Logik, computere og kunstig intelligens

Af Lektor Thomas Bolander, Professor Jørgen Fischer Nilsson
og Lektor Jørgen Villadsen, DTU Informatik

Med anvendt matematik tænkes almindeligvis på udvikling af matematiske modeller, der direkte bringes i anvendelse uden for matematikken. Det kan f.eks. være opstilling af en matematisk model for vindforløbet omkring en vindmøllevinge med henblik på optimering af vingens evne til at omdanne vindenergi til elektrisk energi. Sådanne modeller vil ofte danne grundlag for beregning på computere med et stort antal beregningsoperationer og tilhørende numeriske (talberegningsmæssige) overvejelser over nøjagtighed af resultatet.

Dette kapitel handler ikke om *anvendt matematik* vedrørende fysiske systemer, men derimod om *anvendt logik* og specielt, hvordan logik udmøntes på computere især i form af såkaldte logik-programmeringssprog. Anvendt logik må her ikke forveksles med, at computerens kredse i sig selv virker efter logiske principper, hvori der regnes med to værdier 0 og 1 eller falsk og sand. Her beskæftiger vi os med, hvordan udsagn om et anvendelsesområde formuleret logisk kan behandles i computeren og tjene til at ræsonnere logisk om forhold i samme anvendelsesområde.

Hvad er logik?

Logik er læren om at ræsonnere i dagligdagen og i videnskaberne. Dermed anvendes logik både inden for praktiske fagområder såsom jura og medicin samt i abstrakte videnskaber som matematik. Logikkens arbejdshypotese er, at der findes generelle kerneprincipper for ræsonnering, der er fælles for alle fagområder. Det er logikkens hovedopgave at afdække og formulere disse principper for korrekt ræsonnering og specielt at udmønte dem i ræsonneringslove i form af slutningsregler. Dette er i parallel til de eksakte naturvidenskabers hovedopgave at afdække naturlovene og formulere disse præcist.

Dette er i lighed med matematikkens talbegreber og regneregler, der er universelt anvendelige og gyldige, uanset hvad vi tæller, når vi bruger matematikken. Principperne for ræsonnering og disses realisering på computer er hovedemnet i dette kapitel. Computeraspektet er så væsentligt, at vi kalder logikken for datalogisk logik eller blot "datalogik" (eng. computational logic). Matematisk logik er derimod specielt læren om at ræsonnere om matematiske beviser og i bredere forstand læren eller filosofien vedrørende matematikkens grundlag.

Lidt historie

Logikken har en lang forhistorie tilbage til de gamle græske filosoffer og til middelalderens øvelser i argumentation. Sidstnævnte kunne af og til omhandle meget verdensfjern, spidsfindig argumentation vedrørende eksempelvis, hvor mange engle der kunne danse på spidsen af en knappenål! I det 19. århundrede beskrev George Boole i "Laws of Thought", hvordan man kunne slutte på udsagn ved at knytte tal til dem og dermed regne på dem. I det 20. århundrede har der været en kamp mellem matematikere og logikere. Matematikere har forsøgt at gøre logikken til en gren af matematikken, mens logikere har forsøgt at reducere hele matematikken til anvendt logik. Denne langvarige videnskabelige strid har været meget frugtbar for videreudvikling af logikken (og sikkert også matematikken). Specielt har den ledt til afgørende resultater angående logikkens begrænsede muligheder for at underlægge sig matematikken og gøre matematik til logik.

Siden fremkomsten af computere er der kommet nye spillere på banen, nemlig softwareingeniører inden for feltet datalogi. Det har nemlig vist sig, at logikken danner det teoretiske grundlag for al maskinel beregning, dvs. alt det, der kan udføres på computer. Hvis man vil studere computervidenskab i dag, så skal man tilegne sig logik helt på linje med, at beherskelse af faget fysik forudsætter godt kendskab til matematik.

Logik er regning med udsagn

Hvilke udsagn? – er den naturlige reaktion på denne overskrift. Og svaret er i sin ubeskedne form alle udsagn, uanset hvad de handler om, blot de rummer en påstand. Logikken behandler udsagn, der kan manipuleres meningsfuldt uden at kende meningen med udsagnene. Logikken manipulerer symboler ved hjælp af slutningsregler (også kaldet inferensregler). Den vigtigste af disse kan skrives

$$\frac{p \leftarrow q \quad q}{p}$$

hvor der er to udsagn over stregen og et resulterende udsagn under stregen. Det første udsagn $p \leftarrow q$ læses: "p hvis q". Operatoren $\leftarrow$, som vi læser "hvis", kaldes omvendt implikation i logikken. Denne regel siger, at hvis vi har de to udsagn $p \leftarrow q$ og q, så slutter vi det nye udsagn p.

