

Alan Turing: Het Enigma

Jan Van den Bussche
Universiteit Hasselt

lezing Seniorenuniversiteit
17 maart 2014

Samenvatting

We geven een bloemlezing uit de biografie van Alan Turing, de uitvinder van de computer.¹ We gaan kort in op zijn jeugdijaren; we bespreken zijn uitvinding van de computer en zijn fundamentele bijdragen aan de wiskundige logica en de grondslagen van de wiskunde; we beschrijven de oorlogsjaren (Tweede Wereldoorlog) waarin hij een leidende rol speelde als codebreker; we bekijken de ontwikkelingen vlak na de oorlog wanneer de eerste werkende digitale computers werden gemaakt; en we bespreken zijn ideeën waarin hij als eerste het begrip van de Kunstmatige Intelligentie introduceerde. Jammer genoeg heeft hij maar 42 jaar geleefd; we bespreken tot slot dan ook kort hoe de Alan Turing van het verleden zich verhoudt tot het heden.

1 Inleiding

Op 25 mei 2011, tijdens zijn staatsbezoek aan Groot-Brittannië, sprak president Obama van de Verenigde Staten het Brits parlement toe in Westminster Hall. Hij noemde daarin de drie allergrootste Britse wetenschappers, met name Isaac Newton, Charles Darwin, en Alan Turing. Iedereen heeft wel gehoord van de evolutietheorie van Darwin, en elke leerling van het Algemeen Secundair Onderwijs heeft in de lessen fysica geleerd over de wetten van Newton. Van Alan Turing heeft echter bijna niemand gehoord. In deze lezing zal duidelijk worden dat Alan Turing de uitvinder is van de computer en van de Kunstmatige Intelligentie. Als we beschouwen hoe de informatica onze samenleving zo radicaal heeft veranderd, zult u hem zijn plaats naast Newton en Darwin wel gunnen.

¹*Alan Turing: The Enigma*, van Andrew Hodges, uitgegeven door Vintage, 2012.

In het jaar 2012 was de honderste verjaardag van zijn geboorte de gelegenheid voor de wereldwijde informaticagemeenschap om Turing uitgebreid te herdenken.² Overall zijn er toen evenementen georganiseerd waarin Turings intellectuele erfenis is gevierd. In Groot-Brittannië is Turing toen ook uitgebreid in de media gekomen. Het docudrama *Codebreaker* over het leven van Alan Turing kwam uit een jaar eerder, in 2011. Maar al in 1986 schreef Hugh Whitmore het toneelstuk *Breaking the Code*, gebaseerd op dezelfde biografie van Turing die wij in deze lezing zullen volgen. *Breaking the Code* kwam in 1996 ook uit als TV-film. Tenslotte werken de Amerikanen aan hun eigen langspeelfilm over Turing, getiteld *The Imitation Game*, die nog dit jaar zal uitkomen.

Hoe komt het echter dat hij zo lang zo onbekend is gebleven bij het algemeen publiek? Een reden is zeker dat zijn belangrijkste wetenschappelijk werk werd gepubliceerd vlak voor de Tweede Wereldoorlog, die zijn wetenschappelijke carrière drastisch onderbrak. Zoals de titels van voormelde films al suggereren, heeft Turing tijdens de oorlog uiterst geheim werk verricht als codebreker, dat pas in de jaren 1970 en 1980, lang na zijn vroegtijdige dood, is vrijgegeven. Hij heeft dus altijd in de schaduw van de Britse geheime dienst geleefd. Een belangrijkere reden echter is dat Turing, genie dat hij was, weinig sociale, diplomatieke, of managerstalenten had. Hij kon zichzelf niet ‘verkopen’. Heel belangrijk ook is dat Turing op een zeer tragische wijze is gestorven, waarin de Britse staat mogelijk een onverkwikkelijke rol heeft gespeeld.

Een laatste reden die ik persoonlijk nog wil aangeven is de afwezigheid van informatica als wetenschap in het secundair onderwijs. Terwijl in de lessen fysica de naam van Newton zeker zal vallen, bestaat er in ons secundair onderwijs geen vak over de informatica als wetenschap. Als er al een vak informatica wordt gegeven, wordt het ingevuld als ICT en gaat het over het praktisch gebruik van computers en software, maar het wordt zeker niet beschouwd als wetenschapsvak. De naam van Alan Turing zal op school dus niet vernoemd worden. Als informatica al niet de wetenschap van de 20ste eeuw was, dan is ze zeker die van de 21ste eeuw. Het is dus dringend nodig dat informatica als wetenschap serieus wordt genomen in ons onderwijs. Gelukkig is er een beweging met dit doel op gang gekomen, waar mijn collega professor Frank Neven een leidende rol in speelt.³

²2012 *The Alan Turing Year, A Centenary Celebration of the Life and Work of Alan Turing*, <http://www.turingcentenary.eu>

³*Forum voor Informaticawetenschappen*, <http://www.i22n.org>

2 Jeugd

Alan Turing wordt geboren op 23 juni 1912 in een moederhuis in Paddington, in west-Londen. Zijn ouders zijn dan op verlof uit Madras, Indië, waar vader Julius Turing werkt als koloniaal ambtenaar. John, de oudere broer van Alan, was vier jaar voorheen al geboren in Indië. De familie van Alan behoort tot een sociale klasse die de auteur George Orwell omschreef als ‘lower upper middle class’, die voornamelijk actief was in de Britse ambtenarij of het leger. De ouders kiezen ervoor om Alan en John niet in Indië te laten opgroeien, maar plaatsen hen bij de Wards, een gepensioneerd koppel dat jonge kinderen opvangt en opvoedt in hun huis in St.-Leonards-on-Sea, aan de zuidkust tussen Dover en Brighton.

Het is al snel duidelijk dat Alan een genie is. Als jonge kleuter heeft hij onmiddellijk door wanneer volwassenen hem opzettelijk laten winnen bij gezelschapsspelletjes. Op vijfjarige leeftijd leert hij zichzelf lezen uit een boek *Reading Without Tears*. Tijdens wandelingen op straat wil hij absoluut stoppen bij elke lantaarnpaal om dan luidop het serienummer af te lezen.

De opvoeding van John en Alan is gericht op één enkel doel: hun middelbaar onderwijs krijgen in een *public school*, de misleidende Engelse naam voor het systeem van elitaire kostscholen waarvan Eton de bekendste is. Met dit doel gaat Alan naar St.-Michaels, een privé-lagere school waar hij al vanaf zes jaar Latijn leert. Op zijn negende gaat hij op kostschool in Hazelhurst, waar de kinderen rechtstreeks klaar worden gestoomd voor public school. Dit hele privé-onderwijssysteem is tot op zekere hoogte zelfs vandaag nog de norm voor de hogere sociale klassen in Groot-Brittannië, en families werken zichzelf soms zwaar in schulden om de opvoeding van hun kinderen te financieren. Een heel contrast met het onderwijs bij ons in Vlaanderen, waar kwalitatief onderwijs van kleuterschool tot en met universiteit door de gemeenschap wordt gedragen! Wij staan daar misschien te weinig bij stil dat dit absoluut geen vanzelfsprekendheid is.

