1'')\\
&= (\bar{x}\cdot y)\cdot\bar{y} && \text{ifølge } (14'')\\
&= \bar{x}\cdot(y\cdot\bar{y}) && \text{ifølge } (15'')\\
&= \bar{x}\cdot 0 && \text{ifølge } (4'')\\
&= 0 && \text{ifølge } (8'')
\end{aligned}$$

1 af værdierne = 0

Disse to mellemregninger viser, at $\bar{x}\cdot\bar{y}$ er et komplement til $x+y$. Ifølge sætning 6 er $\overline{x+y}$ det eneste komplement til $x+y$. Altså må $\overline{x+y}$ være lig med $\bar{x}\cdot\bar{y}$. Hermed er (16') bevist. På sædvanlig måde giver dualitetsprincippet, at også (16") er sand. Dette afslutter beviset for sætning 16. □

De regler, der kommer til udtryk i sætning 16, er opkaldt efter en amerikansk logiker, *De Morgan*, som levede fra 1806 til 1871, altså nogenlunde samtidigt med George Boole.

De Morgans regler har vidtrækkende betydning inden for alle specialtilfælde af den generelle booleske algebra, og ikke mindst inden for digitalalgebraen.

ØVELSE 33. Anvend De Morgans regler til en forenkling af udsagnet $\neg(t>3 \vee t \notin Z_+)$. For hvilke reelle tal, t, er udsagnet sandt?

ØVELSE 34. Oversæt De Morgans regler til betegnelserne fra logikalgebraen, og illustrér dem ved hjælp af sandhedstabeller.

ØVELSE 35. Oversæt De Morgans regler til betegnelserne fra mængdealgebraen, og illustrér dem ved hjælp af Venn-diagrammer.

# 4. Digitalalgebra

I første del (kapitel 1-3) så vi, hvorledes den booleske algebra opstod som en generalisation af logikalgebraen og mængdealgebraen. I denne anden del (kapitel 4-6) går vi den modsatte vej. Med udgangspunkt i den generelle booleske algebra definerer vi et specialtilfælde, nemlig digitalalgebraen. Ved behandlingen af denne nye algebraiske struktur kan vi benytte hele formelapparatet fra kapitel 3. Her viser sig netop en af fordelene ved at udvikle teorien under så generelle betingelser som muligt.

DEFINITION 1. (*Digitalalgebra*).
En digitalalgebra er en boolesk algebra med netop to elementer.

Denne definition virker meget elegant i al sin enkelthed, men den afføder straks nogle spørgsmål om eksistens og entydighed: "Findes der en digitalalgebra?", og "Hvor mange digitalalgebraer findes der?". Det er berettiget at stille sig tvivlende over for eksistensen af en digitalalgebra. For eksempel kan man ret let bevise, at der *ikke* findes nogen boolesk algebra med præcis *tre* elementer. (Dette skal vi behandle i en senere øvelse). De følgende sætninger giver et klart svar på spørgsmålet om eksistensen af en digitalalgebra, og et delvis svar på spørgsmålet om entydigheden af en digitalalgebra.

SÆTNING 1. (*Eksistens af en digitalalgebra*).
Der findes en digitalalgebra.

BEVIS: Lad E være en mængde, og lad M være mængden af alle delmængder af E. Så er $(M,\cup,\cap,\complement)$ som bekendt en boolesk algebra. Antag nu, at E indeholder præcis ét element. Så har E præcis to delmængder, nemlig $\emptyset$ og E. Men det betyder jo, at M har præcis to elementer. Altså er $(M,\cup,\cap,\complement)$ en digitalalgebra. Hermed er sætning 1 bevist. □

Efter således at have bekræftet eksistensen af en digitalalgebra, begynder vi at undersøge entydigheden:

SÆTNING 2. De neutrale elementer over for henholdsvis addition og multiplikation i en digitalalgebra er forskellige, og disse er de eneste elementer i digitalalgebraen.

BEVIS: Vi kalder digitalalgebraen for $(D,+,\cdot,\bar{\ })$. De neutrale elementer over for henholdsvis addition og multiplikation kalder vi 0 og 1. Antag nu, at 0 og 1 er ens. For et vilkårligt element, x, i D gælder så:

$$\begin{aligned} x &= x\cdot 1 && \text{ifølge (3") i kapitel 3} \\ &= x\cdot 0 && \text{ifølge antagelsen} \\ &= 0 && \text{ifølge (8") i kapitel 3} \end{aligned}$$

Dette viser, at ethvert element i D er lig med 0. Derfor har D kun ét element. Men det er i modstrid med definitionen på en digitalalgebra. Altså må antagelsen, om at 0 og 1 er ens, være forkert. Men så må 0 og 1 være forskellige. Desuden må 0 og 1 være de eneste elementer i D, da D ifølge definitionen ikke har mere end to elementer. Hermed er sætning 2 bevist. □

ØVELSE 1. Findes der en boolesk algebra, hvori de neutrale elementer over for henholdsvis addition og multiplikation er ens ?

SÆTNING 3. I en digitalalgebra, $(D,+,\cdot,\bar{\ })$, må additionstabellen og multiplikationstabellen samt komplementtabellen nødvendigvis være

| x | y | x+y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| x | y | x·y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| x | $\bar{x}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

hvor 0 og 1 betegner de neutrale elementer over for henholdsvis addition og multiplikation.

BEVIS: Ifølge sætning 2 består D præcis af elementerne 0 og 1. Tabellerne skal derfor kun indeholde disse elementer. Ifølge (3') i kapitel 3 er $x+0 = x$. Derfor kan første og tredie række i additionstabellen kun udfyldes på den angivne måde. Ifølge (8') i kapitel 3 er $x+1 = 1$. Altså må også anden og fjerde række i additionstabellen nødvendigvis udfyldes som angivet. Hermed er det bevist, at der ikke er andre muligheder for additionstabellen end den angivne. Det overlades til læseren at bevise, at det samme er tilfældet for multiplikationstabellen. Ved at benytte additionstabellen og multiplikationstabellen fås herefter, at

$$1+0 = 1 \wedge 1\cdot 0 = 0.$$

Men det betyder jo, at 0 er et komplement til 1. Ifølge sætning 6 i kapitel 3 er $\bar{1}$ det eneste komplement til 1. Altså er $\bar{1} = 0$. Derfor må anden række i komplementtabellen nødvendigvis være udfyldt som angivet. Ved at komplementere begge sider af ligningen $\bar{1} = 0$ får man $\bar{\bar{1}} = \bar{0}$. Ifølge sætning 10 i kapitel 3 kan vi heri erstatte $\bar{\bar{1}}$ med 1. Altså gælder $1 = \bar{0}$. Derfor må også første række i komplementtabellen nødvendigvis være udfyldt som angivet. Dette afslutter beviset for sætning 3. □

Det er værd at bemærke den store lighed, der hersker mellem tabellerne i en digitalalgebra og de tilsvarende tabeller i logikalgebraen. Denne lighed er årsagen til, at man ofte bruger betegnelsen *digitallogik* som et synonym for digitalalgebra.

ØVELSE 2. Ved at anvende den teknik, der blev benyttet i beviserne for sætning 2 og 3, kan man bevise, at der *ikke* findes nogen boolesk algebra med præcis tre elementer. Gennemfør dette bevis.

Vi har nu alt i alt bevist, at der findes en digitalalgebra, og at den regnemæssige struktur i en digitalalgebra er entydigt bestemt. Selvfølgelig kan vi vælge andre symboler end netop

$$0,\ 1,\ +,\ \cdot,\ \bar{\ }$$

som repræsentanter for elementerne og operatorerne. For eksempel kunne vi (jævnfør beviset for sætning 1) vælge at benytte symbolerne

$$\emptyset,\ E,\ \cup,\ \cap,\ \complement$$

hvor E er en mængde med præcis ét element. Imidlertid er betegnelserne uinteressante, idet vi kun skal beskæftige os med den regnemæssige struktur. Derfor vælger vi fortsat at benytte symbolerne

$$0,\ 1,\ +,\ \cdot,\ \bar{\ }$$

der iøvrigt netop er de symboler, der traditionelt anvendes inden for digitalteknikken. På denne baggrund kan vi påstå, at *i al væsentlighed findes der netop én digitalalgebra* , - bortset fra valget af symboler.

EKSEMPEL 1. På de følgende sider skal vi se på et eksempel på en anvendelse af digitalalgebraen. Anvendelsesområdet er meget stort. Dette hænger sammen med, at der i dagligdagen er så mange systemer, der kan befinde sig i to stabile tilstande:

- En jernbanebom kan være oppe eller nede.
- En advarselslampe kan være tændt eller slukket.
- En kørende bilist kan standse eller fortsætte sin kørsel.

Ofte indvirker sådanne systemer på hinanden:

- En kørende bilist, der nærmer sig en jernbaneoverskæring, fortsætter kun sin kørsel, såfremt jernbanebommen er oppe, og advarselslampen er slukket.

Det ses, at de to variable

- Jernbanebommen,
- Advarselslampen,

*styrer* den tredie variable

- Bilisten.

I forbindelse med undersøgelsen af sådanne *styringssystemer* er digitalalgebraen et kærkomment og praktisk redskab. For at indse dette skal vi først gøre de variable tilgængelige for matematisk behandling. Dette hedder at *erklære* de variable. *Erklæringen* for jernbanebommen kan eksempelvis lyde:

> Jernbanebommen er en variabel, som kaldes x. Dens tilstande er:
> x=0 : Jernbanebommen er oppe.
> x=1 : Jernbanebommen er nede.

Erklæringen for advarselslampen vælges til at lyde:

Advarselslampen er en variabel, som kaldes y. Dens tilstande er:

$y=0$ : Advarselslampen er slukket.

$y=1$ : Advarselslampen er tændt.

Og endelig erklærer vi bilisten således:

Bilisten er en variabel, som kaldes z. Hendes tilstande er:

$z=0$ : Bilisten standser sin kørsel.

$z=1$ : Bilisten fortsætter sin kørsel.

Teksten, der fortalte om bilistens reaktion på jernbanebommens og advarselslampens tilstande, kan nu oversættes ordret til:

$$z=1 \Leftrightarrow x=0 \wedge y=0.$$

Ved at negere begge sider af biimplikationen fås:

$$\neg(z=1) \Leftrightarrow \neg(x=0 \wedge y=0).$$

Ved anvendelse af en af De Morgans regler får vi:

$$\neg(z=1) \Leftrightarrow \neg(x=0) \vee \neg(y=0),$$

som reduceres til:

$$z=0 \Leftrightarrow x=1 \vee y=1.$$

Vi har nu fuld kontrol over z's afhængighed af x og y:

$$z=0 \Leftrightarrow x=1 \vee y=1,$$
$$z=1 \Leftrightarrow x=0 \wedge y=0.$$

Resultatet kan systematiseres ved hjælp af en *tabel:*

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

eller ved hjælp af et *udtryk:*

$$z = \overline{x+y}.$$

ØVELSE 3. Gør rede for, at tabellen for z og udtrykket for z er i overensstemmelse med hinanden.

ØVELSE 4. Oversæt hver af tabellens rækker til naturligt sprog.

Vi har nu foretaget en undersøgelse af bilistens reaktion på jernbanebommen og advarselslampen. En simpel styringsmekanisme som denne berettiger egentlig ikke til en så grundig undersøgelse. Eksemplet er kun medtaget for at vise nogle perspektiver i anvendelsen af digitalalgebraen. Styringsmekanismerne i for eksempel en lommeregner eller i en microcomputer er helt anderledes komplicerede. I forbindelse med konstruktion og analyse af den slags systemer er digitalalgebraen ikke blot velegnet, men også aldeles nødvendig.

ØVELSE 5. I nedenstående policebetingelse skal du foretage en erklæring af hver af de variable, og derefter oversætte policebetingelsen til en tabel og til et udtryk.

> Forsikringsselskabet betaler kun for reparation af skader i forbindelse med et sprængt vandrør, såfremt skadesanmelderen har tegnet en rørskadeforsikring, og såfremt skaden ikke er en følge af krigshandlinger.

ØVELSE 6. I nedenstående tekst gemmer sig fire digitale variable. Den ene af disse er styret af de tre andre. Prøv at finde og erklære de variable. Opstil derefter en tabel og et udtryk for teksten.

> De år, hvis årstal er delelige med fire, er skudår, dog kun de hundredeår, hvor årstallet er deleligt med firehundrede. År 1900 var således ikke et skudår, hvorimod år 2000 vil være det.

Til slut i dette kapitel indfører vi en række praktiske skrivemåder, som sikrer os en mere fri notation, navnlig hvad angår det enorme forbrug af parenteser.