Eksempel: Lad der være givet "det er tirsdag i morgen", hvis "det er mandag i dag". Givet yderligere udsagnet "det er mandag i dag" slutter vi udsagnet "det er tirsdag i morgen" ved reglen ovenfor. Hvilke udsagn svarer til p og q? Bemærk, at denne slutning ikke afhænger af sandhedsværdien af udsagnene. Om det faktisk er mandag i dag er ligegyldigt for slutningens logiske korrekthed.

Lad os generalisere denne slutningsregel til

$$\frac{p \leftarrow q_1 \wedge q_2 \wedge \cdots \wedge q_n \quad q_1 \quad q_2 \quad \cdots \quad q_n}{p}$$

som vi også kan skrive

$$\frac{p \textbf{ hvis } q_1 \textbf{ og } q_2 \textbf{ og } \ldots \textbf{ og } q_n \quad q_1 \quad q_2 \quad \ldots \quad q_n}{p}$$

Den første slutningsregel fremkommer nu som specialtilfælde ved at sætte $n = 1$, kan vi bemærke. Vi putter altså $n + 1$ udsagn ind i slutningsregneregleen og får 1 udsagn ud. Det lyder som en dårlig forretning. Men vi har naturligvis stadig de givne udsagn til rådighed, så vi får 1 udsagn ekstra, som vi kan bruge igen.

Eksempel: X er i Y **hvis** X er i Z **og** Z er i Y

Dette kan i formel logisk symbolik skrives $\forall X \forall Y\,(i(X,Y) \leftarrow \exists Z(i(X,Z) \wedge i(Z,Y)))$, hvor vi benytter alkvantoren $\forall$ og eksistenskvantoren $\exists$, og hvor vi har indført prædikatet "i" for at være inden for noget.

Lad os supplere med udsagnene "København er i Danmark" og "Danmark er i Skandinavien". Disse kan skrives

i(København, Danmark)
i(Danmark, Skandinavien)

- to udsagn af en enkel atomisk form, der kaldes logiske fakta.

Vi kan nu bruge det først givne udsagn med X, Y og Z til i et trin at slutte, eller som det kaldes i logikken, at inferere eller deducere:

København er i Skandinavien

Vi behøver blot at sætte X=København og Z= Danmark og Y=Skandinavien. Hvis vi gør det, får vi jo udsagnet "København er i Skandinavien", **hvis** "København er i Danmark" **og** "Danmark er i Skandinavien". Og dette udsagn kan vi bruge i slutningsreglen med p = "København er i Skandinavien", q_1 = "København er i Danmark" og q_2 = "Danmark er i Skandinavien". Unægtelig ikke nogen dybsindig slutning, men kun en begyndelse til noget, der kan kræve et stort antal slutningstrin at udlede, som vi skal se eksempler på senere.
I logikken regner vi ikke meget på tal, men mere på data. I ovenstående udledning satte vi X=København osv. Disse datamanipulationer er trættende for mennesker at udføre, så meget desto mere at (de fleste) mennesker kan slutte sig til det rigtige i dagligdagens overvejelser uden at gennemføre logikkens omstændelige manipulationer med symboler. Sandt at sige har logikkens praktiske anvendelighed været stærkt begrænset indtil for nylig.

Logik og computerberegning

Omkring 1970 skete der et gennembrud i forståelse af logik til computerberegning. Man havde allerede da på det rent teoretiske og spekulative plan skabt forbindelse mellem logiske slutninger og computerberegning, men i 1970'erne indså man, at logik i tilpasset form kunne bruges som computerprogrammeringssprog. Faktisk nåede man en erkendelse, der er endnu mere fundamental og vidtrækkende: Al computerberegning er i princippet logisk ræsonnering.

Så længe man "blot" bruger computeren til talberegninger, hvad enten det er videnskabelige eller finansielle udregninger, er denne indsigt uden særlig betydning og værdi. Men når man f.eks. bruger computere til databaser, er det en væsentlig erkendelse, at svar på databaseforespørgsler (databaseopslag) kan forstås som logiske udledninger af konsekvenser ved at opfatte databasens indhold som logiske fakta. Forståelsen af computerberegninger som logiske ræsonneringer etablerer logikken som det teoretiske grundlag for computervidenskab.
Logikken afklarer spørgsmålet om, hvad der overhovedet kan beregnes på computere, og giver metoder for ræsonnering om egenskaberne for computerprogrammer. Dermed er logikken kommet til at spille en rolle for computervidenskab, som er lig den matematiske analyses grundlag for fysikken.