In Hazelhurst toont Alan zich als een heel intelligent kind. Zijn leraars klagen echter dat hij slordig schrijft. Hij is heel inventief en vindt allerlei tuigen uit, zoals een cinematoestel waarmee hij zelfgemaakte tekenfilmpjes kan afspelen. In een brief aan zijn ouders schrijft hij over het uitvinden van een typemachine. Ook is hij heel graag bezig met scheikundige proeven: voor zijn Kerstmis krijgt hij een chemiedoos. Op tienjarige leeftijd krijgt hij het boek *Natural Wonders Every Child Should Know* van Edwin Brewster. Dit boek laat kinderen de wonderen van de natuur kennen en legt in vrij veel detail uit hoe planten, dieren, en ook wij mensen in elkaar zitten, als het ware hoe wij ‘werken’ alsof we machines waren. Dit fascineerde Alan bovenmatig; hij sleurde zijn boek overal mee. Op dat ogenblik is het eigenlijk al duidelijk

dat Alan wetenschapper wil worden. Zijn vader denkt daar het zijne van, maar zijn moeder kan zich daar goed in vinden: een overgrootnonkel van Alan langs moeders kant, George Johnstone Stoney, was fysicus, Fellow of the Royal Society, en uitvinder van het elektron.

In 1926 start Alan dan zijn middelbaar onderwijs in Sherborne public school, een dorpje in zuid-Engeland 80 km ten westen van Southampton. Hij woont op dat ogenblik bij zijn ouders in Rouen; zijn vader was toen al op vervroegd pensioen gegaan, en had zich in Rouen gevestigd omdat hij dan geen belastingen hoefde te betalen noch aan Frankrijk noch aan Engeland. Als twaalfjarige jongen wordt Alan in St.-Malo op de boot gezet richting Southampton, waar hij geacht wordt verder de trein te nemen naar Sherborne. Op de boot verneemt hij echter dat er een algemene staking aan de gang is in Engeland. Hij had gelukkig zijn fiets bij zich en fietste dan maar op eigen kracht in twee dagen naar Sherborne. Deze uitzonderlijke zin voor initiatief was vrij sensationeel en werd gerapporteerd in de regionale krant *The Western Gazette* die nu nog altijd bestaat. In het 2012 Turing jaar hebben leerlingen van Sherborne public school (die ook nog altijd bestaat) trouwens deze fietstocht overgedaan als eerbetoon aan hun meest beroemde alumnus.

Het Victoriaanse kostschoolleven was voor Alan, die sociaal helemaal niet zo vaardig was, niet zo aangenaam, maar hij was sterk en hij redde zich. De belangrijkste teamsport op school was hockey, maar hij stond ervoor bekend dat hij tijdens de hockeytraining liever de madeliefjes zag groeien. Waar hij wel heel graag aan deelnam waren de lange wandelingen in de natuur die regelmatig georganiseerd werden. Hij blonk uit in scheikunde en wiskunde, waarover hij soms meer wist dan zijn leraars. Hij werd echter zwak bevonden in Frans en Latijn, en ook zijn slordig werk, waardoor hij ook veel fouten maakte zelfs voor wiskunde, werd in elk rapport negatief vermeld.

Door zijn voorliefde voor wiskunde en zijn scheikundige experimenten wordt Alan als een zonderling beschouwd door zijn medeleerlingen. In zijn vrije tijd, die hij dan ook meestal alleen doorbrengt, leest hij het populariserend boek over de relativiteitstheorie dat Einstein een tiental jaren voordien had geschreven. In brieven aan zijn moeder leidt hij zelfstandig de wiskundige wet af die de vrije beweging van lichamen volgens de relativiteitstheorie beschrijft. Einstein had die wiskundige formule in zijn populariserend boek weggelaten.

Heel belangrijk voor Alan is de vriendschap, vanaf 1928, met Christopher Morcom, een leerling van een jaar hoger. Christopher Morcom komt uit een welgestelde familie: zijn grootvader was een ondernemer in stoommachines, en zijn vader had het bedrijf overgenomen. Uit zijn militaire dienst had vader Morcom de graad van Kolonel overgehouden. Moeder Morcom was een

artistieke dame die zich bezighield met een veelheid aan projecten. Christopher had een veel oudere broer, Rupert, die gestudeerd had in Cambridge en nu in Zürich aan een doctoraat bezig was in de fysica. Rupert en Christopher hadden thuis een heus laboratorium, en een eigen observatorium met telescoop om naar de sterren te kijken. Christopher speelde prachtig piano, was heel beleefd, gedroeg zich altijd heel netjes, en werkte heel ordelijk en nauwkeurig. Op deze wijze had Christopher een positieve invloed op Alan. Maar het meest belangrijke was dat Alan merkte dat hij niet alleen was, dat hij vrienden kon hebben, en dat er nog mensen gepassioneerd waren door de wetenschap.

Christopher wijst Alan ook de weg naar Cambridge. In 1929 gaan ze samen naar Cambridge om gedurende een week lang deel te nemen aan examens, met als doel het winnen van een beurs. Zo'n beurs, waardoor je gratis kon studeren, is niet alleen gigantisch prestigieus: je krijgt een kamer en maaltijden op een College, en je krijgt ook nog een inkomen genoeg om rond te komen als student. Christopher is succesvol, maar Alan, die tenslotte een jaar jonger is, zal pas het volgende jaar een beurs verwerven. Jammer genoeg sterft Christopher vroegtijdig op 13 februari 1930 aan tuberculose. Alan zal nog vele jaren in contact blijven met de familie Morcom.

3 De Turing Machine

In september 1931 begint Alan dan zijn studies wiskunde aan de universiteit van Cambridge. Hij is verbonden aan King's college. Alan voelt zich daar als een vis in het water. In King's college is eccentriciteit eerder de norm dan de uitzondering; het enige wat telt is hoe briljant je bent in je studies.

De duur van een normale universitaire studie, wat we nu een bachelorstudie zouden noemen, duurt drie jaar in Engeland. Als beursstudent wordt van Alan verwacht dat hij het zwaarste curriculum volgt. Hij schrijft zich in voor Schedule B, wat we nu een master zouden noemen, waardoor hij in een extra vierde jaar geavanceerde vakken zal volgen die hem zullen brengen tot op het peil van het actueel wetenschappelijk onderzoek. Ondertussen blijkt al snel zijn wiskundig talent. Alan heeft de gewoonte om de dingen voor zichzelf te ontdekken en uit te vinden, zonder eerst uitvoerig in de literatuur uit te zoeken wat er al bekend en gedaan is. Op deze manier ontdekt hij op eigen kracht een stelling in de kanstheorie waarvan hij pas achteraf verneemt dat ze al bekend was als de befaamde Centrale Limietstelling, eerst aangetoond door de Finse wiskundige Lindeberg in 1922.

