Alan Turing
Cientista Universal

José Carlos Espírito Santo (editor)

UMinho Editora

Coleção Ciência e Cultura para Todos

UMinho Editora
Ciência e Cultura para Todos

EDIÇÃO
José Carlos Espírito Santo

A produção deste livro foi apoiada pela FCT
através do projeto UID/MAT/00013/2013

CAPA
Cláudia Mendes Araújo

DESIGN
Tiago Rodrigues

PAGINAÇÃO
Carlos Sousa | Bookpaper

IMPRESSÃO e ACABAMENTOS Tipografia Priscos, Lda.

EDIÇÃO UMinho Editora

LOCAL DE EDIÇÃO Braga 2019

DEPÓSITO LEGAL N.º 464875/19

ISBN impresso 978-989-8974-02-0

ISBN digital 978-989-8974-03-7

DOI: https://doi.org/10.21814/uminho.ed.5

Os conteúdos apresentados (textos e imagens) são da exclusiva responsabilidade dos respetivos autores.
© Autores / Universidade do Minho – Proibida a reprodução, no todo ou em parte, por qualquer meio, sem autorização expressa dos autores.

DEPARTAMENTO DE MATEMÁTICA – UNIVERSIDADE DO MINHO

Alan Turing: cientista universal

Este volume celebra Alan Turing (1912 - 1954), o grande matemático e lógico cujo trabalho direta ou indiretamente acabou por revolucionar as nossas vidas. Seja porque em 1936, ainda estudante, definiu matematicamente o conceito moderno de computador e estabeleceu e exemplificou os limites da computabilidade; seja pelo contributo, por muitos considerado decisivo, que deu nos serviços secretos britânicos durante a II Guerra Mundial; seja porque, imediatamente após a Guerra, participou na construção dos primeiros computadores britânicos; seja porque, precocemente, discutiu se os computadores poderiam pensar, e para isso inventou o "jogo da imitação"; seja por muitas outras contribuições para a ciência, o trabalho de Turing tem proporções universais: porque cobre vastas áreas do saber sem respeito por barreiras disciplinares; porque rasga novos horizontes, criando novas disciplinas; porque é um fermento de ideias que, combinadas com a revolução tecnológica das últimas décadas, mudou para sempre a face da sociedade.

O reconhecimento de Turing pelos meios científicos e académicos foi imediato e crescente. A partir dos anos 1950, o estudo matemático da computabilidade e da complexidade computacional funda-se nos instrumentos deixados por Turing. Nos anos 1960 é criado o *Turing Award,* considerado o Prémio Nobel da Computação. Em 1997 um computador vence o campeão do mundo de xadrez, prenúncio do triunfo da Inteligência Artificial que observamos nos nossos dias, que é também uma vindicação tardia dos primeiros contributos de Turing nessa direção, que incluem os esforços, à altura excêntricos, de criar programas que jogassem xadrez. E, em 2012, todo o mundo académico celebrou o *Turing Year*, o centenário do nascimento: organizaram-se conferências locais e globais, editaram-se inúmeros volumes, descerraram-se placas e batizaram-se edifícios.

Fora dos meios académicos, o reconhecimento de Turing tem sido muito mais lento. Durante décadas, sobretudo antes da publicação da biografia de Andrew Hodges em 1983, um manto de obscuridade encobriu Turing. Para isso certamente contribuíram os contornos da perseguição judicial de que foi alvo, e o carácter secreto do seu trabalho durante o período da II Guerra Mundial. Mas também a muito lenta perceção na sociedade de que o trabalho de Turing é uma das raízes da revolução tecnológica que vivemos – esta última associada mais facilmente a outros protagonistas bem reconhecidos na esfera económica e mediática. Entretanto, Turing foi alvo de uma reabilitação oficial pelas autoridades do seu país e até já protagoniza filmes do cinema comercial. Por estes dias, Turing foi escolhido pelo Banco de Inglaterra para figurar na nova nota de 50 libras.

Este volume pretende contribuir para um reconhecimento mais alargado e informado de Alan Turing. Reúne-se aqui um conjunto de reflexões sobre a figura, trabalho, legado e impacto do cientista, que se pretendem acessíveis a um público não-especialista, e representativas da academia portuguesa. A abrir, explica-se por que razão "Turing está entre nós", e faz-se uma viagem abrangente do trabalho de Turing e suas implicações presentes e futuras. A fechar, dois testemunhos, inspirados por Turing. No miolo do volume, seis capítulos, organizados em dois grupos, todos eles aprofundando aspetos do capítulo inicial.

No primeiro desses grupos, constituído pelos capítulos 2 e 3, discutem-se certas impossibilidades demonstradas por Turing no início da sua carreira, especificamente a impossibilidade de se reconhecer algoritmicamente se uma dada fórmula é uma verdade lógica, a qual, por sua vez assenta na impossibilidade de se decidir algoritmicamente se a execução de um dado programa desencadeada por um dado *input* vai terminar ou não.

No segundo desses grupos, constituído pelos capítulos 4 a 7, agregam-se textos que, ao invés, exploram as vastas possibilidades abertas por Turing, curiosamente podendo todos eles ser vagamente associados à área da Inteligência Artificial, de que Turing foi um pioneiro. Por um lado, temos a possibilidade de a mente estar para o cérebro tal como o programa está para o computador. Como se pode ler no capítulo 4, esta ideia de a mente ser uma entidade computacional (e matematizável – ver o capítulo 6) é muito fecunda em Filosofia da Mente, e tem raízes antigas (ver o capítulo 5). Por outro lado, temos a possibilidade de se mecanizar vastas áreas do processo dedutivo, com implicações óbvias na forma como se ensina e faz a Matemática. Uma ilustração é dada para a área da Geometria no capítulo 7.

Veja-se a Figura 1 para uma descrição esquemática da organização do volume.

O editor gostaria de agradecer: aos autores, por terem correspondido ao desafio de contribuir para este livro e, depois, terem esperado pacientemente que ele viesse a lume; a Manuel Curado, pela sugestão inicial de produzir este volume; a Cláudia Mendes Araújo, pelo design da capa; a Filipe Mena e Maria Antónia Forjaz, pelo trabalho de revisão; ao Centro de Matemática da Universidade do Minho e à Fundação para a Ciência e Tecnologia, pelo apoio institucional[1]; à UMinho Editora, por ter imediatamente acarinhado a ideia de editar este livro; e sobretudo à Tony, pelo forte apoio na fase final de produção.

José Carlos Espírito Santo

Braga, 3 de Setembro de 2019

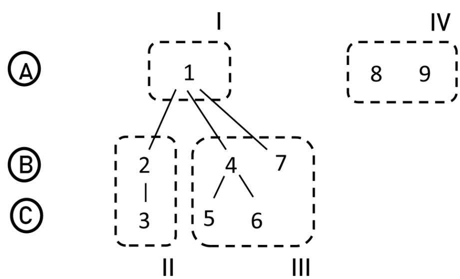

Figura 1. Como ler este volume.

Os capítulos que compõem o volume estão numerados sequencialmente de 1 a 9 e organizados em quatro partes temáticas, numeradas de I a IV. O conjunto de textos está ainda estratificado em três níveis de "detalhe", designados por A, B e C. Por exemplo, o nível A é constituído pelos capítulos 1, 8 e 9. Todos os capítulos são autocontidos e podem, por isso, ser lidos independentemente. Porém, cada capítulo no nível B (respectivamente C) pode ser lido como detalhando ou complementando algum aspecto daquele capítulo no nível A (respectivamente B) a que ele está ligado, de acordo com o diagrama.

1 A produção deste livro foi apoiada pela FCT através do projeto UID/MAT/00013/2013.

LISTA DE AUTORES

José António Alves, Doutor em Filosofia da Mente,
Universidade do Minho

E. R. de Arantes e Oliveira, Professor Catedrático Aposentado,
Instituto Superior Técnico

José Carlos Espírito Santo, Professor Auxiliar, Universidade do Minho

Fernando Ferreira, Professor Catedrático, Universidade de Lisboa

Sofia Miguens, Professora Associada, Universidade do Porto

Luís Moniz Pereira, Professor Catedrático Aposentado,
Universidade Nova de Lisboa

Olga Pombo, Professora Auxiliar com Agregação Aposentada,
Universidade de Lisboa

Pedro Quaresma, Professor Auxiliar, Universidade de Coimbra

José Manuel Valença, Professor Catedrático, Universidade do Minho

Parte I – Turing

1. Turing Está Entre Nós[1]

Luís Moniz Pereira[2]

Resumo

A relevância presente e futura de Turing advém da intemporalidade das questões que ele atacou e a luz inovadora que ele lançou sobre elas.

Turing, em primeiro lugar, definiu os limites algorítmicos da computabilidade, quando determinada por mecanismo *efetivo*, e mostrou a generalidade da sua definição ao provar a sua equivalência com outras

1 Este capítulo foi originalmente publicado em: Luís Moniz Pereira, "Turing is among us", Journal of Logic and Computation, 22(6), 2012. A tradução é da responsabilidade do editor.

2 O autor agradece o suporte do projeto NOVA-LINCS UID/CEC/04516/2013.

formulações gerais, mas menos algorítmicas, não mecânicas, mais abstratas, da computabilidade.

Na verdade, a sua originalidade impressionou muito Gödel, pela simplicidade do mecanismo invocado – aquilo que hoje em dia nós chamamos de Máquina de Turing (ou programa) – e pela demonstração da existência de uma Máquina de Turing Universal (aquilo que nós chamamos de computador digital) – a qual pode demonstravelmente imitar qualquer outra Máquina de Turing, ou seja, executar qualquer programa. De facto, as Máquinas de Turing baseiam-se simplesmente num autómato de estados finitos (como uma máquina de venda automática) e numa fita ilimitada de papel constituída por quadrículas discretas (como um rolo de papel), com, no máximo, um símbolo que se pode reescrever em cada quadrícula.

Turing foi também o primeiro a introduzir implicitamente a perspetiva do "funcionalismo" – embora ele não tenha usado a palavra, a qual foi introduzida mais tarde por Putnam, inspirado pelo trabalho de Turing – ao mostrar que o que conta é a capacidade de realização de funções, independentemente do *hardware* que as incarna.

E essa capacidade assenta na simplicidade do mecanismo por si idealizado, o que ele então chamou de máquinas-A (mas agora tem o seu nome), o qual depende apenas da manipulação de símbolos – tão discretos como os dedos de uma mão – em que quer os dados quer as instruções são representadas com símbolos, ambos sujeitos a manipulação. O par, os dados, bem como as instruções, estão armazenados em memória, com as instruções no duplo papel de dados e regras de ação – a ideia de programa armazenado em memória.

Ninguém até hoje inventou um processo mecânico computacional com tais propriedades gerais que não possa ser aproximado teoricamente com precisão arbitrária por alguma Máquina de Turing em que as interações são capturadas pelo conceito inovador de oráculo de Turing.

Nestes dias de quantização de tempo-discreto, processos biológicos computacionais, e demonstração de universo em permanente expansão – o autómato e a fita – a Máquina de Turing reina suprema. Mais ainda, o funcionalismo universal – outra essência de Turing – é o que possibilita a inevitável congregação dos fantasmas nas várias máquinas em que se encarnam (baseadas em silício, biológicas, extraterrestres ou outras) para promover a sua simbiótica coevolução epistémica, pois tais fantasmas compartilham o mesmo funcionalismo teórico.

Turing está verdadeiramente e para sempre entre nós.

1. Alan Turing e a Computação

Alan Turing atreveu-se a perguntar se a máquina podia pensar. As suas contribuições para compreender e responder a esta e outras questões desafiam uma classificação convencional. No início deste século XXI, o conceito de 1936 de máquina de Turing surge não apenas em matemática e ciências da computação, mas também em ciências cognitivas e biologia teórica. O seu artigo "Computing Machinery and Intelligence" (Turing 1950), descrevendo o chamado teste de Turing, é a pedra angular da teoria da Inteligência Artificial (Hodges 1997). Pelo meio, Turing desempenhou um papel vital no desenlace da Segunda Guerra Mundial, e produziu sozinho um plano visionário para a construção e uso de um computador eletrónico (Hodges 1983). Ele pensou e viveu uma geração à frente do seu tempo, e ainda assim os atributos do seu pensamento, que rompeu as fronteiras dos 1940s, estão bem vivos entre nós hoje.

O trabalho de Gödel deixou por resolver a questão hilbertiana da *decidibilidade*, o *Entscheidungsproblem*, ou seja a questão da existência de um método bem determinado que, pelo menos em princípio, pode ser aplicado a uma dada proposição para decidir se a proposição é demonstrável. Esta questão sobreviveu à análise de Gödel porque a sua resolução requeria uma definição precisa e convincente de *método*. Isto é o que Turing alcançou, trabalhando sozinho um ano até Abril de 1936; a sua ideia, agora conhecida por *Máquina de Turing*, seria publicada mesmo no fim de 1936, no artigo *On Computable Numbers, with an Application to the Entscheidungsproblem* (Turing 1936). É característico de Turing que ele tenha refrescado a questão de Hilbert reformulando-a em termos, não de provas, mas de números computáveis. A reformulação sustentava a afirmação de ter encontrado uma ideia central para a matemática. Tal como o título dizia, o *Entscheidungsproblem* era apenas uma aplicação de uma nova ideia, a de computabilidade (Hodges 1997).

Aquilo que hoje em dia chamamos de *Máquina de Turing* é um autómato de estados finitos suplementado com uma "fita" (o análogo do papel) correndo através dele, e dividida em secções (chamadas de "quadrículas"), cada uma capaz de conter um "símbolo". Em cada momento, há apenas uma quadrícula que "está na máquina". Podemos chamar a esta quadrícula a "quadrícula examinada". Ao símbolo na quadrícula examinada podemos chamar "símbolo examinado". O símbolo examinado é o único do qual a máquina está, por assim dizer, "diretamente ciente". De seguida, Turing especifica o reportório de ações à disposição da máquina imaginada. Cada ação é completamente determinada pela "configuração" em que a máquina está e pelo símbolo que ela está nesse momento a examinar. É esta completa determinação que faz dela uma "máquina". A ação é limitada ao seguinte: em cada passo, ela (1) ou apaga o símbolo ou imprime um símbolo especificado, (2) move uma quadrícula para a esquerda ou para a direita, (3) muda para uma nova configuração. Versões ligeiramente diferentes da ideia de Turing são dadas em vários livros de texto, e a forma técnica precisa que ele deu originalmente não é importante; a essência é que a ação é completamente dada pelo que Turing chama de "tabela de comportamento" para a máquina, ditando o que ela fará em cada configuração e símbolo examinado: este determinismo é o que a torna "mecânica". Cada "tabela de comportamento" é uma máquina de Turing diferente. As ações são altamente restritas em forma, mas a tese de Turing é que elas formam um conjunto de elementos atómicos a partir dos quais todas as operações matemáticas podem ser compostas. De facto, num estilo muito inusual para um artigo matemático, é dada argumentação em termos muito gerais, justificando a suficiência das ações da máquina de Turing para abranger o método mais geral possível de computar (Casti & DePauli 2000, Leavitt 2011, Minsky 1967, Pereira 1970):

> "Vou também supor que o número de símbolos que podem ser impressos é finito. Se permitíssemos uma infinidade de símbolos, então haveria símbolos diferindo numa extensão arbitrariamente pequena. O efeito desta restrição no número de símbolos não é grave. É sempre possível usar sequências de símbolos no lugar de símbolos individuais." (Turing 1936)

Deste modo, Turing afirma que um reportório finito de símbolos na realidade permite uma infinidade contável de símbolos, mas não uma infinidade de símbolos imediatamente reconhecíveis. Note-se também que a fita tem de ter tamanho ilimitado, embora em cada momento o número de símbolos na fita seja finito. Os números computáveis são então definidos como aquelas dízimas infinitas que podem ser escritas por uma máquina de Turing, começando com uma fita em branco.