Logik og databaser

Lad os vende tilbage til geografieksemplet ovenfor og tilføje information om naboskab (grænser) mellem stater. Lande med fælles grænse kan formuleres som følger

nabo(Danmark, Tyskland)
nabo(Sverige, Norge)
nabo(Sverige, Finland)
nabo(Finland, Rusland) osv.

Disse logiske fakta udgør en database. Hvis vi vil udtrykke en forespørgsel, kan vi gøre det med omvendt implikationsudsagn som benyttet ovenfor, f.eks. "nabotilSverige(X) hvis nabo(Sverige, X)".

Her bør vi huske, at naboprædikatet er symmetrisk, dvs.

nabo(X,Y) hvis nabo(Y,X)

Vi kan nu udregne forespørgslen ved at efterlyse vædier for X. Formelt logisk sker det ved at bevise påstanden, at der findes et X så at "nabotilSverige(X)". Ved at bruge vores generelle slutningsregel fås svaret X=Norge sammen med X=Finland.
Vi kan også udtrykke, at to lande er indirekte naboer via et tredje land, f.eks. med

nabotilnabotil(X,Z) hvis nabo(X,Y) og nabo(Y,Z) og forskellige(X,Z).

Dette lille eksempel illustrerer, at databaser, specielt relationsdatabaser, og forespørgsler til disse kan forstås logisk, hvilket kan udtrykkes med et slogan:

Datalogik = logik + computerberegning

Der er således en direkte sammenhæng mellem logik og databasesystemer. Og databasesystemer danner grundlaget for alle internetsteder med søgeoperationer, alle finansielle transaktioner og alle tele- og it-systemer.

Logikprogrammering

For at introducere til logikprogrammering fortsætter vi med endnu et eksempel inden for geografien, nemlig farvelægning af landkort.
Vi vil starte med landene i Norden: Danmark, Finland, Island, Norge og Sverige. Vi benytter bogstaverne D, F, I, N og S herfor.

Idéen er at farvelægge et landkort over Norden, således at lande, der grænser direkte op til hinanden, får forskellige farver. I denne forbindelse grænser Danmark og Sverige ikke direkte op til hinanden, da der er vand imellem Danmark og Sverige. Vi kan således udtrykke kravet til farvelægningen på følgende måde:

korrekte-farver hvis F-farve ≠ N-farve og F-farve ≠ S-farve og N-farve ≠ S-farve.

Bemærk, at D-farve og I-farve ikke indgår.

Kravet er ovenfor udtrykt på en form for dansk under anvendelse af det matematiske symbol ≠ ("forskellig fra"). Vi kan i stedet udtrykke kravet i logikprogrammeringssproget Prolog (vi benytter her Prolog således som defineret af standardiseringsorganisationen ISO). Det er her hensigtsmæssigt at kode ≠ som prædikatet diff (eng. difference), og vi vil benytte engelske ord for farverne, da dette er nemmest i Prolog. Endvidere angives ordet "hvis" som :- og ordet "og" som komma.

I den første linje i logikprogrammet nedenfor udtrykkes, at kortet er farvelagt korrekt, hvis F og N har forskellig farve samt at F og S har forskellig farve samt yderligere at N og S har forskellig farve.

```
colorable :- diff(F,N), diff(F,S), diff(N,S).
diff(red,green).
diff(red,blue).
diff(green,red).
diff(green,blue).
diff(blue,red).
diff(blue,green).
```

Ovenfor er landebetegnelserne F, N og S nu variable (jvf. X, Y og Z ovenfor), der tillægges farveværdier ved udregning af logikprogrammet ved hjælp af vores slutningsregel. Prolog-systemet giver følgende svar:

Yes: colorable

Det er således tilstrækkeligt med 3 farver i eksemplet.

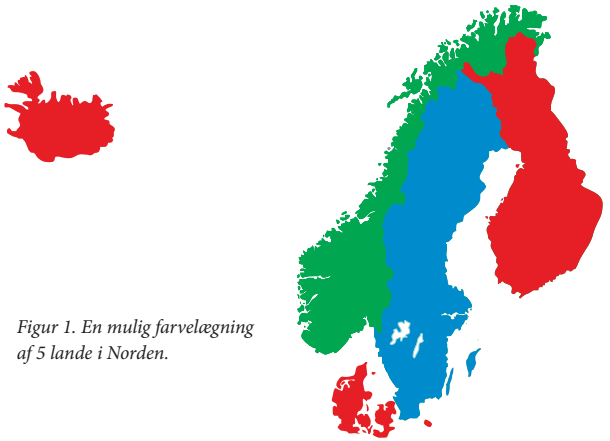
Det er også muligt at få computeren til at angive forslag til farverne på landene. Her er det dog nødvendigt at sikre sig, at også Danmark og Island får tildelt farver i løsningen. Dette gøres nemmest ved, at variablen D begrænses til en af de 3 farver (kravet color(D) i det følgende program), og tilsvarende for de resterende variable F, I, N og S.

```
solution(D,F,I,N,S) :-
  color(D), color(F), color(I), color(N), color(S),
  diff(F,N), diff(F,S), diff(N,S).
diff(red,green).
diff(red,blue).
diff(green,red).
diff(green,blue).
diff(blue,red).
diff(blue,green).
color(red).
color(green).
color(blue).
```