Stel dat je een bepaald experiment, dat we E noemen, een willekeurig aantal keren onafhankelijk van elkaar kan uitvoeren. De uitkomst van E

wordt telkens gemeten wat resulteert in een getal. Denk bijvoorbeeld aan het experiment waarbij je een willekeurig persoon van de bevolking uitkiest en de lengte van die persoon meet. Of denk aan het experiment waarbij je met een dobbelsteen rolt en bepaalt dat de meting 1 zal zijn als je zes hebt gegooid, en 0 anders. Voor een willekeurig natuurlijk getal N , bijvoorbeeld $N = 100$, herhaal je het experiment N keer, en je berekent het gemiddelde van alle metingen. Dus, je bepaalt de gemiddelde lengte van N willekeurige personen uit de bevolking. Of, je gooit N keer met een dobbelsteen en deelt het aantal keer dat je zes hebt gegooid door N . Merk op dat we deze procedure zelf ook kunnen bekijken als een omvangrijk experiment op zichzelf: we duiden dit aan als E_N . We kunnen nu E_N op haar beurt zo dikwijls uitvoeren als we willen, en een lijst bijhouden van alle resultaten. Wat Alan wiskundig aantoonde is dat deze resultaten zich zullen concentreren rond een centrale waarde, en dat de afwijkingen tegenover deze centrale waarde zijn verdeeld volgens de bekende Gausscurve. Het bijzondere aan deze stelling is dat ze geldt voor eender welk experiment E , zolang N maar groot genoeg is. Wanneer E het meten is van de lengte van een willekeurig persoon, zal de centrale waarde de gemiddelde lengte zijn over de gehele bevolking. Wanneer E het rollen van een dobbelsteen is, zal de centrale waarde $1/6$ bedragen, wat de kans is om zes te gooien.

Alhoewel het resultaat dus al bekend was, had Alan zijn wiskundig bewijs onafhankelijk gevonden en kon hij dit gebruiken als afstudeerwerk, dat hij indiende in het begin van zijn vierde en laatste studiejaar. Op basis van dit afstudeerwerk, en zijn behaalde examenresultaten, zal hij een vervolgbeurs verwerven als Fellow van King's College. Met die fellowship zal hij na zijn afstuderen nog zes jaar kunnen blijven aan de universiteit, inclusief maaltijden en wonen op kamers in de College. Naast het lesgeven aan studenten zal hij zich vrij kunnen wijden aan wetenschappelijk onderzoek.

Ondertussen volgt Alan in zijn vierde jaar de beruchte Schedule B vakken waaronder een vak van Max Newman (zelf een Fellow van King's College) over wiskundige logica en de grondslagen van de wiskunde. Newman doceert daar over het beruchte *Entscheidungsproblem* gesteld door David Hilbert, de absolute paus van de wiskunde in de periode rond de wisseling tussen de 19de en 20ste eeuw. In die periode ging de wiskunde door een soort van identiteitscrisis, wat zich concreet vertaalde in een aantal fundamentele vragen die gesteld werden over de grondslagen van de wiskunde. Men zegt wel eens dat de biologie gebaseerd is op de chemie; de chemie op de fysica; de fysica op de wiskunde. Maar waarop is de wiskunde gebaseerd? Uiteindelijk doet een wiskundige niets anders dan bewijzen zoeken van stellingen. Deze bewijzen zijn redeneringen, ketens van gevolgtrekkingen, die voldoen aan strikte logische regels, en die starten van een aantal basisaannames. Deze

basisaannames noemen we axioma's. De Italiaanse wiskundige Giuseppe Peano had zulke axioma's geformuleerd voor wiskundige beweringen over de natuurlijke getallen: de elementaire rekenkunde. Een voorbeeld van zulke bewering is "voor elk natuurlijk getal x bestaat er een getal dat strikt groter is", wat kan bewezen worden via het axioma dat $x + 1$ strikt groter is dan x .

Voorheen, reeds in 1900, had David Hilbert een basisvraag gesteld over de elementaire rekenkunde. Is de rekenkunde *volledig*? Met volledigheid bedoelen we dat voor elke mogelijke uitspraak P , we ofwel P kunnen bewijzen ofwel haar negatie. Volledigheid zou ons garanderen dat, tenminste in principe, elke vraag over de elementaire rekenkunde wiskundig kan beslecht worden. Of wel is de uitspraak waar, en kunnen we dit bewijzen; ofwel is de uitspraak niet waar, en kunnen we dat bewijzen. Hilbert was er van overtuigd dat de rekenkunde inderdaad volledig is: "in de natuurwetenschappen is er geen *ignorabimus*."

In 1931, hetzelfde jaar waarin Alan zijn eerste jaar aan de universiteit aanvangt, gooit echter een jonge Weense logicus, Kurt Gödel, alle dromen van David Hilbert over de consistentie en de volledigheid van de wiskunde aan diggelen. Hij toonde aan dat de elementaire rekenkunde niet volledig is. Gödel toonde de onvolledigheid aan door een variant te gebruiken van de paradox van de leugenaar, al bekend door de oude Grieken: "deze zin is niet waar". Hij verving dit door "deze zin is niet bewijsbaar" en de technische krachttoer bestond er natuurlijk in zulke uitspraak te herleiden tot een uitspraak louter over de elementaire rekenkunde. De onvolledigheidsstelling van Gödel sloeg in als een bom, en tot op vandaag breken filosofen zich het hoofd over de gevolgen ervan.

Hilbert had echter nog een andere vraag gesteld, die Gödel had opengelaten. In zijn befaamde Entscheidungsproblem, letterlijk vertaald Beslissingsprobleem, vroeg Hilbert of de wiskunde kon gemechaniseerd worden. Is er een effectieve methode waarmee we, gegeven een stel axioma's en een uitspraak P , kunnen beslissen of P kan bewezen worden uit de axioma's? Hilbert zelf was er vast van overtuigd dat dit kon. In principe zou dan iedereen wiskundige kunnen worden: het zou dan volstaan om nauwgezet de methode te volgen om aldus tot het resultaat te komen, hetzij dat P bewijsbaar is, hetzij dat dit niet zo is. Natuurlijk, er wordt hier niets gezegd over hoe lang het uitvoeren van die beslissingsmethode zou duren: misschien duurt het met de snelste computer van vandaag nog altijd langer dan 100 jaar, of erger, langer dan de nog verwachte levensduur van ons zonnestelsel bijvoorbeeld! Zelfs al zou zulke beslissingsmethode bestaan, zou er dus in de praktijk nog altijd een plaats zijn voor 'echte' wiskundigen die misschien via een briljant inzicht of andere creatieve redenering sneller tot een bewijs kunnen komen dan door slaafs de algemene methode te volgen.

In zijn geloof in het bestaan van een algemene beslissingsmethode deelde Hilbert de overtuiging van de grote 17de-eeuwse filosoof Gottfried Wilhelm Leibnitz. Die geloofde dat alle logische redeneringen konden herleid worden tot een wiskundige berekening, zodat elk meningsverschil zou kunnen opgelost worden door te zeggen: “Calcuemus!”