Assim, Turing abordou a questão das funções computáveis pela direção oposta à usual, isto é, pelo ponto de vista dos números produzidos como resultado, não pelo ponto de vista de quais funções podem ser construídas a partir de um conjunto de funções primitivas. Turing começou com a ideia informal de computador – que em 1935 significava, não uma máquina calculadora, mas um ser humano equipado com lápis, papel e tempo. Depois, ele substituiu componentes por componentes não-ambíguos, até que nada além de uma definição formal de "computável" restou. Com este objetivo, Turing introduziu duas assunções fundamentais: discretização do tempo e do estado mental. Deste modo, a máquina de Turing incorpora a relação entre uma sequência ilimitada de símbolos no espaço, e uma sequência de eventos no tempo, regulada por um número finito de estados mentais. O título "On Computable Numbers" (em vez de "On Computable Functions") assinala uma mudança fundamental. Antes de Turing, faziam-se coisas aos números. Depois de Turing, os números começaram a fazer coisas. Ao mostrar que uma máquina podia ser codificada como um número, e um número descodificado como uma máquina, "On Computable Numbers" levou a números (agora chamados de "software") que eram "computáveis" de um modo que era inteiramente novo (Dyson 2012). Ele esboçou a demonstração de que π é um número computável, juntamente com todos os números reais definidos pelos métodos comuns de equações e limites em matemática.

Munido com a sua nova definição de computável, é depois fácil de mostrar que existem números incomputáveis. O ponto crucial é que a tabela de comportamento de qualquer máquina de Turing é finita. Logo, todas as possíveis tabelas de comportamento podem ser listadas em ordem alfabética: isto mostra que os números computáveis são contáveis. Como os números reais são incontáveis, segue que a maioria deles é incomputável (Hodges 1983).

Uma inspeção mostra que o problema vem de identificar aquelas máquinas de Turing que não produzem uma infinidade de dígitos. Esta não é uma operação computável: isto é, não há máquina de Turing que inspecione a tabela de uma outra qualquer máquina e decida se ela vai produzir uma infinidade de dígitos ou não. Isto pode ser visto mais diretamente: se existisse tal máquina, ela poderia ser aplicada a si própria, e esta ideia pode ser usada para obter uma contradição. Hoje em dia, isto é conhecido como o facto de que o *problema da paragem* não pode ser decidido por uma máquina. A partir desta descoberta de um problema que não pode ser decidido por uma máquina, não é um passo difícil empregar o cálculo formal da lógica matemática, e responder ao hilbertiano *Entscheidungsproblem* pela negativa (*ibidem*).

Alonzo Church, o eminente lógico americano em Princeton, anunciou a mesma conclusão ou tese sobre o *Entscheidungsproblem*. A tese de Church era a afirmação de que a calculabilidade efetiva podia ser identificada com as operações do elegante e surpreendente formalismo churchiano, o cálculo-lambda (a partir do qual a linguagem de programação “Lisp” surgiu). Turing escreveu um apêndice (Turing 1936, 1995) relacionando o seu resultado com o de Church, sob quem estudou em Princeton por um período, a partir de 1936, depois de já ter produzido o seu esboço de artigo. Em Março de 1937, Alonzo Church reviu “On Computable Numbers” para o *Journal of Symbolic Logic*, e cunhou o termo *máquina de Turing*. “Computabilidade por uma máquina de Turing”, escreveu Church, “tem a vantagem de fazer a identificação com a efetividade no sentido comum (não explicitamente definido) imediatamente evidente.”

A tese de Church é agora, por vezes, chamada de tese de Church-Turing, mas a tese de Turing é diferente, trazendo o mundo físico para a figura, com uma afirmação sobre o que pode ser feito. “On Computable Numbers” não só resolveu uma questão maior e proeminente posta por Hilbert, e abriu o novo campo matemático da computabilidade, e ofereceu uma nova análise da atividade mental, mas também teve uma implicação prática: lançou os princípios do computador através do conceito de máquina de Turing universal (M. Davis 1958). De facto, as várias e diversas tentativas de formal e rigorosamente caracterizar a, *a priori*, “intuitiva” noção de função efetivamente computável, por abordagens muito diferentes, mostraram-se no fim todas equivalentes, desse modo sustentando o consenso de que a intuição fora definitivamente capturada. Entre estas abordagens temos: definibilidade-lambda de Church, computabilidade de Turing, os sistemas canónicos de Post, os sistemas elementares formais de Smullyan, recursividade geral de Herbrand-Gödel-Kleen (Davis 1958, Minsky 1967, Pereira 1970).

A ideia da máquina Universal é facilmente sugerida. Logo que a especificação de qualquer máquina de Turing é dada por uma tabela de comportamento, seguir o rasto da operação dessa máquina torna-se uma questão mecânica de procurar entradas na tabela e imitá-las. Porque é mecânica, a máquina de Turing pode fazê-lo: quer dizer, uma única máquina de Turing pode ser desenhada para a ter a propriedade de que, uma vez que lhe seja fornecida a tabela de comportamento de outra máquina de Turing, ela fará o que quer que essa outra máquina teria feito. Turing chamou a essa máquina, a máquina Universal (Dyson 2012, Hodges 1983, Leavitt 2011, Minsky 1967, Pereira 1970).

A máquina Universal concede a Turing o crédito de ter inventado o princípio do computador – e não só de um modo abstrato. Não se pode estudar as máquinas de Turing sem as ver como programas de computador executáveis, armazenados em símbolos, juntamente com os dados na memória da fita – "o programa armazenado em memória modificável". A máquina universal sendo o computador no qual qualquer programa pode correr. Quer as instruções do programa, quer as ações que elas desencadeiam são descritas por símbolos na máquina universal. Mais ainda, os programas podem eles próprios ser manipulados na fita, mesmo auto-modificáveis – uma possibilidade que a IA só recentemente começou a explorar (Alferes *et al.* 2012, Pereira & Lopes 2009, Pereira & Han 2009).

A construção da máquina de Turing mostrou como tornar "mecânicas" todas as demonstrações formais; e no artigo de 1936 tais operações mecânicas haveriam de ser escolhidas como triviais, pondo ao invés sob o microscópio os passos não-mecânicos remanescentes. Ele de seguida abandona a ideia de que os momentos de intuição correspondem a operações incomputáveis. Ao invés, decidiu ele, o âmbito do computável abrangia muito mais do que o que podia ser capturado por explícitas notas de instrução, e mais do que suficiente para incluir tudo o que os cérebros humanos faziam, por mais criativo e original que fosse.

Máquinas suficientemente complexas teriam a capacidade de evoluir para comportamento nunca antes explicitamente programado. A afirmação de Turing é que as únicas características do cérebro relevantes para o pensamento ou a inteligência são aquelas que caem dentro do nível de descrição da máquina de estados discretos.

O Turing do pós-guerra afirma que as máquinas de Turing podem imitar o efeito de *qualquer* atividade da mente, não apenas da mente cumprindo um "método bem determinado". Turing é claro quanto a máquinas de estados discretos incluírem máquinas com capacidades de aprendizagem e auto-organização, e enfatiza o facto destas ainda caberem no âmbito do computável. Turing chama a atenção para o aparente conflito com o facto da definição das máquinas de Turing terem tabelas de comportamento fixas, mas esboça uma demonstração de que as máquinas auto-modificáveis ainda são de facto definidas por um conjunto de instruções inalterado. Turing advoga duas abordagens diferentes – em linguagem moderna *top-down* e *bottom-up* – que de facto derivam da sua descrição de modelo de máquina de 1936. Notas de instrução explícitas tornam-se programação explícita; estados mentais implícitos tornam-se os estados da máquina atingidos por aprendizagem e experiência de auto-organização (Hodges 1983, McDermott 2001).

2. Gödel, Computabilidade e Turing

A terceira questão hilbertiana dirigida ao congresso internacional de Bolonha em 1928, a da *decidibilidade*, veio a ser formulada em termos de *demonstrabilidade*, em vez de *verdade*, uma vez que os resultados de Gödel não tinham eliminado a possibilidade de poder existir uma forma de distinguir as asserções demonstráveis das não-demonstráveis. Isto significava que a peculiar asserção auto-referente de Gödel poderia ser separada de algum modo das restantes. Poderia haver um *método* bem-definido, *i.e.* um procedimento *mecânico*, que pudesse ser aplicado a qualquer afirmação matemática, e que pudesse decidir se a afirmação era, ou não era, derivável num dado sistema formal? (Pereira 2007, Wang 1973, Webb 1980).

Entre 1929 e 1930, Gödel já tinha resolvido a maior parte dos problemas fundamentais levantados pela escola de Hilbert. Uma das questões que restavam era a de encontrar um conceito preciso que caracterizaria a noção intuitiva de computabilidade. Mas não era claro, na altura, que este problema admitiria uma resposta definitiva. Provavelmente, Gödel foi surpreendido pela solução de Turing, que era mais elegante e conclusiva do que ele esperava. Gödel compreende perfeitamente, no início dos '30s, que o conceito de sistema formal está intimamente ligado ao de procedimento mecânico, e ele considera o trabalho de Alan Turing de 1936 (sobre os números computáveis) como um importante complemento ao seu próprio trabalho sobre os limites da formalização.

Em 1934, Gödel deu palestras no Institute of Advanced Studies em Princeton, onde visitou Alonzo Church (que comunicava com o orientador inglês de Turing e viria a ser seu orientador de PhD em Princeton) e recomendou a análise de Turing de procedimentos mecânicos (publicada em 1936) como um avanço essencial que poderia vir a elevar os seus teoremas da incompletude a uma forma mais acabada. Em 1964 Gödel acrescenta sobre isso um *Postscriptum* às suas palestras de 1934. Em consequência do trabalho de Turing, esses teoremas podem ser vistos como "aplicáveis a *qualquer* sistema formal consistente que contenha parte da teoria dos números finitária". Ao longo dos anos, Gödel reconheceu regularmente o artigo de Turing de 1936 como o trabalho definitivo que captura o conceito intuitivo de computabilidade, e o único autor a apresentar argumentos persuasivos sobre a adequação do conceito preciso que ele próprio definiu (Wang 1987).

Relativamente ao conceito de procedimento mecânico, os teoremas da incompletude de Gödel também pediam naturalmente uma definição exata (como a que Turing veio a produzir), através da qual se pudesse dizer que eles se aplicavam a qualquer sistema formal, *i.e.*, a qualquer sistema no qual demonstrações pudessem ser verificadas por meio de um procedimento automático. Na realidade, o programa de Hilbert incluía o *Entscheidungsproblem* (literalmente "problema de decisão"), que visava determinar se haveria um procedimento para decidir se, em lógica elementar (também conhecida por lógica de primeira-ordem), uma dada proposição era derivável ou não pelas regras de Frege. Isto exigia um conceito preciso de procedimento automático, caso a resposta fosse negativa (o que é o caso) (*ibidem*).

Com este objetivo, em 1934, Gödel introduz o conceito de função recursiva geral, que mais tarde se demonstrou capturar o conceito intuitivo de computabilidade. Gödel sugeriu o conceito e Kleene trabalhou nele. A génese do conceito de função recursiva geral estava implícita na demonstração de Gödel da incompletude da aritmética. Quando Gödel provou que o conceito de demonstração usando regras "do tipo do xadrez" era um conceito "aritmético", ele estava de facto a dizer que uma demonstração podia ser realizada por um método "bem-definido". Esta ideia, uma vez formalizada e de algum modo estendida, deu origem à definição de "função recursiva". Foi mais tarde verificado que estas eram exatamente equivalentes às funções computáveis (Casti & DePauli 2000, Wang 1973, Webb 1980).

Após algum tempo, Gödel veio a reconhecer que a conceção de "máquina de Turing" oferece a definição mais satisfatória de "procedimento mecânico". De acordo com o próprio Gödel, o trabalho de Turing de 1936 sobre os números computáveis é o primeiro a apresentar uma análise convincente de tal procedimento, mostrando a perspetiva correcta segundo a qual se pode claramente entender o conceito intuitivo. "Com este conceito conseguiu-se dar pela primeira vez uma definição absoluta … não dependendo do formalismo escolhido", admitiu ele em 1946 (Dyson 2012).

Sobre a definição rigorosa do conceito de computabilidade levada a cabo por Turing, Gödel diz, esclarece Wang: "Esta definição certamente não era supérflua. No entanto, se o termo 'mecanicamente computável' não tivesse tido já uma definição clara anterior, se bem que não-analítica (i.e. sintética), a questão de a definição de Turing ser ou não adequada não teria sentido, embora a sua definição tenha indubitavelmente uma resposta pela positiva" (Wang 1987).

Logo que o conceito rigoroso, tal como definido por Turing, seja aceite como o correto, um passo simples é suficiente não só para para ver que os teoremas da incompletude de Gödel se aplicam a sistemas formais em geral, mas também para mostrar que o *Entscheidungsproblem* é insolúvel. A demonstração desta insolubilidade pelo próprio Turing mostrou que uma classe de problemas que não podem ser resolvidos pelas suas ``máquinas-A" (de "máquinas Automáticas", hoje conhecidas por máquinas de Turing) pode ser expressa por proposições em lógica elementar (Hodges 1983).

Na sua tese de doutoramento, completada em Maio de 1938 e publicada como “Sistemas de Lógica baseados em Ordinais” em 1939, Turing tentou encontrar uma saída para o poder dos teoremas da incompletude de Gödel. A ideia fundamental era a de adicionar ao sistema inicial sucessivos axiomas, de modo que ele ficasse incrementalmente mais completo, fazendo passos não-deterministas de vez em quando ao consultar “uma espécie de oráculo, por assim dizer, que não pode ser uma máquina”. Cada asserção “verdadeira mas indemonstrável” é adicionada como novo axioma. No entanto, desta forma, a aritmética adquire a natureza de uma Hidra, porque, uma vez adicionado o novo axioma, uma nova asserção de tal tipo será produzida para ser então tomada em consideração. Não é, assim, suficiente, adicionar um número *finito* de axiomas, antes é necessário adicionar um número infinito, o que era claramente contra o sonho finitista de Hilbert. Se fosse possível produzir um gerador finito de tais axiomas, então a teoria inicial também seria finita e, como tal, sujeita ao teorema de Gödel. Turing mostrou, porém, que afirmações indecidíveis, resistentes à assistência de um oráculo externo, poderiam ainda ser construídas, e o *Entscheidungsproblem* permaneceria insolúvel (Webb 1980).

Outra questão é que existe um número infinito de possíveis sequências pelas quais se podem adicionar tais axiomas, conduzindo a diferentes e mais complexas teorias. Turing descreveu as suas diferentes extensões aos axiomas da aritmética como “lógicas ordinais”. Nestas, a regra para gerar os novos axiomas é dada por um “processo mecânico” que pode ser aplicado a uma “fórmula ordinal”, mas a determinação se a fórmula é “ordinal” *não* é mecânica. Turing comparou a identificação de uma fórmula “ordinal” ao trabalho da intuição, e considerou os seus resultados dececionantemente negativos, pois embora existissem “lógicas completas”, elas sofriam do defeito de não se poder contar quantos passos “intuitivos” eram necessários para demonstrar um teorema (Hodges 1983).

Este trabalho, no entanto, teve um efeito lateral persistentemente agradável, a saber, a introdução do conceito de “máquina de Turing com oráculo”, precisamente para que lhe fosse permitido, a partir dela, perguntar e obter do exterior a resposta a um problema insolúvel (como o da identificação de uma “fórmula ordinal”). Ele introduziu a noção de computabilidade relativa, ou insolubilidade relativa, que abriu um novo domínio em lógica matemática, e em ciências da computação. A ligação, feita por S. A. Cook em 1971, entre máquinas de Turing e o cálculo proposicional levaria ao estudo de questões centrais sobre complexidade computacional (*ibidem*).

3. *Mens ex Machina*

Vale a pena mencionar que, ao contrário de Turing, Gödel não estava interessado no desenvolvimento de computadores. A sua mecânica está tão ligada às operações e aos conceitos da lógica, que, hoje em dia, é comum lógicos estarem envolvidos, de uma forma ou outra, no estudo de computadores e ciências da computação. No entanto, os famosos teoremas da incompletude de Gödel demonstraram e estabeleceram a inexorabilidade da matemática e as limitações dos sistemas formais e, de acordo com alguns, dos programas de computador. Isto relaciona-se com a bem conhecida questão sobre se a mente supera a máquina.

Assim, o interesse crescente dado aos computadores a à Inteligência Artificial (IA) conduziu a um incremento generalizado no interesse sobre o trabalho do próprio Gödel. Mas, assim o reconhece o próprio Gödel, o seu teorema não resolve a questão de saber se a mente supera a máquina. De facto, o trabalho de Gödel nesta direção parece favorecer (em vez de contrariar) a posição *mecanicista* (até mesmo *finitista*) como uma abordagem à automação de sistemas formais (Pereira 2007, Wang 1987).