Løsningerne fås i lighed med databaseeksemplerne i forrige afsnit ved hjælp af den tidligere givne slutningsregel. Her er den første af i alt 54 løsninger:

Yes: solution(red,red,red,green,blue)

Her er Danmark (D), Finland (F) og Island (I) tildelt farven rød, Norge (N) farven grøn og endelig Sverige (S) farven blå. Se figur 1.



Figur 1. En mulig farvelægning af 5 lande i Norden.

De resterende 53 løsninger udregnes og udskrives af Prolog på forlangende. Eksemplet med landene i Norden er ikke stort. Programmet kan udmærket udvides med flere lande, men det bliver hurtigt uoverskueligt at holde styr på de mange krav. Desuden er det besværligt at ændre, hvilke farver der ønskes benyttet. Det ville være nemmere blot et enkelt sted at angive listen af farver [red,green,blue]. Dette kan naturligvis også gøres i Prolog.

Logikprogrammering, især programmeringssproget Prolog, er et nyttigt værktøj til at løse matematik- og logikproblemer, da man skriver programmer ved at opstille en sammenhæng af logiske udsagn. Det er imidlertid også muligt at bruge Prolog til at udvikle store computersystemer, f.eks. til spil, databaser og internetapplikationer. En stor fordel ved logikprogrammering er, at anvendelsen af logik medfører et højere abstraktionsniveau, så avancerede programmer kan udvikles hurtigere og med et mere pålideligt resultat.

Logik og modeller

Vi benytter matematiske formler til at beskrive eksempelvis planetbevægelser, spredning af lydbølger og elektriske kredsløbs opførsel. Logiske formler er også et beskrivelsesværktøj, ligesom andre matematiske formler er det. Vi kan opfatte en mængde af logiske udsagn – det vi ovenfor kaldte en database – som en model af et stykke af verden. I eksemplet med farvning af landene havde vi en mængde af udsagn, som beskrev landenes indbyrdes beliggenhed. Denne mængde af udsagn giver en logisk beskrivelse af et stykke af vores virkelighed – og kan opfattes som en model af denne virkelighed.

Figur 2. Lille bord med bøger og mandarin.
(foto: Thomas Bolander)

Et andet eksempel på modellering af virkeligheden ved logiske formler er følgende. Betragt billedet i figur 2.

Dette lille stykke af verden kan beskrives ved følgende logiske udsagn:

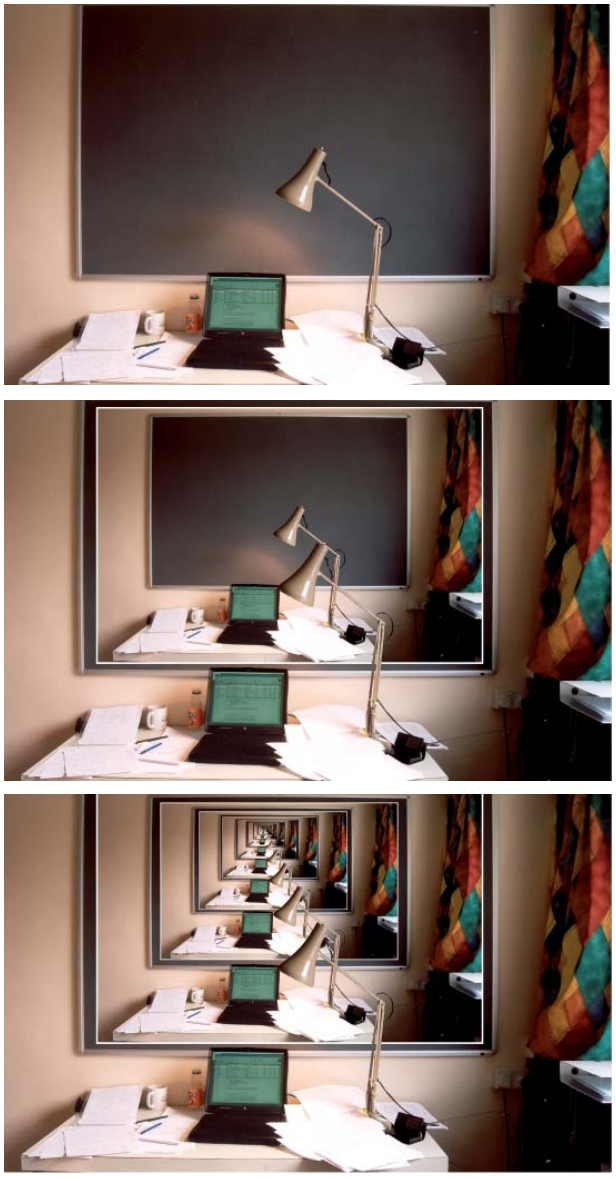
på(bord, store bog)
på(store bog, lille bog)
på(lille bog, mandarin).