Ik sprak daarnet over de snelste computer, maar we mogen niet vergeten dat er in die tijd nog geen duidelijke notie van computer bestond. Het is op dit vlak dat Alan Turing, als jonge Fellow in 1935, een spectaculaire verandering zou brengen. Alan ging graag en lang lopen. Na een jog langs de rivier Cam lag hij wat uit te rusten in het gras van de Grantchester Meadows. Het is toen dat hij een theoretische machine bedacht die hij de *computing machine* noemde. Hij vond op dat moment de computer uit en later zou zijn theoretisch concept bekend worden als de Turing machine.

Alan zag een computing machine een beetje als een automatische typmachine met een lange rol papier. De rol is verdeeld in vakjes zodat in elk vakje één letter kan staan. Zoals een echte typmachine kan de machine heen en weer bewegen over de rol. Er is een correctielint, zodat een letter kan uitgewist worden en overschreven door een andere letter. Het bijzondere echter is dat de machine niet enkel kan schrijven maar ook de letter in het huidige vakje kan lezen. Verder is de machine automatisch: ze volgt een eindige lijst van instructies. Elke instructie specificeert voor elke mogelijke letter a uit een gekozen eindig alfabet (inclusief de blanco) wat er moet gebeuren indien het huidige vakje de letter a bevat:

1. Met welke letter moet a worden overschreven? (Dit kan a zelf zijn zodat a gewoon blijft staan.)
2. Bewegen we een stap naar rechts, naar links, of blijven we staan?
3. Met welke instructie gaan we nu verder?

De instructies worden traditioneel met genummerde q 's genoteerd, en de machine start met instructie q_0 . Om de machine te kunnen laten stoppen laten we nog een extra halt-instructie toe: wanneer deze instructie bereikt wordt, stopt de machine.

De volledige lijst van instructies kan neergeschreven worden in de vorm van een tabel, die er een beetje uitziet als een tafel van vermenigvuldiging. Een voorbeeld wordt gegeven in Tabel 1.

De rol, waarop letters geschreven staan, is het geheugen van de machine; de instructietabel is het programma van de machine. Wanneer de machine stopt is wat er dan geschreven staat op de rol, het resultaat of de uitvoer van de berekening. De machine kan gestart worden op een volledig blanco rol,

Tabel 1: *Instructietabel van een Turing machine met 4 instructies q_0, q_1, q_2 en q_3 . Wanneer gestart op een invoer bestaande uit N eentjes, gevolgd door een nul, gevolgd door M eentjes, zal de machine werken totdat de uitvoer bestaat uit $N + M$ eentjes. Bijvoorbeeld, de invoer ‘11101111’ zal getransformeerd worden in de uitvoer ‘1111111’. De machine kan aldus beschouwd worden als een machine voor optelling. Een blanco wordt voorgesteld door het vierkantje.*

| instructie | gelezen letter | geschreven letter | beweging | volgende instructie |
|------------|----------------|-------------------|----------|---------------------|
| q_0 | 1 | 1 | rechts | q_0 |
| q_0 | 0 | 0 | rechts | q_1 |
| q_1 | 1 | 1 | rechts | q_1 |
| q_1 | □ | □ | links | q_2 |
| q_2 | 1 | □ | links | q_3 |
| q_3 | 1 | 1 | links | q_3 |
| q_3 | 0 | 1 | blijf | halt |

maar evengoed kan ze gestart worden op een rol waarop reeds een rij letters is getypt: dit vormt dan de invoer van de berekening.

In zijn artikel, dat gepubliceerd werd in de Proceedings of the London Mathematical Society in 1936, argumenteert Alan zeer zorgvuldig dat eender welke effectieve methode die je maar kan bedenken, kan nagebootst worden door een Turing machine. In het bijzonder beschrijft Alan de *universele* Turing machine. Immers, het slaafs uitvoeren van de instructietabel van een gegeven Turing machine op een gegeven rij van letters als invoer, is evengoed een mechanische procedure. Ook deze procedure kan nagebootst worden door een Turing machine die we universeel noemen. Deze universele machine krijgt als invoer een instructietabel van een specifieke Turing machine M , gevolgd door een invoer W voor M , en bootst de werking van M op W na. We noemen zulke machine universeel omdat ze in één enkele machine alle mogelijke machines samenbalt. Turing geeft in zijn artikel een gedetailleerde instructietabel voor een universele machine. Hij geeft daarmee het eerste concrete ontwerp van de digitale, programmeerbare computer. Merk op dat zowel de instructies M als de invoer W samen in hetzelfde geheugen zitten (namelijk, ze staan samen op de rol). Dit idee was baanbrekend en zal het cruciaal onderdeel worden van de zogenaamde von Neumann-architectuur van moderne computers.⁴

Maar dat is nog niet alles. In zijn artikel bewijst Turing dat er problemen zijn die niet kunnen opgelost worden met een computer. Als een machine

⁴Zie het hoofdstuk Princeton.

M wordt losgelaten op een invoer W , zijn er twee mogelijkheden: ofwel zal de berekening ooit de halt-instructie bereiken en automatisch stoppen, ofwel niet. Beschouw nu het zogenaamde stop-probleem: we krijgen de instructietabel van een machine M en een mogelijke invoer W . We noteren dit gegeven als (M, W) . Onze taak is te beslissen of M , wanneer gestart op invoer W , zal stoppen of niet. We kunnen dit probleem niet gewoon oplossen met de universele machine, want als M niet stopt op W zal de universele machine op invoer (M, W) evenmin stoppen en ons dus geen uitsluitsel geven. Turing toont nu aan dat er simpelweg geen enkele machine H bedacht kan worden die het stop-probleem oplost. Concreet eisen we van zulke machine H dat ze, wanneer gestart op invoer (M, W) , uiteindelijk zal stoppen, met uitvoer 1 als M stopt op invoer W , en met uitvoer 0 anders.

Het bewijs gaat als volgt. Laat ons van de onderstelling uitgaan dat zulke H toch bestaat. Dan kunnen we een nieuwe machine D maken, met H als onderdeel, en met volgende werking. Als D gestart wordt op een invoer X , zal D het volgende doen. Eerst zal D de machine H nabootsen op invoer (X, X) . Als de uitvoer van H op invoer (X, X) gelijk is aan 1, gaat D naar een instructie die zichzelf altijd blijft herhalen, zodat de berekening in dat geval niet zal stoppen. Als de uitvoer van H op invoer (X, X) echter gelijk is aan 0, wordt de berekening gestopt. We vragen ons nu af: als we D uitvoeren op haar eigen instructietabel, dus met invoer D , wat gebeurt er dan? We komen nu tot een logische tegenspraak. Als D op invoer D stopt, kan dat enkel als H op invoer (D, D) de uitvoer 0 produceert. Maar omdat H het stop-probleem oplost, betekent dit dat D *niet* stopt op uitvoer D ! Op gelijkaardige wijze komen we tot de conclusie dat, als D op invoer D niet stopt, dat D dan op invoer D *wel* stopt. We komen dus op een logische tegenspraak, wat alleen maar kan betekenen dat onze onderstelling, dat er een machine H bestaat die het stop-probleem oplost, vals was.