VEDTÆGT 1. Som tidligere omtalt, kan vi i kraft af de associative regler undvære parenteser i rent additive udtryk og i rent multiplikative udtryk. Vi vedtager at benytte os af denne frihed fremover. For eksempel kan vi tillade os at skrive $x \cdot y \cdot z \cdot w$, idet udtrykket giver samme mening, uanset hvorledes man sætter parenteser i det.

VEDTÆGT 2. I sædvanlig talregning er der tradition for, at parenteser om produkter er underforståede. For eksempel betyder $7+2 \cdot 3+1$ faktisk, at 2 og 3 skal multipliceres *før* additionerne udføres. Det vil sige, at der er underforstået parentes om $2 \cdot 3$, altså $7+(2 \cdot 3)+1$. Vi vedtager at overføre denne tradition til digitalalgebraen. Herefter kan vi eksempelvis skrive:

$$\begin{aligned} (u+v)+((w \cdot x)+y) &= u+v+(w \cdot x)+y && \text{ifølge vedtægt 1} \\ &= u+v+w \cdot x+y && \text{ifølge vedtægt 2.} \end{aligned}$$

VEDTÆGT 3. I den sædvanlige talregning er der også en tradition for, at multiplikationstegn mellem talvariable er underforståede. Når vi skriver $7x+3xy$, mener vi i virkeligheden $7 \cdot x+3 \cdot x \cdot y$, der igen skal forstås som $(7 \cdot x)+(3 \cdot x \cdot y)$. Også denne tradition vedtager vi at overføre til digitalalgebraen. For eksempel kan vi tillade os at skrive:

$$\begin{aligned} & u+((v\cdot w)+(x\cdot(y+z))) \\ & = u+(v\cdot w)+(x\cdot(y+z)) && \text{ifølge vedtægt 1} \\ & = u+v\cdot w+x\cdot(y+z) && \text{ifølge vedtægt 2} \\ & = u+vw+x(y+z) && \text{ifølge vedtægt 3.} \end{aligned}$$

Bemærk, at parentesen om y+z *ikke* kan undværes.

ØVELSE 7. Skriv de distributive regler så enkelt som muligt ved anvendelse af de tre vedtægter.

ØVELSE 8. Hvilke af udtrykkene $xy+z$, $x(y+z)$, $(xy)+z$ er ens?

ØVELSE 9. Udregn værdien af $\overline{(1+0\cdot 1+1)\cdot 1+\overline{0\cdot 1}}$.

ØVELSE 10. Opskriv De Morgans regler så enkelt som muligt.

# 5. Digitale funktioner

I det foregående kapitel så vi et eksempel på, hvorledes bilistens reaktion var styret af jernbanebommens og advarselslampens tilstande. Dette vil vi fremover udtrykke ved at sige, at bilistens reaktion er en *funktion* af jernbanebommens og advarselslampens tilstande. Her er der tale om en funktion af to variable. Nedenstående definition er en anelse mere generel:

> DEFINITION 1. (*Digital funktion*).
> Lad n være et positivt helt tal. En variabel, w, i digitalalgebraen siges at være en *funktion* af de n variable, $x_1, x_2, x_3, \dots, x_n$, såfremt værdien af w er entydigt bestemt af værdierne af $x_1, x_2, x_3, \dots, x_n$. Den variable, w, kaldes den *afhængige variable*, fordi dens værdi afhænger af værdierne af $x_1, x_2, x_3, \dots, x_n$. I modsætning hertil kaldes $x_1, x_2, x_3, \dots, x_n$ de *uafhængige variable*.

Når man skal arbejde med en digital funktion, w, af n variable, $x_1, x_2, x_3, \dots, x_n$, er man nødt til at gøre præcist rede for, hvorledes værdien af w afhænger af værdierne af $x_1, x_2, x_3, \dots, x_n$. I eksemplet med bilisten har vi allerede set tre forskellige formuleringer af en sådan afhængighed:

-En *sproglig formulering*:

En kørende bilist, der nærmer sig en jernbaneoverskæring, fortsætter kun sin kørsel, såfremt jernbanebommen er oppe, og advarselslampen er slukket.

-En *funktionstabel*:

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

-Et *funktionsudtryk*:

$z = \overline{x+y}$.

Den sproglige formulering er kun velegnet i forbindelse med ganske simple funktioner, idet man meget let løber ind i problemer med dagligsprogets mangetydighed. I praksis foretrækker man derfor at foretage en erklæring af alle de variable, således at man hurtigst muligt kommer bort fra den rent sproglige formulering. Derved opnår man også, at funktionen bliver tilgængelig for behandling med matematiske redskaber, det vil sige alle formlerne fra kapitel 3. I det følgende vil vi fortrinsvis se på funktioner, som er givet ved en tabel eller et udtryk.

ØVELSE 1. Opstil tabellen for den funktion, der har funktionsudtrykket $w = xy+\bar{z}$.

ØVELSE 2. Hvor mange rækker er der i tabellen for en funktion af n variable ?

ØVELSE 3. Find et funktionsudtryk for den funktion, der har tabellen:

| x | y | z |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

ØVELSE 4. Reducér udtrykket for funktionen $w = \bar{x}yz+\bar{x}\bar{y}z+xy$.

Vi skal nu se på nogle metoder, der kan anvendes ved overgangen fra funktionsudtrykkene til tabellerne.

EKSEMPEL 1. Vi vil finde tabellen for den funktion, der har funktionsudtrykket

$$w = \bar{x}(y+z).$$

Tabellen har formen

| x | y | z | w |
|---|---|---|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | |
| 0 | 1 | 0 | |
| 0 | 1 | 1 | ? |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | |
| 1 | 1 | 0 | |
| 1 | 1 | 1 | |

Den mest ligefremme metode består i at udfylde tabellen række for række. For hver af de otte rækker sætter vi de pågældende værdier af x, y og z ind i udtrykket for w, og regner værdien af w ud. For første række fås:

$$\begin{aligned} & x = 0 \wedge y = 0 \wedge z = 0 \\ \Rightarrow\ & w = \bar{0}(0+0) \\ \Rightarrow\ & w = 1 \cdot 0 \\ \Rightarrow\ & w = 0 \end{aligned}$$

Derfor skal tabellens første række udfyldes med et 0. Således kunne man fortsætte med hver af de resterende syv rækker.

En anden, og måske lidt mere elegant metode, ligner den, der blev benyttet ved opstilling af sandhedstabellerne i kapitel 1. Ud fra de tre indgangssøjler i tabellen konstrueres søjler for henholdsvis $\bar{x}$ og y+z og $\bar{x}(y+z)$:

| x | y | z | $\bar{x}$ | y+z | w |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |

En tredie metode benytter det faktum, at w kun kan antage værdierne 0 og 1. Vi kan regne ud, i hvilke tilfælde w antager værdien 0:

$$
\begin{aligned}
& w = 0 \\
\Leftrightarrow\ & \bar{x}(y+z) = 0 \\
\Leftrightarrow\ & \bar{x} = 0 \vee y+z = 0 \\
\Leftrightarrow\ & x = 1 \vee (y = 0 \wedge z = 0)
\end{aligned}
$$

Af denne analyse følger det, de rækker i tabellen, hvori w antager værdien 0, præcis er de rækker hvori enten x er 1 eller såvel y som z er 0. Alle de øvrige rækker skal derfor udfyldes med værdien 1 for w. Denne fremgangsmåde, der i visse sammenhænge er meget hurtig, er i virkeligheden en oversættelse af problemet fra digitalalgebraen til logikalgebraen.

De tre beskrevne metoder har hver deres fordele. Hvilken metode man vælger, afhænger vist mest af vanen, men fælles for de tre metoder er, at det altid kan betale sig at forsøge at reducere funktionsudtrykket *inden* tabellen skrives.

ØVELSE 5. Reducér funktionsudtrykket $z = (u+x)(u+\bar{x})(u+v)+x$, og opstil derefter tabellen for funktionen.

ØVELSE 6. Reducér funktionsudtrykket $w = \overline{(x+y)}x$, og opstil derefter tabellen for funktionen.

Til slut i dette kapitel skal vi se på en generel metode, der er anvendelig ved overgangen den modsatte vej, altså ved overgangen *fra* en funktionstabel *til* et funktionsudtryk. Vi starter med et eksempel:

EKSEMPEL 2. Vi vil finde et funktionsudtryk for den funktion, der har tabellen:

| x | y | z | w |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Det ses, at w har værdien 1 i tredie og syvende række:

$$w=1 \Leftrightarrow (x=0 \wedge y=1 \wedge z=0) \vee (x=1 \wedge y=1 \wedge z=0).$$

Vi omskriver nu deludsagnene, således at alle 0'erne forsvinder fra implikationens højre side:

$$w=1 \Leftrightarrow (\bar{x}=1 \wedge y=1 \wedge \bar{z}=1) \vee (x=1 \wedge y=1 \wedge \bar{z}=1).$$

Ved anvendelse af egenskaberne ved multiplikationen fås så:

$$w=1 \Leftrightarrow \bar{x}y\bar{z}=1 \vee xy\bar{z}=1,$$

som igen kan omskrives ved anvendelse af egenskaberne ved addition:

$$w=1 \Leftrightarrow \bar{x}y\bar{z}+xy\bar{z}=1.$$

Heraf ses, at w har værdien 1, netop når $\bar{x}y\bar{z}+xy\bar{z}$ har værdien 1. Men så må det også gælde, at w har værdien 0, netop når $\bar{x}y\bar{z}+xy\bar{z}$ har værdien 0, thi der er jo ikke andre elementer end 0 og 1 i digitalalgebraen. Under alle omstændigheder har w og $\bar{x}y\bar{z}+xy\bar{z}$ samme værdi. Men så er de jo ens:

$$w = \bar{x}y\bar{z}+xy\bar{z}.$$

Hermed har vi fundet et funktionsudtryk for w som funktion af x, y og z. Tilbage har vi kun at reducere dette funktionsudtryk:

$$\begin{aligned} w &= \bar{x}y\bar{z}+xy\bar{z} \\ &= (\bar{x}+x)y\bar{z} \\ &= 1\cdot y\bar{z} \\ &= y\bar{z}. \end{aligned}$$

Altså er det endelige udtryk for w givet ved:

$$w = y\bar{z}.$$

Det fremgår af dette funktionsudtryk, af w faktisk slet ikke afhænger af x. Dette er en egenskab ved w, som man næppe ville have opdaget alene ved en betragtning af tabellen.

Den beskrevne metode kan virke lidt "tung". I praksis forløber overgangen fra tabel til udtryk dog langt mere elegant. Det er faktisk muligt at opskrive et funktionsudtryk direkte ud fra tabellen. Hvis vi nemlig ser efter, hvad der egentlig foregik i den beskrevne metode, kan fremgangsmåden systematiseres således:

Først markeres med * de rækker i tabellen, hvori w har værdien 1:

| | x | y | z | w |
|---|---|---|---|---|
| | 0 | 0 | 0 | 0 |
| | 0 | 0 | 1 | 0 |
| * | 0 | 1 | 0 | 1 |
| | 0 | 1 | 1 | 0 |
| | 1 | 0 | 0 | 0 |
| | 1 | 0 | 1 | 0 |
| * | 1 | 1 | 0 | 1 |
| | 1 | 1 | 1 | 0 |

For hver af de markerede rækker skrives et rent multiplikativt udtryk bestående af de uafhængige variable (hvis de har værdien 1) eller deres komplement (hvis de har værdien 0):

| | x | y | z | w | hjælpesøjle |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 1 | 0 | |
| * | 0 | 1 | 0 | 1 | $\bar{x}y\bar{z}$ |
| | 0 | 1 | 1 | 0 | |
| | 1 | 0 | 0 | 0 | |
| | 1 | 0 | 1 | 0 | |
| * | 1 | 1 | 0 | 1 | $xy\bar{z}$ |
| | 1 | 1 | 1 | 0 | |

Til slut adderes de multiplikative udtryk:

| | x | y | z | w | hjælpesøjle |
|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 1 | 0 | |
| * | 0 | 1 | 0 | 1 | $\bar{x}y\bar{z}$ |
| | 0 | 1 | 1 | 0 | |
| | 1 | 0 | 0 | 0 | |
| | 1 | 0 | 1 | 0 | |
| * | 1 | 1 | 0 | 1 | $xy\bar{z}$ |
| | 1 | 1 | 1 | 0 | |
| | $w = \bar{x}y\bar{z}+xy\bar{z}$ | | | | |

Det ses, at resultatet svarer til det tidligere fundne udtryk for w som funktion af x, y og z. Læseren opfordres til at sammenligne denne sidste metode punkt for punkt med den tidligere metode.