Existe uma conhecida ambiguidade entre a noção de *mecanismo* confinado ao mecânico (no sentido preciso de computável ou recursivo) e a noção de *mecanismo* materialista. Gödel enuncia duas proposições: (i) O cérebro opera basicamente como um computador digital. (ii) As leis da física, nas suas consequências observáveis, têm um limite finito de precisão. Ele é de opinião que (i) é muito plausível, e que (ii) é praticamente certa. Talvez a interpretação que Gödel atribui a (ii) seja o que a faz compatível com a existência de leis físicas não-mecânicas, e do mesmo passo ele liga-a a (i), no sentido em que, tanto quanto podemos observar do comportamento do cérebro, ele funciona como um computador digital, como uma máquina de Turing Universal (*ibidem*).

4. A Intuição Matemática é Algorítmico?

Roger Penrose (1994) afirma que *não*, e sustenta muito do seu argumento, tal como J. R. Lucas antes dele (Lucas 1969), no teorema da incompletude de Gödel: é a *intuição* que nos permite *ver* que a asserção de Gödel, indecidível num dado sistema formal, é por isso mesmo verdadeira. Como poderia esta intuição ser o resultado de um algoritmo? Penrose insiste que o seu argumento teria sido "certamente considerado pelo próprio Gödel nos 1930's e nunca foi devidamente refutado desde então ..." (Davis 1990, 1993).

No entanto, na sua palestra Gibbs, proferida perante a American Mathematical Society em 1951, Gödel contradiz abertamente Penrose (Casti & DePauli 2000):

"Por um lado, na base do que tem sido demonstrado até agora, permanece possível que uma máquina demonstradora de teoremas, de facto equivalente à intuição matemática, possa existir (e até ser descoberta empiricamente), embora isso não possa ser *demonstrado*, nem mesmo demonstrado que ela apenas obtém teoremas corretos da teoria de números finitária."

Na realidade, durante os 1930's, Gödel foi especialmente cauteloso em evitar afirmações controversas, limitando-se ao que podia ser demonstrado. No entanto, a sua palestra Gibbs foi uma verdadeira surpresa. Gödel argumentou insistentemente que o seu teorema tinha importantes implicações filosóficas. Apesar disso, e tal como a citação acima deixa claro, ele nunca afirmou que a intuição matemática podia ser demonstrada não-algorítmica.

É plausível que Gödel concordasse com o juízo de Penrose de que a intuição matemática não podia ser o produto de um algoritmo. De facto, Gödel aparentemente acreditava que a mente humana nem podia ser o produto da evolução natural. No entanto, Gödel nunca afirmou que tais conclusões fossem consequência do seu famoso teorema (Wang 1987).

Devido à prevalecente tendência presente de restringir a discussão sobre os limites da racionalidade, em contraposição ao discernimento, aos bem-definidos e surpreendentes avanços na ciência, tecnologia e programação de computadores, a perspetiva de considerar a razão em termos de capacidades mecânicas recebeu muita atenção nas últimas décadas. Tal é reconhecido como sendo o núcleo do estudo da IA, o que é claramente relevante para o desejo de Gödel, e enraizado no sucesso de Turing, de separar mente e máquina.

Nesta posição, a IA estaria primariamente interessada no que é fazível do ponto de vista da computabilidade, cujo interesse formal envolve apenas uma parte limitada da matemática e da lógica. No entanto, o estudo das limitações da IA não pode ser reduzido a esta restrição do seu âmbito. A este respeito, é essencial distinguir entre *algoritmos para solução de problemas*, e *algoritmos simpliciter*, enquanto conjunto de regras a seguir de um modo sistemático e automático, que são possivelmente auto-modificáveis, como Turing ousou, e *sem* necessariamente terem um problema específico e bem-definido para resolver (Davis 1990, Pereira 2007).

5. Prolegómenos de Neurologia Artificial

Pontes potenciais entre neurologia e ciências da computação merecem ser exploradas, com especial ênfase na distinção *harware/software*. Primeiro, algumas definições.

Aquilo que é essencialmente o resultado de uma intenção é artificial, mesmo que tenha intencionalidade – as suas intenções são artificiais. A intenção sob a qual não existe outra é natural. No entanto, não há nada mais artificial do que a definição de "natural" (Pereira 1979,1988).

Por exemplo, se o universo foi intencionado por um ser todo-poderoso que desse modo realizou as suas intenções, então todo ele é artificial, incluindo é claro as nossas intenções. Outro exemplo: um computador construído por um ser humano, e que manifeste intenções, é inteiramente artificial, independentemente do ser humano, por sua vez, ser ou não artificial.

Reconhecidamente, o uso das palavras muda. Consigo antever que a palavra "natural", em virtude do desenvolvimento da Inteligência Artificial, possa ser identicamente aplicada a muitos computadores, cujas intenções, originalmente artificiais, já não serão vislumbráveis pelos seus construtores humanos. Então, a criatura confisca independência ao seu criador.

Esta secção refere-se a um mundo artificial onde existem cérebros. Portanto, eu omitirei sempre a palavra “artificial”, e por isso escreverei “neurónio” em vez de “neurónio artificial”, “intenção” em vez de “intenção artificial”, etc.

O mundo artificial a que aludimos, onde existem cérebros, foi construído por seres intencionais, indeterminados e anónimos, como um laboratório experimental para a possível confirmação das suas conceções sobre o seu próprio mundo. Em particular, os cérebros nesse lugar são usados como dispositivos epistemológicos onde os construtores desse mundo testam, *in vitro*, algumas das suas conceções sobre os seus próprios processos cognitivos, metafísica, etc.

Tais cérebros funcionam de acordo com princípios previamente testados pelos construtores do mundo nos seus próprios computadores, embora o substrato fisiológico e físico de uns (os computadores) e outros (os cérebros) difiram substancialmente, o par diferindo ainda do substrato material suportando a atividade mental dos construtores do mundo.

Este estado de coisas não evita alguns serem tomados como modelos de outros. Pelo contrário, é precisamente no princípio da independência do *software* do *harware*, amplamente confirmado pela sua Ciência da Computação, que os construtores do mundo baseiam as suas construções. Consequentemente, ao modelarem os seus processos cognitivos, eles exploram as potencialidades do modelo para aperfeiçoarem, por um ciclo de *feedback* cognitivo, as suas próprias capacidades mentais.

O recurso à nomenclatura, paradigmas e técnicas das ciências da computação para a modelação do cérebro não era novo para eles. De facto, as ciências da computação tinham sido, no processo, enriquecidas com a nomenclatura, paradigmas e técnicas das ciências do cérebro.

Historicamente, essa fertilização cruzada começou quando foi reconhecido que só com a ajuda de um instrumento com a complexidade acumulada e organizada de um sistema computacional (computador, periféricos e programas) seria possível lidar de forma rigorosa com a complexidade dos processos cerebrais e as estruturas dos construtores do mundo e, subsequentemente, com as dos cérebros artificiais que eles desejavam construir.

Em particular, os modelos implementados em computador tornam-se bem-definidos, eminentemente observáveis na sua formulação e sua dinâmica, e podem ser transformados incrementalmente de uma forma expedita. Por outro lado, a dinâmica observável dos modelos liberta a neurologia da sua excessiva ênfase na patologia das lesões, e permite-lhe adotar uma metodologia mais em linha com o estudo do funcionamento normal do cérebro. Mas a própria patologia ganha um novo ímpeto com a possibilidade de simular, no modelo, uma gama de lesões específicas e provocadas.

6. A Distinção *Software/Hardware*

Recordemos, de seguida, o reconhecidamente mais importante princípio originador da introdução do computador como laboratório para modelos do cérebro (Pereira 1979, Pereira 1988).

O princípio da distinção entre *software* e *hardware* – entre função e forma, afinal de contas –, na sua versão mais simples, e que de algum modo está presente em toda a máquina – surge finalmente cristalina com o advento do computador digital e o seu precursor conceptual, a máquina de Turing Universal.

A diversidade de tecnologias empregues para obter a mesma função, desde os primeiros computadores, de facto confirma-o. Um mesmo programa é executável em diferentes máquinas, precisamente porque, ao nível de discurso do programa, os detalhes da sua execução, abaixo de um certo nível averiguável de análise, são irrelevantes, desde que um idêntico nível de discurso seja produzido para os resultados. Numa analogia grosseira, podemos dizer que a cor da tinta e a caligrafia são irrelevantes para a mensagem a ser transmitida.

Esta distinção, diga-se, é suscetível de níveis. Aquilo que é *hardware* não é necessariamente coisas físicas, mas antes aquilo que, a um certo nível de análise, é considerado fixo, dado, e cuja análise ou não analisabilidade pode ser irrelevante para um certo fim – e.g. instruções RISC, ou o DNA das células da glia.

Historicamente, nos primeiros computadores, esse nível coincidia com o nível das partes físicas da máquina – daí a confusão. Subsequentemente, esse nível moveu-se em direções opostas, e a sua relatividade tornou-se clara. Por outro lado, o conceito de máquina abstrata foi introduzido, isto é, uma dada e não analisável coleção fixa de instruções, definida matematicamente, capaz de suportar um conjunto de funções de *software*, independentemente dos processos físicos e detalhes que decretam a implementação da máquina abstrata na máquina física. Por outro lado, os componentes físicos fixos de uma geração de computadores deram lugar, na geração seguinte de computadores, a componentes parcialmente programáveis, cujas funções são determináveis por *software* (microprogramação). Uma antiga analogia grosseira neste caso seria uma máquina de escrever IBM com uma "typeball" selecionável. Melhor ainda, algumas funções de *software* previamente definidas, tais como aritmética de vírgula flutuante, tornaram-se completamente *hardware* disponíveis.

Uma das consequências principais, para a neurologia artificial, deste sempre presente princípio – cujos precursores foram as primeiras escolas de pensamento axiomático – consiste numa melhor focagem no nível de análise neurológica mais apropriado para responder às suas questões.

Outras das consequências principais tem sido, é claro, a crescente popularidade, entre os neurologistas, do computador como instrumento de simulação, dada a vantagem de ser capaz de escolher o nível de abstração da simulação, incluindo ao nível do neurónio. Nesta tarefa, eles serviram-se da teoria das caixas pretas de abstração, sucessivamente desenvolvida com sucesso pelos cientistas da computação.

Muitos neurologistas, de facto, não tiveram dificuldade em ajustar-se à ideia de simulação do neurónio porque eles reconheceram que a sua própria conceção do neurónio já era um modelo muito abstrato dele, em contraste com os cientistas da computação, os quais podem conhecer intimamente os "circuitos" do computador, desde que estes sejam construídos de acordo com especificações.

A distinção *software/hardware* é rica em consequências para a neurologia artificial. Nomeadamente, ela explica por que razão a correspondência entre função e o seu suporte físico não é compulsória. O *harware* físico não é específico de nenhuma função de *software* de alto nível. Pelo contrário, ela autoriza a execução de uma variedade de funções, de uma forma distribuída e não-localizada, exceção seja feita ao *hardware* específico para o *interface* com órgãos periféricos, e para codificação/descodificação de informação externa, como no caso do sistema nervoso (Churchland 2002).

À medida que os processos cerebrais ganham em abstração e nível de integração de diversas fontes sensoriais, os neurónios que os suportam tornam-se menos específicos e independentes da origem da informação. Como poderia ser a integração possível se não fosse assim?

É um facto que, para além do *interface* com o exterior, existe *hardware* específico que também é especializado. Mas essa especificidade podia concebivelmente ser realizada de outras formas, apesar da química orgânica, de modo que nenhum *software* requer efetivamente um *hardware* específico.

Outra consequência da distinção *harware/software* concerne a noção de nível apropriado de explicação. Um programa pode ser compreendido, na sua função ou disfunção, em termos dele próprio, ou do seu nível de discurso. É claro, a sua disfunção pode originar numa disfunção do *harware* subjacente, mas nesse caso ela manifesta-se através de um comportamento bizarro, não compreensível ao nível de discurso do programa, e não específico desse programa.

Complementarmente, a sua função pode ser descrita recorrendo ao nível do *hardware*, mas tal descrição não constitui um nível apropriado de explicação, porque, ao ser demasiado detalhado, não pode ser generalizado. A analogia seguinte, adaptada de Putnam, é um exemplo do que quero dizer.

Imagine um painel rígido, com um buraco circular de 10 cm de diâmetro, e um buraco quadrado com 10 cm de lado. Pretende-se explicar por que razão um cubo rígido, de 9 cm de lado, atravessa um dos buracos mas não o outro. O nível adequado de explicação recorre aos conceitos e princípios geométricos envolvidos. Um possível mas inapropriado nível de explicação consideraria as propriedades físico-quânticas dos materiais presentes, digamos vidro do painel e alumínio do cubo, e explica a impossibilidade da passagem do cubo através do círculo em termos de resistência mecânica, para cada trajetória de aproximação.

Tal explicação, porque demasiado específica, só com dificuldade se generaliza a outros materiais constituintes, digamos ferro e granito. Há um nível de generalidade abaixo do qual a explicação perde em generalidade e se torna desnecessária, desde que todas propriedades relevantes estejam garantidamente fixas a esse nível – no exemplo dado, a invariância da forma dos elementos em presença, relativamente a qualquer trajetória seguida e/ou a sua composição física.

Finalmente, num computador, o *software* prevalece, em geral, sobre o *hardware*. Embora o *hardware* suporte e cause a execução do *software*, a iniciativa pertence, mais vezes sim do que não, ao *software*. É o *software* que escolhe e provoca a entrada em atividade do *hardware* apropriado em cada passo – o programa armazenado em memória de Turing.

Tal atividade consiste em consultar as instruções armazenadas em memória, e em executar as instruções do *software* no *hardware*, com o resultado de que o *hardware* selecionado por instruções é levado à atividade, fechando o círculo. Deste modo, a teleologia do *software* é mantida ao comando, não obstante a subjacente causalidade do *hardware* físico.

Logo que a máquina de Turing leia o primeiro símbolo examinado da sua fita, as instruções em presença assumem o controlo. Similarmente, logo que o computador é inicializado, ele começa a executar as instruções na sua memória. Interrupções exteriores e atualizações à sua memória através de oráculos podem ocorrer, apenas para que o mesmo padrão seja seguido inexoravelmente. Afinal de contas, queremos que o computador faça aquilo que o *software* lhe diz para fazer.

7. Lógica e Consciência

Se se perguntar "Como podemos introduzir o inconsciente nos computadores?", alguns responderão que os computadores são totalmente inconscientes. Na verdade, o que não sabemos é como introduzir consciência nos algoritmos, porque nós usamos os computadores como apêndices inconscientes da nossa própria consciência. A questão, como tal, é prematura, tendo em vista que nós só poderemos referir-nos à conceção humana de inconsciência depois de introduzirmos consciência na máquina. De facto, nós compreendemos muito melhor a inconsciência computacional do que a nossa própria inconsciência (Pereira 2007).

No início dos anos 1980's, William Reinhardt estava interessado em quanto uma máquina de Turing poderia saber sobre ela própria. Ele conjeturou que em aritmética epistémica – aritmética de Peano enriquecida com um operador de conhecimento – uma máquina de Turing pode demonstrar, com certeza matemática, a frase "Eu sei que eu sou uma máquina de Turing". Timothy Carlson deu a primeira demonstração da conjetura de Reinhardt em meados dos anos 1990's (Buechner 2007).

Estas férteis questões indicam a complexidade dos nossos processos de pensamento, incluindo os da criatividade e intuição, que em grande medida nós não compreendemos, e põem um desafio muito maior à IA, que nos pode ajudar providenciando um espelho indispensável, não apenas epistemológico, mas também simbiótico.

A tradução, para uma construção computacional, de algum modelo funcional, como aludido acima, de uma consciência introspetiva e por isso auto-referente, seria permitida usando quaisquer metodologias e paradigmas de programação presentemente ao nosso dispor. À luz desta constatação, poder-se-ia estar inclinado a perguntar por que razão o uso de um paradigma baseado em lógica (via programação em lógica, digamos) é precisamente aquele que preferimos. Há muitos argumentos que podem ser levantados contra o seu uso. Por isso, vamos tentar reproduzir nesta secção os mais relevantes, replicar-lhes, e apresentar os nossos próprios argumentos a favor desse paradigma.

O primeiro argumento a ser levantado nestas discussões é que o raciocínio humano regular não usa lógica, existindo processos complexos e não-simbólicos no cérebro que supostamente não podem ser emulados pelo processamento simbólico.