Met een technisch moeilijker bewijs toont Turing vervolgens aan dat, als er een Turing machine zou bestaan die Hilberts Entscheidungsprobleem oplost, dat er dan ook een Turing machine zou bestaan die het stop-probleem oplost. Aangezien we net gezien hebben dat dit laatste onmogelijk is, is het eerste ook onmogelijk. De conclusie is dus dat het Entscheidungsprobleem onbeslisbaar is: er is geen effectieve methode die men consequent kan volgen om te controleren of een wiskundige uitspraak kan bewezen worden. De wiskunde is niet automatiseerbaar. De 22-jarige Alan zet de grote Hilbert in zijn hemd.

4 Princeton

Toen Alan zijn bevindingen voorlegde aan zijn leraar Max Newman zag die meteen het grote belang ervan in. Newman suggereerde dat Alan zou gaan samenwerken met Alonzo Church, een vooraanstaand logicus aan de universiteit van Princeton in de Verenigde Staten. Church had immers gelijkaardige ideeën ontwikkeld, die echter minder ver gingen dan die van Turing. Princeton was (en is nog steeds) een van de meest rijke en prestigieuze universiteiten van de Verenigde Staten, en met de opkomst van de nazi's in Duitsland en Oostenrijk zochten vele Europese topwetenschappers, dikwijls met Joodse wortels, een nieuw onderdak in Princeton. Einstein was er aangekomen in 1935, en Gödel zou enkele jaren later arriveren.

Alan zou uiteindelijk twee jaren in Princeton verblijven, van 1936 tot 1938. Hij werkt er met professor Church en behaalt er het PhD diploma.⁵ Belangrijker echter is dat hij John von Neumann ontmoet, die ook professor was in Princeton.

Von Neumann was de absolute topwetenschapper in de Verenigde Staten in die tijd. Hij was bekend voor zijn razendsnel brein en fotografisch geheugen, waardoor hij elk boek dat hij ooit gelezen had letterlijk kon reciteren. Hij had wiskunde gestudeerd bij David Hilbert. Afkomstig uit Budapest, van een rijke Joods-Hongaarse familie, was hij in 1931 met de opkomst van het nazisme verhuisd naar Princeton.⁶

Alan heeft zijn idee van de universele machine zeker besproken met von Neumann. De Amerikanen waren heel geïnteresseerd in rekenmachines, veelal met militaire toepassingen in gedachten zoals ballistiek, en tijdens de Tweede Wereldoorlog, het Manhattan Project, waar von Neumann een sleutelrol in speelde. De berekeningen voor de eerste atoombom zijn nog allemaal met de hand gemaakt, maar vlak na de oorlog schreef von Neumann het invloedrijke *Draft Report on the EDVAC* dat de architectuur beschrijft van een programmeerbare digitale computer. Het principe dat instructies voor de computer samen met de gegevens van de computer in hetzelfde geheugen zitten, is prominent aanwezig in de von Neumann-architectuur en het leidt weinig twijfel dat von Neumann hiervoor sterk geïnspireerd is geweest door de universele

⁵Wat misschien niet zo bekend is, is dat de graad van PhD, wat wij hier nu het doctoraat noemen, eigenlijk een Amerikaanse uitvinding is, die nadien in Europa is overgenomen. In Europa werd er lange tijd niet echt een kwalitatief onderscheid gemaakt tussen de titels licentiaat, master, of doctor; welke titel juist gebruikt werd hing eerder af van de universiteit, of van de discipline.

⁶Johnny (zo werd hij genoemd) was altijd perfect opgekleed: het verhaal gaat dat hij de Grand Canyon bezocht, waar je op een muilezel de canyon kan afdalen. Johnny deed dat in stijl, gekleed in een driedelig krijtstreepak.

Turing machine.⁷

In Princeton heeft Alan ook tijd om zijn uitvinderstalenten te ontwikkelen. Hij speelt al lang met het idee om een machine te maken die getallen kan vermenigvuldigen aan de hand van elektrische schakelingen. Via een vriend krijgt hij toegang tot de werkplaats van het fysicadepartement van de Princeton universiteit. Daar kan hij zijn ontwerp in elkaar knutselen en testen. Zijn ontwerp is gebaseerd op de binaire schrijfwijze van getallen. Wanneer we een willekeurig getal, b.v. 3704, opschrijven, gebruiken we de decimale schrijfwijze. Dit betekent dat 3704 bestaat uit 3 duizendtallen, 7 honderdtallen, 0 tientallen, en 4 eenheden, oftewel⁸

$$3704 = \underline{3} \times 10^3 + \underline{7} \times 10^2 + \underline{0} \times 10^1 + \underline{4} \times 10^0.$$

Er is echter niets bijzonder aan de basis 10, behalve misschien dat wij precies zoveel vingers hebben! Een machine heeft geen vingers en als we met elektrische schakelingen werken, die uit of aan kunnen staan, dus twee mogelijkheden hebben, is de basis 2 zoveel handiger. We noemen dit de binaire schrijfwijze, waar we niet denken in tientallen, hondertallen, duizendtallen, enzovoort, maar wel in tweetallen, viertallen, achttallen, enzovoort. De binaire schrijfwijze van het getal 3704 is 111001111000, want

$$\begin{aligned} 3704 = & \underline{1} \times 2^{11} + \underline{1} \times 2^{10} + \underline{1} \times 2^9 + \underline{0} \times 2^8 + \underline{0} \times 2^7 \\ & + \underline{1} \times 2^6 + \underline{1} \times 2^5 + \underline{1} \times 2^4 + \underline{1} \times 2^3 + \underline{0} \times 2^2 + \underline{0} \times 2^1 + \underline{0} \times 2^0. \end{aligned}$$

Een getal van N binaire cijfers, dus van N éénen en nullen, stelde Alan voor door N elektrische schakelaars: een 1 wordt voorgesteld door de schakelaar in te schakelen, en een 0 door hem uit te schakelen. Op dezelfde manier kunnen we nu twee getallen vermenigvuldigen door de schakelaars te verbinden in een elektrisch circuit van bijkomende schakelingen die verbonden zijn zodat de klassieke cijfermethode voor vermenigvuldiging (die evengoed voor de binaire schrijfwijze werkt als voor de decimale schrijfwijze) wordt uitgevoerd.

5 Oorlog

Alan keert in juni 1938 terug naar Engeland. Nu het meer en meer duidelijk wordt dat er een oorlog met Duitsland staat aan te komen, begint hij na

⁷We zullen echter zien dat de eerste computers niet door de Amerikanen, maar door de Britten zijn gebouwd.