ØVELSE 7. Find et udtryk for den funktion, der har tabellen:

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

ØVELSE 8. Find et udtryk for den funktion, der er givet ved tabellen:

| x | y | z | w |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

og reducér det fundne udtryk.

Metoden for overgangen fra tabellen til funktionsudtrykket vil altid give et additivt udtryk med lige så mange led, som der er 1'ere i funktionstabellens udgangssøjle. Hvis antallet af 1'ere er meget stort, kan det fremkomne udtryk derfor virke uoverskueligt. I sådanne tilfælde kan det betale sig *først* at finde et udtryk for den komplementerede funktion, og *derefter* for selve funktionen.

EKSEMPEL 3. Vi ønsker at finde et udtryk for den funktion, der har tabellen:

| x | y | z |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Da der er flere 1'ere end 0'ere i tabellen for z, finder vi tabellen for $\bar{z}$:

| x | y | z | $\bar{z}$ |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

Derefter finder vi et udtryk for $\bar{z}$:

| | x | y | z | $\bar{z}$ | hjælpesøjle |
|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | |
| | 0 | 1 | 1 | 0 | |
| * | 1 | 0 | 0 | 1 | $x\bar{y}$ |
| | 1 | 1 | 1 | 0 | |
| | | | $\bar{z} = x\bar{y}$ | | |

Altså er

$$\bar{z} = x\bar{y}.$$

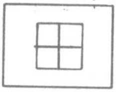
Ved komplementering findes udtrykket for z:

$$\bar{\bar{z}} = \overline{x\bar{y}} \Leftrightarrow z = \bar{x}+\bar{\bar{y}} \Leftrightarrow z = \bar{x}+y.$$

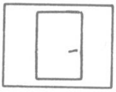
ØVELSE 9. Find et funktionsudtryk for den funktion, der er fastlagt ved følgende tabel:

| x | y | z | w |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

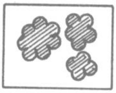
ØVELSE 10. En virksomhed med én dør, ét vindue og ét lagerlokale skal have installeret et alarmanlæg, der træder i funktion, hvis døren eller vinduet åbnes uden for normal arbejdstid, eller hvis der opstår røg i lagerlokalet. Erklæringerne for de variable fremgår af figuren:



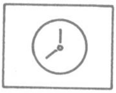
v=0: Vinduet er åbent.
v=1: Vinduet er lukket.



d=0: Døren er lukket.
d=1: Døren er åben.



r=0: Ingen røg i lageret.
r=1: Røg i lageret.



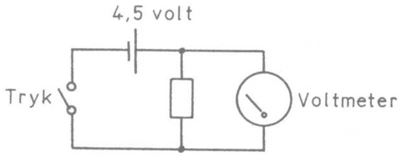
u=0: Inden for arbejdstid.
u=1: Uden for arbejdstid.

a=0: Ingen alarm.
a=1: Alarm.

Opstil en tabel for a som funktion af v, d, r og u. Find dernæst på grundlag af tabellen et udtryk for funktionen, og reducér dette udtryk. Overvej, om du kunne have fundet udtrykket direkte fra den sproglige formulering.

# 6. Digitale diagrammer

Nedenstående tegning viser et simpelt elektronisk kredsløb:

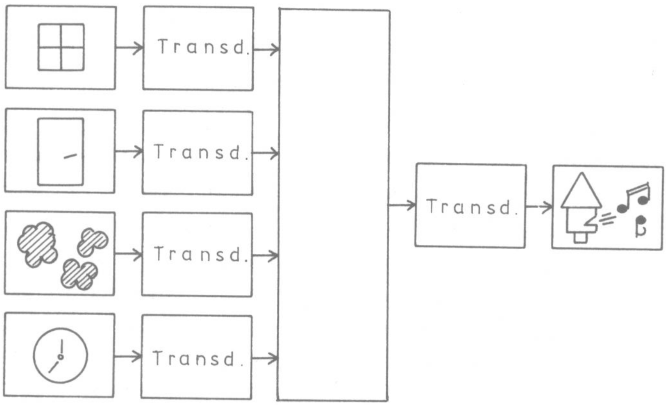



Hvis man trykker på kontakten, viser voltmeteret 4,5 volt. Ellers viser det 0 volt. Kontakten tænkes nu anbragt i en vindueskarm, således at vinduet lige netop trykker på kontakten, når det er lukket. Dermed får vi vinduets tilstand oversat til en elektrisk spænding. Informationen, om hvorvidt vinduet er åbent eller lukket, kan genfindes i den elektriske spænding: 0 volt betyder, at vinduet et åbent, mens 4,5 volt betyder, at vinduet er lukket. Informationen kommer blot til udtryk på forskellige måder. Et apparat, der oversætter en information fra én udtryksform til en anden, kaldes generelt en *transducer*. Kredsløbet ovenfor er altså en transducer.

I øvelse 10 i kapitel 5 så vi på den matematiske planlægning af et alarmanlæg. Når dette skal konstrueres i praksis, kan man forsyne vinduet, døren, lagerlokalet og uret med transducere, som oversætter tingenes tilstande til elektriske spændinger. Desuden kan man forsyne dampfløjten med en transducer, hvormed en elektrisk spænding kan åbne eller lukke for damptilførslen til fløjten. Transduceren kunne for eksempel være en elektromagnetisk ventil.

Alle disse transducere, som fysikerne og ingeniørerne forsyner verden med, har deres sidestykke i den matematiske verden, nemlig i form af de erklæringer, der foretages for de variable. Man kan på en måde sige, at erklæringerne er den matematiske model af den fysiske verdens transducere.

Alarmanlægget kan nu tænkes udformet som et digitalelektronisk netværk, der bearbejder de spændingsinformationer, der kommer fra vinduet, døren, lagerlokalet og uret. Resultatet af bearbejdningen fremkommer derefter i form af en elektrisk spænding, der sættes til at styre dampfløjtens transducer. Dette system er skitseret nedenfor:

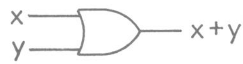



Den digitale funktion, vi har beregnet for alarmanlægget, kan betragtes som den matematiske model af det digitalelektroniske netværk, der tænkes at befinde sig inde i den store kasse på tegningen. Funktionens uafhængige variable, v, d, r og u, svarer til signalerne fra de fire indgangstransducere, mens den uafhængige variable, a, svarer til det elektriske signal, der tilføres dampfløjtens transducer. I resten af dette kapitel skal vi koncentrere os om at forbedre den matematiske model, således at de matematiske resultater bliver mere umiddelbart anvendelige i forbindelse med den praktiske konstruktion af det digitalelektroniske netværk.

Et digitalelektronisk netværk er opbygget af små operationsenheder, der ganske nøje svarer til digitalalgebraens operatorer. Derfor er det en nærliggende tanke at lade diagramsymbolerne for operationsenhederne overtage rollen som operatorsymboler i digitalalgebraen. Når en digital funktion omskrives til denne nye symbolik, får den en form, der direkte angiver, hvorledes det tilsvarende digitalelektroniske netværk skal loddes sammen.

Diagramsymbolet for additionsoperatoren hedder en *OR-gate*, og den fungerer således, at de to variable, for eksempel x og y, der skal adderes, optræder på de to indgange, mens resultatet, x+y, fremkommer på udgangen:

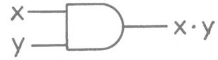
OR-GATE




Da additionen er en kommutativ operator, er der ingen grund til at skelne mellem de to indgange.

Diagramsymbolet for multiplikationsoperatoren hedder en *AND-gate*. Den er forsynet med to indgange ligesom OR-gaten, da de begge repræsenterer dyadiske operatorer. Virkningen af AND-gaten fremgår af diagrammet:

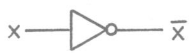
AND-GATE




Heller ikke ved AND-gaten er der nogen grund til at skelne mellem de to indgange.

Diagramsymbolet for komplementoperatoren hedder en *NOT-gate* eller en *inverter*. Den har kun én indgang, da komplementoperatoren jo er en monadisk operator. Dens virkemåde er fastlagt således:

NOT-GATE
INVERTER

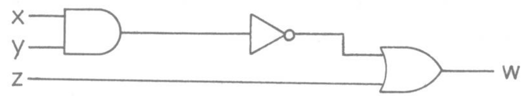


Foruden den i praksis ubrugelige sproglige formulering råder vi nu over tre formuleringsmåder, hvad angår digitale funktioner. Disse tre er:

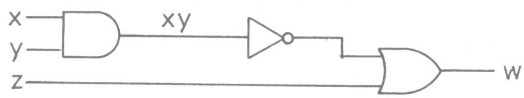
- Funktionstabeller.
- Funktionsudtryk.
- Funktionsdiagrammer.

I kapitel 5 beskæftigede vi os med konverteringer mellem tabeller og udtryk. Nu skal vi se på konverteringer mellem udtryk og diagrammer.

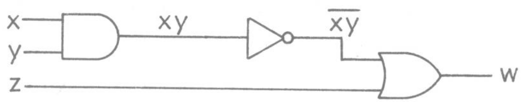
EKSEMPEL 1. Vi skal finde et funktionsudtryk for den digitale funktion, der har diagrammet:



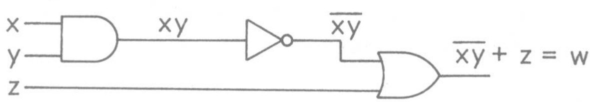

Først lader vi x og y "glide" igennem AND-gaten:




Derefter glider xy igennem inverteren:




Til slut glider $\overline{xy}$ og z igennem OR-gaten:

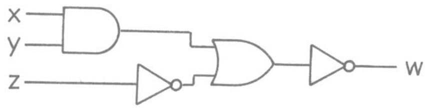



Hermed har vi fundet det søgte funktionsudtryk:

$$w = \overline{xy}+z.$$

Metoden kaldes *dekoration af diagrammet*. Det er indlysende, at et dekoreret diagram er mere forbrugervenligt end et, der ikke er dekoreret. For at opnå en overskuelig dekoration, er det nødvendigt at tegne et "luftigt" diagram, med tilstrækkelig plads til dekorationen.

ØVELSE 1. Find et funktionsudtryk for den funktion, der har diagrammet:



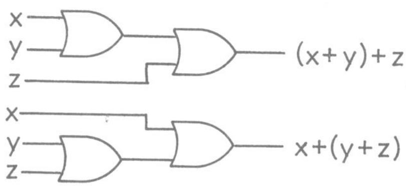

Reducér derefter det fundne udtryk, og vis, hvorledes funktionen kan repræsenteres af et diagram med kun tre gates.

ØVELSE 2. Gør rede for, at nedenstående to diagrammer repræsenterer den samme funktion:

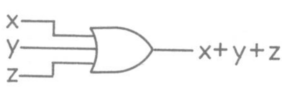
DE MORGAN

ØVELSE 3. Tegn et diagram for alarmanlægget i øvelse 10 i kapitel 5.

Den associative regel for addition kan udtrykkes ved at sige, at de to diagrammer:

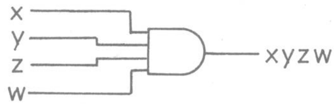



udfører samme funktion. Af denne grund benytter man ofte symbolet:

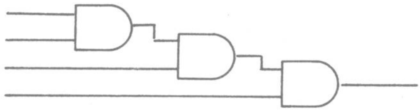


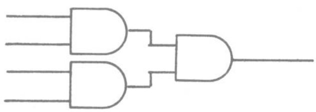

som erstatning for et hvilket som helst af de to diagrammer. Diagramsymbolet hedder en *3-input-OR-gate*.

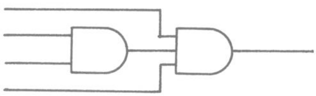
På tilsvarende måde kan man definere *n-input-OR-gates* og *n-input-AND-gates* for et vilkårligt antal inputs, n. For eksempel ser diagramsymbolet for en 4-input-AND-gate således ud:



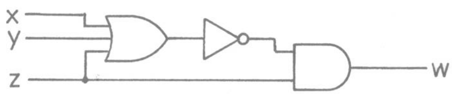

ØVELSE 4. Gør rede for, at hvert af nedenstående tre diagrammer udfører funktionen svarende til en 4-input-AND-gate som beskrevet i diagrammet ovenfor:







ØVELSE 5. Find et udtryk for den funktion, der har følgende diagram, og reducér dernæst det fundne udtryk:




ØVELSE 6. Reducér udtrykket for funktionen,

$$w = (x+y+z)(x+\bar{y}+z),$$

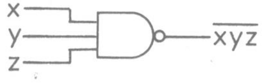
og tegn derefter et diagram for funktionen.