Seguindo esta linha de pensamento, muitos modelos têm sido produzidos baseados em redes neuronais artificiais, em propriedades emergentes de sistemas puramente reativos, e muitos outros, numa tentativa de escapar à tirania da GOFAI (“Good Old Fashioned AI”), enraizada no computacionalismo simbólico de Turing – ele próprio surgido para responder negativamente ao hilbertiano *Entscheidungsproblem* sobre a decidibilidade da lógica aumentada com aritmética. No entanto, há um embaraço nestes modelos: a sua implementação pelos seus proponentes, acaba por ser, sem qualquer desconforto, num computador, o que só pode ajudar a usar processamento simbólico para simular esses outros paradigmas.

A relação deste argumento com a lógica é assegurada pela perspetiva filosófica do funcionalismo: a lógica pode ela própria ser implementada sobre um sistema de processamento simbólico, independentemente do substrato físico que o suporta. E as redes neuronais podem implementar uma máquina de Turing Universal, quando não a lógica diretamente.

Logo que um processo seja descrito em lógica, podemos usar a sua descrição para sintetizar um artefacto com aquelas mesmas propriedades. Conquanto seja um modelo computacional, qualquer tentativa de escapar à lógica demonstrar-se-á não ser ela própria inerentemente mais poderosa.

Na realidade, trabalho recente de Valiant, d’Avila Garcez, Woods e outros (Valiant 2000, 2003, d’Avila Garcez *et al* 2006, 2007, Bruza *et al* 2009) demonstrou as redes neuronais artificiais e o raciocínio lógico poderem ser combinados num modelo cognitivo computacional unificado. De facto, Valiant, d’Avila Garcez e Woods mostraram que há modelos fundados nos quais a computação é executada por um modelo neural (conexionista) que implementa diversas regras de raciocínio (não)-clássicas. A lógica, nestes sistemas, é usada como uma linguagem de representação de conhecimento, e pode também ser empregue para oferecer explicações sobre o processo de raciocínio. A lógica robusta de Valiant e a Lógica Modal Conexionista de d’Avila Garcez e outros combinam ambas a aparente natureza estatística da aprendizagem com a natureza lógica do processo de raciocínio, este por vezes expresso por fragmentos de linguagens de programação em lógica. Em certa medida, ao invocar os seus grafos de conexão, o trabalho de Kowalski (Kowalski 2011) também dá suporte ao desenvolvimento de modelos cognitivos computacionais que integrem aprendizagem (automática) e raciocínio lógico.

Por outro lado, há uma óbvia capacidade humana de compreender o raciocínio lógico, uma capacidade desenvolvida no decurso da evolução do cérebro. A sua expressão mais poderosa hoje é a própria ciência, e o conhecimento acumulado por numerosas disciplinas, cada uma das quais com as suas próprias nuances lógicas dedicadas ao raciocínio no seu domínio. Das leis de estado nacionais à física quântica, a lógica, no seu sentido geral, tornou-se o pilar sobre o qual o conhecimento humano é construído e melhorado, a recompensa definitiva pelo nosso domínio da linguagem.

Os humanos podem usar a linguagem sem aprender gramática. No entanto, se temos de aprender linguística, saber a lógica da gramática, sintaxe e semântica é vital. Os humanos usam a gramática sem qualquer conhecimento explícito dela, mas isso não significa que ela não possa ser descrita. Similarmente, ao falar do movimento dos eletrões, certamente não queremos dizer que um eletrão em particular conhece as leis que ele segue, mas nós certamente estaremos a usar linguagem simbólica para descrever o processo, e é até mesmo possível usar a descrição para implementar um modelo e uma simulação que exiba precisamente o mesmo comportamento. Analogamente, mesmo que a consciência humana não *opere* diretamente sobre lógica, isso não significa que nós não sejamos forçados a usar lógica, entre nós, para fornecer uma *descrição* rigorosa desse processo. E, se usarmos programação em lógica para esse fim, tal descrição pode funcionar também como uma especificação executável.

Uma vez obtida uma descrição suficientemente rigorosa do sistema da consciência, estaremos supostamente na posse de todo o *nosso* conhecimento corrente (temporário) desse sistema, reduzido a conexões entre caixas pretas minimais, dentro das quais nós não sabemos ainda como encontrar outros mecanismos essenciais. Presentemente, ninguém conseguiu dividir adequadamente a caixa preta da nossa consciência sobre a consciência no cérebro, mas talvez nós possamos fornecer para ela uma descrição suficientemente rigorosa de modo que possamos modelar um sistema funcional que lhe seja equivalente. Se uma divisão dessa caixa preta cerebral epistémica numa diversidade de outras for conseguida mais tarde, estaremos certos de conseguir, nesta perspetiva, descrever novos modelos computacionais equivalentes ao modelo inerentemente funcional.

8. Funcionalismo

A tese da múltipla realizabilidade diz que um estado mental pode ser "realizado" ou "implementado" por diferentes estados físicos. Seres com diferentes constituições físicas podem, por isso, estar no mesmo estado mental, e podem portanto simbioticamente cooperar ao nível epistémico.

De acordo com o funcionalismo clássico, a múltipla realizabilidade implica que a psicologia é autónoma: por outras palavras, os factos biológicos do cérebro são irrelevantes para ela (Boden 2008). Como afirma um computacionalista: "(a) generalizações interessantes (...) podem com frequência ser feitas sobre eventos cujas descrições físicas nada têm em comum; (b) é frequente que o facto de as descrições físicas dos eventos subsumidos por tais generalizações terem ou não terem alguma coisa em comum seja inteiramente irrelevante para a verdade das generalizações, ou para o seu interesse, ou para o seu grau de confirmação, ou, de facto, para qualquer das suas propriedades epistemológicas importantes" (Fodor 1974). Esta doutrina ainda é usada como um argumento para responder à objeção de computadores de metal e silício serem (fisicamente) muito diferentes de neuroproteínas; bem como uma forma de evitar questões neurocientíficas para as quais, por enquanto, não se pode dar respostas.

Os sistemas de processamento de informação definidos pelos conexionistas são grosso modo inspirados pelo cérebro. No entanto, muitos dos sistemas existentes são, de facto, imensamente diferentes do cérebro. Em geral, as unidades componentes são computacionalmente demasiado simples em comparação com neurónios reais. Mais ainda, a matemática que define as regras de aprendizagem é usualmente imensamente irrealista. O método popular de retro-propagação, por exemplo, baseia-se na capacidade das unidades transmitirem informação em duas direções – o que neurónios reais não podem fazer. A "autonomia" teórica da psicologia/computação permanece, no sentido em que se pode – por vezes se deve – considerar aquilo que os cientistas da computação chamariam de máquina virtual da mente-cérebro, sem se preocupar com os detalhes da sua implementação biológica (Boden 2008).

A ampla posição do "funcionalismo" pode ser articulada em muitas e diferentes variedades. A primeira formulação de uma teoria funcionalista da mente foi proposta por Hilary Putnam (Putnam 1960, 1975). Esta formulação, que agora é apelidada de funcionalismo de estado-da-máquina, ou apenas funcionalismo de máquina, foi inspirada pelas analogias que Putnam e outros notaram entre a mente e as "máquinas" teóricas ou computadores capazes de computar qualquer algoritmo dado, que foram desenvolvidos por Alan Turing – e agora chamados de máquinas de Turing Universais (Wikipedia 2012). A esta luz, as máquinas são modelos físicos de processos abstratos (Minsky 1967).

Num artigo seminal (Turing 1950), A. M. Turing propôs que a pergunta "Podem as máquinas pensar?" fosse substituída pela pergunta "É teoricamente possível que um computador digital de estados finitos, equipado com uma tabela de instruções grande mas finita, ou programa, forneça respostas a questões que levem um interrogador desconhecedor a pensar que é um ser humano?"

Hoje em dia, em deferência para com o seu autor, esta questão é frequentemente expressa como "É teoricamente possível que um computador digital de estados finitos (convenientemente programado) passe o Teste de Turing?". Ao argumentar que esta questão é um substituto legítimo da original (e especulando que a resposta é "sim"), Turing identifica pensamentos com estados de um sistema definidos somente pelos seus papéis em produzir outros estados internos e *outputs* verbais, uma visão que tem muito em comum com teorias funcionalistas contemporâneas. Na verdade, o trabalho de Turing foi invocado explicitamente por muitos teóricos durante os estádios iniciais do funcionalismo do séc. XX, e foi a declarada inspiração para uma classe de teorias, as teorias do "estado da máquina" firmemente associadas com Hilary Putnam (1960), que tiveram um papel importante nos desenvolvimentos iniciais da doutrina (Levin 2010).

O funcionalismo é fundamentalmente uma ampla tese metafísica que se opõe a uma estreitamente ontológica. Isto é, o funcionalismo não está tão preocupado com *o que existe*, quanto com o que é que caracteriza um certo tipo de estado mental, *e.g.* dor, como o tipo de estado que é. Todas as tentativas anteriores de responder ao problema mente/corpo tentaram resolvê-lo respondendo a *ambas* as questões: o dualismo diz que há duas substâncias e que os estados mentais são caracterizados pela sua imaterialidade; o *behaviorismo* afirma que há uma substância e que os estados mentais são disposições comportamentais; o fisicalismo afirma a existência de um tipo de substância e caracteriza os estados mentais como estados físicos (como em "dor = disparos de fibras-C") (Wikipedia 2012).

Neste entendimento, o fisicalismo de tipo pode ser visto como incompatível com o funcionalismo, uma vez que ele afirma que o que caracteriza os estados mentais (*e.g.* dor) é que eles são físicos por natureza, enquanto o funcionalismo diz que o que caracteriza a dor é o seu papel funcional/causal e a sua relação com gritar, "ai", etc. No entanto, qualquer tipo mais fraco de fisicalismo, que faça a afirmação ontológica simples que tudo o que existe é feito de matéria inorgânica, é perfeitamente compatível com o funcionalismo. Mais ainda, muitos funcionalistas que são fisicalistas exigem que as propriedades sobre as quais se quantifica em definições funcionais sejam propriedades físicas. Assim, eles *são* fisicalistas, ainda que a tese geral do funcionalismo ela própria não os comprometa a ser (*ibidem*).

Em 1984, Putnam elaborou um argumento engenhoso como suporte principal ao seu assalto ao computacionalismo, e introduziu a *Interpretação Relacional* (e outras noções relacionadas). O seu argumento emprega não só raciocínio demonstrativo, ao qual os teoremas de Gödel se aplicam, mas também emprega raciocínio não-demonstrativo. Putnam encontrou uma forma de aplicar ao seu argumento os teoremas de Gödel e, tal como é apresentado em (Putnam 1988), assim pretende mostrar que todos os métodos epistémicos usados no inquérito humano são, se formalizados, suscetíveis aos teoremas de Gödel. Na sua interpretação, podem ser empregues (com menos do que certeza matemática) métodos fracamente formalizáveis para demonstrar que um programa de computador para a mente humana é consistente, por seres humanos que não são suscetíveis a Gödel, mas não podem ser empregues por agentes que o são (Buechner 2007).

O argumento engenhoso de Putnam falha ainda assim. De facto, o que os matemáticos e filósofos avaliaram mal é que os teoremas de Gödel mostram que ninguém, suscetível a Gödel ou não, pode demonstrar a consistência da aritmética de Peano com certeza matemática sem construir uma árvore de prova infinita, o que põe limitações aos finitários humanos. Esta é a verdadeira razão pela qual os anti-funcionalistas que desejam empregar os teoremas de Gödel estão condenados a falhar. A menos que os seres humanos possam construir árvores de prova infinitas, estamos limitados pelos teoremas de Gödel, mesmo que não sejamos máquinas computacionais às quais eles diretamente se aplicam. Este simples ponto tem resistido à apreciação pelos anti-funcionalistas.

O argumento de Putnam afirma que todos os métodos de inquérito do mundo empregues por uma máquina de computar são suscetíveis a Gödel. Disto segue facilmente que a máquina não pode saber a verdade de nenhuma das suas frases de Gödel em nenhuma das modalidades epistémicas gerais definidas por Putnam, e sob as quais ela conduz o inquérito do mundo.

Mas este resultado limitativo também se aplica aos seres humanos, mesmo que não sejamos suscetíveis a Gödel. Se uma máquina de computar executando um programa é incapaz de determinar a sua correção em alguma modalidade epistémica porque é suscetível a Gödel, então nenhum ser humano finitário pode tão pouco determinar a sua correção naquela modalidade epistémica, mesmo que os humanos não sejam suscetíveis a Gödel.

A menos que consigamos construir árvores de prova infinitas, nenhum dos nossos métodos finitários de inquérito do mundo podem mostrar, na sua característica modalidade epistémica, a verdade da frase de Gödel que surge na formalização desses métodos de inquérito. E se não usarmos um método de inquérito formalizável, nenhum método de inquérito não-formalizável que usemos pode demonstrar a verdade da frase de Gödel de um método de inquérito formalizável na modalidade epistémica dessa formalização (*ibidem*).

Apesar da rejeição do funcionalismo de Putnam, ele tem continuado a florescer e sido desenvolvido em numerosas versões por pensadores tão diversos como David Marr, Daniel Dennett, Jerry Fodor e David Lewis (Fodor 1974, Dennett 2005). O funcionalismo ajudou a lançar as fundações da moderna ciência cognitiva, e é a teoria da mente dominante em filosofia hoje.

Na parte final do séc. XX, o funcionalismo ergueu-se como a teoria dominante dos estados mentais. Tal como o behaviorismo, o funcionalismo retira os estados mentais do domínio do "privado" ou subjetivo, e dá-lhes um estatuto aberto a investigação científica.

Mas, ao contrário do behaviorismo, a caracterização dos estados mentais do funcionalismo em termos dos seus papéis na produção de comportamento concede-lhes a eficácia causal que o senso comum lhes atribui. E, ao permitir que os estados mentais sejam multiplamente realizados, o funcionalismo oferece uma explicação dos estados mentais que é compatível com o materialismo, sem limitar a classe das mentes a criaturas com cérebros como os nossos (Levin 2010).

9. Emergência e Funcionalismo

A evolução biológica é caracterizada por um conjunto de processos altamente entrelaçados, que produzem uma espécie extraordinária de inovação combinatória complexa. Um termo genérico, frequentemente usado para descrever esta vasta categoria de processos geradores de ordem, fracamente previsíveis e espontâneos, é "emergência" (ou "imanência").

Este termo tornou-se uma espécie de sinal para referir os paradigmas de investigação sensíveis aos fatores sistémicos. Os sistemas dinâmicos complexos podem assumir espontaneamente padrões de comportamento ordenados que não são previamente imagináveis a partir das propriedades dos seus elementos componentes, nem a partir dos seus padrões de interação. Existe imprevisibilidade nos fenómenos auto-organizadores – preferencialmente chamados de "evolucionários", como Turing fez (Turing 1950) – com consideravelmente diversos e variáveis níveis de complexidade.

"Complexidade" refere-se ao estudo da emergência de propriedades coletivas em sistemas com muitos componentes interdependentes. Estes componentes podem ser ou macromoléculas num contexto físico ou biológico, e pessoas, máquinas ou organizações num contexto socioeconómico.

O que emerge? A resposta não é algo definido fisicamente mas antes algo como uma forma, padrão ou função – tal como no funcionalismo. O conceito de emergência é aplicável a fenómenos nos quais as propriedades relacionais predominam sobre as propriedades dos elementos componentes na determinação das características do conjunto. Os processos emergentes surgem devido a configurações e topologias, não devido a propriedades dos elementos (Deacon 2003). Este funcionalismo é, quase por definição, anti essência substancial, anti princípio vital, anti *qualia* monopolizadores.

10. Psicologia Evolutiva e Lógica

A lógica, sustentamos nós, fornece a cúpula conceptual global que, como um módulo genérico, articula conjuntamente de forma fluida os módulos emergidos específicos identificados pela psicologia evolucionária (Pereira 2012). A esse respeito, ela é refletida pela computabilidade geral da máquina Universal de Turing, a qual pode executar qualquer programa, computar qualquer função computável.

A relação deste argumento com a lógica é assegurada pela perspetiva filosófica do funcionalismo: a própria lógica pode ser implementada sobre um sistema de processamento de símbolos, independentemente do substrato físico que o suporta. Logo que um processo é descrito em lógica, podemos usar a descrição para sintetizar um artefacto com aquelas mesmas propriedades.

No entanto, a definição canónica de conhecimento científico procurada avidamente pelos positivistas lógicos não é um problema filosófico, nem pode ser atingida, como eles esperavam, simplesmente por análise lógica e semântica. É também uma questão empírica, que apenas pode ser respondida pelo exame contínuo da funcionalidade possível do próprio processo de pensamento e da sua base física.

