Hilbert forestillede sig, undslipper ikke paradokserne: den fuldstændige formalisering er umulig.

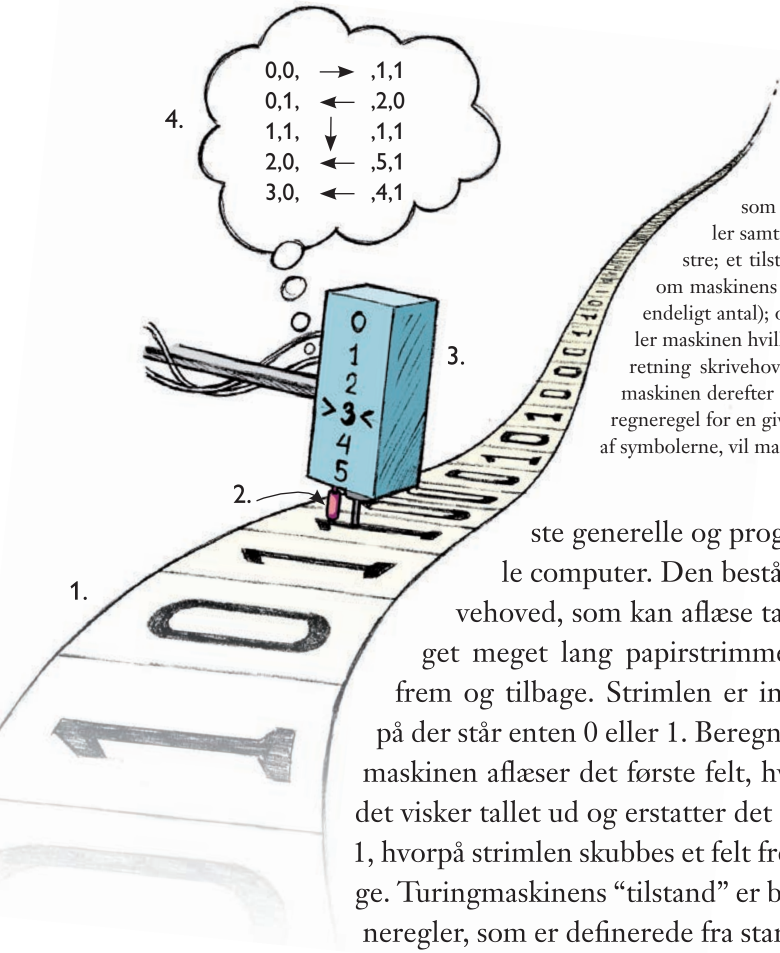
Reaktionerne var til at starte med stor forbløffelse. Logikkens og matematikkens fundamenter var pludselig blevet gennemhullet, og mange matematikere mistede en hel del entusiasme til at fortsætte ud af den vej, som Hilbert havde udstukket for dem. Men det viste sig i løbet at meget kort tid, at matematikernes normale hverdag slet ikke blev berørt af Gödels resultater. Man kunne lige så godt ignorere problemerne og fortsætte som altid, fordi man kun ville komme i problemer i de mest obskure og mærkelige udkanter af matematisk forskning. Selv i dag er Gödels bevis mere berømt blandt videnskabsteoretikere og filosoffer end blandt praktiserende matematikere.

Det er også interessant at iagttage, at Brouwers konstruktivistiske matematiske filosofi er den eneste, som undgår paradokser og inkonsistenser. Det skyldes, at Brouwer er meget forsigtig med hensyn til, hvad man ifølge ham kan udtale sig om. Men på trods af det har den matematiske konstruktivisme altid været den mest marginaliserede blandt de matematiske skoler, der opstod i begyndelsen af 1900-tallet. Måske skyldes det den omstændighed, at tal og matematiske beviser virker så reelle, så virkelige, når man sidder og arbejder med dem dagen lang, at de fleste matematikere får et meget "platonisk" forhold til deres fag. Matematikkens urimelige effektivitet med hensyn til at beskrive verden kan nærmest tvinge én til at tro, at matematik er lige så konkret og virkeligt som den stol, man sidder og arbejder på.