ØVELSE 7. Tegn et diagram for den funktion, der har følgende tabel:

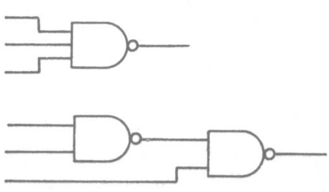
| x | y | z | w |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Når udgangen på en OR-gate forbindes til indgangen på en NOT-gate, fremkommer der en såkaldt *NOR-gate*. Hvis den OR-gate, der anvendes, er en 2-input-OR-gate, kaldes NOR-gaten en *2-input-NOR-gate*. Den er så hyppigt anvendt, at den har sit eget diagramsymbol, der fremkommer ved en sammensmeltning af diagramsymbolerne for henholdsvis OR-gaten og NOT-gaten:


Hvis udgangen på en AND-gate forbindes til indgangen på en NOT-gate, fremkommer en såkaldt *Nand-gate*. Diagramsymbolet for en *3-input-NAND-gate*, eksempelvis, ser således ud:

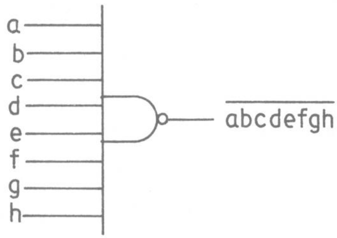



ØVELSE 8. Vis, at følgende to diagrammer *ikke* udfører samme funktion:



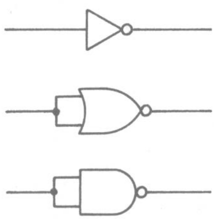
ØVELSE 9. Opstil tabellerne for henholdsvis en 2-input-NOR-gate og en 2-input-NAND-gate.

Ofte benytter man gates med mange indgange. Så kan der opstå nogle tegnetekniske problemer med at få plads til dem alle. I sådanne tilfælde gør man brug af en lidt speciel tegnemåde. For eksempel kan en 8-input-NAND-gate tegnes således:

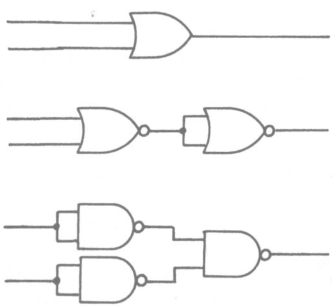



ØVELSE 10. Gør rede for, at følgende tre diagrammer udfører samme funktion:

5.9', 9"



ØVELSE 11. Gør rede for, at følgende tre diagrammer udfører samme funktion:

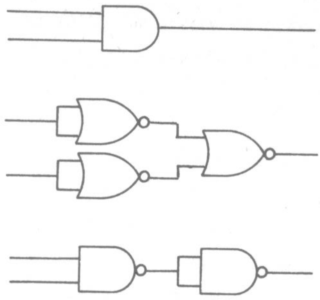


x+y

x+y

$\overline{\bar{x}\cdot\bar{y}} = x+y$

ØVELSE 12. Gør rede for, at følgende tre diagrammer udfører samme funktion:



SÆTNING 1. En vilkårlig digital funktion har et diagram bestående udelukkende af 2-input-NOR-gates og et diagram bestående udelukkende af 2-input-NAND-gates.

BEVIS: En digital funktion er entydigt bestemt af sin tabel. Som vi så i kapitel 5, er det altid muligt at finde et udtryk for funktionen ud fra tabellen. Hermed har vi vist, at enhver digital funktion har et funktionsudtryk. Dette udtryk er foruden de variable opbygget af operatorerne addition, multiplikation og komplementering. Svarende til disse operatorer har vi diagrammerne for henholdsvis 2-input-OR-gate, 2-input-AND-gate og NOT-gate. Når operatorerne i funktionsudtrykket erstattes af disse gates, fås et diagram for funktionen. Dette diagram består udelukkende af 2-input-OR-gates, 2-input-AND-gates samt NOT-gates. Ifølge resultaterne fra øvelserne 10, 11 og 12 kan alle disse gates erstattes af 2-input-NOR-gates. Altså har funktionen et diagram bestående udelukkende af 2-input-NOR-gates. På helt samme måde fås, at funktionen har et diagram bestående udelukkende af 2-input-NAND-gates. Hermed er sætningen bevist. □

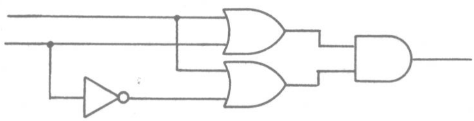
Ifølge sætningen kan man altså realisere en hvilket som helst digital funktion med et tilpas stort antal *ens* gates. Denne egenskab ved digitalalgebraen er af fundamental betydning, idet den danner grundlag for masseproduktion af *ens* digitalelektroniske operationsenheder. Denne masseproduktion er baggrunden for den meget lave markedspris. I en årrække har den såkaldte 7400-serie af digitalelektroniske operationsenheder været dominerende. Denne serie er netop bygget op med en 2-input-NAND-gate som en slags stamfader.

Ud over denne rent praktiske konsekvens af sætningen må man som matematiker også have lov til at nyde dens æstetiske kvalitet. Sætningen kan betragtes som en påvisning af, at der er et "grundstof" i digitalalgebraen. Ethvert "molekyle" er opbygget af dette grundstof. En sådan stræben efter enkelhed er slet ikke ukendt i videnskabshistorien. Den har blandt andet resulteret i de gamle græske naturfilosoffers teorier, om at alt er sammensat af et ganske lille antal forskellige atomer.

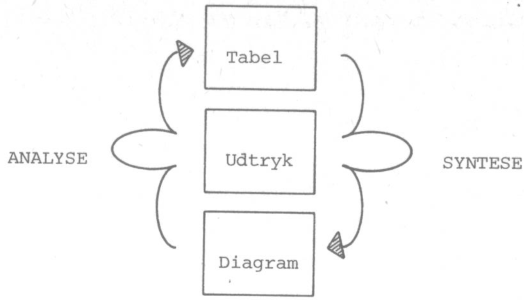
ØVELSE 13. Undersøg, om en vilkårlig digital funktion har et diagram bestående udelukkende af 2-input-OR-gates, eller et diagram bestående udelukkende af 2-input-AND-gates.

ØVELSE 14. Omskriv udtrykket for funktionen $w = x(y+\bar{z})$, således at det fremgår, hvordan funktionen kan realiseres af et diagram bestående udelukkende af 2-input-NAND-gates.

ØVELSE 15. Omskriv nedenstående diagram, således at det ender med at bestå udelukkende af 2-input-NOR-gates.



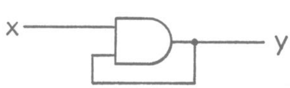
Vi har set, at en digital funktion kan repræsenteres af enten en *tabel* eller et *udtryk* eller et *diagram*. Desuden har vi lært at konvertere mellem disse tre forskellige repræsentationsformer. Nogle hyppigt anvendte arbejdsrutiner er skitseret på omstående tegning:




ANALYSE: Med udgangspunkt i et givet diagram for en digital funktion opskrives et udtryk for funktionen. Udtrykket reduceres så meget som muligt, hvorefter funktionstabellen opskrives. Ved hjælp af tabellen kan man svare på alle spørgsmål angående funktionens virkemåde.

SYNTESE: Med udgangspunkt i nogle krav til et ønsket diagram opstilles en tabel for den tilsvarende digitale funktion. Ved hjælp af tabellen findes et funktionsudtryk, der reduceres og eventuelt omskrives. Reduktionen er nødvendig, hvis man er interesseret i at bruge så få operatorer i udtrykket som muligt. Dette er man ofte interesseret i, da antallet af operatorer i udtrykket jo er lig med antallet af gates i det endelige diagram. Desuden er omskrivningen nødvendig, hvis man er interesseret i at bruge *én bestemt* type gates, for eksempel 2-input-NAND-gates. Når funktionsudtrykket har fået en tilfredsstillende form, kan diagrammet tegnes.

Til slut i dette kapitel skal det nævnes, at der findes diagrammer, der *ikke* repræsenterer nogen digital funktion. Som et eksempel på et sådant diagram kan nævnes:




På de to indgange af AND-gaten ligger signalerne x og y. Altså må udgangen have værdien xy. Men da udgangssignalet på forhånd vides at være y, kan vi opstille ligningen for diagrammet:

$$y = xy.$$

Hvis x tildeles værdien 0, ses det let, at også y får værdien 0. Men hvis x tildeles værdien 1, er det umuligt at sige, hvilken værdi y antager, thi ligningen bliver i dette tilfælde:

$$y = y.$$

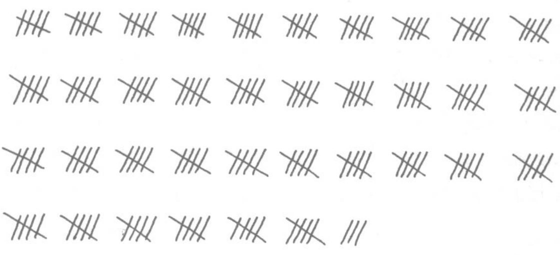
Denne ligning vil være tilfredsstillet af såvel værdien 0 som værdien 1. Det betyder, at y ikke er entydigt bestemt af værdien af x. Derfor er y *ikke* en funktion af x.

I det undersøgte diagram optræder der en såkaldt *loop*. En loop er generelt en rute i diagrammet bestående af forbindelseslinier, der fører fra en udgang til en indgang og videre gennem en gate til dennes udgang. Derfra til en ny indgang og så videre, indtil man ender det sted, man startede. Sådanne loops i diagrammer giver en *tilbagekobling* af værdier fra et "senere" sted i diagrammet til et "tidligere" sted. Tilbagekobling er grundlaget for digitalelektroniske *hukommelsesceller*. Disse skal ikke behandles her. De diagrammer, der opstår som følge af syntese af en digital funktion, vil altid være *loop-frie*, idet den samme variabel aldrig vil optræde på begge sider af lighedstegnet i funktionsudtrykket.

# 7. Talsystemer

Talbehandling foregår ofte i digitalelektroniske netværk. Dette er for eksempel tilfældet med digitalarmbåndsuret og lommeregneren. Desværre er det sædvanlige talsystem uegnet til maskinel behandling. I dette kapitel skal vi derfor se på andre muligheder for talrepræsentation.

Når man tæller får, er det praktisk at benytte det velkendte princip:



Når man senere skal indberette antallet af får til landbrugsministeriet, bør man vælge skrivemåden:

183.

Nu er antallet angivet i overensstemmelse med det sædvanlige 10-talsystem, *det decimale talsystem*. Det mest betydende ciffer angiver antallet af hundreder, næste ciffer angiver antallet af tiere, mens det mindst betydende ciffer er antallet af enere:

$$183 = 1\cdot10^2+8\cdot10^1+3\cdot10^0.$$

Dybt inde i landbrugsministeriets computer er antallet af får imidlertid repræsenteret på formen:

$$(10110111)_2.$$

Her er benyttet 2-talsystemet, *det binære talsystem*, men ellers er princippet det samme som for 10-talsystemet:

$$(10110111)_2 = 1\cdot2^7+0\cdot2^6+1\cdot2^5+1\cdot2^4 +0\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0.$$

Cifrene i det binære talsystem kan kun antage værdierne 0 og 1. Derfor er det specielt dette talsystem, der benyttes ved talbehandling i digitalelektroniske netværk.

Desværre er det binære talsystem ikke særligt menneskevenligt. Det skyldes blandt andet, at antallet af cifre i et binært tal er cirka 3 gange større end antallet af cifre i det tilsvarende decimale tal. Derfor benytter man ofte 16-talsystemet, *det hexadecimale talsystem*, som en yndet rasteplads på overgangen mellem menneskets decimale talsystem og computerens binære talsystem. Hvis antallet af får udtrykkes i det hexadecimale talsystem, ser det således ud:

$$(B7)_{16}.$$

B står for det hexadecimale ciffer 11, således at

$$(B7)_{16} = 11\cdot16^1+7\cdot16^0.$$

En lignende rolle spiller det såkaldte BCD-system. Navnet er en forkortelse af *det binært kodede decimale talsystem*. Princippet i dette system er, at hvert enkelt ciffer i et decimalt tal udtrykkes i det binære talsystem med fire binære cifre. Antallet af får kan i BCD-systemet skrives på følgende måde:

$$(0001.1000.0011)_{BCD}$$

Fortolkningen af denne skrivemåde er:

$$(0001.1000.0011)_{BCD} =$$

$$(0001)_2 \cdot 10^2 + (1000)_2 \cdot 10^1 + (0011)_2 \cdot 10^0$$

Efter denne indledende præsentation af de mest benyttede talsystemer, skal vi til at undersøge deres matematiske indhold. Vi skal beskæftige os med mængden af de tal, der kan opfattes som *antal*, det vil sige mængden af de hele tal, der er større end eller lig med nul. Denne mængde vil vi betegne med $N_o$. Det er sundt i det følgende at opfatte elementerne i $N_o$ som nogle objekter, der fører en selvstændig abstrakt tilværelse uafhængigt af menneskenes gøren og laden. I modsætning hertil kan talsystemerne opfattes som menneskenes middel til at tale om og beskrive elementerne i $N_o$. Eksempelvis er der næppe nogen, der betvivler, at der er et vist antal får på marken. Dette antal er der uafhængigt af, om vi tæller fårene eller ej. Antallet påvirkes heller ikke af, hvorledes vi eventuelt nedskriver antallet. Med andre ord kan man sige, at tallene er universelle, men talsystemerne er menneskeskabte.