Em alguns casos, as ferramentas cognitivas e os instrumentos de racionalidade descobrir-se-ão independentes do *hardware*. Mesmo então, a adequação do seu uso em circunstâncias e objetivos reais específicos necessitará de ser determinada empiricamente. Não existe uma receita epistemológica universal, mas pode obter-se acordo sobre o sucesso relativo de um dado conjunto de ferramentas.

Em todo o caso, pode procurar-se uma compreensão parcial através da construção de máquinas inteligentes, salvaguardados pelo funcionalismo quando postulando que o substrato material não é frequentemente a essência, que é suficiente realizar funcionalidade equivalente ainda que sobre diferente *hardware*. Mais ainda, diferentes caminhos funcionantes para o mesmo comportamento podem ser seguidos, assim acumulando conhecimento sobre o que a inteligência em geral significa, em direção ao entrelaçamento simbiótico desses caminhos, o mais recente passo em epistemologia evolucionária. O funcionalismo só pode tornar isso mais ágil.

Os procedimentos mais frutuosos certamente incluirão o uso da Inteligência Artificial, teoria e técnicas, ajudados oportunamente pelo ainda embrionário campo da emoção artificial, para simular operações mentais complexas, já antevistas em (Turing 1950). Este sistema de modelagem vai ser articulado com uma neurobiologia do cérebro em rápida maturação, incluindo o exame de alta resolução de redes computacionais ativas em várias formas de pensamento.

Como é que a seleção natural antecipa as nossas necessidades futuras? Criando uma máquina cognitiva, chamada cérebro, que pode criar modelos do mundo, e mesmo dele próprio, e processos hipotéticos, muito como a máquina de Turing Universal pode imitar qualquer outra máquina de Turing, e tal como qualquer computador pode em princípio executar qualquer programa. Esta plasticidade fornece a sua versatilidade universal (Davis 2000).

É útil considerar uma dualidade que designo por “Turing versus Eva”. O matemático Alan Turing representa o computador na essência da sua completa flexibilidade. A máquina de Turing Universal é aquela que pode imitar qualquer computador, qualquer programa: é mimetismo *par excellence*. Tal mimetismo faz-nos pensar sobre o meme e a nossa própria flexibilidade mental, tão vital para complementar a reprodução genética, devido às diferentes calendarizações da reprodução. Na reprodução genética, a diferença espalha-se por gerações, e isso não é suficiente quando a adaptação tem de ser ágil. É a partir dessa necessidade que provém o mecanismo cerebral de reprodução – aqueles memes que saltam de cérebro em cérebro. Na reprodução genética, as mitocôndrias são as estruturas genéticas fixas por excelência, surgindo do lado feminino, que são replicadas sem união de genes. Elas representam metaforicamente a multitude de módulos específicos que nós herdamos em virtude do passado da nossa espécie.

A abordagem de Turing à inteligência mecânica, no seu relatório NPL de 1948 "Intelligent Machinery" (Turing 1948), era tão desembaraçada quanto a sua abordagem aos números computáveis dez anos antes. Ele continuava, uma vez mais, a partir do ponto onde Gödel havia terminado. A incompletude dos sistemas formais limita a capacidade dos computadores em duplicar a inteligência e criatividade da mente humana? Turing sumariza a essência deste argumento retorcido em 1947, dizendo que "por outras palavras, se uma máquina deve ser infalível, ela não pode ser inteligente". Em vez de tentarmos construir máquinas infalíveis, nós devíamos estar a desenvolver máquinas falíveis capazes de aprender com os próprios erros (Dyson 2012).

Ele sugeriu incorporar um gerador de números aleatórios para criar uma "máquina de aprender", concedendo ao computador a capacidade de fazer uma suposição e, ou reforçar, ou descartar os consequentes resultados. Se as suposições fossem aplicadas em modificações das instruções do próprio computador, a máquina poderia aprender a ensinar-se a si própria. "O que queremos é uma máquina que possa aprender com a experiência", escreveu ele. "A possibilidade de a máquina alterar as suas próprias instruções fornece o mecanismo para isso".

Turing traçou um paralelo entre inteligência e "a busca genética ou evolucionária através da qual uma combinação de genes é procurada, o critério sendo o valor de sobrevivência. O sucesso notável desta busca confirma em certa medida a ideia de que a atividade intelectual consiste sobretudo em diferentes formas de busca". Precursor da Biologia Matemática, ele procurou com sucesso reduções matemáticas de fenómenos emergentes particulares a soluções de equações diferenciais, no seu artigo sobre morfogénese (Turing 1952). Mais geralmente, a computação evolucionária levaria a máquinas verdadeiramente inteligentes. "Em vez de se tentar produzir um programa para simular a mente adulta, por que não ao invés tentar produzir um que simule a da criança?" perguntou ele. O caminho para a inteligência artificial, sugeriu Turing, é construir uma máquina com a curiosidade de uma criança, e deixar a inteligência evoluir, com a ajuda de oráculos externos.

A máquina de busca da Internet é um autómato determinista de estados finitos, exceto naquelas junções onde as pessoas, individual ou colectivamente, fazem uma escolha não-determinista sobre que resultados são selecionados como significativos e recebem um *click*. Estes *clicks* são imediatamente incorporados no estado da máquina determinista, a qual cresce assim incrementalmente mais inteligente com cada *click*. Isto é o que Turing definiu como uma máquina com oráculo (Dyson 2012).

11. Lógica Computacional e IA

O trabalho de Turing sobre a computabilidade leva a uma questão profunda: "A computação com símbolos discretos dá uma explicação completa da nossa conceção do mundo físico? Por outras palavras, é o mundo, como o vemos, computável?".

Em IA, um agente é qualquer entidade, embutida num mundo real ou artificial, que pode observar o mundo em mudança e realizar ações no mundo para se manter numa relação harmoniosa com o mundo. A Lógica Computacional (LC), tal como usada em IA, é a linguagem de pensamento do agente. As frases expressas nesta linguagem representam as crenças do agente sobre o mundo tal como ele é, e sobre os seus objectivos acerca da forma como ele gostaria que o mundo fosse. O agente usa os seus objectivos para controlar o seu comportamento. Ele pode usar também LC para guiar as suas comunicações públicas com outros agentes.

O mais influente e largamente citado argumento contra a lógica vem de experiências psicológicas sobre raciocínio com frases em linguagem natural em forma condicional. A interpretação mais popular destas experiências é que as pessoas não têm uma capacidade de uso geral natural para raciocinar logicamente – como uma máquina de Turing Universal – mas apenas desenvolveram ao invés, através dos mecanismos da evolução darwiniana, algoritmos especializados para resolver problemas típicos que surgem no seu ambiente. Um dos problemas destas experiências é que elas falham em reconhecer que a forma em linguagem natural de uma frase condicional é apenas uma aproximação à forma lógica do seu significado pretendido. Outro problema é que a interpretação destas experiências é baseada sobre um entendimento desadequado da relação entre conhecimento e raciocínio. Em contraste, o entendimento da LC sobre o pensamento humano pode ser expresso livremente como “pensamento = conhecimento especializado + raciocínio de uso geral” (Kowalski 2011), muito em linha com o ponto de vista de Turing sobre a IA.

Consequentemente, no seu esforço para obter essa descrição rigorosa, o campo da Inteligência Artificial tornou viável o propósito de tornar a lógica uma linguagem de programação (Pereira 2002). A lógica pode presentemente ser usada como uma linguagem de especificação que, não só é executável, mas também em cima da qual podemos demonstrar propriedades e fazer demonstrações de correção que validam as descrições que produzimos com ela. Ao enfrentar o desafio, a IA desenvolveu a lógica para lá dos confins da cumulatividade monótona, muito distante dos paraísos artificiais do bem-definido, e decididamente para o alcance da incompletude do mundo real, contraditório, discutível, revisável, aprendível, distribuído, atualizável, assim exigindo consequentemente, entre outros, o estudo e desenvolvimento de lógicas não-monótonas, e da sua implementação em computador.

Ao longo dos anos, uma enorme quantidade de trabalho foi desenvolvida sobre estes tópicos individuais, tais como semântica de linguagens de programação em lógica, revisão de crenças, preferências, programas evolutivos com atualizações, aprendizagem indutiva, e muitos outras questões que são cruciais para uma arquitetura computacional da mente. Estamos hoje na presença de um estado-da-arte de onde podemos começar a visar as questões mais gerais com as ferramentas ao dispor, unificando esses esforços para obter implementações poderosas, exibindo novas e promissoras propriedades computacionais. A LC mostrou-se capaz de evoluir para corresponder às exigências das difíceis descrições que está a tentar visar.

O uso do paradigma da LC também permite-nos ver o sistema de cada um a um nível de abstração e generalidade suficientemente elevado para possibilitar discussões interdisciplinares produtivas quer sobre a sua especificação, quer sobre as suas propriedades derivadas.

Tal como mencionado previamente, a linguagem da lógica é usada quer pelas ciências naturais, quer pelas humanidades, e, mais geralmente, está no cerne de qualquer fonte de conhecimento comum derivado humanamente, de modo que ela fornece-nos um terreno conjunto para raciocinar acerca das nossas teorias. Uma vez que o campo da ciências cognitivas é essencialmente um esforço concertado da parte de vários campos distintos do conhecimento, acreditamos que tais esforços de unificação de linguagem e vocabulário são, não apenas úteis, mas também obrigatórios.

A filosofia de Alan Turing pode parecer um reducionismo último, na sua atomização do processo mental, o seu desprezo pelo suporte não-material. Ainda assim, vista à luz acima, ela depende de uma síntese de visão que se opõe ao costumário mundo intelectual dividido entre muitos especialismos verbais, ou matemáticos, ou técnicos. Os seus muitos interesses: do computável à morfogénese, passando pela mecânica quântica e a mente cognitiva, mostram uma ambição profunda por uma abrangente síntese de filosofia natural.

12. Epistemologia Simbiótica

Quando se considera o conhecimento científico, se o processamento computacional do genoma humano levou-nos à Bio-informática, então, por analogia, podemos afirmar que o "cognoma" vai ser a base da futura "Cogno-tecnologia", aplicável em qualquer ciência. Deste modo, o futuro da IA está ligado à característica dela ser um instrumento epistemológico, não apenas para um agente autónomo, mas para um simbiótico, que ajudará os humanos a levar a cabo a própria ciência. E não estou a falar apenas de mineração de dados, reconhecimento de padrões, construção de ontologias, embora nestes campos possamos abordar aspetos mais estruturados da epistemologia. Estou a pensar naquilo que todo o cientista faz, que é abduzir, inventar e profetizar teorias, testá-las, criar experiências, retirar conclusões para sustentar observações adicionais, discutir essas observações e as suas conjeturas com outros cientistas.

Existe uma meta-argumentação em progresso sobre o que é bom raciocínio, quais são as conclusões que podemos retirar de uma discussão (i.e. uma semântica), que é inerente a toda a atividade científica. O computador vai ser usado cada vez mais como um ajudante de investigação, não apenas para automatizar, mas também para propor experiências e hipóteses; e, ao fim de contas, ao tornar as nossas próprias conceções sobre a aplicação da epistemologia repetíveis e externalizadas, o computador torna tais conceções também mais objetivas (Pereira 2012).

Para compreender – e daí para aperfeiçoar – a inteligência humana, tem que se exprimir como é que ela funciona, e o computador é o dispositivo experimental para testar o nosso próprio entendimento, modelando-o em detalhe na extensão que pudermos. A lógica, no sentido lato de lógico, é o veículo natural e partilhado para o fazer de um modo científico preciso. E o computador, a nossa máquina computacional privilegiada por excelência, é sem dúvida o nosso veículo partilhado artificial para objetivamente demonstrar a valia desse entendimento. A Lógica Computacional abrange ambos, e beneficia simbioticamente de ambos.

Verdadeiramente, a capacidade de cognição é o que nos permite antecipar o futuro, pré-adaptar e imaginar cenários de evoluções possíveis – do mundo e de nós próprios enquanto agentes cognitivos –, fazer escolhas, usar preferências sobre alguns mundos hipotéticos e os seus futuros, e meta-preferências, i.e. preferências sobre que preferências empregar e como fazê-las evoluir. A atividade de prospetar o futuro é vital e característica da nossa espécie e sua capacidade de entender o mundo real e nós próprios, vivendo em sociedade, onde a cognição distribuída é a forma normal e regular de fazer ciência.

A consciência prospetiva permite-nos pré-adaptar-nos ao que vai acontecer. Para isso, a capacidade de simular, de imaginar "o que aconteceria se", i.e. pensamento hipotético, torna-se necessária. Tal pensamento é indispensável em ciência; pois ele dá-nos regras para prever e explicar o que vai ou pode acontecer, sem o qual a tecnologia não seria possível.

Ultimamente, temos trabalhado no sentido de automatizar esta capacidade, implementando programas que conseguem imaginar os seus futuros, fazendo escolhas informadas acerca deles, e depois mesmo modificando-se eles próprios – tal como Turing permitia e predizia que haveria de acontecer – para promulgar essas escolhas, os pressentimentos da vontade livre. Chamamos-lhe computação prospetiva (Alferes *et al* 2002, Pereira & Lopes 2009, Pereira & Han 2009).

A epistemologia terá finalmente a capacidade de ser partilhada, seja com robôs, alienígenas ou outra qualquer entidade que precise de facto de executar cognição para continuar a existir e programar o seu futuro. Criar computadores e robôs situados significa executar a nossa própria evolução cognitiva por outros meios. Com a virtude de engendrar ciclos auto-aceleradores, simbióticos, em co-evolução.

Os robôs computadorizados retificam as nossas teorias científicas, tornando-as objetivas, repetíveis, e parte de uma realidade externa construída em comum, edificada sobre ciência unificada multidisciplinar.

A Inteligência Artificial e as Ciências Cognitivas, ao construir tais entidades, fornecem um passo enorme e estimulante na direção da promoção da unidade da ciência, através precisamente do esforço dessa construção.

Nestes dias de quantização do tempo discreto, processos biológicos computacionais, e evidência do universo em permanente expansão – os autómatos e a fita – a Máquina de Turing reina suprema. O funcionalismo Universal – a essência de Turing – é o que possibilita a inevitável reunião dos fantasmas adentro das diversas máquinas corpóreas (baseadas em silício, biológicas, extraterrestres ou outras) para promover a sua coevolução epistémica simbiótica, uma vez que esses fantasmas compartilham o mesmo funcionalismo.

Assim, a posição funcionalista computacional geral, estimulada primeiramente por Alan Turing, é extremamente útil. Turing está verdadeiramente e para sempre entre nós.

Obras Consultadas e Referências

[Os meus trabalhos estão disponíveis em http://centria.di.fct.unl.pt/~lmp/publications/Biblio.html]

J. J. Alferes, A. Brogi, J. A. Leite, L. M. Pereira (2002). Evolving Logic Programs. *Procs. Of the 8th European Conf. On Logics in Artificial Intelligence (JELIA'02)*, S. Flesca *et al.* (eds.), pp. 50-61, LNAI 2424. Berlin: Springer.

M. A. Boden (2008). Information and Cognitive Science. *Philosophy of Information*. P. Adriaans & J. van Bentham (eds.), pp. 741-761. Amsterdam: North-Holland, Elsevier.

P. D. Bruza, D. Widdows, J. Woods (2009). A Quantum Logic of Down Below. *Handbook of Quantum Logic and Quantum Structures: Quantum Logic*. K. Engesser, D. M. Gabbay, D. Lehmann (eds.), pp. 625-660. Amsterdam: North-Holland, Elsevier.

J. Buechner (2007). *Gödel, Putnam, and Functionalism: a New Reading of Representation and Reality*. Cambridge: The MIT Press.

J. L. Casti, W. DePauli (2000). *Gödel – A life of Logic*. Cambridge: Perseus.

P. S. Churchland (2002). *Brain-Wise: Studies in Neurophilosophy*. Cambridge: The MIT Press.

A. S. d'Avila Garcez, L. C. Lamb, D. M. Gabbay (2006). Connectionist computations of intuitionistic reasoning. *Theor. Comput. Sci.,* 358(1): 34-55.

A. S. d'Avila Garcez, L. C. Lamb, D. M. Gabbay (2007). Connectionist modal logic: Representing modalities in neural networks. *Theor. Comput. Sci*., 371(1-2): 34-53.

