

Human IT

Tidskrift för studier av IT
ur ett humanvetenskapligt perspektiv

Alan Turing og informationsteori

av [Ellen Bonnevie](#)

This paper presents a discussion of Alan Turing's ideas on computational methods for solving Hilbert's "Entscheidungsproblem". The computational process is demonstrated by the Turing Machine which is a theoretical machine, a description of how a mathematical problem can be solved by a mechanical process. The Turing Machine is a universal machine as it can take the set of rules of other machines as input and perform the same computations as these. In this way Turing demonstrates some principles behind the symbolic problem solving process. The same ideas have influenced Niels Ole Finnemann whose ideas on symbol theory are introduced. Touched upon is also Alan Turing's influence on artificial intelligence which concerns "the thinking machine" and the nature of intelligence.

The relevance of Turing's ideas for library and information science is discussed from the perspective of the library school curriculum and the scientific mapping of library and information science.

Innehåll

[1. Turingmaskinen og afgørighedsproblemet](#)

[2. Turingmaskinen og Finnemanns symbolteoretiske betragtninger](#)

[3. »Den tænkende maskine«](#)

[4. Konkluderende bemærkninger](#)

[Om författaren](#)

(One of the) Ten Biggest Thinkers of the 20th Century:

Alan Turing

Invented the Turing machine and the Turing test – who

else has two of the most important ideas of computer science named after them? (Jakob Nielsen)¹

...you (can) only really understand something when you can build it. (Justin Leiber)²

Alan Turing er ikke et navn der har en fremtrædende plads på en biblioteksskoles curriculum eller på et datalogisk institut eller en IT-højskole. Det skyldes formodentlig at Turing som en af pionererne inden for computer science identificeres med en computerorienteret tilgang til informationsteorien og med begrebet kunstig intelligens. Det hænder dog at man støder ind i navnet, f.eks. i forbindelse med de bibliometriske love³ eller i forbindelse med diskussion af informationsbegrebet eller videnskabsteoretiske diskussioner generelt inden for biblioteks- og informationsvidenskab.⁴ Turing findes ikke som selvstændigt opslagsord i Birger Hjørlands leksikon *Informationsvidenskabelige grundbegreber*,⁵ men er nævnt under »Datalogi, teoretisk«. I *International Encyclopedia of Information and Library Science* er en kort omtale.⁶ Her krediteres Turing især for at have udviklet ideen om turingmaskinen som på idéplanet er i stand til at modellere den proces der ligger bag menneskelig beregning og som danner basis for computerens beregningsmetoder.

Alan Turing (1912–1954) var professor i matematik ved Cambridge Universitetet, og er især kendt for to banebrydende artikler: Den første af disse er »*On Computable Numbers, with an Application to the 'Entscheidungsproblem'*«⁷ fra 1936, hvori han fremsætter de matematiske teorier som førte frem til opbygningen af datamaskinen. Den udkom 10 år før den første amerikanske computer blev bygget, men Turing viste her i princippet hvordan en computer fungerer ved demonstrationen af turingmaskinen som er en ideel datamaskine. Turing brugte selv udtrykket »a computing machine«. Ordet »computer« alene betyder i Turings terminologi et menneske der udfører en beregning. Det er i denne artikel at hans indflydelse især gør sig, og »On Computable Numbers« er den mest vægtige af Turings værker. Den kom som et svar på Hilberts afgørelsesproblem, Hilberts 3. spørgsmål.

En anden væsentlig indflydelse fik han p.g.a. de tanker han fremførte i 1950 i sin artikel »*Computing Machinery and Intelligence*«.⁸ Denne artikel indvarsler diskussionerne om »den tænkende maskine« og er af mere kognitionsvidenskabelig karakter.

Uden at ville argumentere for at de to nævnte værker bør indgå i en biblioteksskoles pensum og curriculum, vil det være nyttigt overordnet set at kende til de tanker der ligger bag konstruktionen af turingmaskinen, uanset at vi inden for biblioteksverdenen i højere grad beskæftiger os med anvendt computer science.

([Åter](#) till början av artikeln)

1. Turingmaskinen og afgørelsesproblemet

Inden turingmaskinen beskrives, må vi tilbage til begyndelsen af århundredet for at finde det som Niels Ole Finnemann⁹ kalder »det symbolteoretiske spor«. Hermed mener han dels udviklingen af en matematisk logik, dels tegnteorien som blev udviklet af Saussure

og Charles S. Peirce omtrent på samme tid. Den matematiske logik blev udformet bl.a. af matematikerne Hilbert, Frege og Russell. De forsøgte at give matematikken et logisk fundament med formuleringen af et nyt formelt symbolsprog. Kort fortalt mente Frege og Russell, at matematik er logik, mens Hilbert mente at man kunne behandle matematiske problemer med en meta-matematisk logik.

Disse teorier er nødvendige forudsætninger for Turing, idet de indebar en axiomatisering af matematikken. Det vil sige, at man med nogle axiomer (grundsætninger) ved hjælp af deduktioner kunne bygge hele det matematiske system op.

Også algoritmisering er en væsentlig forudsætning for turingmaskinen. Ordet algoritme henføres sædvanligvis til Leibnitz (1646–1716). Hans definition af algoritme lyder således:

*En algoritme er en endelig mængde af regler, der tillader en fuldstændig mekanisk udførelse af en proces med symboler.*¹⁰

I forbindelse med en datamaskine forstås en algoritme som »regler, der fortæller, hvorledes de korrekte inddata transformeres til korrekte ud-data«¹¹ Algoritmen determinerer på forhånd hvad der skal ske uden at resultatet nødvendigvis er kendt på forhånd. Det fungerer på den måde at man analyserer den proces man ønsker at udføre, oversætter den til symboler som kan forstås af en maskine, og bestemmer de regler hvorefter symbolerne kan manipuleres. Hvis vi får nogle resultater fra processen i form af symboler, har der fundet en algoritmisk informationsbehandling sted.