⁸ 10^3 is tien verheven tot de derde macht, dus $10 \times 10 \times 10 = 1000$. We hebben $10^2 = 100$, $10^1 = 10$, en $10^0 = 1$ (een getal tot de nulde macht is altijd 1, per definitie).

te denken over geheime codes die de Duitsers niet zouden kunnen kraken. Zijn vermenigvuldigingsmachine was ook vanuit die motivatie ontstaan. Stel dat we een boodschap willen versturen, die we kunnen voorstellen als een getal. In plaats van het getal zomaar door te sturen, vermenigvuldigen we het eerst met een zorgvuldig geheim gehouden getal van een groot aantal cijfers. Het product is wat we doorsturen. De verzonden boodschap is in principe niet te kraken door iemand die het geheim getal niet kent. Iemand die het geheim getal wel kent, kan de ontvangen boodschap eenvoudig delen door het geheim getal om de oorspronkelijke boodschap te verkrijgen. Het geheim getal noemen we de *sleutel*. De wetenschappelijke discipline die het versleutelen van boodschappen bestudeert noemt men *cryptografie*.

Op aanbeveling van zijn professoren in Cambridge gaat Alan cryptografielessen volgen in de Government Code and Cypher School (GCCS), een nieuw opgerichte eenheid van de Britse geheime dienst. Het doel van GCCS is het kraken van de geheime codes gebruikt door de Duitsers. De school is gevestigd in Bletchley Park, een grote villa in het dorpje Bletchley 40 km ten noorden van Londen.⁹ Bij het uitbreken van de oorlog, in september 1939, meldt Alan zich aan voor dienst in Bletchley Park. Naast Alan bestaat het kernteam uit een handvol jonge briljante wiskundigen. Alan steekt er echter al snel bovenuit en wordt door iedereen aangesproken als “the Prof”.

Alle militaire communicatie gebeurde in Morse-code via radio-uitzending: iedereen die op de gebruikte frequentie afstemde kon meeluisteren. Daarom werden alle boodschappen versleuteld. Zowel de Duitsers als de Engelsen deden dit, en iedereen gebruikte hiervoor een gelijkaardig toestel. De Duitsers noemden hun toestel de Enigma; de Engelsen noemden hun toestel de Typex. De versleuteling van een boodschap door een Enigmatoestel gebeurde letter per letter. Voor elke letter is er een toets, zoals bij een elektrische typemachine. De toetsen hebben verbindingen naar een eerste rotor, die de 26 verbindingen door elkaar haspelt naar 26 andere. Deze 26 nieuwe verbindingen gaan naar een tweede rotor, die ze opnieuw door elkaar haspelt. Tenslotte is er ook een derde rotor. De verbindingen die uit de derde rotor komen worden er langs een andere weg ook terug ingestuurd; wat daar uitkomt gaat dan terug door de tweede rotor, en wat daar uitkomt gaat verder terug door de eerste rotor. De verbindingen die uiteindelijk uit de eerste rotor komen gaan naar lampjes die kunnen oplichten en het resultaat van de versleuteling tonen. Na elke letter draait de eerste rotor een slag door, waardoor een andere verhaspeling ontstaat. Wanneer de eerste rotor een volledige ronde heeft gedaan, draait de tweede rotor een slag door, en wanneer die een volledige ronde heeft gedaan,

⁹Bletchley Park is tegenwoordig ingericht als museum over het cryptografisch werk gedaan tijdens de oorlog, en zeker een toeristische aanrader.

draait de derde rotor een slag door. Dit is perfect te vergelijken met een kilometer teller.

Nu hadden Poolse cryptologen al heel belangrijk voorbereidend werk gedaan. Ze waren via Franse spionage te weten gekomen hoe de drie rotors precies geschakeld waren. Ze wisten dus hoe de verhaspeling gebeurde. Onbekend echter is de beginstand van de rotors wanneer de Duitse verzender begint te versleutelen. Dit zijn echter “slechts” $26 \times 26 \times 26 = 17576$ mogelijkheden. De Polen hadden daarvoor een speciale machine gebouwd die een Enigmatoestel nabootste en automatisch door alle standen kon lopen. Die machine werd de Bombe genoemd.

Was het echter maar zo simpel! Een eerste complicatie was dat de drie rotors verwisselbaar waren, en er een keuze was uit acht verschillende rotors. Een tweede complicatie was dat de Duitsers het Enigmatoestel hadden uitgebouwd met een stekkerbord, waardoor letters, alvorens ze naar de eerste rotor gingen, eerst nog konden verwisseld worden met een andere letter; ook de letters die uiteindelijk terug uit de eerste rotor kwamen werden nog een laatste keer door het stekkerbord verwisseld vooraleer ze naar de lampjes gingen. Er waren tien stekkerkabeltjes waarmee deze willekeurige verwisselingen konden ingesteld worden. Het breken van zulke ingewikkelde code kon alleen opgelost worden door een combinatie van geluk, militaire actie, genialiteit, en industrialiteit.

Op het niveau van industrialiteit had Winston Churchill himself ervoor gezorgd dat fabrieken gespecialiseerd in het maken van mechanische rekenmachines en telefooncentrales werden omgeturnd om Bombes te fabriceren. Op kruissnelheid hadden ze bijna honderd Bombes die 24 uur op 24 uur draaiden. Ze werden in shiften bediend door een heus legertje operatoren, meestal vrouwen van de Women’s Royal Naval Service (de ‘Wrens’). Later kwamen ook de Amerikanen in het spel en ook zij fabriceerden nog bijkomend honderd Bombes.

Maar zelfs honderden Bombes op zich waren volstrekt onvoldoende. Het aantal mogelijke combinaties van verwisselingen die de Duitser in zijn stekkerbord kon maken bedraagt immers meer dan honderd biljoen. De redding kwam door de genialiteit van Alan Turing. Hij zette een systeem op van logische gevolgtrekkingen, waardoor, wanneer men een bepaalde combinatie kon uitsluiten, men onmiddellijk miljoenen andere combinaties eveneens kon uitsluiten. Het was een beetje te vergelijken met het zoeken in een woordenboek. In een woordenboek staan duizenden woorden en we willen opzoeken of een bepaald woord er instaat. Doordat we de eerste letter van het woord kennen, kunnen we meteen de duizenden woorden die niet met die letter beginnen uitsluiten. Dit is slechts een analogie natuurlijk; in het geval van de ingewikkelde Enigma had Turing maanden koortsachtig denkwerk nodig om

zijn systeem van uitsluitingen op punt te zetten. De Wrens kregen gedetailleerde instructies voor de bediening van hun Bombes om het hele systeem te implementeren.

Maar Alans systeem van uitsluitingen op zich was nog steeds niet genoeg: er moesten eerst hints gevonden worden in de geheime boodschappen op basis waarvan de eerste uitsluitingen konden bepaald worden, die dan door Alans systeem een kettingreactie van nieuwe uitsluitingen zouden teweegbrengen. Hiermee was een flinke dosis geluk en militaire operatie gemoeid. Er werden een aantal geheime codeboeken bemachtigd door het overmeesteren van twee Duitse weerscheperen voor de kust van Noorwegen, en Engelse duikers zijn er ook in geslaagd de codeboeken uit een zinkende U-boot te recupereren zonder dat de Duitsers dit doorhadden. Met deze boeken kon het hele systeem dan opgestart worden.