DEFINITION 1. Lad g være et helt tal, der er større end eller lig med 2. Vi definerer følgende betegnelser:

1) Tallene 0,1,2,...,g-1 kaldes *g-cifrene*.

2) En uendelig følge, $(a_0, a_1, a_2, \ldots)$, bestående af g-cifre vil vi kalde en *g-følge*, såfremt kun endelig mange af følgens elementer er forskellige fra nul.

3) Mængden af samtlige g-følger vil vi kalde *g-talsystemet*, og vi giver denne mængde betegnelsen $T_g$.

4) Tallet g kalder vi *grundtallet* for talsystemet $T_g$.

ØVELSE 1. Lav en liste over samtlige 2-cifre, samtlige 10-cifre og samtlige 16-cifre.

ØVELSE 2. Er g et g-ciffer?

ØVELSE 3. Undersøg, hvorvidt de følgende to udsagn om en uendelig talfølge er ækvivalente:

1) Kun endelig mange af følgens elementer er forskellige fra nul.
2) Uendelig mange af følgens elementer er lig med nul.

ØVELSE 4. Hvilke af nedennævnte talfølger er 16-følger?

1) (1,2,3,1,2,3,1,2,3,1,...)
2) (16,16,16,16,16,16,.....)
3) (15,15,15,15,15,15,.....)
4) (0,0,0,0,0,0,0,0,0,0,...)
5) (1,0,0,15,0,2,0,0,0,0,..)
6) (2,7,$\frac{1}{2}$,4,0,0,0,0,0,0,...)
7) (0,0,0,1,0,1,0,0,0,0,...)

ØVELSE 5. Hvor mange elementer har $T_{\bar{g}}$, hvor g betegner et vilkårligt helt tal, der er større end eller lig med 2 ?

ØVELSE 6. Hvis vi tillod g at have værdien 1, hvor mange elementer ville $T_g$ så indeholde ?

Vi skal forsøge at etablere en én-entydig korrespondance mellem elementerne i $T_g$ og elementerne i $N_o$. Hvis det lykkes, kan vi nemlig bruge $T_g$ som en *model* af $N_o$. Den formelle behandling af $T_g$ og $N_o$ vanskeliggøres lidt af, at vi ikke så godt kan tale om elementerne i $N_o$ uden at benytte et af de talsystemer, vi er i færd med at udvikle, for eksempel det decimale talsystem. Ikke desto mindre vedtager vi, at når vi nævner konkrete tal i $N_o$, så er det underforstået, at de er angivet i det decimale talsystem, med mindre andet er nævnt.

**DEFINITION 2.** (*Heltalsfunktionen, int*).
Heltalsfunktionen, int, er den funktion, der er defineret ved:

$$\text{int} : \mathbf{R} \to \mathbf{Z}.$$
$$\forall x \in \mathbf{R} : \text{int}(x) = \max\{i \in \mathbf{Z} \mid i \leqq x\}.$$

Heltalsfunktionen er altså den funktion, der til et vilkårligt reelt tal, x, tilordner det største hele tal, der er mindre end eller lig med x. Betegnelsen, int, er en forkortelse af *integer* (latin: ren, hel). Funktionen har, som vi skal se, stor betydning i alt arbejde vedrørende de hele tal.

**ØVELSE 7.** Find følgende funktionsværdier:

1) int(7,02)
2) int(7)
3) int(0,4)
4) int(0)
5) int(-0,4)
6) int(-7)
7) int(-7,02)
8) int($\pi$).

**SÆTNING 1.** For vilkårlige reelle tal, x og y, gælder:

(1) $\text{int}(x) \leqq x < \text{int}(x)+1$.
(2) $x \leqq y \Rightarrow \text{int}(x) \leqq \text{int}(y)$.

BEVIS: For et vilkårligt reelt tal, x, definerer vi:

$$M_x = \{i \in \mathbf{Z} \mid i \leqq x\}.$$

Så gælder der:

$$\begin{aligned} & \text{int}(x) = \max M_x \\ \Rightarrow\ & \text{int}(x) \in M_x \\ \Rightarrow\ & \text{int}(x) \leqq x. \end{aligned}$$

Hermed er den første ulighed i (1) bevist. Antag nu, at int(x)+1 er mindre end eller lig med x. Så får vi:

$$\begin{aligned} & \mathrm{int}(x)+1 \leqq x \\ \Rightarrow\ & \mathrm{int}(x)+1 \in M_x \\ \Rightarrow\ & \mathrm{int}(x)+1 \leqq \max M_x \\ \Rightarrow\ & \mathrm{int}(x)+1 \leqq \mathrm{int}(x) \\ \Rightarrow\ & 1 \leqq 0. \end{aligned}$$

Da udsagnet, $1 \leqq 0$, er falsk, må også udsagnet, $\mathrm{int}(x)+1 \leqq x$, være falsk. Derfor er int(x)+1 større end x. Hermed er den sidste af ulighederne i (1) bevist. For at bevise (2) antager vi, at x er mindre end eller lig med y. Så gælder:

$$\begin{aligned} & \mathrm{int}(x) = \max M_x \\ \Rightarrow\ & \mathrm{int}(x) \in M_x \\ \Rightarrow\ & \mathrm{int}(x) \leqq x \\ \Rightarrow\ & \mathrm{int}(x) \leqq y \\ \Rightarrow\ & \mathrm{int}(x) \in M_y \\ \Rightarrow\ & \mathrm{int}(x) \leqq \max M_y \\ \Rightarrow\ & \mathrm{int}(x) \leqq \mathrm{int}(y). \end{aligned}$$

Hermed er også (2) bevist. Dette afslutter beviset for sætningen. □

ØVELSE 8. Bevis, at følgende formel er gyldig for et vilkårligt reelt tal, x,:

$$\mathrm{int}(x+1) = \mathrm{int}(x)+1.$$

ØVELSE 9. Gør rede for, at formlen

$$\mathrm{int}(-x) = -\mathrm{int}(x)$$

*ikke* gælder generelt. For hvilke værdier af x er formlen gyldig ?

Fremover antager vi, at g er et helt tal, der er større end eller lig med 2. Så er g grundtallet for talsystemet $T_g$. Hvis $(a_0, a_1, a_2, \ldots)$ er et element i $T_g$, så er der kun endelig mange af cifrene, $a_0$, $a_1$, $a_2$, ... , der er forskellige fra nul. Derfor kan vi uden vanskelighed tale om summen $a_0 g^0 + a_1 g^1 + a_2 g^2 + \ldots$, idet vi blot ser bort fra hele "halen" af nul'er. Da denne sum er et ikke-negativt helt tal, er den et element i $N_o$. Hermed er der skabt en forbindelse fra elementerne i $T_g$ til elementerne i $N_o$. Disse iagttagelser formaliserer vi i nedenstående definition.

DEFINITION 3. (*Sumfunktionen*).
Sumfunktionen, $(\ )_g$, er den funktion, der er defineret ved:

$$(\ )_g : T_g \to N_o$$
$$(a_0, a_1, a_2, \ldots)_g = a_0 g^0 + a_1 g^1 + a_2 g^2 + \ldots$$

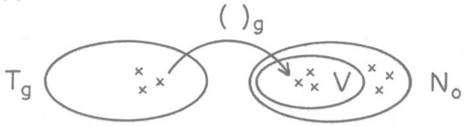
ØVELSE 10. Beregn funktionsværdierne:

1) $(3,8,1,0,0,0,0,0,0,0,0,\ldots)_{10}$
2) $(1,1,1,0,1,1,0,1,0,0,0,\ldots)_2$
3) $(7,9,13,0,0,0,0,0,0,0,\ldots.)_{16}$

Vi skal bruge sumfunktionen til at formidle kontakten mellem elementerne i $T_g$ og elementerne i $N_o$. Det er derfor af største vigtighed at få fastslået, at den er bijektiv. Det sker i de kommende tre sætninger.

SÆTNING 2. Sumfunktionen, $(\ )_g$, afbilder $T_g$ surjektivt på $N_o$.

BEVIS: Antag, at sumfunktionen *ikke* er surjektiv, og lad V betegne dens værdimængde. Så findes der elementer i $N_o$, som ikke tilhører V:



Da enhver ikke-tom delmængde af $N_o$ indeholder et mindste element, kan vi definere t til at være det mindste element i $N_o$, som ikke tilhører V. Så gælder:

(1) $t \in N_o$

(2) $t \notin V$

(3) $n \in N_o \wedge n < t \Rightarrow n \in V$

Lad nu k og r være de tal, der er defineret ved:

(4) $k = \text{int}(t/g)$

(5) $r = t - gk$

Det ses let, at disse tal er hele tal:

(6) $k \in Z$

(7) $r \in Z$

Vi beviser nu, at k tilhører $N_o$:

| | |
|---|---|
| $t \geqq 0$ | ifølge (1) |
| $\Rightarrow t/g \geqq 0$ | da $g>0$ |
| $\Rightarrow \text{int}(t/g) \geqq \text{int}(0)$ | ifølge sætning 1(2) |
| $\Rightarrow k \geqq 0$ | ifølge (4) |
| $\Rightarrow k \in N_o$ | ifølge (6) |

Altså gælder:

(8) $k \in N_o$.

Vi skal herefter bevise, at k er mindre end t. Da 0 er et element i V (bevis selv dette), og da t ifølge (2) ikke er et element i V, kan t ikke være lig med 0. Da t ifølge (1) tilhører $N_o$, og da t ikke er 0, må t være positiv. Dette benyttes nedenfor:

| | |
|---|---|
| $k = \text{int}(t/g)$ | ifølge (4) |
| $\leqq t/g$ | ifølge sætning 1(1) |
| $< t$ | da $t>0$ og $g>1$ |

Altså har vi:

(9) $k<t$

Af (8) og (9) i kombination med (3) følger, at k tilhører billedmængden, V. Derfor findes der en g-følge, som afbildes i k ved sumfunktionen. Lad $(a_0,a_1,a_2,\ldots)$ være en sådan g-følge:

(10) $(a_0,a_1,a_2,\ldots) \in T_g$

(11) $(a_0,a_1,a_2,\ldots)_g = k$

Efter således at have undersøgt k, vender vi blikket mod r. Vi skal bevise, at r er et g-ciffer:

| | |
|---|---|
| $\text{int}(t/g) \leqq t/g < \text{int}(t/g)+1$ | iflg. sætn. 1(1) |
| $\Rightarrow 0 \leqq t/g-\text{int}(t/g) < 1$ | |
| $\Rightarrow 0 \leqq t-g\cdot\text{int}(t/g) < g$ | da $g>0$ |
| $\Rightarrow 0 \leqq t-gk < g$ | ifølge (4) |
| $\Rightarrow 0 \leqq r < g$ | ifølge (5) |

Dette sammen med (8) giver det ønskede:

(12) r er et g-ciffer.

Af (10) og (12) følger så:

(13) $(r,a_0,a_1,a_2,\ldots) \in T_g$.

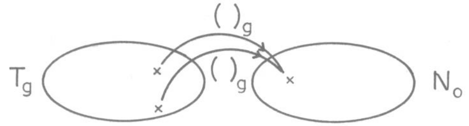
Til slut omskriver vi t således:

| | |
|---|---|
| $t = r+gk$ | ifølge (5) |
| $= r+g\cdot(a_0,a_1,a_2,\ldots)_g$ | ifølge (11) |
| $= r+g\cdot(a_0g^0+a_1g^1+a_2g^2+\ldots)$ | iflg. def. 3 |
| $= rg^0+a_0g^1+a_1g^2+a_2g^3+\ldots$ | |
| $= (r,a_0,a_1,a_2,\ldots)_g$ | iflg. def. 3 |

Denne sidste omskrivning har mening på grund af (13). Omskrivningen viser, at t tilhører værdimængden, V. Men det er i strid med (2). Antagelsen, om at sumfunktionen ikke er surjektiv, må derfor være forkert. Altså *er* sumfunktionen surjektiv. Hermed er sætningen bevist. □

SÆTNING 3. Sumfunktionen, $(\ )_g$, afbilder $T_g$ injektivt på $N_o$.