Hilberts afgørighedsproblem gik ud på spørgsmålet om hvorvidt der eksisterer en generel, finit metode til at afgøre, om en vilkårlig matematisk sætning kan bevises eller ej. David Hilbert fremlagde sine bevisteorier på en kongres i 1930 idet han mente at have bevisliggjort, at det var muligt at afgøre spørgsmålet om hvorvidt matematikken kan betragtes som afgørlig eller bevisbar. Han besvarede afgørighedsproblemet positivt med udtalelsen om, at »ein unlösbares Problem Überhaupt nicht gibt«. Dette var der andre der satte spørgsmålstegn ved. På samme kongres fremsatte Kurt Gödel sit ufuldstændighedsteorem som siger, at et formelt system S indeholder en logisk afledt formel A, sådan at hverken A eller dets negation $\neg A$ kan bevises inden for systemets rammer. Han viste, at »ethvert kendt system var ufuldstændigt, fordi det måtte indeholde mindst én sætning, der ikke kunne bevises inden for systemets rammer.«¹² Gödels bevisførelse er kompliceret, men det væsentligste er at metoden er algoritmisk funderet.

Turing satte sig for at forsøge at løse Hilberts afgørighedsproblem ved at forestille sig en maskine som kunne løse matematiske spørgsmål, og som var i stand til at besvare spørgsmålet om hvorvidt en hvilket som helst matematisk sætning den blev præsenteret for kunne bevises. Turings løsning fremkom altså »by a mechanical process« og blev demonstreret med turingmaskinen. Det vil sige at Turing med sin artikel i 1936 kunne demonstrere afgørighedsproblemet ved hjælp af en mekanisk procedure. Svaret var han kommet til via ideen om »computable numbers«. Turingmaskinen kunne beregne et hvilket som helst tal, også reelle tal hvis de var defineret af en bestemt regel som f.eks. beregningen af » π «. Ganske vist er resultatet et uendeligt decimaltal, men det kunne på

et hvilket som helst tidspunkt repræsenteres ved en finit tabel. Det infinite decimaltal kunne således repræsenteres i en finit liste, der kunne foretages beregningerne med dem, osv.

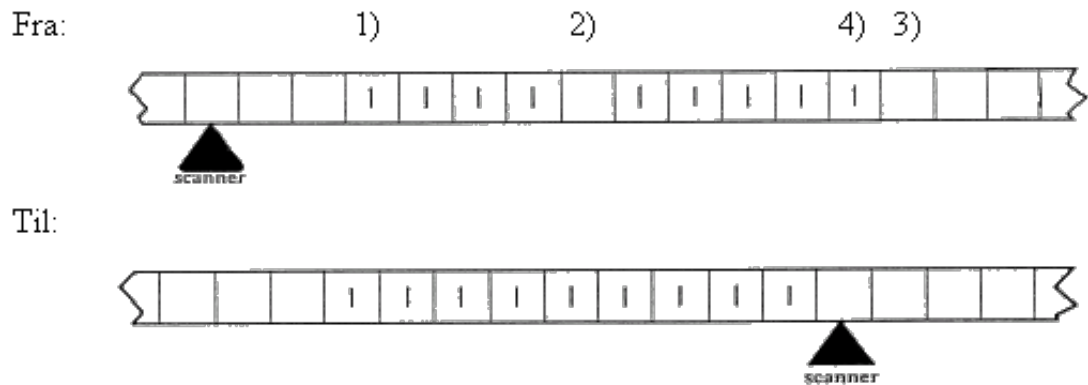
Turing var her inspireret af matematikeren Cantor og hans diagonalbevis med idéen om hvordan rationelle tal, i hans eksempel brøker mellem 0 og 1 udtrykt i decimaltal, kan give anledning til at demonstrere eksistensen af irrationelle tal ved at stille de rationelle tal op i en liste og herefter gå diagonalt ned gennem listen og ændre de pågældende tal for eksempel ved at forøge dem med 1. De fremkomne tal er ikke rationelle, da de adskiller sig fra tallene på listen med ét ciffer. Dette »trick« udnyttede Turing til at vise hvordan det beregnelige (the computable) kan give anledning til at demonstrere det ikke beregnelige. Cantors bevis var et eksempel på en operation som ikke kan mekaniseres eftersom den krævede, at man på forhånd vidste om maskinen ville gå i en uendelig løkke. Der var ingen måde hvorpå man kunne få en garanti for at et diagonaltal ville blive produceret inde for en overskuelig tid. Det uberegnelige diagonaltal kunne ikke beregnes på Turings maskine. Det var et uløseligt problem, han havde fundet. Han kan således afvise Hilberts svar på afgørlighedsproblemet. Turings svar er »nej«. Der kan ikke findes en metode til at afgøre spørgsmålet om hvorvidt man kan løse et vilkårligt, matematisk eller formelt logisk problem med et finit antal operationer. Et formelt system kan ikke være både fuldstændigt og modsigelsesfrit hvilket kan illustreres med »stopproblemet«. En turingmaskine kan ikke afgøre om den vil stoppe, givet et bestemt datasæt. Hvis den tager sin egen beskrivelse som input, vil den gå i en uendelig løkke; der opstår en kontradiktion.¹³ Church var kort forinden kommet med et lignende svar, og derfor kaldes det ofte for *Church/Turing tesen*.