Turing og den universelle maskine

Den engelske matematiker Alan Turing fremlagde i 1936 en endnu dybere begrundelse for ufuldstændighed end Gödel, samtidig med at han udviklede det teoretiske grundlag for den moderne computer. I Alan Turings samtid blev ordet "computer" stadig forstået som værende et menneske, der arbejdede med at beregne. Det var derfor en meget kontroversiel ide at betragte mentale processer som noget, der kunne splittes op i simple mekaniserbare operationer.

En såkaldt Turingmaskine er en abstrakt repræsentation af en regnemaskine. Selvom Turing ikke rent fysisk byggede den første "moderne" computer – det gjorde amerikaneren John Vincent Atanasoff (1903-95) – kan man med god ret sige, at Alan Turing på et teoretisk plan opfandt den før-



En mekanisk version af en Turingmaskine består af en papirstrimmel (1), som er inddelt i felter, hvorpå der står nogle symboler, typisk 0 og 1; et skrivehoved (2), som kan læse, viske ud og skrive symboler samt bevæge sig trinvis til højre eller venstre; et tilstandsregister (3), som gemmer viden om maskinens tilstand (hvoraf der kun kan være et endeligt antal); og et sæt regneregler (4), som fortæller maskinen hvilket symbol, der skal skrives, i hvilken retning skrivehovedet skal rykke, og hvilken tilstand maskinen derefter skal gå over i. Når der ikke er nogen regneregel for en given tilstand og en given kombination af symbolerne, vil maskinen stoppe.

ste generelle og programmerbare digitale computer. Den består af et læse- og skrivehoved, som kan aflæse tal på en tynd og meget meget lang papirstrimmel, der kan skubbes frem og tilbage. Strimlen er inddelt i felter, hvorpå der står enten 0 eller 1. Beregningen begynder, når maskinen aflæser det første felt, hvorefter skrivehovedet visker tallet ud og erstatter det med enten et 0 eller 1, hvorpå strimlen skubbes et felt frem eller et felt tilbage. Turingmaskinens "tilstand" er bestemt af et sæt regneregler, som er definerede fra starten af. Dens aktuelle tilstand ændrer sig altså løbende i forhold til de ettaller og nuller, den møder på vejen. Maskinen kan for eksempel være i tilstand 7, hvor den har en regel om at skrive 0, når den møder et 1-tal, for derefter at bevæge sig et hak til venstre og gå over i tilstand 5. Når den så læser et 0 i tilstand 5, kan reglen være, at den skal skrive 1, gå én til højre og skifte til tilstand 12 osv. Turingmaskinen er altså en mekanisk bureaukrat, som ikke behøver frokostpauser eller toiletbesøg. Den gør præcis de ting, den får besked på ved at slå op i en prædefineret og endelig tabel, som den kan huske, og derefter rykke rundt på ettaller og nuller.

I virkeligheden er en Turingmaskine en abstraktion af et computerprogram, snarere end en maskine – dvs. den er egentlig et stykke software, ikke hardware. Man kan vise, at man kan lave Turingmaskiner, der på trods af deres ekstreme simpelhed kan beregne en hvilken som helst funktion, som en normal digital computer kan beregne. Den skal blot have nok tid og papir. Turingmaskinen er derfor en konkret definition af, hvad en effektiv algoritme er. At en algoritme er effektiv forstås her, som at den skal have et endeligt antal veldefinerede trin, der kan udføres mekanisk, og at den med det samme input altid skal producere det samme output.