BEVIS: Antag, at sumfunktionen *ikke* er injektiv. Så findes der to forskellige elementer i $T_g$, som har samme funktionsværdi ved sumfunktionen:




Lad $(a_0,a_1,a_2,...)$ og $(b_0,b_1,b_2,...)$ være to sådanne elementer. Så gælder:

(1) $(a_0,a_1,a_2,...) \in T_g$

(2) $(b_0,b_1,b_2,...) \in T_g$

(3) $(a_0,a_1,a_2,...) \neq (b_0,b_1,b_2,...)$

(4) $(a_0,a_1,a_2,...)_g = (b_0,b_1,b_2,...)_g$

Af (4) får vi:

$$a_0g^0+a_1g^1+a_2g^2+... = b_0g^0+b_1g^1+b_2g^2+...$$
$$\Rightarrow (a_0-b_0)g^0+(a_1-b_1)g^1+(a_2-b_2)g^2+... = 0$$

altså:

(5) $(a_0-b_0)g^0+(a_1-b_1)g^1+(a_2-b_2)g^2+ ... = 0$

Af (1) og (2) følger, at kun endelig mange af leddene på venstre side af (5) er forskellige fra nul, mens (3) medfører, at der er mindst ét sådant led. Altså findes der et største element, n, i $N_o$, således at leddet $(a_n-b_n)g^n$ er forskelligt fra nul. Med dette valg af n gælder så:

$$(6) \quad (a_n - b_n) \neq 0$$

$$(7) \quad j \in N_o \wedge j > n \Rightarrow (a_j - b_j)g^j = 0$$

Ved at kombinere (5) med (7) får vi:

$$(8) \quad (a_0 - b_0)g^0 + (a_1 - b_1)g^1 + \ldots + (a_n - b_n)g^n = 0$$

Hvis n var lig med nul, ville (6) og (8) være i indbyrdes modstrid. Derfor må n være større end nul, således at der er mere end ét led på venstre side af (8). Altså kan vi omskrive (8) til:

$$(9) \quad (a_0 - b_0)g^0 + \ldots + (a_{n-1} - b_{n-1})g^{n-1} = (b_n - a_n)g^n$$

Vi foretager nu en vurdering af det i'te led på venstre side af (9):

$$\begin{aligned}
& \begin{cases} 0 \leqq a_i \leqq g-1 \\ 0 \leqq b_i \leqq g-1 \end{cases} \\
\Rightarrow & \begin{cases} 0 \leqq a_i \leqq g-1 \\ 0 \geqq -b_i \geqq 1-g \end{cases} \\
\Rightarrow & \begin{cases} 0 \leqq a_i \leqq g-1 \\ 1-g \leqq -b_i \leqq 0 \end{cases} \\
\Rightarrow \quad & 1-g \leqq a_i - b_i \leqq g-1 \\
\Rightarrow \quad & (1-g)g^i \leqq (a_i - b_i)g^i \leqq (g-1)g^i \qquad \text{da } g>0 \\
\Rightarrow \quad & g^i - g^{i+1} \leqq (a_i - b_i)g^i \leqq g^{i+1} - g^i
\end{aligned}$$

Ved successivt at lade i antage værdierne 0,1,2,...,n-1 i denne sidste dobbeltulighed, får vi:

$$\begin{array}{ccccc}
g^0 - g^1 & \leqq & (a_0 - b_0)g^0 & \leqq & g^1 - g^0 \\
g^1 - g^2 & \leqq & (a_1 - b_1)g^1 & \leqq & g^2 - g^1 \\
g^2 - g^3 & \leqq & (a_2 - b_2)g^2 & \leqq & g^3 - g^2 \\
\vdots & & \vdots & & \vdots \\
g^{n-1} - g^n & \leqq & (a_{n-1} - b_{n-1})g^{n-1} & \leqq & g^n - g^{n-1}
\end{array}$$

Ved addition af alle disse dobbeltuligheder og ved samtidig anvendelse af (9) fremkommer følgende vurdering af $a_n - b_n$:

$$1-g^n \leq (b_n - a_n)g^n \leq g^n - 1$$

$$\Rightarrow \frac{1}{g^n} - 1 \leq (b_n - a_n) \leq 1 - \frac{1}{g^n} \qquad \text{da } g>0$$

$$\Rightarrow -1 < b_n - a_n < 1$$

$$\Rightarrow b_n - a_n = 0 \qquad \text{da } b_n - a_n \in Z$$

$$\Rightarrow a_n - b_n = 0$$

Men dette resultat strider mod (6). Antagelsen, om at sumfunktionen ikke er injektiv, må derfor være forkert. Altså *er* sumfunktionen injektiv. Hermed er sætningen bevist. □

SÆTNING 4. Sumfunktionen, $(\ )_g$, afbilder $T_g$ bijektivt på $N_o$.

BEVIS: Ifølge sætning 2 er sumfunktionen surjektiv, og ifølge sætning 3 er den injektiv. Altså er sumfumktionen bijektiv. Hermed er sætningen bevist. □

Sætning 4 er en garanti for, at $T_g$ er en god model af $N_o$. Sætningen siger nemlig, at et vilkårligt element, t, i $N_o$ på præcis én måde kan skrives på formen:

$$t = (a_0, a_1, a_2, \ldots)_g.$$

Dermed kan vi tillade os at *identificere* t med elementet $(a_0, a_1, a_2, \ldots)$ i $T_g$.

Den skrivemåde, vi har benyttet for elementerne i $T_g$ og for deres funktionsværdier ved sumfunktionen, er meget praktisk i forbindelse med en *teoretisk* behandling af talsystemerne, men når man arbejder med konkrete *anvendelser* af talsystemerne, vælger man en anden skrivemåde. Overgangen fra skrivemåden:

$$t = (a_0, a_1, a_2, \ldots)_g$$

til den almindeligt anvendte skrivemåde:

$$t = (a_n a_{n-1} \ldots a_0)_g$$

kan beskrives således:

1) g-cifrene skrives i omvendt rækkefølge.
2) Foranstillede nul'er udelades.
3) Komma'erne udelades.

Hvis det klart fremgår af sammenhængen, hvilket talsystem der anvendes, udelader man som regel også g-suffix'et samt parentesen:

$$t = a_n a_{n-1} \ldots a_0 .$$

For eksempel er siderne i denne bog nummereret efter dette princip.

Hvis grundtallet er større end ti, opstår der et specielt problem, når komma'erne fjernes (hvilket problem?). Dette problem kan dog let undgås, hvis vi vælger nye symboler for de cifre, der er større end ni. De almindeligt accepterede symboler for 16-cifrene fremgår af tabellen:

| 16-ciffer | Symbol |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 10 | A |
| 11 | B |
| 12 | C |
| 13 | D |
| 14 | E |
| 15 | F |

Den tradition, ifølge hvilken man skriver tal med de mindst betydende cifre længst mod højre, kan virke ret ulogisk. For eksempel medfører den, at talkolonner fremtræder med lige højremargin, mens al anden skriftlig fremstilling har lige venstremargin. Desværre må vi nok se i øjnene, at denne tradition er kommet for at blive, -til stor ærgelse for blandt andet konstruktører af bogholderimaskiner.

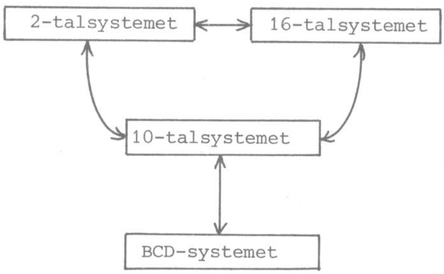
Vort sædvanlige 10-talsystem blev udviklet i Indien for rundt regnet 1400 år siden. Senere blev det kendt i Europa efter påvirkning fra arabiske skribenter og handelsfolk. Allerede for cirka 1000 år siden var systemet i brug ganske enkelte steder i middelhavsegnene. Men det var først for omkring 400 år siden, at dette nye system begyndte at gøre sig rigtigt gældende blandt de mange andre tællesystemer ( for eksempel romertal). I de tidlige indisk-arabiske skrifter er skriveretningen orienteret fra højre mod venstre. Også tallene blev skrevet fra højre mod venstre, og de mindst betydende cifre blev skrevet *først*. Derved kom tallene til at fremtræde netop på den måde, de skrives endnu idag. Det er nærliggende at gætte på, at vi netop her kan finde grunden til, at vi er havnet i en situation med højreløbende tekst og venstreløbende tal.

ØVELSE 11. I mange lande, deriblandt Danmark, har man i tidligere tid anvendt 12-talsystemet (det duodecimale talsystem). Find nogle spor efter dette talsystem.

ØVELSE 12. Hvilke uheldige følger ville det få, hvis sumfunktionen ikke var surjektiv ? - eller hvis den ikke var injektiv ?

# 8. Konvertering mellem talsystemer

Figuren nedenfor viser de talsystemer der for tiden er de mest benyttede:




Pilene på figuren symboliserer konverteringerne mellem talsystemerne. Vi skal først se på overgangen fra henholdsvis det binære og det hexadecimale talsystem til 10-talsystemet.

ØVELSE 1. Konvertér følgende tal til det decimale talsystem:

| | | | |
|---|---|---|---|
| 1) | $(1100101)_2$ | 5) | $(1FC)_{16}$ |
| 2) | $(1011100000)_2$ | 6) | $(9)_{16}$ |
| 3) | $(101000100)_2$ | 7) | $(1A3)_{16}$ |
| 4) | $(101011100)_2$ | 8) | $(195)_{16}$ |

Ved en automatiseret konvertering, for eksempel i en programmérbar lommeregner, benyttes en algoritme, der i alle detaljer beskriver, hvorledes konverteringen skal forløbe. Nedenfor er vist en sådan algoritme. Den forudsætter, at man kender grundtallet, g, samt antallet af cifre, n, og endelig, at man kender selve cifrene, $a_n, a_{n-1}, \ldots, a_0$.

ALGORITME $T_g \to T_{10}$ :

```
start
indlæs g,n,a_n,a_(n-1),...,a_0
t:=0
t:=tg+a_n  <---------+
n:=n-1               |
hvis n≥0 så hop -----+
udlæs t
stop
```

ØVELSE 2. Konvertér $(E27)_{16}$ til det decimale talsystem ved anvendelse af algoritmen.

ØVELSE 3. Konvertér et abstrakt trecifret tal, $(a_2a_1a_0)_g$, til det decimale talsystem ved hjælp af algoritmen.

ØVELSE 4. Hvordan kan det være, at algoritmen "véd", at den skal konvertere til det decimale talsystem, når tallet 10 overhovedet ikke er nævnt i algoritmen ?

ØVELSE 5. Algoritmen kan også bruges til at beregne funktionsværdier for et n'te-grads polynomium. Gør nærmere rede for dette.

ØVELSE 6. Omsæt algoritmen til et computerprogram.

Vi skal nu konvertere den anden vej, altså *fra* det decimale talsystem *til* henholdsvis det binære og det hexadecimale talsystem. Følgende algoritme beregner g-cifrene (på decimal form) for et decimalt tal, t:

ALGORITME $T_{10} \rightarrow T_g$ :

```
start
indlæs g,t
n:=-1
n:=n+1
k:=int(t/g)
a_n:=t-gk
t:=k
hvis t>0 så hop
udlæs a_n,a_(n-1),...,a_0
stop
```

ØVELSE 7. Prøv algoritmen på tallene g=16 og t=183.

ØVELSE 8. Hvilket resultat leverer algoritmen, hvis t=0 ?

ØVELSE 9. Beviset for at sumfunktionen er surjektiv bygger på samme teknik som ovenstående algoritme. Benyt hovedpunkterne i dette bevis til at forklare, at algoritmen nødvendigvis giver det ønskede resultat.

ØVELSE 10. Konvertér følgende decimale tal til såvel det binære som det hexadecimale talsystem: 5, 127, 1000, 16.

Som tidligere nævnt er det binære talsystem særdeles velegnet i forbindelse med digitale netværk. Dette skyldes, at hvert enkelt ciffer i et binært tal kan opfattes som en digital variabel, thi cifferet kan kun antage to forskellige værdier, 0 og 1. Digitale netværk, der er designet til talbehandling, er som regel konstrueret til at arbejde med et bestemt antal bits. Ordet *bit* er en sammentrækning af det engelske *binary digit*, der ganske enkelt betyder binært ciffer.