Turings maskine er en teoretisk konstruktion som med enkle operationer kan beregne matematiske problemer, i princippet alle tænkelige matematiske konstruktioner også meget komplicerede, ved hjælp af et modularitetsprincip. Med disse instruktioner kan alle tænkelige algoritmer udføres. – »Alle matematiske sætninger bevises, al formel logisk bevisførelse kan simuleres, alle former for symbolmanipulation udføres.«¹⁴

Turingmaskinen – den abstrakte mekaniske processor – blev konstrueret ved at analysere og opløse den menneskelige regneproces i delelementer og ved at analysere en regulær mekanisk maskine: skrivemaskinen. Skrivemaskinen er bl.a. karakteriseret ved at den kan have 2 forskellige konfigurationer: »uppercase« og »lowercase«, og ved at typerne er variable i forhold til papiret. Det var dette forhold Turing abstraherede ved at forestille sig en maskine »som på et hvilket som helst tidspunkt ville være i én ud af et finit antal af mulige konfigurationer.«¹⁵ Det er her løsenet til turingmaskinen ligger. På et bestemt tidspunkt kan maskinen karakteriseres ved at være i en bestemt, determineret tilstand. Det er på dette beskrivelsesniveau, man kan tale om determinisme, afgørlighed, »the quality of being fixed in advance«, som Turings biograf Andrew Hodges udtrykker det. Det er denne egenskab som Turing abstraherer og relaterer til symbolmanipulation. Selvom hans maskine er finit defineret, vil den have ubegrænset adgang til papir at arbejde på. Det forestillede Turing sig i form at et bånd, inddelt i felter, hvor maskinen kunne læse (Turing brugte ordet scanne), viske ud, og skrive symboler. Maskinen kunne kun bevæge sig til enten højre eller venstre. Den arbejdede automatisk i følge dette begrænsede sæt af ordrer som stod på båndet. Der var ikke tale om nogen menneskelig indgriben, selvom Turing dog nævnedes, at man kunne tænke sig »valgmaskiner«, hvor mennesker greb ind i processen. Ordrene

kunne opstilles i form af en »adfærdstabel« af begrænset, finit størrelse, udtrykt ved de mulige konfigurationer. Disse ordrer kan betragtes som en del af maskinen.

En additionsmaskine som lægger 4 og 5 sammen ser således ud:¹⁶



Skanneren bevæger sig et trin ad gangen efter sin instruks med i alt 4 konfigurationer. 1) Når maskinen møder et tegn skal den bevæge sig til højre. 2) Når den møder et blankt felt, skriver den et tegn. 3) Herefter går maskinen i en tredje konfiguration, hvorefter den bevæger sig til højre så længe der er et tegn, men når den møder et blankt felt skal den gå til venstre. 4) I den fjerde konfiguration sletter maskinen tegnet. Slut. Udgangspunktet ses som den øverste strimmel. Resultatet af proceduren ses som den nederste strimmel.

Hvis maskinen skal beregne » π « som er et uendeligt decimaltal, følger den blot de gældende regneregler, hvorefter den tilføjer decimal for decimal i en række finite operationer. Båndet den arbejder på er uendeligt, og processen kan gå videre i det uendelige, men beregningen af de enkelte decimal kan udtrykkes i en finit tabel, en adfærdstabel.

Turings adfærdstabeller er todimensionale, men Turing forestillede sig at hver tabel kunne få et beskrivelsesnummer som maskinen kunne kodes med – et nummer eller symbol som kaldte den pågældende tabel. Det er således man skal forstå det Turing beskriver som *universalmaskinen*: En bestemt turingmaskine kan simulere en hvilket som helst anden maskine. Det foregik ganske enkelt ved at maskinen kunne kodes til at læse beskrivelsesnummeret – symbolet – afkode det til en tabel, udføre den pågældende operation etc.

I forestillingen om den universelle computer ligger ikke blot det forhold at et hvilket som helst matematisk problem kan udføres af maskinen, men også at det med passende programmering kan gøres på én og samme maskine, idet universalmaskinen kan simulere en hvilket som helst anden turingmaskine.¹⁷ En anden generel observation er, at en algoritme implementeret i én computer kan implementeres i en hvilken som helst anden computer.¹⁸

Når man taler om universalmaskiner, er det vigtigt at gøre opmærksom på at det ikke betyder at datamaskiner kan gøre alt. Der er spørgsmål en datamaskine ikke kan besvare; svar der ikke kan produceres af nogen effektiv procedure – og her tænkes ikke alene på afgørlighedsproblemet. Det er Joseph Weizenbaum,¹⁹ en af kritikerne af den rationalistiske tankegang, som har denne indvending og følger tankegangen op ved

at gøre opmærksom på at sandhed ikke er ækvivalent med formel bevislighed. Her hentyder han til »den levende sandhed«, alt det der ikke kan siges med ord. Det er den form for sandhed (og heraf afledt intelligens) som en datamaskine ikke kan simulere.

([Åter](#) till början av artikeln)

2. Turingmaskinen og Finnemanns symbolteoretiske betragtninger

Turingmaskinen er traditionelt defineret ud fra et matematisk/proceduralt perspektiv. Niels Ole Finnemann gør op med dette og indfører en symbolteoretisk betragtningsmåde. Dermed introducerer han en ny tværvideenskabelig synsvinkel, her specifikt på turingmaskinen, som han beskæftiger sig med i grænsfeltet mellem naturvidenskab og humanisme, som det i øvrigt også fremgår af titlen på hans bog *Tanke, sprog, maskine*.²⁰

Når vi ser på turingmaskinen er det vigtigt at understrege, at den ikke er en maskine i fysisk forstand. Det er en teoretisk konstruktion udviklet med det specifikke formål at vise, hvordan man kan løse matematiske problemer mekanisk. Det er dette forhold Finnemann griber fat i og gør opmærksom på, at turingmaskinen er en *dedikeret* maskine, hvor der er et dybere niveau – den *fysiske* dimension.