Na een periode van heel wat tegenslagen kon men vanaf 1943 bijna de gehele communicatie van de U-boten ontcijferen. Ironisch genoeg hadden de Duitsers wel door dat er iets mis was, maar waren ze ervan overtuigd dat het ging om spionage van binnenin, dus Duitse verraders die posities van schepen en dergelijke verklikten aan de geallieerden. Het was voor hen ondenkbaar dat de geheime codes zelf ontcijferd werden. Ze zullen nooit gedacht hebben dat de Britten en Amerikanen een heuse industrie van Bombes zou ontwikkelen alleen maar voor het breken van codes. We mogen echter nooit vergeten dat deze industrie enkel iets heeft kunnen opbrengen door het geniale systeem bedacht door Alan Turing. De U-boten waren een echt probleem omdat ze systematisch de Amerikaanse transportkonvooien over de Atlantische Oceaan tot zinken brachten, alsook de gehele bevoorrading van Engeland bijna tot stilstand brachten. Eens dit probleem was overmeesterd, zo rond het begin van 1943, kon Amerika zijn volle macht inzetten. Eisenhower heeft later gezegd dat het team van codebrekers in Bletchley Park het einde van de oorlog zeker twee jaar vervroegd hebben.

Alan deed nog veel ander werk tijdens de oorlog. Hij ontwikkelde de statistische methoden voor het kraken van een geheel andere code, die het Duitse opperbevel hanteerde. Hiervoor werd ook Max Newman, zijn oude leraar logica aan Cambridge, aangetrokken. Het team van Newman ontwikkelde een nieuw Bombe-achtig toestel, de Colossus genoemd, dat revolutionair was omdat het met electronica werkte, wat veel sneller ging dan de mechanische werking van de Bombes. De Colossus was de eerste elektronische rekenmachine ooit gemaakt.

In de laatste jaren, toen de hele codebrekersindustrie op volle toeren draaide, was Alan een beetje overbodig geworden. The Prof werd nog wel geconsulteerd voor wiskundige vragen, maar het kleine team van wiskundigen waarmee alles was begonnen was uitgegroeid tot een groot industrieel appa-

raat waar hij zijn draai niet meer kon vinden. Hij was niet gepromoveerd tot manager binnen dat hele apparaat omdat hij de nodige managerstalenten miste; hij rendeerde enkel in een kleinschalige, intellectuele omgeving. Hij speelde wel nog een heel belangrijke rol als liaison tussen Engeland en Amerika. Hij ging naar Amerika om de systemen voor het breken van de codes ginder over te brengen. In de beroemde Bell Laboratories, kreeg hij een speciale toelating van het Witte Huis om het onderzoek in alle departementen te mogen observeren. De Amerikanen waren ook bezig met het versleutelen van spraak: ze hadden een versleutelde telefoonlijn opgezet tussen het Witte Huis en Winston Churchill. Hierdoor geïnspireerd begon Alan, eens terug in Engeland in de laatste oorlogsjaren, een klein projectje op zichzelf, waarin hij zich vrij kon uitleven als uitvinder. Samen met nog één collega ontwikkelden ze een betere machine voor het versleutelen van spraak. Uiteindelijk mogen we wel zeggen dat Alan best gelukkig was tijdens zijn periode in Bletchley Park. In zijn vrije tijd ging hij nog dikwijls lopen en zoeken naar paddestoelen in de vrije natuur.

6 De eerste computers

De oorlog had grote technologische progressie gezien, en de verwachtingen waren hoog gespannen om dit in vreedstijd verder te zetten. Vlak na de oorlog hadden de Amerikanen onder leiding van John von Neumann plannen om een eerste echte computer, equivalent aan de universele Turing machine te bouwen. De Britten waren echter vast van plan om de Amerikanen voor te zijn. Ze hadden tenslotte Turing zelf in hun rangen. Het eerste initiatief komt van Charles Darwin, de kleinzoon van 'de' Charles Darwin van de evolutietheorie. Darwin is directeur van het National Physical Laboratory en trekt de wiskundige John Womersley aan als departementshoofd. Womersley had het von Neumann rapport in handen gekregen en was vastbesloten de Amerikan voor te zijn. Womersley had eerder al het artikel van Turing gelezen en biedt Alan een post aan als Temporary Senior Scientific Officer. Alan hapt toe. Hij schrijft een omstandig rapport waar een concreet ontwerp wordt voorgesteld voor de ACE: de Automated Computing Engine.

Alans visie op de ACE is, uiteraard, sterk gebaseerd op zijn universele Turing machine. De belangrijkste vraag is het geheugen: wat moet er in de plaats komen van de rol papier van de theoretische machine? Er moest een elektronisch geheugen komen dat genoeg capaciteit had, en dat snel kon werken. Magnetische band was al uitgevonden, maar dat was veel te traag om te dienen als inwendig geheugen voor de computer. Door de moderne chip-technologie zijn de hedendaagse computergeheugens extreem snel, extreem

klein, en hebben ze een extreem grote capaciteit. Het is echter interessant ons te herinneren hoe, voor de uitvinding van de transistor en de chip, de allereerste snelle computergeheugens werden gerealiseerd.

Er waren twee alternatieven: de delay line en de beeldbuis. De delay line was gebaseerd op het principe dat, als je aan één eind van een metalen staaf een tikje geeft, de geluidsgolf zich zal verplaatsen door de staaf naar de andere kant. Als je met een stethoscoop aan het andere eind zou luisteren, zou je het tikje pas een tijdje later horen. Gedurende die tijd zou je kunnen zeggen dat de staaf de puls heeft ‘opgeslagen’. Een getal in binaire schrijfwijze kunnen we dus opslagen door regelmatig, bijvoorbeeld elke microseconde, een geluidspuls aan het linkereind van de staaf aan te brengen. Meer specifiek, voor elke 1 in het getal geven we een puls; voor elke 0 geven we geen puls. Deze pulsen reizen door de staaf en kunnen aan de rechterkant opgevangen en afgelezen worden; om ze niet te verliezen worden ze ook terug naar links gestuurd zodat alles herhaald wordt.

De beeldbuis, zoals we ze kennen van de eerste TV's, bestaat uit een raster van puntjes. Elk puntje kan beschoten worden met elektronen, waardoor het oplicht. Een opgelicht puntje heeft dus een elektrische lading en komt overeen met het opslaan van een 1, terwijl een donker puntje geen lading heeft en overeenkomt met 0. Door een metalen plaat achter de beeldbuis te plaatsen kunnen we deze ladingen uitmeten.

Alhoewel Alans rapport reeds medio 1946 klaar is (hij kiest voor delay lines omdat die goedkoper zijn en hij daardoor een groter geheugen kan hebben), ondervindt de bouw van de ACE grote vertraging. De oorlog is voorbij en de bureaucraten hebben het terug voor het zeggen, zeker in een staatsapparaat zoals het National Physical Laboratory. Alan kan zijn frustraties hierover verwerken in het lopen. Hij wordt lid van een atletiekclub en legt zich serieus toe op het marathonlopen. Hij wordt echt goed en is op weg om zich te kwalificeren voor de Olympische Spelen van Londen 1948. Een kwetsuur aan de heup beslist daar uiteindelijk anders over.