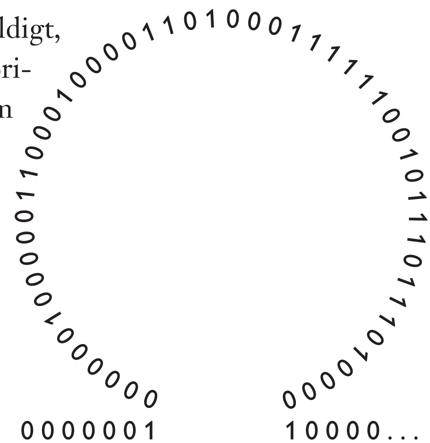
På et idehistorisk plan var Turingmaskinen meget vigtig, fordi den var en konkret realisering af Hilberts ide om at transformere hele matematikken til en formel mekanisk proces, hvor man ikke behøver andet end slavisk at følge et endeligt antal opskrifter for at bevise et hvilket som helst teorem. Dette var formuleringen af Hilberts "Entscheidungsproblem", og Turings svar var, at der ikke findes nogen "Entscheidung", dvs. at der ikke findes nogen effektiv metode eller mekanisk procedure, ikke nogen computer eller noget computerprogram, som på forhånd kan afgøre, om et vilkårligt andet computerprogram stopper, dvs. får regnet sig færdig til et resultat eller ej. Dette kaldes Turings "stop-problem". Gödels ufuldstændighedsteorem er så blot en naturlig følge af stop-problemet, fordi hvis man kan vise, at det er umuligt at afgøre, om et computerprogram stopper, så kan der heller ikke findes et fuldstændigt og konsistent sæt af aksiomer, med hvilke man kan slutte sig frem til, om en matematisk sætning kan bevises eller ej, idet aksiomerne jo ville kunne oversættes til en effektiv algoritme.

Gödels og Turings arbejder rejste væsentlige spørgsmål for grænserne for formel tænkning. Selv det at antage, at noget var "falskt" eller "sandt" i matematisk forstand, var blevet problematisk, fordi begge kategorier var blevet beviseligt ubeviselige. En platonist som Gödel kunne måske godt tro på, at kontinuumshypotesen var falsk, ligesom Cantor mente, at den var sand, men nu var det ikke længere en sag, der kunne afgøres ved hjælp af matematik eller logik. Den amerikanske matematiker Paul Joseph Cohen (1934-2007) kunne i forlængelse af Gödels og Turings arbejde i 1960'erne vise, at kontinuumshypotesen for den sags skyld både kunne være sand og falsk, uden at det ville påvirke mængdelærens øvrige aksiomer. Nogle matematikere begyndte derfor at anse Cantors mængdelære som en forkert måde at gribe sagen an på. I stedet for at reducere et linjestykke til et sæt af individuelle punkter og tal, for derpå at konstruere et kontinuum, forsøgte man at forstå kontinuet ud fra en mere helhedsorienteret tilgang, hvor det snarere er strukturer og relationer, som betragtes som de fundamentale objekter i analysen, og ikke uendelige mængder af punkter.

En anden væsentlig pointe i Turings arbejde var, at hans matematiske argument afhang af fysikkens natur – i dette tilfælde af beregnelighedens grænser via en effektiv computeralgoritme. Det åbnede op for en meget tættere forbindelse mellem fysik og matematik end tidligere antaget. For hvis de logiske grænser for beregnelighed også gælder for alle regnemaskiner, vil en lang række problemer heller aldrig kunne løses i praksis. Turings stop-problem og bestemte aritmetiske beslutningsproblemer vil aldrig kunne finde en løsning, selv med den størst tænkelige computer. En lang række beregninger, som f.eks. sorteringen af elementer i en liste, vil kun kunne løses inden for et tidsrum, der er afhængig af problemets størrelse. Det vil sige, at hvis problemet er for stort, vil det aldrig kunne løses i endelig tid.