Turings formål med maskinen var at forsøge at løse afgørighedsproblemet, og han anser maskinens fysiske egenskaber for om ikke irrelevante, så knap så relevante. Finnemann analyserer, hvad der sker i den »universelle computer« og giver en ny tolkning der ligger udover den Turing selv foretog. Han »læser« Turing på en ny måde og gør opmærksom på, at den fysiske dimension har en afgørende indflydelse på forståelsen af hvad der foregår i maskinen. Den kan ikke køre uden at være dedikeret et eller andet formål, idet programmet er en del af maskinen, men fundamentalt set er det en maskine, som kører – ikke på benzin men på diskrete binære elementer med en primitiv semantik. Finnemanns nytænkning ligger i at han viser hvad der foregår i turingmaskinen på det grundlæggende niveau, hvor det fysiske og det symbolske interagerer på en hidtil uset måde. Det er her den universelle egenskab ligger.

Vigtigheden for denne tolkning ligger i det Finnemann kalder *det informationelle notationssystem* – det er dette notationssystem som gør, at computeren kan anvendes på en mangfoldighed af måder, betinget af et nyt forhold mellem fysisk/mekaniske og symbolske procedurer. Dette beskriver han som den symbolske opløsning af den mekaniske determination.

Den klassiske naturteori er deterministisk. Der er nogle generelle love, der virker i hele systemet, og ethvert trin i processen er bestemt af begyndelsesbetingelserne. I turingmaskinen er det kun de enkelte trin i processen, der er deterministiske.

Turingmaskinen fungerer anderledes end klassiske maskiner, betinget af strimlens dobbeltfunktion som dels en del af maskinen og det materiale maskinen bearbejder. Dvs. at: 1) *grænsen mellem maskine og materiale opløses*. Det er den ene forskel mellem klassiske maskiner og turingmaskinen. Den anden er, at 2) *den arbejder skridt for skridt*. Her spiller kombinationen af finite og infinite procedurer ind. Den infinite

er repræsenteret af den uendelige strimmel med symboler, som maskinen fodres med. Maskinen derimod er beskrevet således, at den arbejder skridtvis, i en række af finite tilstande. Determinismen befinder sig på denne skridtvise begrænsede relation.

Finnemann konkluderer, at eftersom maskinens determinisme kun er lokal, kan turingmaskinen defineres som »en indeterministisk, mekanisk maskine«. Den kan reguleres gennem »løbende nye input« (instruksen).

Det betyder, at man må opgive den klassiske forestilling om, at maskiner ikke er *intensionelle*. Turings maskine viser, at enhver maskine er styret af en »formålsstyret afgrænsning«. Enhver anvendelse af en mekanisk regel beror på en begrænsning af dens generelle deterministiske karakter, dvs. der sker en opløsning af den konceptuelle sammensmeltning af begreberne om mekanik og determination.²¹ I flg. Finnemann var Turing ikke selv opmærksom på dette, og måske præsenterer Finnemann med denne analyse en nytænkning, som kan udvide forståelsen af de processer, som betinger forholdet mellem computer og menneske.

Den universelle egenskab ligger som omtalt i at det fysiske og symbolske interagerer på en måde, som ikke tidligere er set. Regnereglerne kan selv blive genstand for beregning – i form af strimlen, »hvor grænsen mellem maskine og materiale flyder ud, og hvor overgangen mellem det fysiske og det symbolske finder sted«.²² Maskinen bruger strimlen til at regulere sig selv på en måde som ikke er bygget ind i maskinens struktur. På sin vis gør han på denne måde maskinen analog med et menneske, idet de finite deterministiske regneprocedurer som maskinen kan udføre, er de samme som dem et menneske kan udføre med mekaniske midler, og den uendelige strimmel kan så sammenlignes med den menneskelige bevidsthed med de problemer det giver. På det enkelte skridts niveau arbejder computeren mekanisk på nogle symboler, som kan fortolkes som hvad som helst. Maskinen reagerer på det fysiske plan på et hvilket som helst symbol (som form), blot det kan skelnes fra de andre symboler i systemet – uanset hvad det betyder, uanset deres funktion. De enkelte symboler har en mekanisk-fysisk virkning i maskinen og en funktionel betydning i det symbolske udtryk.

Det er væsentligt her at pege på, at den informationelle notation er indholdsmæssig og semantisk tom. Regel og data udtrykkes i samme notationssystem, hvor den enkelte udtryksenhed (bittet) er mindre end den enkelte indholdsværdi, siger Finnemann. Den symbolske determination er hermed opløst i atomer. Dette forhold bliver først fuldstændiggjort med indførelsen af det binære system. Turing selv brugte en ret vilkårlig notation. Med den binære notation bliver maskinen også mere vanskelig at aflæse.

Turings biograf, Andrew Hodges, mente at Turing havde en opfattelse af menneskets bevidsthed som et finit system – dvs. en maskine – men det skal ikke omvendt forstås således, at turingmaskinen kan sammenlignes med den menneskelige bevidsthed. Turing brugte den menneskelige regnemetode som model for sin maskine.