Her bruger vi udsagn af typen på(X,Y) til at betegne, at Y ligger direkte oven på X. Sammenlign disse logiske udsagn med naboudsagnene i afsnittet om logik og databaser. Strukturen er naturligvis den samme, vi benytter blot i stedet udsagnene til at repræsentere den verden, som består af objekterne på det lille bord (bøger og en mandarin) samt relationerne mellem disse objekter (deres placering i forhold til hinanden).

Det logiske sprog kan altså lige så vel bruges til at beskrive relationerne mellem objekterne på bordet som naborelationer mellem lande. De to eksempler er forholdsvis simple og relativt ens. De antyder dog alligevel, at vi kan bruge mængder af udsagn til at beskrive en lang række forskelligartede egenskaber ved verden. Disse mængder af logiske udsagn kan opfattes som modeller af de pågældende dele af verden. Eksempelvis udgør de 3 udsagn ovenfor en simpel matematisk model af den verden, som er repræsenteret i figur 2.

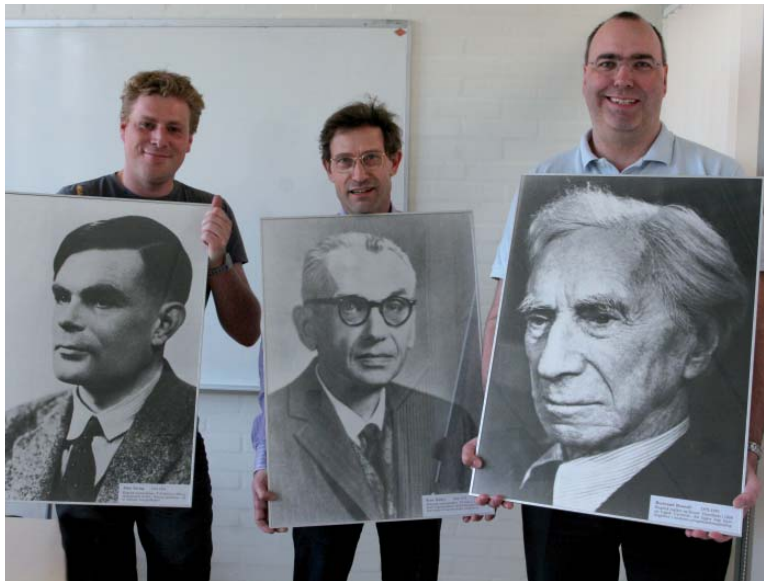
Logikkens grænser

Hvor meget af verden kan vi beskrive med logiske udsagn? Er det en særlig begrænset del af verden, særlige typer af egenskaber ved verden, eller er der måske ingen grænser? Man kunne starte med at være optimistisk og forestille sig, at der ingen grænser er, så hele verden kan beskrives med logiske udsagn. Lad os prøve at lege lidt med dette tankeeksperiment. Vi forestiller os altså, at vi kan skabe en mængde af logiske udsagn, som giver en model af hele verden. Lad os kalde denne mængde af logiske udsagn for L. L er naturligvis selv en del af verden, så hvis L er en model af hele verden, må en del af L altså udgøre en model af L selv. L er således i en vis forstand indeholdt i sig selv. Dette er naturligvis en ret abstrakt tanke, så lad os prøve at starte med at give en lidt simplere analogi.