Et tredje aspekt ved Gödels og Turings resultater er, at ikke blot uvidenhed, men også tilfældighed sniger sig ind ad bagdøren på det, man troede var den mest eksakte af alle videnskaber. Den amerikanske matematiker Gregory Chaitin (f. 1947) kunne i sine analyser af Gödels og Turings resultater gå så vidt som til at konstruere et tal, der er absolut umuligt at kende. Tallet, som han kaldte Omega, var pr. definition så tilfældigt, at ingen computer nogen sinde ville være i stand til at finde alle dets decimaler, selv hvis den var uendelig hurtig. Ved hjælp af informationsteoretiske overvejelser om, hvordan tilfældighed og kompleksitet bør defineres matematisk, kunne han vise, at alle reelle tal er tilfældige tal – samtidig med at han beviste, at det er umuligt at udpege et eneste tilfældigt tal. Kompleksiteten af et tal kunne Chaitin definere som længden af det korteste computerprogram, der kan beregne tallet, mens et tilfældigt tal er et tal, hvis kompleksitet svarer til dets bitlængde, dvs. et tal, som ikke kan komprimeres.

Tallet Omega er bemærkelsesværdigt. Det kan defineres, men er ikke beregneligt. Det er tilfældigt, men tælleligt. Det indeholder den mest komprimerede information om Turings stop-problem



Omega er et ikke-beregneligt tal, dvs. et tal, hvor man ikke kan lave noget program, der kan beregne sig til alle dets bits. Trods disse vanskeligheder lykkedes det i 2002 at beregne de første 64 bits af Chaitins originale Omega. Resultatet er her skrevet ind i Omega-tegnet i binær form og viser, at tallet er ca. 0,78 procent. Med andre ord er sandsynligheden for – ved gentagne forlængelser af en bit-streng med tilfældige bits – at lave et program, som på et eller andet tidspunkt stopper, lig med ca. 0,78 procent.

for alle programmer med maksimalt n bit. Og derfor indeholder Omegas første n decimaler informationen om bevisbarheden for alle matematiske sætninger i et formelt system, som er n-bit stort.

Informationsbegrebet

Resultatet af overvejelserne om ufuldstændighed, tilfældighed, uvidenhed, sandsynlighed osv. førte til, at der i løbet af midten af 1900-tallet udvikledes et helt nyt koncept, hvormed man kunne begynde at forstå fysiske og matematiske strukturer. Dette koncept var *information*. I daglig tale defineres information som noget, der har med overførsel af viden at gøre. Dette er en forståelse af ordet, hvor enhver information kun kan forstås, hvis modtageren har en baggrundsviden om det pågældende sprog, om konteksten, de implicitte antagelser osv. Hvis man derimod forestiller sig, at der er kommet et brev fra stjernesystemet Alfa Centauri, kan man næppe gøre sig forhåbninger om at få en forståelse af indholdet, alle populære myter til trods – man kan måske ikke engang være sikker på, at der er tale om et brev.

I semantisk informationsteori taler man her om et referenceproblem: et objekts potentielle vidensindhold vil udefra kun kunne beskrives som en sandsynlighedsfordeling over alle mulige fortolkninger med én og samme sandsynlighed. Uden anden information vil brevet principielt kunne betyde alt mellem himmel og jord. Først når der opstår referentielle begrænsninger for realiseringen af de enkelte alternativer, vil der kunne opstå betydningsbærende elementer, som så igen kan virke tilbage og favorisere bestemte muligheder, og først da bliver brevet forståeligt. Meningsfyldt information (i f.eks. en binær sekvens) er altså ensbetydende med en ændring i sandsynlighedsfordelingen af mulighederne på grundlag af en yderligere indsnævring, som kun kan opstå ved en berøring med en omverden, fælles referencer, symboldannelse osv. Det betyder, at uden referencer kan et virkelig fremmed signal, hvor man intet kender til afsenderen, ikke være andet end et spejl. Et spejl af vores egne tanker. Man bliver offer for en projektion og begynder at lægge sine egne bekymringer og håb i signalets tydninger. Militæret vil sikkert forvente nye våben, videnskabsmanden ny erkendelse, og skønånden vil håbe på forløsning. Men når alt kommer til alt, vil vi mennesker ikke være i stand til at erkende eller forstå signaler fra sådanne verdener, selv hvis de stod med lysende flammer på firmamentet.