Turingmaskinen fungerer på basis af nogle bestemte forhold vedr. den menneskelige bevidsthed. Turing tager altså en begrænset del af den menneskelige bevidsthed (via en fænomenologisk analyse af regneproceduren hos et menneske), og herudfra opbygger han maskinen. Men det er selvfølgelig ikke det samme som at drage nogen paralleller

på det generelle plan. I sin analyse af menneskets regneprocedure beskæftigede Turing sig med spørgsmålet om, hvorvidt den menneskelige bevidsthed arbejder på samme måde som turingmaskinen, nemlig opløser »de komplekse udtryk«. Det mente Turing, vi gjorde. Turingmaskinen fungerer på den måde at den kun kan foretage beregninger hvis processen automatiseres, men sådan er det ikke for mennesker. Vi bruger mest denne metode til meget simple opgaver, mens mere komplicerede kræver en kombination af forskellige metoder og redskaber. En turingmaskine opløser udtrykkene videre, end det er menneskeligt muligt, mens mennesker ikke opererer med enkelte symboler, som ikke længere har en fast (tal)værdi. I turingmaskinen er den fysiske definition af symbolerne udskilt fra definitionen af symbolernes værdi. Maskinen er ikke i stand til at foretage denne definition; det er kun mennesket, der har evnen til at definere symbolets fysiske form, uafhængigt af dets funktion og værdi. Netop dette er et kriterium på forskellen mellem menneske og maskine, mener Finneemann.

([Åter](#) till början av artikeln)

3. »Den tænkende maskine«

Turing uddyber sine tanker om menneske/maskin-relationen i sin 1950-artikel: »Computing Machinery and Intelligence«, hvor hovedspørgsmålet lyder: Kan maskiner tænke? Som »bevis« opstiller Turing en test, som står uimodsagt den dag i dag – på sine præmisser. Sherry Turkle²³ beskriver testen således (i oversættelse):

Man går ind i et rum og ser to terminaler. Én er forbundet med en computer, og den anden med en person som kan tale gennem den fra et andet rum. Man kan skrive spørgsmål, antagelser, fornærmelser, alt hvad man ønsker på begge terminaler. Formålet er at finde ud af hvilken terminal der er koblet til en computer under den antagelse, at den person der er tilknyttet gør sit bedste for at hjælpe med at finde løsningen. Maskinen derimod har ingen bindinger; dens eneste job er at simulere et menneske, så godt den kan (også evt. ved at snyde). Hvis man under disse vilkår ikke kan afgøre, hvilken af de to computere, der er sluttet til en datamaskine, ja så må man konkludere, at maskinen er intelligent.

Der findes endnu ikke nogen systemer, der har bestået Turings test, hvilket ikke betyder, at det ikke vil være muligt med tilstrækkeligt gode programmer. Finneemann siger meget rigtigt, at man hermed orienterer sig mod mulighedsbegrebet og ikke sandhedsbegrebet.²⁴

Turing-testen har givet anledning til mange modargumenter, hvor Searles »kinesiske værelse« er det mest berømte. Searle sætter sig i maskinens sted og simulerer ved hjælp af ordbøger, opslagsbøger, lærebøger o. lign. et totalt kendskab til kinesisk som han i øvrigt ikke forstår. Han består på denne måde Turing-testen. Argumentet er at simulering af intelligens i en turingmaskine ser bort fra »forståelse«. Det er ikke »tænkning« der finder sted i hhv. Searle eller en computer. Der er tale om et formelt program som ikke har tilknyttet nogen mening eller fortolkning. Dvs. et formelt program er ikke tilstrækkeligt til at give en forståelse, og det som gælder for kinesisk vil også gælde for andre mentale fænomener. Det er syntaks/semantik problematikken, der her

bliver aktualiseret. Searle udtrykker det selv på denne måde:

En computer-simulering af tanke er ikke tanke, fordi computeren simpelthen følger regler, den ikke »forstår«. Den mangler bevidsthed.²⁵

Diskussionen gik videre, og der kom mange modargumenter til Searle, f.eks. »systemargumentet.« Fortalerne for Turing-testen henviste nemlig til, at det ikke var Searles forståelse, som var essentiel. Systemet i sin helhed besad den nødvendige viden. Searles svar til dette var at det ingen forskel ville gøre, selvom manden i det kinesiske værelse besad den viden der var tilstede i rummet (dvs. internaliserede ordbøger), han ville stadig ikke forstå kinesisk. Forskere fra den systemtankegangen, som f.eks. Minsky ville kalde dette synspunkt for »præ-videnskabeligt«, idet det forudsætter at kun et »jeg« kan tænke. »Simulation af tanke er tanke«, siger han.²⁶ I den sidste ende bliver spørgsmålet dog: Hvad er det egentlig der bliver testet? Er det ikke snarere forsøgspersonens dygtighed til at stille gode spørgsmål? Eller med Hofstadters ord: Hvis en maskine består Turing-testen, vil det så ikke sige, at maskinen er god til at simulere tankevirksomhed!²⁷