M. Davis (1958). *Computability and Unsolvability*. New York: McGraw-Hill.

M. Davis (1990). Is mathematical insight algorithmic? *Behavioral and Brain Sciences*, 13 (4), 659-60.

M. Davis (1993). How subtle is Gödel's theorem. More on Roger Penrose. *Behavioral and Brain Sciences*, 16, 611-61.

M. Davis (2000). *The Universal Computer: The Road from Leibniz to Turing*. New York: W.W. Norton & Co.

T. W. Deacon (2003). The Hierarchic Logic of Emergence: Untangling the Interdependence of Evolution and Self-Organization. *Evolution and Learning: The Baldwin Effect Reconsidered*. H. W. Weber, D. J. Depew (eds.), pp. 273-308. Cambridge: The MIT Press.

D. C. Dennett (2005). *Sweet Dreams: Philosophical Obstacles to a Science of Consciousness*. Cambridge: The MIT Press.

G. Dyson (2012). *Turing's Cathedral: The Origins of the Digital Universe*. London: Allen Lane.

J. A. Fodor (1974). Special Sciences, or the Disunity of Science as a Working Hypothesis. *Synthèse,* 28: 77-115.

A. Hodges (1983). *Alan Turing – the Enigma*. New York: Simon and Schuster.

A. Hodges (1997). *Alan Turing: one of The Great Philosophers*. London: Phoenix.

R. Kowalski (2011). *Computational Logic and Human Thinking: How to be Artificially Intelligent*. Cambridge: Cambridge University Press.

D. Leavitt (2006). *The Man Who Knew Too Much: Alan Turing and the Invention of the Computer*. London: Phoenix.

J. Levin (2010). Functionalism. *The Stanford Encyclopaedia of Philosophy*, E.N. Zalta (ed.), http://plato.stanford.edu/archives/sum2010/entries/functionalism/.

J. R. Lucas (1996). Minds, Machines, and Gödel: A Retrospect. *Machines and Thought (vol. 1),* P. Millican, A. Clark (eds.), pp. 103-124. Oxford: Oxford University Press.

D. McDermott (2001). *Mind and Mechanism*. Cambridge: The MIT Press.

M. Minsky (1967). *Computation: Finite and Infinite Machines*. Englewood Cliffs: Prentice-Hall.

R. Penrose (1994). *Shadows of the Mind: a search for the missing science of consciousness*. Oxford: Oxford University Press.

L. M. Pereira (1970). Introdução aos Autómatos Infinitos e Teoria da Computabilidade. *Técnica*, vol. 395, pp. 265-273 & vol. 396, pp. 321-328. Lisboa: Instituto Superior Técnico.

L. M. Pereira (1979). Prolegómeno a uma Neurologia Artificial. *Análise Psicológica*, vol. 11(4), pp. 519-522.

L. M. Pereira (1988). Inteligência Artificial: Mito e Ciência. *Revista Colóquio/Ciências*, vol. 3, pp. 1-13, Lisboa: Fundação Calouste Gulbenkian.

L. M. Pereira (2002). Philosophical Incidence of Logical Programming. *Handbook of the Logic of Argument and Inference,* D. Gabbay *et al.* (eds.), pp. 425-448, Studies in Logic and Practical Reasoning series, vol. 1. Amsterdam: Elsevier Science.

L. M. Pereira (2007). Gödel and Computability. *Progress in Artificial Intelligence*, J. M. Neves *et al.* (eds.), pp. 63-72, LNAI 4874. Berlin: Springer-Verlag.

L. M. Pereira, G. Lopes (2009). Prospective Logic Agents. *International Journal of Reasoning-based Intelligent Systems (IJRIS)*, 1(3/4):200-208.

L. M. Pereira, T. A. Han (2009). Evolution Prospection in Decision Making. *Intelligent Decision Technologies (IDT)*, 3(3):157-171.

L. M. Pereira (2012). Evolutionary Psychology and the Unity of Sciences-Towards na Evolutionary Epistemology. *Special Sciences and the Unity of Science*, O. Pombo, J. M. Torres,

J. Symons, S. Rahman (eds.), Series on Logic, Epistemology, and the Unity of Science, Vol. 24, pp. 163-175. Dordrecht: Springer.

H. Putnam (1960). Minds and Machines. Reprinted in Putnam (1975).

H. Putnam (1975). *Mind, Language, and Reality*. Cambridge: Cambridge University Press.

H. Putnam (1988). *Reality and Representation*. Cambridge: The MIT Press.

A. M. Turing. On computable numbers, with na application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* ser. 2, 42 (1936) pp. 230-65, correction ibid. 43 (1937), pp. 544-6.

A. M. Turing (1948). Intelligent Machinery. Report to the National Physics Laboratory. Alan Turing papers, King's College Archives, Cambridge, UK.

A. M. Turing (1950). Computing Machinery and Intelligence. *Mind* 59:433-460.

A. M. Turing (1952). The chemical basis of morphogenesis. *Phil. Trans. Of the Royal Society of London. Series B, Biological Sciences 237*, 641, 37-72.

A. M. Turing, J.-Y. Girard (1995). *La Machine de Turing*. Paris: Éditions du Seuil.

L. G. Valiant (2000). A neuroidal architecture for cognitive computation. *J. ACM* 47(5): 854--882.

L. G. Valiant (2003). Three problems in computer science. *J. ACM* 50(1): 96-99.

H. Wang (1973). *From Mathematics to Philosophy*. New York: Routledge.

H. Wang (1987). *Reflections on Kurt Gödel*. Cambridge: The MIT Press.

J. C. Webb (1980). *Mechanism, Mentalism and Meta-mathematics*. Dordrecht: Reidel.

Wikipedia(2012). http://en.wikipedia.org/wiki/Functionalism_(philosophy_of_mind)

Parte II –

Incomputabilidades

2. O problema da decisão e a máquina universal de Turing[1]

Fernando Ferreira

Em Setembro de 1928, realizou-se em Bolonha o VI Congresso Internacional de Matemáticos. Foi o primeiro congresso em que participou

1 O editor deste volume pediu-me para escrever um texto para um público não especialista. Os meus instintos de profissional impeliram-me a querer dar referências, a elaborar, a abordar pontos subtis, a não condescender com formulações mais apelativas mas menos corretas. Mas, o que é facto, é que aceitei o pedido. Acabei por escrever um texto escorrido, quase sem elaborações e sem notas de pé de página. Juntei dois apêndices com informação adicional que não quis que figurasse no texto principal. Observações e referências (que tentei que fossem em português) foram colocados num anexo denso intitulado "Apontamentos complementares".

a Alemanha depois da primeira guerra mundial. À frente da delegação alemã (a segunda maior, depois da italiana) figurava David Hilbert, talvez o maior matemático ao tempo, professor na Universidade de Göttingen e já perto da reforma. Trabalhou e obteve resultados fundamentais e de grande influência em álgebra, teoria dos números, geometria, análise e física teórica, mas também se dedicou à lógica matemática e aos fundamentos de matemática. Na sua comunicação ao congresso, fez um apelo:

> " É claro que, para o completo desenvolvimento desta tarefa, é necessária a fervorosa colaboração da geração mais jovem de matemáticos."

De que tarefa falava Hilbert? Tratava-se, sem mais nem menos, de pôr a matemática em solo absolutamente seguro. Sem dúvida que o leitor estar-se-á a questionar: – Mas não é a matemática uma ciência completamente rigorosa? Não é a matemática absolutamente segura?

É pouco conhecido entre o público em geral que a matemática sofreu uma grande transformação na viragem do século XIX para o século XX, apenas comparável ao aparecimento da axiomática na Grécia antiga ou à descoberta do cálculo infinitesimal por Leibniz e Newton. A transformação consistiu essencialmente na aceitação sem pejo do infinito atual (o infinito tomado como acabado e não em potência, como quando se toma a sucessão de todos os números naturais 1, 2, 3, … como algo fechado e completo e não como um processo sem fim) e no abandono, por completo, da intuição como método de *justificação* matemática. Foi, também, acompanhada pela aceitação de abstrações cada vez maiores e cada vez mais afastadas das fontes da física e das aplicações. Podemos dar como exemplo os trabalhos dos matemáticos Richard Dedekind e Karl Weierstrass na fundamentação rigorosa do cálculo infinitesimal e integral. Estes trabalhos puseram em evidência a necessidade do infinito atual no tratamento rigoroso dos números reais. Um número real (um ponto do contínuo da reta) é agora visto como um ente de natureza infinitária (um número real é dado por uma dízima *infinita*). Um ponto é um ente infinitário: note o leitor o quão longe estamos da conceção euclidiana, onde um ponto é aquilo que não tem partes! Por sua vez, a criação por Georg Cantor de uma teoria extremamente bela e fecunda do infinito atual, em que se distinguem vários tipos de infinitos – a teoria dos conjuntos – leva o tratamento do infinito atual até às suas últimas consequências.

O infinito atual tem sido objeto de desconfiança por parte de filósofos e matemáticos. Em 1902 cai sobre a comunidade matemática uma espécie de bomba: o paradoxo de Russell. Dada uma qualquer condição (p. ex., a condição de ser homem) podemos considerar a sua extensão (o conjunto de todos os homens). Normalmente, um conjunto não é membro de si próprio. O conjunto de todos os homens não é um homem. Mas, por vezes, é-o: o conjunto de todas as abstrações é, ele próprio, uma abstração. Bertrand Russell considerou o conjunto de todos os conjuntos que não são membros de si próprios. Põe-se a questão: é o conjunto de Russell membro de si próprio? Se é, então dada a sua definição (trata-se do conjunto formado pelos conjuntos que *não são* membros de si próprios), não é. Mas se não é, é. Chegamos a uma situação paradoxal: o conjunto de Russell nem pode ser membro de si próprio, nem pode deixar de o ser!

A descoberta do paradoxo de Russell provocou reações díspares e, por vezes, emotivas, entre os matemáticos. Uns, os tradicionalistas, que se opunham à emergente teoria dos conjuntos, viram na antinomia de Russell a prova de que o novo caminho era um erro. Outros, os modernos, atarefaram-se a tentar compreender as razões para o aparecimento do paradoxo e propuseram programas de fundamentação da matemática. Hilbert encontrava-se entre estes últimos e, nos anos vinte do século passado, propôs o seu próprio programa de fundamentação da matemática: o formalismo. Este programa competia com o programa logicista do próprio Russell e com o denominado programa intuicionista do matemático holandês L. E. J. Brouwer.

O programa de Hilbert concebia a matemática como uma espécie de jogo. No jogo de xadrez há a partida propriamente dita, jogada de acordo com regras exatas. Há, também, a teoria *sobre* o jogo de xadrez. Faz parte desta teoria a afirmação de que não é possível forçar xeque-mate com apenas dois cavalos a um rei isolado. Já não se trata agora de jogar uma partida de xadrez mas sim de investigar o jogo em si. É *metaxadrez*. A matemática, quando vista formalmente, também tem regras bem definidas. Elas são dadas pelos axiomas e pelas regras de inferência. Os axiomas correspondem ao tabuleiro inicial duma partida de xadrez – as peças na sua posição inicial – e as regras de inferência correspondem às jogadas permitidas. A ideia fundamental do formalismo é ver a matemática como um jogo (dedutivo) simbólico. As peças deste jogo simbólico são as fórmulas da linguagem da matemática. Para que este jogo tenha regras exatas, como acontece no xadrez, é necessário simbolizar (formalizar) completamente a linguagem da matemática. Não basta enunciar o primeiro axioma da geometria euclidiana dizendo que por dois pontos passa uma reta. É necessário formulá-lo numa linguagem exata. Na notação da lógica simbólica, o axioma de Euclides escreve-se assim:

$$\forall x \forall y (Px \land Py \rightarrow \exists z (Rz \land Ixz \land Iyz))$$

Se o leitor não tem treino lógico e não percebe a fórmula acima tem, do ponto de vista estritamente formalista, uma vantagem sobre mim. Não consegue interpretar a fórmula acima. Mas esse é precisamente o ponto do formalismo: a fórmula deve ser vista como não querendo dizer nada, como sendo uma mera sequência de símbolos. Se o leitor tiver treino lógico adivinha que se pode interpretar os objetos de tipo P como pontos, os objetos de tipo R como retas e a relação I como a relação de incidência entre um ponto e uma reta (diz-se que um ponto *incide* numa reta se o ponto está na reta). Mas, para efeitos do programa formalista da fundamentação da matemática, não o devemos. Os axiomas de Euclides são vistos apenas como fórmulas de uma dada linguagem formal. Estas fórmulas são construídas de forma precisa a partir de símbolos primitivos (P, R e I) e de operadores lógicos como *e*, *ou*, *não*, *se … então …*, *para todo* e *existe* (na simbologia da lógica, $\land$, $\lor$, $\neg$, $\rightarrow$, $\forall$ e $\exists$, respetivamente).

Para além dos axiomas próprios da teoria em questão (no nosso exemplo, os axiomas de geometria euclidiana), o sistema formal inclui também axiomas comuns a todos os sistemas: os axiomas da lógica (mais precisamente, os axiomas do denominado cálculo de predicados). Por exemplo, as fórmulas da forma "se A, então A" (na simbologia da lógica, "$A \rightarrow A$") podem fazer parte duma axiomática da lógica. As regras de inferência permitem gerar (deduzir) novas fórmulas a partir de outras fórmulas. Uma regra de inferência básica é a regra do *modus ponens*. Por exemplo:

Se x incide sobre z, então x é um ponto

x incide sobre z

Logo, x é um ponto

Na notação da lógica, podemos inferir a configuração simbólica "Px" a partir das configurações "$Ixz \rightarrow Px$" e "Ixz". O "jogo de fórmulas" consiste em gerar certas sequências de símbolos (os teoremas) a partir de determinadas sequências dadas inicialmente (os axiomas) por intermédio das regras de inferência. Hilbert escreve:

> (...) não devemos todavia perder de vista o pré-requisito essencial do nosso procedimento. Pois existe uma condição, uma só mas absolutamente necessária, a que está submetida a utilização do [nosso] método (...), e essa condição é a *demonstração de consistência*.

Hilbert exige que, no cálculo axiomático formal, não se chegue a contradições, i.e., não se chegue a fórmulas da forma "*A* e *não-A*" (simbolicamente "$A \wedge \neg A$"). O "jogo", neste caso, deixaria de ter interesse até porque, duma contradição, é possível inferir qualquer fórmula. Por isso o paradoxo de Russell é tão devastador: toda a fórmula seria um teorema.

Na sua palestra em Bolonha, Hilbert menciona vários problemas relacionados com o seu programa de fundamentação da matemática. Um é o *problema de consistência* que acabámos de discutir. Hilbert pretendia mostrar que as regras do jogo formal das axiomáticas importantes para a matemática – um jogo cujas regras são completamente exatas, como é o xadrez – nunca levariam à dedução de contradições. Deduzir teoremas a partir duma axiomática é fazer matemática mas mostrar que a axiomática da aritmética não leva a contradições é fazer *metamatemática*. A metamatemática ou teoria da demonstração foi a disciplina inventada por Hilbert para levar a cabo o seu programa. Outro dos problemas mencionados na palestra foi o *problema da completude*. No caso da axiomática da aritmética, este problema consiste em mostrar que, para qualquer asserção simbólica da linguagem formal da aritmética, ou ela ou a sua negação é um teorema. Um terceiro problema foi também mencionado em 1928, mas num livro de Hilbert com o seu aluno Wilhelm Ackermann. É o *problema da decisão*, também conhecido pelo seu nome alemão: *Entscheidungsproblem*. Hilbert e Ackermann escreveram que "*o Entscheidungsproblem deve ser considerado o problema principal da lógica matemática*" (em itálico no original).

O problema da decisão é o problema de encontrar um método *efetivo* (também se diz *mecânico* ou *algorítmico*) de acordo com o qual, dada uma fórmula da linguagem do cálculo de predicados, se determina se essa fórmula é, ou não, um teorema da lógica (i.e. deduzível apenas a partir dos axiomas do cálculo de predicados). Um método ou procedimento é *efetivo* se:

1. puder ser descrito através dum número finito de instruções exatas;
2. produzir o resultado desejável ao fim dum número finito de passos (desde que se sigam as instruções sem erro);
3. puder, em princípio, ser executado por um ser humano apenas com a ajuda de papel e lápis;
4. não exigir nem criatividade nem perspicácia por parte do ser humano.