Figur 3, 4 og 5.
(foto: Thomas Bolander)

Figur 3 forestiller et kollegieværelse. Lad os antage, at vi ønskede at friske værelset lidt op med et stort billede på den grå opslagstavle. Hvis vi var lidt fantasiløse (eller måske netop ikke), kunne vi få den tanke at ophænge et billede, som forestillede kollegieværelset selv. Vi kan tænke på et sådan billede af værelset som en model af værelset. I første omgang ville vi måske tro, at det var nok at tage et foto af kollegieværelset og hænge det op på opslagstavlen. Resultatet af at gøre dette er illustreret i figur 4. Men billedet på opslagstavlen er faktisk ikke et billede af kollegieværelset selv, det er kun et billede af kollegieværelset, som det så ud, før billedet blev hængt op. Det, der er forkert, er, at billedet på opslagstavlen indeholder en tom opslagstavle – ikke en opslagstavle med et billede på. For at få et billede, som korrekt repræsenterer hele kollegieværelset – inklusive billedet selv – bliver vi nødt til at udfylde det grå område med endnu en kopi af værelset. Og denne proces bliver vi nødt til at fortsætte, indtil vi til sidst ender med situationen i figur 5. På figuren har vi uendeligt mange kopier af værelset indlejret i hinanden.



Artiklens forfattere med fotos af berømte matematikere og logiske tænkere. Fra venstre lektor Thomas Bolander med et foto af den engelske matematiker Alan Turing (1912-1954), professor Jørgen Fischer Nilsson med et foto af den østrigske matematiker Kurt Gödel (1906-1978) og lektor Jørgen Villadsen med et foto af den engelske logiker og filosof Bertrand Russell (1872-1970).

Løgner-paradokset

Samme overraskende fænomen opstår, når vi igennem logiske udsagn forsøger at skabe en model af en verden, som indeholder modellen selv, altså hvis vi eksempelvis forsøger at skabe en model af *hele* verden. Det er dog ikke nødvendigvis et stort problem i sig selv. En uendelig række af modeller indlejret i hinanden kunne man vel leve med, men det bliver faktisk værre endnu. Hvis en logisk model indeholder en model af sig selv, giver man den mulighed for at tale om sig selv, dvs. udtrykke egenskaber ved sig selv. Det leder til det, man kalder selvreference, som er notorisk problematisk. Et af de simpleste eksempler opstår i det såkaldte løgner-paradoks. Løgner-paradokset opstår ved at betragte følgende selvrefererende sætning:

> "Denne sætning er falsk".

Problemet er, at uafhængigt af om vi antager, at sætningen er sand eller falsk, ledes vi til en modstrid. Antag først, at sætningen er sand. Hvis den er sand, er det, den siger, korrekt. Men sætningen siger, at den er falsk, altså må den så være falsk. Men hvis vi antager, den er falsk, kan det, den udtrykker, ikke være sandt. Det vil sige, det er ikke sandt, at den er falsk. Og så må den altså alligevel være sand. Men det kunne den jo ikke, for hvis vi antog, at den var sand, fandt vi ud af, at den måtte være falsk.

Vi har ingen chance for at undslippe en sådan modstrid. Situationen kaldes derfor et paradoks. Det specielle ved løgner-paradokset er, at det er bygget på en sætning, som refererer til sig selv, nemlig sætningen "Denne sætning er falsk". Når logiske systemer modellerer sig selv, får de en tilsvarende mulighed for at referere til sig selv, og dermed løber de ind i problemer svarende til løgner-paradokset.

Logik og naturlige tal

Nu behøver man måske ikke tage tingene alt for tungt. Hvor tit vil vi overhovedet være interesseret i at skabe logiske modeller, som kan modellere sig selv? Umiddelbart ville man måske tro, at dette skulle være meget sjældent, men det viser sig at optræde ganske ofte. En af de ting, man kan bruge logiske udsagn til at beskrive, er aritmetik, dvs. læren om egenskaber ved de naturlige tal (0, 1, 2, 3 osv.). Man kan altså skabe en model af de naturlige tal og deres egenskaber igennem en række logiske udsagn. Blandt de logiske udsagn, som er nødvendige for at beskrive egenskaberne ved de naturlige tal, finder man bl.a. følgende:

$$0 = 0$$
$$X + 0 = X$$
$$X + 1 \neq 0$$
$$X = Y \text{ hvis } X + 1 = Y + 1$$

Her betyder eksempelvis X + 0 = X, at ethvert tal adderet til 0 giver tallet selv. Umiddelbart skulle man ikke tro, at der kunne opstå problemer ved at give en logisk beskrivelse af de naturlige tal. Vi forsøger at skabe en model af de naturlige tal, men denne model indeholder vel ikke en model af sig selv, eftersom naturlige tal ikke i sig selv er logiske udsagn? Det problem, der imidlertid opstår, er, at vi faktisk kan bruge tal til at sige noget om logiske udsagn. Helt simpelt kunne vi f.eks. vælge at opstille alle logiske udsagn på en række, og hvert logisk udsagn ville da få tilknyttet et tal svarende til, hvilket nummer

det havde i rækken. Dermed kunne vi opfatte et symbol '5' som ikke kun refererende til tallet 5, men også til et udsagn, nemlig det 5. udsagn i rækken. Pludselig vil man så kunne benytte logiske udsagn indeholdende tal til at snakke om andre logiske udsagn, idet man kan opfatte tallene som refererende til andre udsagn. Og i sidste instans kan man så også få udsagn til at referere til sig selv, hvormed vi havner i problemerne omkring selvreference og paradokser nævnt ovenfor.