Denne diskussion foregriber spørgsmålet om, hvad intelligens er. Om det siger Turing selv, at han ikke kan definere intelligens, og at det derfor på sin vis er meningsløst at stille spørgsmålet: »Kan maskiner tænke?« I det hele taget kan man diskutere hvad man forstår ved at tænke, anfører Jack Copeland.²⁸ Hidtil har mennesker haft eneret på at være tænkende væsener, men med computerens fremkomst er vi blevet konfronteret med artefakter som er svære at skelne fra mennesker, når det gælder evnen til problemløsning, planlægning, at tage beslutninger og lignende. Kan mennesker i disse tilfælde tage patent på begrebet »at tænke«? Copeland mener at vi må se på formålet med at bruge begrebet tænkning, når vi skal afgøre om en hensigtsmæssigt programmeret computer er en genstand der tænker eller ej. Her citerer han Wittgenstein: »Look at the word 'to think' as a tool.«²⁹ Copelands definition på tænkning som begreb er det, der kan henføres til »massivt adaptable enheder« som kan planlægge, analysere situationer, ræsonnere, undersøge analogier, revidere sine opfattelser i lyset af erfaringer, afveje modstridende interesser, formulere hypoteser og matche dem mod beviser, foretage fornuftige beslutninger på basis af ufuldstændig information osv., altså lidt mere end Turing-testen. For øvrigt er det Copelands opfattelse, at der ikke er noget i vejen for, at der en dag kan fremstilles robotter, som matcher mennesker i »adapabilitet«, dvs. at de kan tænke i bogstavelig forstand. Det Wittgensteinsk-inspirerede synspunkt er muligvis udtryk for et mere indsnævret intelligensbegreb, men det lægger op til den fornuftige holdning, at fordi computere skal fungere som mennesker i nogle henseender, skal de ikke nødvendigvis »være« mennesker i holistisk forstand. De skal være rationelle.

Når Dreyfus³⁰ påpeger, at der er emner en computer ikke kan diskutere, f.eks. følelser og intuition, så accepterer han, at man kun i indsnævret forstand kan tale om kunstig intelligens hos computeren som den kendes i dag. Derimod udelukker han ikke muligheden af, at hvis maskinen bliver forbedret på disse punkter, så vil man på længere sigt kunne tale om egentlig kunstig intelligens, dvs. »den tænkende maskine«. Dybest set accepterer Dreyfus Turing-testens legitimitet. Det samme gør Hofstadter,

idet han tilføjer en biologisk dimension: »Når vi skaber et program som kan bestå Turing-testen vil vi se et hjerte, selvom vi ved at det ikke findes«. ³¹ Når den kunstige intelligens bliver en realitet, vil den blive både mekanisk og organisk, eller vi vil opfatte den som organisk.

Denne tankegang genkendes med computeren Hal fra Stanley Kubricks *Rumrejsen År 2000* og hører hjemme inden for science-fiction. Imidlertid er der også inden for den anerkendte forskning, f.eks. på det nyoprettede Nano Science Center på Københavns Universitet ledet af forskningsprofessor Thomas Bjørneholm, tanker fremme om engang med tiden at kunne bygge organiske computere. Det går ud på at anvende DNA-molekylets evne til selvorganisering og initiering af kredsløbsstrukturer ved at forsøge at sammenkoble dem med uorganiske transistorpartikler, som efterfølgende kan bruges i en computer.

Andre er inde på samme tankegang, f.eks. Sherry Turkle som nævner bevidsthed som det væsentligste begreb i diskussionen om forskellen på menneske/maskine: »I det øjeblik man kan tale om at en maskine besidder bevidsthed, så vil den være levende, og mennesket har skabt en ny Golem«. ³²

I sin egen argumentation tager Turing for øvrigt selv forbehold. Han siger at der ikke er tale om et bevis. I stedet forsøger han at komme med modargumenter mod nogle modargumenter, som han selv opstiller! Finnemann anfører, at der er en logisk/videnskabelig brist i den måde, hvorpå Turing opstiller en hypotese, og herefter forvandler den til en videnskabelig teori. Hermed stilles man overfor den fulde bevisbyrdes problem, i stedet for at gøre brug af heuristik og »det kan ikke udelukkes«-argumentet.

Turing-testen er omdiskuteret, og i lyset af den kritik der er refereret her, kan man undre sig over, at Turings artikel »Computing Machinery and Intelligence« stadig er aktuel – i betragtning af artiklens indlysende mangler. Måske er det fordi emnet langt fra er udtømt og i øvrigt er spændende, og fordi Turing var den der gav bolden op og på den måde indledte diskussionen om kunstig intelligens.

Turing omtaler selv et af kriterierne for menneskelig intelligens, som maskiner ikke besidder, nemlig »diskriminativ kompetence«. En sammenligning mellem menneskelig bevidsthed og en maskine der kan bestå Turing-testen holder ikke vand, for det er to ikke-ækvivalente systemer der beskrives. Maskinen kan beskrives præcist, men det kan bevidstheden ikke. Turings egen universalmaskine viser netop, at det ikke kan lade sig gøre. Han modbeviser jo også, at den menneskelige bevidsthed kun kan arbejde som et diskret, mekanisk symbolsystem (pga. stop- og begyndelsesbetingelserne, nogle begrænsninger mennesker ikke har) og heller ikke kan være realiseret i et diskret fysisk system. Bevidstheden er unddraget den fysiske verdens endelighed ifølge Finnemann, som her bevæger sig på grænsen til en teologisk forestillingsverden. Den menneskelige bevidsthed med evnen til at definere symbolers fysiske og semantiske værdier er transcendentalt givet, en metafysisk forudsætning, mener han. Denne kompetence har mennesket tilsyneladende, fordi »den symbolske struktur ikke falder sammen med eller er kongruent med den fysiske struktur«, ³³ og alligevel kan man ikke se bort fra det fysiske-fysiologiske niveau.

Finnemann definerer intelligens som en »symbolgenerativ kompetence« og foreslår, at det bruges som testkriterium i bedømmelse af intelligens hos menneske/maskine. Den symboldannende kompetence defineres som »bevidsthedens, tænkningens og sprogets minimumsbetingelse«. Turing er inde på nogle af de samme tanker, hvor han skriver om »den kritiske tærskel« – et begreb, han henter fra fysikken, idet han siger, at den kritiske tærskel er karakteriseret ved en eksplosiv produktion af nye forestillinger.