Os algoritmos que as crianças aprendem para efetuar as operações básicas da aritmética são exemplos de procedimentos efetivos. Para quem sabe um pouco de lógica, o método das tabelas de verdade para decidir se uma fórmula do cálculo proposicional é uma tautologia é um procedimento efetivo.

Hilbert e Ackermann propuseram o seguinte problema: encontrar um método efetivo para separar os teoremas da lógica das outras fórmulas. Para teorias importantes da matemática, uma solução positiva para o *Entscheidungsproblem* permitiria decidir, de modo efetivo, se uma fórmula é um teorema dessa teoria.

Uma solução positiva para os três problemas hilbertianos da completude, consistência e decisão subscreveria uma visão magnífica da matemática. A matemática poderia ser vista como um grandioso cálculo formal – consistente, completo e decidível. A visão leibniziana dum *calculus ratiocinator* no domínio da matemática teria sido plenamente justificada. Perante um problema matemático, por mais difícil que fosse, bastaria "pegar na pena e sentar-se ao ábaco e (na presença de um amigo, se se quiser) dizer um para o outro: *calculemus*". O cálculo seria efetivo, cego, como na multiplicação em numeração decimal, e daria sempre resposta. A matemática seria segura (consistência) e responderia a todas as questões (completude). Não só não haveria problemas insolúveis em matemática como as suas soluções seriam encontradas de modo efetivo. Numa comunicação em Königsberg (hoje a cidade russa de Kaliningrado) em 1931, por ocasião dum grande encontro de cientistas e médicos e da sua agraciação como cidadão honorário, Hilbert profere uma vez mais a sua fé inabalável no poder da razão humana com as seguintes palavras: *Wir müssen wissen, wir werden wissen* (temos de saber, havemos de saber). Estas palavras estão gravadas no seu túmulo em Göttingen. Era esta a genial visão hilbertiana da matemática. O apelo de Hilbert à nova geração não era um apelo qualquer. Era um apelo cheio de ambição, um apelo à resposta a questões de natureza fundamental. Era um apelo à resolução definitiva dos problemas dos fundamentos da matemática e à "*honra do próprio conhecimento humano*".

Não se tinham passado ainda três anos sobre a palestra de Hilbert em Bolonha quando o jovem matemático austríaco Kurt Gödel (nascido em Brno, hoje na República Checa, em 1906) mostra que a axiomática da aritmética, desde que seja consistente, não é completa. Umas semanas mais tarde, mostra que não é possível demonstrar a consistência da aritmética por meio de métodos aceitáveis para Hilbert (os métodos da metamatemática). Curiosamente, Gödel fez o anúncio público do primeiro destes resultados também em Königsberg, precisamente na véspera da comunicação de Hilbert de 1931. A *volte-face* foi totalmente inesperada e apanhou de surpresa o mundo matemático. Os resultados de Gödel demoraram algum tempo a ser plenamente apreciados e compreendidos. Afinal, o programa de Hilbert não era possível de levar a cabo. Afinal o grande matemático estava errado.

M. H. A. (Max) Newman ensinava em Cambridge na altura do congresso de Bolonha. Esteve presente no congresso e certamente terá ouvido o apelo de Hilbert. Newman não era lógico, era um topologista. A topologia era, então, uma disciplina nova que estava a unificar e a generalizar várias partes da matemática e em cuja base a teoria dos conjuntos desempenhava um papel importante. Não é de admirar que Max Newman também se interessasse pelos fundamentos da matemática. Em Cambridge, a lógica e os fundamentos da matemática não eram propriamente uma novidade pois foi uma universidade pioneira da lógica moderna. Russell e o seu co-autor Alfred North Whitehead foram *fellows* no Trinity College de Cambridge. O matemático e filósofo Frank Ramsey (prematuramente falecido, com um pouco menos de vinte e sete anos, em 1930) também foi *fellow* em Cambrige (do King's College) e o célebre aluno de Russell, o filósofo Ludwig Wittgenstein, ensinava em Cambridge nos anos trinta. Max Newman decidiu oferecer um curso de fundamentos da matemática, mas não ao estilo tradicional de Cambridge (i.e., incidindo sobre o programa russelliano do logicismo). Ensinou um curso ao estilo de Hilbert. A terceira parte do curso, lecionada na primavera de 1935, terminava com a exposição dos resultados de Gödel.

O jovem inglês Alan Mathison Turing, nascido em 1912, estudava em Cambridge desde 1931 e foi assistir à terceira parte do curso de fundamentos da matemática de Newman. Nesse curso foi exposto ao programa de Hilbert, aos teoremas de Gödel e ao *Entscheidungsproblem*. Como iremos descrever, Turing vai resolver o problema da decisão pela negativa. Numa entrevista nos anos setenta, Max Newman afirma:

> Acredito que tudo tenha começado porque ele [Turing] frequentou um curso de fundamentos da matemática e de lógica dado por mim... Creio ter dito durante este curso que, por um processo construtivo, se entendia um processo puramente mecânico – e penso que terei mesmo dito que uma máquina o podia efetuar.

O leitor deve refletir no seguinte. A descrição de processo efetivo (ou mecânico) que mencionámos atrás é uma descrição um tanto imprecisa. Não é certamente uma definição matemática. Descreve, em traços largos, um conceito *fundamental*. Esta descrição é geralmente suficiente para sabermos que estamos perante um processo efetivo quando um deles nos é apresentado. Não é necessário saber, com rigor matemático, o que é um processo efetivo para nos convencermos, p. ex., que o algoritmo de multiplicação que aprendemos em criança é efetivo. Uma resposta negativa para o *Entscheidungsproblem* carece, porém, de uma abordagem diferente e Turing vai propor uma caracterização matematicamente precisa de processo efetivo. Newman acrescenta na citada entrevista:

> É claro que isto levou [Turing] a um novo desafio – que espécie de máquina – e isto inspirou-o a tentar resolvê-lo e a dizer o que seria uma máquina computacional perfeitamente geral.

O artigo de Turing de 1936 "On computable numbers with an application to the Entscheidungsproblem" (Sobre números computáveis com uma aplicação ao Entscheidungsproblem) é brilhante. Contém um análise conceptual do que é um processo efetivo e, concomitantemente, propõe uma definição matematicamente rigorosa do que se pode calcular com tais processos. Contém a descrição duma "super-máquina" com poder computacional para executar qualquer processo efetivo. Mostra que há problemas de decisão (i.e., de resposta "sim" ou "não") que não têm soluções através de processos efetivos. Finalmente, contém a solução para o *Entscheidungsproblem*.

O artigo introduz um simples e elegante dispositivo abstrato de natureza computacional, hoje conhecido por *máquina de Turing*. Trata-se de um dispositivo que opera de acordo com uma lista finita de instruções. O leitor deve ter em mente uma *imagem instantânea* da operação duma máquina de Turing como algo do género:

	...							1	1	1								...
q								Δ										

Tem-se uma fita infinita (em ambas as direções), dividida em quadrados adjacentes (casas). Cada casa da fita pode ter impresso um *símbolo* ou está em branco. A máquina tem uma cabeça de leitura e escrita que, a cada momento, examina uma casa. Na figura acima, a cabeça da fita (Δ, na figura) está a examinar (ler) uma casa que tem o símbolo 1. Todas as casas à sua esquerda estão em branco. À direita da casa da cabeça estão dois símbolos 1 seguidos de casas em branco. A cabeça pode mover-se, de cada vez, uma casa para a esquerda ou para a direita. Finalmente, em cada momento, a (imagem instantânea da) máquina encontra-se num determinado estado discreto (há um número finito, pré-fixo, de estados, dependendo da máquina). No exemplo acima, a máquina encontra-se no estado q. A imagem instantânea da máquina pode ser representada por q: $\ldots\ 00000000\underline{1}110000000\ \ldots$

onde os zeros representam casas em branco e o traço diz-nos onde se encontra a cabeça da máquina.

Uma máquina de Turing, propriamente dita, é uma lista finita de instruções, cada qual da seguinte forma: se a máquina está no estado E e a cabeça está a ler o símbolo X, então a máquina entra no estado F, substitui o símbolo X por Y e move a cabeça para a casa da esquerda (ou da direita). A lista não deve ser ambígua, i.e., não deve ter mais do que uma instrução aplicável quando a máquina está num determinado estado a ler um determinado símbolo. Por conveniência, permite-se também que Y possa ser o "símbolo branco", caso em que a instrução simplesmente apaga X. Como exemplo, considere-se a seguinte máquina (que denominamos M_1) com três instruções:

q	1	q	1	←
q	0	r	1	←
r	0	p	0	→

Acima, o símbolo 0 representa o "símbolo branco" e as setas são as instruções para mover a cabeça para a esquerda ou para a direita. As imagens instantâneas desta máquina podem encontrar-se num de três estados: q, r e p. A segunda instrução diz que se a (imagem instantânea da) máquina está no estado q com a cabeça a examinar uma casa em branco, então a máquina entra no estado r, imprime 1 e move-se uma casa para a esquerda. Se a configuração inicial da máquina é a imagem instantânea que discutimos acima, fica-se com a seguinte sequência:

q: $\ldots\ 00000000\underline{1}110000000\ \ldots$

q: $\ldots\ 0000000\underline{0}1110000000\ \ldots$

r: $\ldots\ 000000\underline{0}11110000000\ \ldots$

p: $\ldots\ 0000000\underline{1}1110000000\ \ldots$

O leitor está a ver como é que a "máquina" funciona? Note que a máquina para na quarta configuração acima, pois não há nenhuma instrução que comece por "p1". O leitor deve convencer-se de que, perante uma configuração inicial no estado q, com a cabeça sobre o símbolo mais à esquerda duma sequência finita de 1s (uns) consecutivos, o resultado da execução da lista de instruções é adicionar um 1 imediatamente à esquerda do primeiro 1, parando sobre a casa desse 1.

Vamos dar mais alguns exemplos de máquinas de Turing. Com alguma disciplina e concentração, estes exemplos não são difíceis de seguir. O leitor menos interessado pode ignorar a análise dos exemplos sem deixar de compreender o essencial mas, a meu ver, perde a oportunidade de adquirir "sensibilidade" sobre o que é um processo efetivo, sobre o que é um *programa*. Considere-se a máquina M_2:

q	1	r	1	→
r	1	q	1	→
q	0	q	0	→
r	0	p	0	←

Quando M_2 começa no estado q, com a cabeça sobre o símbolo mais à esquerda duma sequência finita (palavra) de 1s consecutivos, para se o número de símbolos é ímpar. Caso contrário, a cabeça prossegue para a direita sem nunca parar. O leitor beneficiará se experimentar alguns exemplos simples para se convencer de que M_2 opera como se diz (experimente com 111 e 1111).

Uma pequena modificação desta máquina, com dois estados especiais s (para "sim") e n (para "não"), decide se o número de 1s é par ou não, parando no estado correspondente. É a máquina M_3:

q	1	r	1	→
r	1	q	1	→
q	0	s	0	→
r	0	n	0	←

Estas três máquinas são bastante simples. No Apêndice I damos a lista de instruções duma quarta máquina M_4, um pouco mais complicada. É uma máquina que duplica o número de 1s quando se inicia no estado q com a cabeça no símbolo mais à esquerda duma sequência consecutiva de 1s.

Nesta altura, é conveniente introduzir alguma terminologia. Dada uma máquina de Turing M e uma palavra α (a *entrada*), considere-se a configuração inicial que está no estado q (o estado no canto superior esquerdo da lista de instruções, designado por *estado inicial*) e com a cabeça a examinar o símbolo mais à esquerda de α (o resto da fita está em branco). Se a máquina parar, escreve-se M(α)↓. Se não parar, escreve-se M(α)↑. P. ex., M_2(111)↓ e M_2(1111)↑.

Uma *máquina de decisão* é uma máquina de Turing M que para sempre (seja qual for a palavra de entrada) e que, quando para, fica num dos estados especiais s (resposta afirmativa) ou n (resposta negativa). A máquina M_3, considerada acima, é uma máquina de decisão. Para no estado s se a entrada tem um número par de 1s e para no estado n no caso contrário.

Dada uma máquina M e uma entrada α, se a máquina parar com a cabeça numa casa que não está em branco, a *saída* é a palavra que se inicia nessa casa e se prolonga para a direita até encontrar a primeira casa em branco. Se λ é a saída, escrevemos M(α) = λ. A primeira máquina considerada, a máquina M_1, adiciona sempre uma unidade à entrada. P. ex., M_1(111) = 1111. Em geral, M_1(α) = α+1. A máquina M_4 "duplica" a entrada: M_4(α) = 2α.

Não há dificuldade conceptual em estender as definições anteriores a máquinas que aceitam duas ou mais entradas. As entradas podem ser dadas separando-as por uma casa em branco. A seguinte máquina, denominada de M_5, efetua a soma de dois inteiros positivos (dados como sequências finitas de 1s). P. ex., quando a configuração inicial da fita é

q: … 0000$\underline{1}$1101111111000000 …

(entradas 3 e 7) a máquina para na configuração final

p: … 0000$\underline{1}$1111111110000000 …

(a saída é 10). O programa de M_5 é:

q	1	q	1	→
q	0	r	1	→
r	1	r	1	→
r	0	u	0	←
u	1	v	0	←
v	1	v	1	←
v	0	p	0	→

Este programa faz o seguinte: preenche com o símbolo 1 o espaço em branco que separa as duas entradas, apaga o último símbolo 1 da segunda entrada e, finalmente, coloca a cabeça no símbolo 1 mais à esquerda da fita. Escrevemos M_5(111,1111111) = 1111111111. Em geral, $M_5(\alpha,\beta) = \alpha+\beta$.

A grande ideia de Turing foi ver que as máquinas podem, *elas próprias*, ser consideradas entradas para máquinas de Turing. A máquina M_1, com três instruções, é descrita pela palavra q1q1←q∗r1←r∗p∗→ (para se evitar possíveis ambiguidades, denota-se o símbolo em branco das instruções da máquina em causa – aqui M_1 – pelo símbolo∗). Nada impede que se considere a configuração

...			q	1	q	1	←	q	*	r	1	←	r	*	p	*	→		...

E Δ

e que se tenha uma lista de instruções num alfabeto (finito) que contenha os símbolos q, r, p, ∗, 1, → e ← e estados E e outros. Mais interessantemente, podemos considerar uma configuração inicial com duas entradas, a primeira das quais uma lista de instruções duma máquina de Turing e a segunda uma "entrada" para a máquina cuja lista de instruções consta da primeira entrada. No caso da máquina M_1 e da entrada 111, ficar-se-ia com a seguinte configuração inicial:

... 00000$\underline{q}$1q1←q∗r1←r∗p∗→01110000 ...

Turing imagina agora uma lista de instruções que permita *seguir listas de instruções*. Uma máquina U programada assim (uma máquina *universal*), com entradas α_M e β – onde α_M é a lista de instruções de uma dada máquina de Turing M – tem como saída (quando há saída, quando $M(\beta)\downarrow$) a palavra $M(\beta)$. Intuitivamente, a máquina U opera sobre β seguindo as instruções da lista α_M. A lista de instruções de U está concebida de tal modo que isto aconteça. Temos, por exemplo, $U(\alpha_{M1},\beta) = \beta+1$ e $U(\alpha_{M4},\beta) = 2\beta$. Em geral, temos a equação:

$$U(\alpha_M,\beta) = M(\beta).$$

Por exemplo:

$$U(q1q1\longleftarrow q{*}r1\longleftarrow r{*}p{*}\longrightarrow,111) = 1111.$$

Na secção 7 do seu artigo, Turing esboça em pouco mais de três páginas como construir uma lista de instruções duma máquina universal. Como o leitor imaginará, não é simples dar uma descrição detalhada duma máquina universal de Turing dado que é uma tarefa muito propensa a "erros de programação". Quem já programou sabe perfeitamente que é comum cometer lapsos e até erros de conceção, especialmente se a linguagem de programação for muito pouco flexível (como é o caso das listas de instruções das máquinas de Turing). Na programação duma máquina universal, a procura e execução da instrução relevante da máquina de entrada é um processo cheio de passos artificiosos. Também é muito artificioso alocar memória (apesar de haver todo o espaço do mundo, visto que a fita é infinita).

O artigo de 1936 contém erros de programação que foram parcialmente emendados pelo próprio Turing numa correção publicada um ano depois. Mas lapsos e erros subsistiram e um colaborador de Turing (Donald W. Davies) apontou-lhe em 1947 alguns erros graves de programação:

> Quando disse isto a Turing, ele ficou impaciente e disse-me claramente que estava a perder o meu tempo e o dele com esforços sem valor.