Ondertussen is Max Newman, die in Bletchley Park ervaring had opgedaan met de Colossus, professor geworden in de universiteit van Manchester. Hij start er zijn eigen computerproject op met steun van de Royal Society. Ze hebben een eerste werkend prototype van een computer in juni 1948 (dit prototype gebruikte een beeldbuis als geheugen). Wanneer de minister van defensie een demonstratie krijgt, is hij laaiend enthousiast en investeert met een tienduizend pond om een commerciële model te bouwen in samenwerking met het Manchesterse electronicabedrijf Ferranti.

Newman biedt Alan de positie van Deputy Director van de Royal Society Computing Laboratory aan in Manchester, en Alan hapt toe, gefrustreerd door de vertragingen van zijn eigen ACE project. Ze ontwerpen een tweede

prototype dat dan door Ferranti zal gecommercialiseerd worden. Alan is eigenlijk de eerste professionele computerprogrammeur ter wereld. Hij schrijft de eerste handleiding voor programmeurs.

7 Kunstmatige Intelligentie

Uiteindelijk zal in 1950 toch een pilootmodel van de ACE gerealiseerd worden, maar Alan is ondertussen al veel verder in zijn denken. Hij denkt na over intelligente machines en schrijft hierover twee rapporten, waarvan het tweede zal gepubliceerd worden in het filosofisch tijdschrift *Mind*. Met dat artikel zal hij minstens even beroemd worden als met zijn artikel over de Turing machine. Hij start op eigen houtje de discipline van de Kunstmatige Intelligentie (Artificial Intelligence, AI).

Een belangrijk inzicht dat Alan heeft is dat computers kunnen leren. Je zou kunnen argumenteren dat dit onmogelijk is: een computer voert enkel slaafs zijn instructies uit, heeft dus geen creativiteit, kan niet bijleren. Turing wijst ons echter op de eigenschap die we reeds herhaaldelijk hebben benadrukt: de instructies voor de computer zitten in hetzelfde geheugen als de gegevens. Je kan een computer zeker zo programmeren dat hij tijdens zijn werking de gegevens in het geheugen leest en aanpast. Op dezelfde wijze kan de computer dus ook zijn instructies aanpassen. Dit kan beschouwd worden als leergedrag.

Turing behandelt dan de vraag “kunnen machines denken?” Hij stelt dat deze vraag zinloos is, omdat ze te vatbaar is voor interpretatie: we zouden oeverloos kunnen blijven discussiëren over wat het betekent dat een machine denkt. In de plaats daarvan ontwerpt hij een concrete test. Hij vindt eigenlijk een vorm van communicatie uit die we nu kennen als ‘chatten’: twee personen zitten aan een toetsenbord met scherm, in verschillende kamers, en corresponderen schriftelijk met elkaar. Turing stelt nu voor een computer intelligent te noemen als hij één van de twee personen kan vervangen zonder dat de andere persoon dat doorheeft. Turing stelt wel voor het gesprek voornamelijk te laten gaan over intellectuele onderwerpen zoals schaken, kaartspelen, talen, of wiskunde. Op die manier hoeft het niet te gaan over lichamelijke onderwerpen.

Turing was erg optimistisch dat een intelligente, lerende machine kon gebouwd worden. Uiteraard zou zulke computer snel moeten kunnen werken, en een groot geheugen moeten hebben. In 1950 voorspelde hij dat in het jaar 2000 een computer met een geheugen van 100 Megabyte zou kunnen gebouwd worden, die minstens 30% zou halen op een intelligentietest van vijf minuten. Hij bedoelde daarmee dat als je de computer met 100 willekeurige personen

zou laten corresponderen gedurende vijf minuten, er minstens 30 personen zouden geloven dat ze met een andere persoon bezig waren. Die 100 Megabyte hebben we gemakkelijk gehaald, maar de Turing intelligentietest is voorlopig nog een onbereikbaar doel.

Er is wel degelijk al veel vooruitgang geboekt. In 1997 versloeg de computer Deep Blue wereldkampioen Garry Kasparov in een schaakwedstrijd over zes partijen. Sindsdien is de computer sterker dan de mens in schaken. Dat zou Alan Turing veel plezier hebben gedaan, want hij is altijd gefascineerd geweest in computerschaak. In 2011 versloeg de computer Watson twee winnaars van het Amerikaans kwisprogramma Jeopardy. Dit is een kwis gebaseerd op feitenkennis. Maar computers die creatief zijn, flexibel bijleren, kortom, intelligent zijn, zijn er zeker nog niet. Het is ook absoluut niet duidelijk of dit ooit zal gebeuren.

8 Dood

Turing is homosexueel, wat destijds illegaal was in Groot-Brittanië. Hij wordt in 1952 veroordeeld tot een hormonentherapie, en verliest zijn beschermde status bij de geheime dienst. Alan was immers nog steeds verbonden met de geheime dienst, alhoewel we daar niets van weten. Trouwens zijn werk tijdens de oorlog moest ook top secret blijven; het verhaal van Bletchley Park zou pas publiek worden in de jaren 1970 en 1980. Hij doet nog baanbrekend wetenschappelijk onderzoek naar morfogenese (de vraag hoe precies vormen en patronen kunnen ontstaan in het biologisch groeiproces). Zijn artikel hierover verschijnt in 1952 in de Transactions of the Royal Society. Anno 2014 heeft dit artikel zelfs meer citaties in de wetenschappelijke literatuur dan zijn artikels over de Turing machine en over intelligente machines! Het toont opnieuw zijn veelzijdigheid en zijn groot genie aan. Kortom, hij houdt de schijn op bij collega's en vrienden. In werkelijkheid echter is hij zeer zwaar aangeslagen door de veroordeling en kan hij niet leven met het besef dat zijn geaardheid illegaal is. In volle koude oorlog en periode van homofobie is hij aangeschoten wild tegenover de Britse geheime dienst: de man die teveel weet. Hij pleegt zelfmoord in 1954. Het is een schok voor iedereen. Zijn moeder zal altijd blijven volhouden dat het om een ongeluk ging. Alan had cyanide in huis voor zijn scheikundeproeven. Hij is gevonden op bed met naast hem een half opgegeten appel. Alan at namelijk elke avond nog een appel. Die op 7 juni 1954 is zijn laatste geweest.

9 Erfenis

In 1962 sticht de Association for Computing Machinery, de grootste professionele vereniging van informatici, de ACM Turing Award, als het equivalent voor informatica van de Nobelprijs.

Zelfs de modernste computers anno 2014 voldoen nog steeds aan de principes van de Turing machine zoals bedacht door Alan Turing in 1935.

Elke student informatica aan de universiteit leert over de Turing machine als theoretische grondslag van de computer.

Onderzoek naar machinaal leren en Kunstmatige Intelligentie is nog steeds wereldwijd in volle gang.

Op 24 december 2013 geeft koningin Elizabeth, na herhaaldelijke petitie door de wetenschappelijke wereld, eindelijk een postuum koninklijk pardon aan Alan Turing waardoor zijn veroordeling in 1952 feitelijk wordt opgeheven.