Diskussionen gik videre og er stadig højaktuel set i lyset af fremkomsten af nye supercomputere. Et eksempel på en sådan computer er Deep Blue, som på et begrænset område – skak – i 1997 kunne slå en menneskelig stormester, Garry Kasparov. I skak er udfaldsrummet stort, så det er svært (for et menneske?) at beskrive alle muligheder.

Computerens valg ligner beslutninger der kommer fra en menneskelig tankeproces, så her kan vi måske tale om et område, hvor Turing-testen er bestået. Vi taler her om et meget lille veldefineret grænseområde, men hvis det kan lykkes på ét område, hvorfor så ikke på flere?

Det nyeste skud på stammen hedder Blue Gene, som skal anvendes inden for det humane genom-projekt. Det blev startet i 1999 af IBM's Thomas J. Watson Forskningscenter og forventes at få en kapacitet som 1000-dobler de nuværende computere. Målet nås ved inden for de næste 5 år at opbygge et netværk af en million parallelle processorer. En mindre forsøgsmodel forventes at være færdig år 2002. Om det kan lykkes at udvikle computere som kan overgå mennesket i udvikling af nye idéer og teorier – det står dog tilbage at eftervise.

[\(Åter till början av artikeln\)](#)

4. Konkluderende bemærkninger

På baggrund af en gennemgang af Turings ideer og hans maskine og Niels Ole Finnemanns symbolteoretiske tanker som gør opmærksom på at der er en semantisk overbygning til Turings algoritmiske, matematiske fundament, kan man stille sig det spørgsmål om dette er relevant for biblioteks- og informationsvidenskaben i dag. Svaret er ja.

Turings model er det tidligste eksempel på mekanisering af informationsbehandling. Nogle anvendelser af informationsteknologi i biblioteks- og informationsvidenskaben, f.eks. genfindning af dokumenter, er ikke uproblematisk bl.a. på grund af støj set i lyset af stadig stigende informationsmængder. Med inspiration fra Turing og Finnemann kan løsningen være at udvikle en semantik på et matematisk fundament som er istand til at definere objekter og aktioner. Hvis vi har sådan en semantik, siger Turing at vi kan løse problemerne. Selvom vi ikke kan bygge intelligente maskiner, kan vi bygge formålsbundne maskiner baseret på rationel logik, som kan hjælpe os med at løse konkrete problemer.

Også på et andet område har Turing relevans for biblioteks- og informationsvidenskaben. Det gælder definitionen og identificeringen af området. Selve den opgave at placere biblioteks- og informationsvidenskaben på det videnskabelige

landkort er et emne for diskurs, såvel som for bibliometriske undersøgelser af import og eksport af viden, hvor en videnskab på basis af kvantitative analyser kan placeres ud fra disse parametre. Det vil sige at en videnskab bl.a. kan defineres ud fra målinger af samspillet med andre videnskabelige områder. Ofte bliver biblioteks- og informationsvidenskab defineret som en samfundsvidenskab. Dette er dog genstand for diskussion, og der vil altid være flere opfattelser af den sag. Nogle vil vægte forbindelsen til naturvidenskab højt, andre ser biblioteks- og informationsvidenskab som et humanistisk område.

Biblioteks- og informationsvidenskabelige undervisningsanstalter og institutter kan have forskellig fokus fra sted til sted.³⁴ At drage Alan Turing ind i en biblioteksskoles curriculum vil være mere indlysende på nogle institutioner end på andre, afhængig af sammensætning af fag, de ansatte læreres interesseområder og baggrunde og uddannelsens videnskabelige niveau. Det er gjort i denne artikel ud fra et humanistisk/kognitivt/naturvidenskabeligt perspektiv, med andre ord ud fra en opfattelse af at biblioteks- og informationsvidenskab er en hybrid, et eksempel på et tværvideenskabeligt område med tråde til alle retninger i den traditionelle tredeling, naturvidenskab, samfundsfag, humaniora.

Tværvideenskabelig forskning er et tema der er oppe i tiden og genstand for politisk velvilje hvor den danske IT- og forskningsministerium år 2001 har afsat et større beløb til forskning på tværs af fag, institutioner og grænsen mellem offentlig og privat. I den forbindelse ligger biblioteks- og informationsvidenskab godt placeret som brobygger med inddragelse af elementer fra mange forskningsretninger, herunder også kognitionsforskning og naturvidenskab med Alan Turings forskning som ikke bare er matematiske teorier om databehandling, men også vedrører emner som kunstig intelligens og forholdet mellem tænkning, maskiner, og erkendelse.

([Åter](#) till början av artikeln)

Om författaren

Ellen Bonnevie er cand mag. i humanistisk informatik, ansat på Danmarks Biblioteksskole i København som lektor ved Institut for informationstudier. Hendes primære arbejdsområde er bibliometri, som hun har undervist i siden 1998. Hendes videnskabelige arbejde ligger dels inden for bibliometri, dels inden for informationsvidenskabelige studier, idet hun har skrevet en artikel om Fred I. Dretskes betydning for metateorier inden for biblioteks- og informationsvidenskaben: »Dretske's Semantic Information Theory and Meta-Theories in Library and Information Science«. (In print, Journal of Documentation, July 2001)

([Åter](#) till början av artikeln)

Noter

1. Jakob Nielsens hjemmeside <http://www.useit.com/jakob/>, besøgt 27.2 2001. ([Åter till texten](#))
2. Leiber, J. *An Invitation to Cognitive Science*. Basil Blackwell, 1991, s. 15. ([Åter till texten](#))