Hvis et binært tal har færre cifre end det antal, det digitale netværk er designet til, kan man altid "fylde ud" med foranstillede nul'er:

| | |
|---|---|
| 00100001 | 21 |
| 01000000 | 40 |
| 00000000 | 00 |
| 01111110 | 7E |
| 00100011 | 23 |
| 10000110 | 86 |
| 00100011 | 23 |
| 01110111 | 77 |
| 01110110 | 76 |

Talkolonnen til venstre består af 8-bit tal. Til højre ser vi de tilsvarende tocifrede hexadecimale tal. Kolonnen af 8-bit tal udgør faktisk et additionsprogram for en Z80 microprocessor. Det ses tydeligt, at der er gode muligheder for at begå en fejl, hvis man skal meddele dette program til andre mennesker, eller hvis man blot skal skrive det af på et stykke papir.

Generelt kan man sige, at hexadecimale tal er meget lettere genkendelige og meget nemmere at huske end tilfældet er med binære tal. Dette er én af grundene til, at hexadecimale tal har vundet så stor udbredelse. En anden, og mere væsentlig grund, er, at der findes nogle meget elegante metoder til konvertering mellem det binære og det hexadecimale talsystem. Disse metoder benytter sig af, at 16 er en potens af 2, nemlig $2^4$. Da et 16-ciffer er mindre end 16, kan det skrives med højst fire bits i det binære talsystem. Omvendt vil ethvert binært tal med højst fire bits være et 16-ciffer:

| Hexadecimal | 4-bit binær |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| A | 1010 |
| B | 1011 |
| C | 1100 |
| D | 1101 |
| E | 1110 |
| F | 1111 |

Hvis man skal konvertere $(6E)_{16}$ til det binære talsystem, erstatter man simpelthen hvert hexadecimalt ciffer med det tilsvarende 4-bit binære tal, hvorefter man kan udelade eventuelle foranstillede nul'er:

$$(6E)_{16}$$

$$(01101110)_2$$

$$\downarrow$$

$$(1101110)_2$$

At denne fremgangsmåde er korrekt, kan indses således:

$$\begin{aligned}
(6E)_{16} &= 6\cdot 16^1+E\cdot 16^0 \\
&= (0110)_2\cdot 2^4+(1110)_2\cdot 2^0 \\
&= (0\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0)\cdot 2^4 \\
&\quad +(1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0)\cdot 2^0 \\
&= (0\cdot 2^7+1\cdot 2^6+1\cdot 2^5+0\cdot 2^4) \\
&\quad +(1\cdot 2^3+1\cdot 2^2+1\cdot 2^1+0\cdot 2^0) \\
&= (01101110)_2
\end{aligned}$$

ØVELSE 11. Konvertér tallene $(9B)_{16}$, $(101)_{16}$ og $(16D4)_{16}$ til det binære talsystem.

Konverteringen den modsatte vej er lige så let. Når man skal konvertere $(101011)_2$ til det hexadecimale talsystem, fylder man først ud med foranstillede nul'er, indtil antallet af bits er deleligt med 4. Dernæst erstatter man hver 4-bit blok med det tilsvarende hexadecimale ciffer:

$$(101011)_2$$
$$\downarrow$$
$$(\underbrace{0010}\underbrace{1011})_2$$
$$\downarrow$$
$$(2B)_{16}$$

ØVELSE 12. Konvertér tallene $(1100)_2$, $(1001001)_2$ og $(10)_2$ til det hexadecimale talsystem.

ØVELSE 13. Konvertér tallet $(10111100111)_2$ til det decimale talsystem på hver af følgende to måder:

1) Direkte $T_2 \rightarrow T_{10}$.
2) Indirekte $T_2 \rightarrow T_{16} \rightarrow T_{10}$.

Hvilken af de to metoder forløber lettest ?

Til slut skal vi se på det binært kodede decimale talsystem, eller *BCD-systemet*. Dette er et underligt kompromis mellem det binære og det decimale talsystem. Man siger, at et decmalt tal er *binært kodet*, hvis hvert enkelt ciffer er erstattet af det tilsvarende 4-bit binære tal, for eksempel som vist nedenfor:

$$183$$
$$\downarrow$$
$$(0001.1000.0011)_{BCD}$$

Principppet for BCD-systemet minder således en hel del om princippet for konvertering mellem det binære og det hexadecimale talsystem. Men BCD-systemet adskiller sig ved, at der kan forekomme *forbudte tal* som for eksempel "tallet" $(1100.0110)_{BCD}$. BCD-systemet anvendes mest ved overgangen mellem rene digitale netværk og perifere enheder. Således anvendes det for eksempel ved overgangen fra et tastatur til en microcomputer, og ved overgangen fra en microcomputer til et 7-segment display.

ØVELSE 14. Omskriv følgende decimale tal til BCD-systemet: 34, 801, 99, 0.

ØVELSE 15. Hvilke af følgende tal er forbudte? : $(0101)_{BCD}$, $(0111.1001)_{BCD}$, $(0000.1011)_{BCD}$, $(1010.0001)_{BCD}$.

ØVELSE 16. Ville det være rimeligt, at kalde det binære talsystem for det *binært kodede hexadecimale talsystem* ?

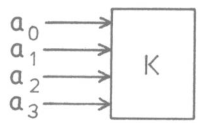
ØVELSE 17. Omskriv $(BCD)_{16}$ til BCD-systemet.

# 9. Digital talbehandling

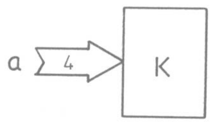
Hvis et digital-elektronisk netværk, K, skal behandle et variabelt tal, a, skal K have tilført information om a. Det sker normalt ved, at hvert af a's binære cifre, det vil sihver bit, opfattes som en digital variabel. Disse digitale variable føres til K via hver sin indgang. Hvis a er et 4-bit binært tal,

$$a = (a_3a_2a_1a_0)_2$$

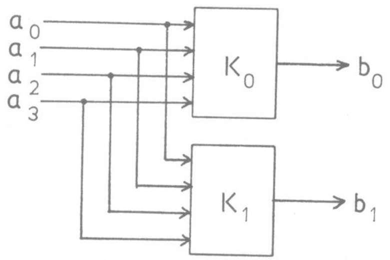
vil K altså være forsynet med fire indgange:



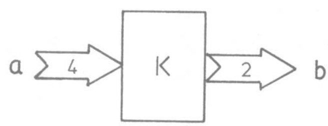

Hele bundtet af forbindelseslinier til indgangene kaldes en *bus*. Bussen transporterer tallet a. I det nærværende tilfælde er der tale om en *4-bit bus*. I blokdiagrammer tegnes en bus ofte som en bred pil med angivelse af antallet af bits i bussen:




Pilen angiver transportretningen. Når K har behandlet tallet, a, foreligger resultatet ofte i form af et nyt tal, b. Indgangstallet, a, og udgangstallet, b, har ikke nødvendigvis samme antal bits. Hvis b for eksempel er et 2-bit binært tal, altså $b = (b_1 b_0)_2$, er såvel $b_0$ som $b_1$ en digital funktion af de fire variable, $a_0$, $a_1$, $a_2$, $a_3$. Det digitalelektroniske netværk, K, udfører således *to* digitale funktioner, én for hver af de to udgangsbits:



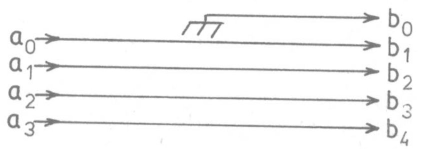

Man kan tit spare en del gates ved at lade $K_0$ benytte nogle af mellemresultaterne i $K_1$ og omvendt. Derfor er det ikke helt realistisk, når ovenstående tegning antyder, at K er sammensat af to separate netværk. Normalt tegnes blokdiagrammet simpelthen således:



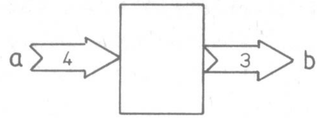

Heraf kan man se det væsentlige, nemlig at K udfører *to* digitale funktioner af *fire* variable.

Digitale talbehandlingsnetværk behøver ikke altid at være komplicerede. Dette fremgår af de følgende to øvelser.

ØVELSE 1. Gør rede for, at nedenstående diagram udfører regneoperationen $b = 2 \cdot a$, hvor $a = (a_3 a_2 a_1 a_0)_2$, og hvor $b = (b_4 b_3 b_2 b_1 b_0)_2$. I diagrammet betegner symbolet den digitale konstant 0.

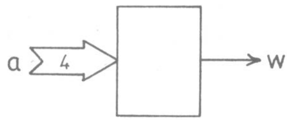


ØVELSE 2. Find et diagram af formen:



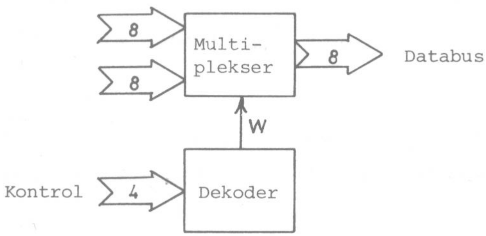

som udfører regneoperationen b = int(a/2).

ØVELSE 3. Find et diagram af formen:



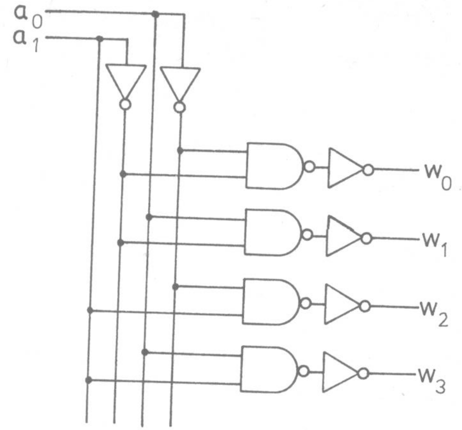

hvori w er 1, præcis når a = $(5)_{16}$.

Diagrammet i øvelse 3 kaldes en *dekoder*. Der er her tale om en 4-bit-$(5)_{16}$-dekoder. Dekodere bruges for eksempel til at kontrollere "busdriften" i større systemer. Udgangssignalet fra dekoderen kan sættes til at styre en *multiplekser* (et skiftespor), således at det tal, der skal ud på systemets databus, hentes fra det ønskede sted i netværket:




Udgangssignalet fra dekoderen kunne også sættes til at styre en *demultiplekser*. En demultiplekser er blot en omvendt multiplekser, idet den er forsynet med én indgangsbus og to eller flere udgangsbusser.

ØVELSE 4. Nedenfor er vist diagrammet for en fuldstændig 2-bit-dekoder. Udfyld tabellen for diagrammet, og forklar, hvad betegnelsen *fuldstændig* dækker over.

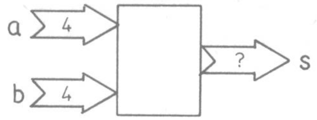



| $a_1$ | $a_0$ | $w_0$ | $w_1$ | $w_2$ | $w_3$ |
|---|---|---|---|---|---|
| 0 | 0 | | | | |
| 0 | 1 | | | | |
| 1 | 0 | | | | |
| 1 | 1 | | | | |

Læg mærke til, at diagrammet i den foregående øvelse indeholder en *intern 4-bit bus*, der transporterer indgangssignalerne samt deres komplementerede signaler. Den øvrige del af netværket tapper den nødvendige information fra den interne bus via et såkaldt *krydsfelt*. Denne måde at konstruere diagrammer på giver ikke altid det mest økonomiske netværk; men metoden har en fin systematik, og diagrammet er derfor let at overskue. Desuden er det nemt, selv i færdige konstruktioner, at ændre på netværkets funktion, hvis dette måtte ønskes. Disse egenskaber prioriteres ofte meget højt.

Resten af dette kapitel er centreret omkring konstruktionen af en *4-bit adder*, det vil sige et netværk, der kan udføre addition af to 4-bit binære tal. Dette eksempel er valgt af to grunde. For det første er heltalsaddition en grundlæggende operation i næsten alle større digitale systemer. For det andet vil eksemplet give mulighed for præsentation af nogle fremgangsmåder, der med held kan anvendes i andre sammenhænge.