Disse tanker og idéer går tilbage til en af det 20. århundredes største matematikere, Kurt Gödel. Han beviste, at enhver logisk beskrivelse, som indeholder aritmetik, vil være ufuldstændig, altså kun give en ufuldstændig model af det, den forsøger at modellere. Måden, han beviser det på, er at vise, at hvis den logiske beskrivelse havde været fuldstændig, da ville modellen indeholde en model af sig selv og dermed give anledning til, at man kunne formulere et selvrefererende paradoks i stil med løgner-paradokset. Givet eksemplet med kollegieværelset ovenfor er det måske forventeligt, at det kan være problematisk at have modeller, som indeholder modeller af sig selv. Mere overraskende er, at der ikke skal mere end en logisk beskrivelse af de naturlige tal til, før man havner i denne problemstilling. Kurt Gödels resultat er kendt som Gödels Ufuldstændighedssætning, og det er en af de mest berømte matematiske sætninger fra det 20. århundrede.

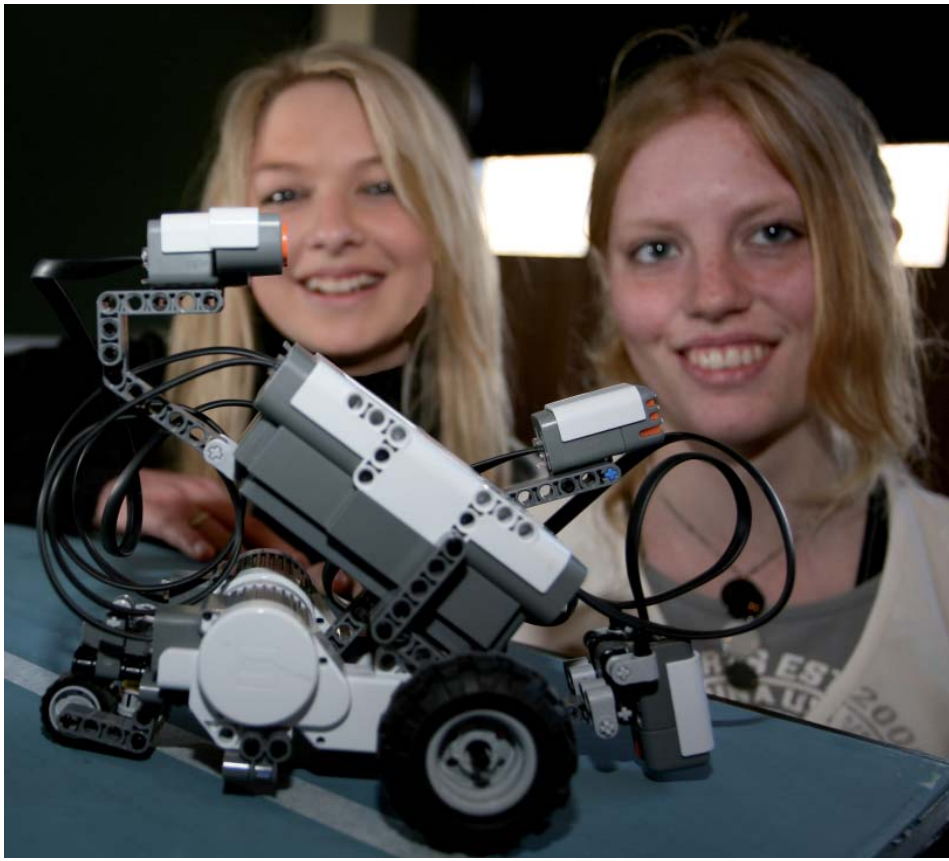
Logik og kunstig intelligens

Tankegangen med at skabe logiske modeller af verden er også central i kunstig intelligens. I kunstig intelligens ønsker man at skabe computerprogrammer eller robotter, som kan tænke selv, lægge planer, træffe beslutninger, handle på egen hånd, lære af sine fejl, kommunikere med andre osv. Her spiller "modelbygning" en central rolle. Vi mennesker har indbygget en "model" af verden, som vi bruger som grundlag for at beslutte, hvilke handlinger vi vil foretage os. Vi bruger vores model af verden til at forudsige konsekvensen af forskellige mulige handlinger og vælger herudfra, hvilke handlinger vi vil foretage. Eksempelvis ville de færreste mennesker vælge at slippe en 1.000-kroners seddel ud over kanten af Rundetårn. Vores model af verden fortæller os, at sedlen næppe vil blive hængende i luften, så vi bare kan tage den ind igen, når vi har lyst. Modellen fortæller os tværtimod, at sedlen på grund af en kombination af tyngdekraft og vind givetvis vil forsvinde for altid. Modellen af den fysiske verden, som omgiver os, tillader os at agere intelligent i verden. Uden denne model ville vi ikke have nogen anelse om, hvad konsekvensen af forskellige handlinger ville være, og vi ville dermed ikke kunne tage de rigtige beslutninger. At vi ikke forsøger at gå ind i et rum ved at passere direkte gennem en væg er f.eks. også en konsekvens af, at vi har den indbyggede model af verden. Mennesker vil i stedet lede efter en dør, som vi kan åbne.

Vores model af verden fortæller os også, at en normal dør enten kan åbne udad eller indad, så vi vil formodentlig prøve begge dele, inden vi giver op. En person, som kun forsøger at åbne døren den ene vej og med det samme giver op, hvis det ikke lykkes, virker naturligvis ikke særlig intelligent. Tilsvarende virker det heller ikke særlig intelligent at forsøge at komme ind i et rum ved at passere direkte gennem væggen. Altså er vores model af verden en central komponent i intelligent ageren.

Robotters intelligens