3. Samuelson, C. Relating Turing's formula and Zipf's law. *Proceedings of the 4th Workshop on Very Large Corpora, Denmark, 1996*. <http://linkage.rockefeller.edu/wli/zipf/>. Besøgt 27.2 2001. ([Åter till texten](#))
4. Umstätter, W. *Zum Thema: Der blinde Springer*. <http://www.ib.hu-berlin.de/~wumsta/lectp.html>. Besøgt 22.11 2000. ([Åter till texten](#))
5. Hjørland, B. *Informationsvidenskabelige grundbegreber*. København: Danmarks Biblioteksskole, 1995. ([Åter till texten](#))
6. *International Encyclopedia of Information and Library Science*. Feather, J. & Sturges, P., eds. London and New York: Routledge, 1996. ([Åter till texten](#))
7. Turing, A. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, vol. 42, 1936–1937. ([Åter till texten](#))
8. Turing, A. Computing Machinery and Intelligence. *Mind*, vol. 59, no. 236, 1950 (Oct. 1950), s. 433–460. ([Åter till texten](#))
9. Finnemann, N.O. *Tanke, sprog, maskine: En teoretisk analyse af computeres symbolske egenskaber*. Århus: Akademisk Forlag, 1994. ([Åter till texten](#))
10. Prebensen, H. Algoritme, sprog og programmer: Blaise ikke på Pascal. *Data Humana*, 5. Studiehæfter udgivet af Humanistisk Edb-Center, Københavns Universitet, 1989, II, 2. ([Åter till texten](#))
11. *ibid.* ([Åter till texten](#))
12. Glenn, J. & Littler, G. *Dictionary of Mathematics*. London: Harper & Row, 1984. ([Åter till texten](#))
13. Andrew Hodges, Turings biograf fortæller, at Turing fandt løsningen på Hilberts spørgsmål mens han lå på en eng efter en løbetur langs floden. Altså snublede han nærmest over det, et eklatant eksempel på »serendipity« med andre ord. Den rette kombination af tanke, krop og natur? ([Åter till texten](#))
14. Prebensen, H. Algoritme, sprog og programmer: Blaise ikke på Pascal. *Data Humana*, 5. Studiehæfter udgivet af Humanistisk Edb-Center, Københavns Universitet, 1989, II, 5. ([Åter till texten](#))
15. Hodges, A. *Alan Turing: The Enigma*. New York: Simon & Schuster, 1983, s. 97. ([Åter till texten](#))
16. *ibid.*, s. 98 og 99. ([Åter till texten](#))
17. Johnson-Laird, Ph. *The Computer and the Mind: An Introduction to cognitive science*. London: Fontana Press, 1988, s. 50 f. ([Åter till texten](#))
18. Goldschlager, L. & Lister, A. *Computer Science: A Modern introduction*. New York, London: Prentice Hall, 1987, s. 70. ([Åter till texten](#))
19. Weizenbaum, J. *Computer Power and Human Reason: From Judgement to calculation*. New York: W.H. Freeman, 1976. ([Åter till texten](#))
20. Finnemann, N.O. *Tanke, sprog, maskine: En teoretisk analyse af computeres symbolske egenskaber*. Århus: Akademisk Forlag, 1994. ([Åter till texten](#))
21. *ibid.*, s. 101. ([Åter till texten](#))
22. *ibid.*, kap. 5, afsnit 5.5. ([Åter till texten](#))

23. Turkle, S. *The Second Self*. London: Granada, 1984. ([Åter till texten](#))
24. Finnemann, N.O. & Stjernfelt, F., eds. *Kognition og sprog*. Århus Universitetsforlag, 1992, s. 89. ([Åter till texten](#))
25. Searle, J. R. Is the Brain's Mind a Computer-Program? *Scientific American*, vol. 262, no. 1, 1990, s. 26–31. ([Åter till texten](#))
26. Minsky, M.L. Matter, Mind and Models. In: Minsky, M.L., ed. *Semantic Information Processing*. Cambridge: M.I.T. Press, 1967. ([Åter till texten](#))
27. Hofstadter, D.R. *Gödel Escher Bach*. New York: Basic Books, 1979. ([Åter till texten](#))
28. Copeland, J. *Artificial Intelligence: A Philosophical introduction*. Oxford: Blackwell, 1993. ([Åter till texten](#))
29. Wittgenstein, L. *Philosophical Investigations*. Oxford: Blackwell, 1963, sect. 360. ([Åter till texten](#))
30. Dreyfus, H.L. & Dreyfus, S.E. *Mind over Machine: The Power of human intuition and expertise in the era of the computer*. New York: Blackwell, 1986. ([Åter till texten](#))
31. Hofstadter, D.R. *Gödel Escher Bach*. New York: Basic Books, 1979, s. 713. ([Åter till texten](#))
32. Meyrink, Gustav. *Der Golem*. 1916. ([Åter till texten](#))
33. Finnemann, N.O. *Tanke, sprog, maskine: En teoretisk analyse af computeres symbolske egenskaber*. Århus: Akademisk Forlag, 1994, kap. 5, afsnit 5.9. ([Åter till texten](#))
34. Ellis, D., Allen, D. & Wilson, T. Information Science and Information Systems: Conjoint subjects disjunct disciplines. *Journal of the American Society for Information Science*, vol. 50, no. 12, 1999, s. 1095–1107. s. 1101. ([Åter till texten](#))

([Åter](#) till början av artikeln)

© Ellen Bonnevie 2001

[Åter till Human IT 1/2001](#)