Det netværk, vi skal konstruere, må være forsynet med to indgangsbusser til de to 4-bit binære tal, samt med én udgangsbus til tallenes sum:



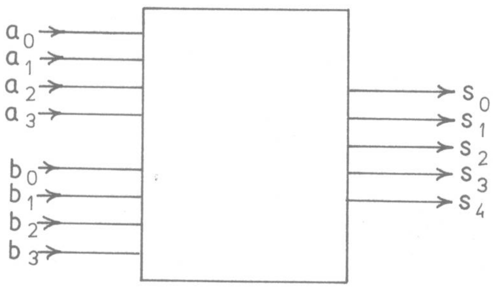

I blokdiagrammet af 4-bit adderen betegner a og b de to tal, der skal adderes, og s betegner deres sum.

ØVELSE 5. Hvad er den største værdi, udtrykt i 10-talsystemet, tallene a og b kan antage ?

For at finde det nødvendige antal bits i udgangsbussen, beregner vi den største værdi, s kan antage:

$$\begin{aligned} s_{max} &= a_{max} + b_{max} \\ &= (1111)_2 + (1111)_2 \\ &= 2\cdot(1111)_2 \\ &= (11110)_2 \qquad \text{(jævnfør øvelse 1)} \end{aligned}$$

Heraf ses, at s-bussen skal indeholde fem bits. Derfor skal netværket udføre fem funktioner af otte variable:




En mulig fremgangsmåde ved konstruktionen af diagrammet ville være at behandle de fem funktioner, $s_0$, $s_1$, $s_2$, $s_3$, $s_4$, hver for sig. Ved behandlingen af den i'te af disse, det vil sige $s_i$, kunne vi gennemføre den sædvanlige syntese:

1) Opstilling af tabellen for $s_i$ som funktion af $a_0$, $a_1$, $a_2$, $a_3$, $b_0$, $b_1$, $b_2$, $b_3$.
2) Opstilling af et udtryk for $s_i$.
3) Reduktion af det fundne udtryk.
4) Konstruktion af et diagram for $s_i$.

ØVELSE 6. Hvor mange rækker vil der være i tabellen for $s_i$?

Efter at have konstrueret diagrammerne for hver af de fem funktioner, $s_0$, $s_1$, $s_2$, $s_3$, $s_4$, kunne vi koble dem alle til a- og b-bussen. Denne fremgangsmåde vil føre til et anvendeligt diagram; men i praksis er metoden meget lidt elegant og næsten ubrugelig. Hver af de fem tabeller er meget store (jævnfør resultatet fra øvelse 6). Derfor vil det sandsynligvis være meget tidsrøvende at reducere funktionsudtrykkene tilstrækkeligt. Desuden er der en god chance for at begå en eller flere fejl undervejs.

I den situation, vi nu står i, er det en god idé at tænke på, hvad vi ville gøre, hvis vi skulle addere a og b manuelt. Derfor prøver vi at addere for eksempel $(1001)_2$ og $(1101)_2$. Vi bruger den sædvanlige additionsrutine, der også er kendt fra 10-talsystemet:

$$\begin{array}{r} {\scriptstyle 1\,0\,0\,1} \\ 1001 \\ 1101 \\ \hline 10110 \end{array}$$

Først adderes de to mindst betydende bits. Det giver en sum på 0 og en mente på 1. Dernæst adderes det næste par af bits samt menten fra den første addition. Således fortsættes, til der ikke er flere betydende bits eller menter.

ØVELSE 7. Addér tallene $(1100)_2$ og $(1011)_2$ efter samme metode som ovenfor, og overvej gyldigheden af denne metode.

Hvis vi vil addere tallene $(a_3a_2a_1a_0)_2$ og $(b_3b_2b_1b_0)_2$, kan vi generalisere omstående regnestykke således:

$$
\begin{array}{ccccc}
 & m_3 & m_2 & m_1 & m_0 \\
 & & a_3 & a_2 & a_1 & a_0 \\
 & & b_3 & b_2 & b_1 & b_0 \\
\hline
 & s_4 & s_3 & s_2 & s_1 & s_0
\end{array}
$$

Der gælder følgende relationer:

$$
\begin{aligned}
&(a_0)_2+(b_0)_2 = (m_0s_0)_2 \\
&(m_0)_2+(a_1)_2+(b_1)_2= (m_1s_1)_2 \\
&(m_1)_2+(a_2)_2+(b_2)_2= (m_2s_2)_2 \\
&(m_2)_2+(a_3)_2+(b_3)_2= (m_3s_3)_2 \\
&(m_3)_2= (s_4)_2
\end{aligned}
$$

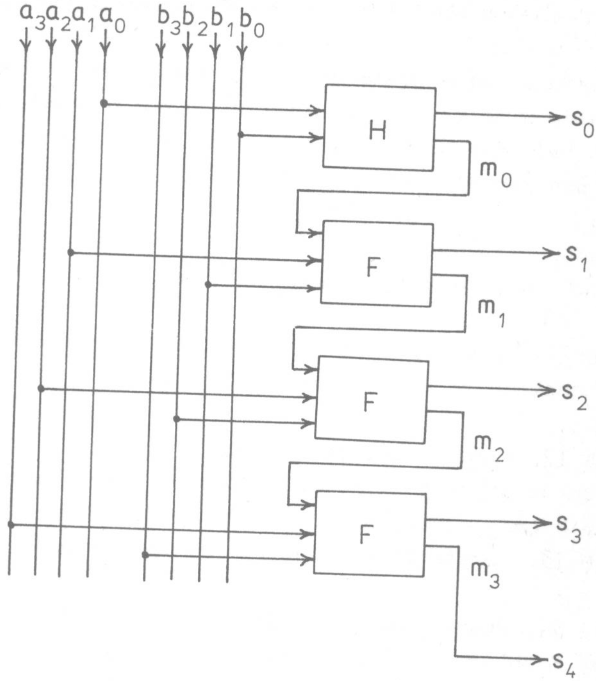
De overførte menter er betegnet med $m_0$, $m_1$, $m_2$, $m_3$ og $m_4$. Det ses, at der én gang blev lagt to enkelte bits sammen, og at der tre gange blev lagt tre bits sammen. Det vi har brug for i diagrammet er altså:

1) Et netværk, der udfører addition af to bits, og leverer resultatet i form af en sum-bit og en mente-bit.
2) Tre ens netværk, der udfører addition af tre bits, og leverer resultatet i form af en sum-bit og en mente-bit.

Et netværk af typen 1) kaldes en *half-adder*. Fornavnet, half-, hentyder til, at netværket ikke kan behandle indkommende menter.

Et netværk af typen 2) kaldes en *full-adder*, og er kendetegnet ved, at kunne behandle såvel de to indkommende bits som den indkommende mente.

Idet vi lader H og F betegne henholdsvis en half-adder og en full-adder, kan vi tegne et lidt mere detaljeret diagram for 4-bit adderen:




ØVELSE 8. Sammenlign diagrammets struktur med strukturen i det tilsvarende regnestykke.

ØVELSE 9. Hvor mange rækker vil der være i tabellen for henholdsvis $s_0$, $m_0$, $s_1$, $m_1$, $s_2$, $m_2$, $s_3$ og $m_3$ ? Sammenlign resultaterne med resultatet fra øvelse 6.

Nu skulle det være klart, at den nye fremgangsmåde er mere velegnet end den tidligere. Desuden rummer ovenstående diagram en oplagt mulighed for en udvidelse til for eksempel en 8-bit adder. Denne mulighed var ikke til stede i det tidligere blokdiagram.

ØVELSE 10. Hvorledes kan man konstruere en 8-bit adder ved anvendelse af half-addere og full-addere ?

Vi er nu kommet dertil, at vores 4-bit adder er færdig, hvis vi blot kan konstruere diagrammet for en half-adder og diagrammet for en full-adder. Konstruktionen af disse diagrammer finder sted i den kommende række af øvelser.

Først ser vi på half-adderen. Den har to indgange, som vi vil kalde x og y, samt to udgange, som vi kalder s og m. Kravet til half-adderen er, at s og m er henholdsvis sum og mente ved addition af x og y. Det vil sige, at der skal gælde:

$$(x)_2 + (y)_2 = (ms)_2.$$

ØVELSE 11. Opstil tabellen for s som funktion af x og y, samt tabellen for m som funktion af x og y.

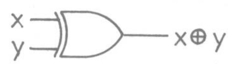
ØVELSE 12. Find et udtryk for s og et udtryk for m som funktioner af x og y. Reducér derefter disse udtryk.

ØVELSE 13. Konstruér et diagram for half-adderen, H.

ØVELSE 14. Der findes et diagram for half-adderen, der kun bruger fem 2-input-NAND-gates. Prøv at lave et sådant.

ØVELSE 15. Den digitale funktion, der er fastlagt af tabellen:

| x | y | $x \oplus y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




er så anvendt, at den produceres som et færdigt digitalelektronisk netværk. Diagramsymbolet ser ud som vist, og kaldes en *2-input-XOR-gate*. Dette er en forkortelse af *2-input-exclusive-or-gate*. Forklar meningen med betegnelsen *exclusive-or*.

ØVELSE 16. Konstruér et diagram for half-adderen, H, ved kun at anvende én 2-input-XOR-gate og én 2-input-AND-gate.

ØVELSE 17. I diagramsymbolet for 2-input-XOR-gate er der ikke gjort forskel på de to indgange. Dette er kun holdbart, såfremt operatoren, $\oplus$, er kommutativ. Bevis, at dette er tilfældet.

ØVELSE 18. Undersøg, om operatoren, $\oplus$, er associativ.

Herefter vil vi gå over til at se på full-adderen. Den har tre indgange, x, y og z, samt to udgange, s og m. Kravet til full-adderen er, at s og m er henholdsvis sum og mente ved additionen af x, y og z. Det vil sige, at der skal gælde følgende ligning:

$$(x)_2 + (\underline{y})_2 + (z)_2 = (ms)_2 .$$

ØVELSE 19. Opstil tabellerne for henholdsvis s og m som funktioner af x, y og z.

ØVELSE 20. Konstruér et diagram for full-adderen, F, efter først at have fundet og reduceret udtryk for s og m som funktioner af x, y og z.

ØVELSE 21. Der findes et diagram for full-adderen, der kun bruger to 2-input-XOR-gates samt tre 2-input-NAND-gates. Prøv at konstruere et sådant.

ØVELSE 22. Tegn et fuldstændigt diagram for 4-bit adderen.

Til slut vil det være på sin plads at nævne, at 4-bit adderen findes som en færdig operationsenhed i 7400-serien.

# Formelsamling

AKSIOMER

( 1') $x+y=y+x$ ( 1") $x \cdot y=y \cdot x$

( 2') $x+(y \cdot z)=(x+y) \cdot (x+z)$ ( 2") $x \cdot (y+z)=(x \cdot y)+(x \cdot z)$

( 3') $x+0=x$ ( 3") $x \cdot 1=x$

( 4') $x+\bar{x}=1$ ( 4") $x \cdot \bar{x}=0$

SÆTNINGER

( 5 ) Entydighed af neutrale elementer.

( 6 ) Entydighed af komplementer.

( 7 ) Dualitetsprincippet.

( 8') $x+1=1$ ( 8") $x \cdot 0=0$

( 9') $x+x=x$ ( 9") $x \cdot x=x$

(10 ) $\bar{\bar{x}}=x$

(11') $(x+y) \cdot (x+\bar{y})=x$ (11") $(x \cdot y)+(x \cdot \bar{y})=x$

(12') $x+(x \cdot y)=x$ (12") $x \cdot (x+y)=x$

(13') $x+(\bar{x} \cdot y)=x+y$ (13") $x \cdot (\bar{x}+y)=x \cdot y$

(14') $\bar{x}+(x \cdot y)=\bar{x}+y$ (14") $\bar{x} \cdot (x+y)=\bar{x} \cdot y$

(15') $x+(y+z)=(x+y)+z$ (15") $x \cdot (y \cdot z)=(x \cdot y) \cdot z$

(16') $\overline{x+y}=\bar{x} \cdot \bar{y}$ (16") $\overline{x \cdot y}=\bar{x}+\bar{y}$

Jørgen Ebert, født 1946, cand. scient. i matematik fra Århus Universitet 1978, siden 1981 ansat på Amtsgymnasiet i Sønderborg.

Matematikserien består af:

NUMERISKE METODER

PLANGEOMETRI

KOMPLEKSE TAL

TRIGONOMETRI

MATEMATISK STATISTIK

POLYNOMIER

KEGLESNIT
af Jens Carstensen

EKSPONENTIAL FUNKTIONERNE OG DERES ANVENDELSE

PLANE KURVER
af Mogens Nørgaard Olesen

ENDELIGE GRAFER
af Hans Chr. Thomsen

MÆNGDELÆRE
af Jesper Frandsen

GEOMETRISKE KONSTRUKTIONER
af Gert M. Felnsberg

BOOLESK ALGEBRA
af Jørgen Ebert



-3