Givet at modeller således er centrale for den menneskelige intelligens, opstår der naturligt følgende spørgsmål: Hvordan kan vi udstyre robotter med samme evne til at opbygge og udnytte modeller af verden? Her kommer logikken og logiske udsagn igen ind i billedet. Vi har set, at logiske udsagn er meget velegnede til at danne beskrivelser (modeller) af dele af verden. Det ville derfor være oplagt at tænke på, om man ikke kunne udstyre robotter med en evne til at håndtere logiske udsagn med henblik på at kunne beskrive og ræsonnere omkring verden. Og det kan man. Netop denne tilgang



Gymnasieelever på besøg på Danmarks Tekniske Universitet. De to piger ses med en robot fra iMARS-laboratoriet på DTU Informatik. Her eksperimenteres med systemer af robotter, som samarbejder om at løse komplicerede opgaver.

til kunstig intelligens udgør den ene af de to store paradigmer (retninger) inden for kunstig intelligens, det såkaldte symbolske paradigme. Idéen er her, at en robot eksempelvis vil kunne repræsentere verden bestående af det lille bord (figur 2) ved hjælp af de logiske udsagn, vi introducerede ovenfor. Hvis robotten ønskede at få fat i den store bog, ville den da ved hjælp af de logiske udsagn kunne ræsonnere sig frem til, at den blev nødt til at fjerne mandarinen og den lille bog først. Man kan også uden videre give logiske beskrivelser, som fortæller, at man ikke kan passere gennem vægge, og at en dør kan åbne enten indad eller udad. Hvis en robot har tilstrækkeligt mange logiske udsagn, som fortæller, hvordan verden hænger sammen, vil den da via ræsonnering med disse logiske udsagn kunne udvise intelligent opførsel. Den vil også kunne lære nye ting om verden og repræsentere disse ting ved nye logiske udsagn.

Det viser sig hurtigt, at intelligent opførsel i den virkelige verden ikke kun kræver en evne til at modellere "døde ting" som mandariner, bøger, vægge og døre. For at vi mennesker kan agere intelligent i forhold til hinanden, kommunikere og samarbejde, bliver vi nødt til også at have dannet mentale modeller af hinanden. Manglende evne til at sætte sig ind i andres situation og forudsige andres tanker og handlinger karakteriserer bl.a. autister, og denne tilstand er naturligvis stærkt hæmmende, når man skal opnå relationer til andre mennesker.

Når man designer kunstig intelligens-systemer og gerne vil opnå så høj grad af intelligent opførsel som muligt, skal man selvfølgelig tage disse ting i betragtning. Hvis en robot skal kunne interagere fornuftigt med mennesker eller andre robotter, bliver den nødt til også at være i stand til at skabe delvise modeller af disse andre mennesker eller robotter. Der arbejdes i dag ihærdigt på at skabe robotter, som via logiske systemer kan modellere ikke kun de "døde ting", men også andre handlende væsener og deres tanker. Der er imidlertid givetvis meget lang vej igen, inden vi opnår noget, der på nogen som helst måde kan måle sig med menneskelig intelligens.

Artiklens forfattere

Lektor Thomas Bolander

Professor Jørgen Fischer Nilsson

Lektor Jørgen Villadsen