Bastante mais tarde, numa entrevista nos anos setenta, Davies reporta que Turing "ficou muito irritado e (lhe) disse furiosamente que a questão realmente não interessava, que a coisa estava correta em princípio". Turing tinha razão: do ponto de vista conceptual estes detalhes não interessam e a ideia de máquina universal é correta. Em 1947, o trabalho já era conhecido pelos especialistas e fora imediatamente aceite como uma obra fundamental. Porém, os detalhes já interessariam a quem quisesse implementar, na prática, uma máquina universal. Se bem que as máquinas de Turing sejam teoricamente muito claras, não são adequadas do ponto de vista prático (p. ex., no que diz respeito a cálculos importantes que devem ser efetuados tão rapidamente quanto o *hardware* o permita) por causa duma série de problemas, alguns já referidos, como os de alocação de memória e sua consulta trabalhosa, alocação de espaço para a computação propriamente dita, grande ineficiência, necessidade de manobras muito artificiosas, etc. Tudo isso torna a programação muito difícil e pouco transparente, cheia de detalhes incidentais e de difícil articulação. Um código de programação ao nível da máquina (i.e., que especifica as operações básicas que a máquina faz) para ser *prático* deve utilizar operações básicas que possam ser realizadas de modo eficiente e confiável por meios eletrónicos e, além disso, ser relativamente fácil de utilizar por humanos. É, porém, crucial que o código seja suficiente para, em princípio (se não na prática, por limitações de memória e tempo), incorporar a ideia de máquina *universal*.

A história do aparecimento dos computadores eletrónicos digitais é fascinante mas este não é o lugar próprio para a contar. Não se deve ficar com a impressão de que Turing não se interessava por aplicações. Turing participou na história da construção de computadores, nomeadamente na construção de um dos primeiros computadores eletrónicos, o ACE (Automatic Computing Engine). O relatório técnico que Turing escreveu para o ACE (provavelmente nos últimos meses de 1945) continha discussões detalhadas de engenharia e, inclusivamente, avançava com uma verba concreta para o custo da construção da máquina. Naturalmente, a construção dum computador digital é em grande parte um empreendimento de engenharia. Porém, ainda que seja interessante falar dos problemas complicados de engenharia e de programação que tiveram que ser superados para construir os computadores eletrónicos assim como mencionar o incrível avanço tecnológico a que temos assistido desde meados dos anos quarenta, não se pode esquecer que venceu uma conceção de computador que desafia o *senso comum*. Ainda em 1956 (vinte anos depois do artigo seminal de Turing!), Howard Aiken, um dos pioneiros da computação e professor na Universidade de Harvard afirmava:

> … se porventura vier a acontecer que as bases lógicas de uma máquina concebida para a solução numérica de equações diferenciais coincida com as lógicas duma máquina concebida para a contabilidade de um armazém, considerá-la-ia a mais espantosa coincidência que alguma vez encontrei.

Este é o senso comum, mas os computadores atuais não são como o senso comum sugere. São computadores de *propósito geral*, capazes de ler *software* adequado, este sim de propósito específico. Os computadores atuais permitem ter como entradas instruções (*software*) desenhadas para desempenhar tarefas específicas: são, em suma, incorporações da ideia de máquina *universal*.

Hoje em dia os computadores são um utensílio doméstico, como uma televisão ou um frigorífico. Há qualquer coisa de pasmoso no facto de que, na raiz dos computadores pessoais que usamos hoje diariamente, esteja uma ideia (a ideia da *universalidade* em computação) que surgiu por intermédio de questões da fundamentação da matemática. Podemos fazer "história virtual" e imaginar um encadeamento de acontecimentos que tivesse levado à conceção e construção de computadores de propósito geral que não tivesse passado pelas discussões e problemas postos pela crise dos fundamentos da matemática do início do século XX. Mas o facto é que a "história real", a que realmente teve lugar, passou por questões extremamente teóricas (fundamentos da matemática). Não é este um caso único na história em que a ânsia pelo conhecimento e a vontade de compreender, assim como a procura desinteressada da verdade, estão na origem de grandes avanços tecnológicos.

Na parte final do seu artigo, como aplicação dos conceitos anteriormente introduzidos, Turing resolve o *Entscheidungsproblem*. O título da última secção é, precisamente, "Aplicação ao *Entscheidungsproblem*". Turing baseia-se em trabalho que efetuou numa secção prévia, onde exibe um problema de decisão que não pode ser decidido de modo efetivo (um problema destes diz-se *indecidível*). O problema é o seguinte: decidir, dadas entradas α_M e β, se a máquina M para com a entrada β, i.e., se $M(\beta)\downarrow$. A este problema chama-se o *problema da paragem* ("halting problem", em inglês). O raciocínio de Turing é por redução ao absurdo. Turing vai supor que o problema da paragem é decidível e, a partir desta suposição, vai chegar a uma contradição.

Suponhamos então, com vista a um absurdo, que o problema da paragem é decidível. Então existe uma máquina de decisão H que decide este problema. Isto quer dizer que o seguinte vale sempre:

$$H(\alpha_M,\beta) \text{ decide afirmativamente (para no estado s) se } M(\beta)\downarrow \text{, e}$$

$$H(\alpha_M,\beta) \text{ decide negativamente (para no estado n) se } M(\beta)\uparrow .$$

Com a ajuda desta máquina H, podemos construir uma máquina *diagonal* D que, com a entrada α_M, procede do seguinte modo:

1. obtém o resultado da decisão $H(\alpha_M, \alpha_M)$;
2. se a decisão é afirmativa, executa para sempre a instrução de mover a cabeça para a direita (portanto, entra numa situação em que nunca para);
3. se a decisão é negativa, para imediatamente.

Por construção, esta máquina goza da seguinte propriedade:

$$D(\alpha_M)\downarrow \text{ se, e somente se, } M(\alpha_M)\uparrow ,$$

qualquer que seja a máquina de Turing M. Ora, se a máquina M for a *própria* máquina D chega-se a uma contradição:

$$D(\alpha_D)\downarrow \text{ se, e somente se, } D(\alpha_D)\uparrow .$$

Por outras palavras, com a entrada α_D, a máquina D para se, e somente se, não parar! Isto é contraditório.

O problema da paragem é o problema indecidível paradigmático. Em geral, mostra-se que um dado problema é indecidível através da *redução* do problema da paragem a esse problema. A ideia é simples. Para mostrar que um problema é indecidível mostra-se que se fosse decidível então o problema da paragem também o seria. *Quod non* (o que não é o caso).

Turing reduz o problema da paragem ao *Entscheidungsproblem*, descrevendo um modo efetivo de associar a cada par M e β uma fórmula $F_{M,\beta}$ do cálculo de predicados de tal sorte que

$M(\beta)\downarrow$ se, e somente se, $F_{M,\beta}$ é um teorema do cálculo de predicados.

A decisão de paragem fica reduzida à decisão de ser teorema do cálculo de predicados. Logo, esta última não se pode fazer de modo efetivo. Com este argumento, Turing mostra que o *Entscheidungsproblem* é indecidível dando, portanto, uma resposta negativa à questão de Hilbert e Ackermann.

Quando Turing estava a terminar o seu artigo na primavera de 1936, tomou conhecimento de um trabalho do lógico americano Alonzo Church que também continha uma solução para o *Entscheidungsproblem*. O artigo intitulava-se "A note on the *Entscheidungsproblem*". Em ciência, é o autor que primeiro chega a um resultado que geralmente fica com o crédito de o ter obtido, mesmo que outro autor tenha independentemente chegado à mesma conclusão. Hoje a solução do *Entscheidungsproblem* é conhecida por *teorema de Church*. Apesar da publicação de Church, Max Newman apoiou e instigou a publicação do artigo de Turing já que o método deste era suficientemente diferente do de Church para merecer ser conhecido e disseminado. Isto é tanto mais verdade porque o artigo de Turing introduz o conceito de máquina universal e argumenta que a noção de máquina de Turing captura a noção informal de efetividade computacional. Na secção 9 do seu artigo, Turing observa que, até à data, não tinha havido nenhuma tentativa de mostrar que as caracterizações matemáticas de processos efetivos incluem, de facto, todos os processos que se podem naturalmente caracterizar de efetivos. É claro que um argumento que procure identificar uma noção matemática com uma noção informal (ou "natural") não pode ter o carácter duma demonstração matemática. No primeiro parágrafo da secção 9, Turing questiona-se: *"Quais são os processos possíveis que podem ser efetuados na computação (...)?"* A secção procura responder a esta pergunta através duma análise detalhada do que significa "um ser humano efetuar uma computação". A análise é necessariamente intuitiva e baseia-se num processo de eliminação e depuração em que detalhes irrelevantes para a computação são descartados e em que o processo computacional é reduzido a operações o mais simples possíveis. No final, Turing chega à sua noção de máquina. A secção 9 do artigo de Turing pode ser considerada um dos mais bem sucedidos casos de *filosofia aplicada*.

A identificação da noção informal de função *efetivamente* computável com uma noção matematicamente rigorosa tinha sido primeiramente proposta, também por Church, num outro artigo do ano de 1936. Church escreve:

> O propósito do presente artigo é propor uma definição de computabilidade efetiva que corresponda de modo satisfatório à noção intuitivamente vaga em cujos termos os problemas desta classe [de problemas efetivos] são frequentemente formulados (...)

E, numa nota de pé de página, acrescenta que "a proposta para identificar estas noções com a noção de computabilidade efetiva é feita pela primeira vez neste artigo". As noções a que Church se refere são duas noções de computabilidade, uma devida a Church e ao seu estudante Stephen C. Kleene (em termos do denominado *cálculo lambda*) e a outra devida a Gödel e ao jovem matemático francês, prematuramente falecido com 23 anos, Jacques Herbrand (esta por meio da noção de *recursividade por equações*). Para efeitos da nossa discussão, não é importante saber o que são estas noções mas apenas que elas são matematicamente rigorosas e que a sua equivalência tinha sido estabelecida recentemente (mais tarde, no apêndice ao seu artigo de 1936, Turing mostra que a sua definição de computabilidade em termos de máquina de Turing coincide com a noção de Church e Kleene: as três noções são, pois, equivalentes). A identificação entre estas noções formais e a noção informal de computabilidade efetiva ficou conhecida na literatura por *tese de Church*. Estes assuntos foram objeto de troca de ideias por parte de Church, Gödel e Kleene. Porém, a identificação da noção informal com a noção de computabilidade por meio do cálculo lambda era "completamente insatisfatória" para Gödel ao tempo. Também, numa carta a Martin Davis em 1965, Gödel explica que, na altura, "não estava de todo convencido que o [meu] conceito de recursão compreendia todas as recursões possíveis". O que convenceu Gödel da correção da tese de Church foi o artigo de Turing. A propósito duma formulação geral dos seus teoremas da incompletude, Gödel escreve em 1965:

> A obra de Turing fornece uma análise do conceito de “processo mecânico” (ou antes “algoritmo” ou “processo de cálculo” ou “processo combinatório finito”). Mostra-se que este conceito é equivalente ao de uma “máquina de Turing”.

Foi o exercício de filosofia aplicada de Turing que convenceu Gödel. O facto de ser possível dar uma definição *absoluta* da noção de computabilidade efetiva é classificado por Gödel como “uma espécie de milagre”. Gödel contrasta a noção de computabilidade efetiva com a noção de *definibilidade*. Para esta noção já parece não haver milagres. A noção de definibilidade é *relativa* à linguagem onde se formula a definição. (Vale a pena analisar a situação com algum detalhe e o leitor pode encontrar uma discussão deste assunto no Apêndice II.)

No caso da computabilidade efetiva, dá-se um “milagre” porque é possível caracterizar os processos mecânicos básicos que dão origem a toda e qualquer computação efetiva, o que está estreitamente ligado a não ser possível decidir efetivamente se uma determinada lista de instruções básicas dá, ou não, origem a um processo que termina. Deixamos o leitor com as seguintes palavras de Gödel, numa comunicação ao bicentenário da Universidade de Princeton em 1946:

> Parece-me que esta importância [do conceito de computabilidade de Turing] é consideravelmente devida ao facto de que, pela primeira vez, se ter conseguido dar uma definição absoluta de uma noção epistemológica com interesse, i.e., absoluta no sentido de não depender do formalismo escolhido.

Apêndice I

A máquina M_4 é dada pela seguinte lista de onze instruções:

q	1	r	0	→
q	0	p	0	→
r	1	r	1	→
r	0	u	0	→
u	1	u	1	→

u	0	v	1	→
v	0	t	1	←
t	1	t	1	←
t	0	w	0	←
w	1	w	1	←
w	0	q	0	→

Esta máquina, quando é iniciada no estado q com a cabeça no símbolo mais à esquerda da sequência 111, tem os seguintes movimentos:

q:	... 000$\underline{1}$110000000000 ...
r:	... 0000$\underline{1}$10000000000 ...
r:	... 00001$\underline{1}$0000000000 ...
r:	... 000011$\underline{0}$000000000 ...
u:	... 0000110$\underline{0}$00000000 ...
v:	... 00001101$\underline{0}$0000000 ...
t:	... 0000110$\underline{1}$10000000 ...
t:	... 000011$\underline{0}$110000000 ...
w:	... 00001$\underline{1}$0110000000 ...
w:	... 0000$\underline{1}$10110000000 ...
w:	... 000$\underline{0}$110110000000 ...
q:	... 0000$\underline{1}$10110000000 ...
r:	... 00000$\underline{1}$0110000000 ...
r:	... 000001$\underline{0}$110000000 ...
u:	... 0000010$\underline{1}$10000000 ...
u:	... 00000101$\underline{1}$0000000 ...
u:	... 000001011$\underline{0}$000000 ...
v:	... 0000010111$\underline{0}$00000 ...
t:	... 000001011$\underline{1}$100000 ...
t:	... 00000101$\underline{1}$1100000 ...
t:	... 0000010$\underline{1}$11100000 ...
t:	... 000001$\underline{0}$111100000 ...
w:	... 00000$\underline{1}$0111100000 ...
w:	... 0000$\underline{0}$10111100000 ...
q:	... 00000$\underline{1}$0111100000 ...
r:	... 000000$\underline{0}$111100000 ...
u:	... 0000000$\underline{1}$11100000 ...
u:	... 00000001$\underline{1}$1100000 ...

u:	$\ldots\ 000000011\underline{1}100000\ \ldots$
u:	$\ldots\ 0000000111\underline{1}00000\ \ldots$
u:	$\ldots\ 00000001111\underline{0}0000\ \ldots$
v:	$\ldots\ 000000011111\underline{0}000\ \ldots$
t:	$\ldots\ 00000001111\underline{1}1000\ \ldots$
t:	$\ldots\ 0000000111\underline{1}11000\ \ldots$
t:	$\ldots\ 000000011\underline{1}111000\ \ldots$
t:	$\ldots\ 00000001\underline{1}1111000\ \ldots$
t:	$\ldots\ 0000000\underline{1}11111000\ \ldots$
t:	$\ldots\ 000000\underline{0}111111000\ \ldots$
w:	$\ldots\ 00000\underline{0}0111111000\ \ldots$
q:	$\ldots\ 000000\underline{0}111111000\ \ldots$
p:	$\ldots\ 0000000\underline{1}11111000\ \ldots$

O leitor está a ver como é que esta máquina opera?

Apêndice II

No que se segue, vamos considerar a noção de definibilidade para sequências infinitas de zeros e uns (sucessão binárias). Um exemplo duma definição destas é: “a sequência que é 1 nas posições ímpares e é 0 nas posições pares”. Esta definição caracteriza a sucessão

1010101010101010101010101 …

As definições são veiculadas por intermédio de frases duma dada linguagem. Ora, as frases definidoras podem ser enumeradas. Podemos, mesmo, *definir* uma tal enumeração. Tomemos, então, uma dessas enumerações

D1, D2, D3, D4, D5, D6, …

onde cada Dn (com n = 1, 2, 3,…) é uma definição que dá origem a uma sucessão binária, designada por sn. Cada uma destas sucessões sn tem um número infinito de posições, cujos valores denotamos assim:

sn1, sn2, sn3, sn4, sn5, sn6, sn7, sn8, sn9,…

em que snk (com k = 1, 2, 3,…) denota o valor da k-ésima posição da sucessão sn. Note que cada snk ou é 0 ou é 1.

Agora podemos “diagonalizar” e *definir* a sucessão d (‘d’ de diagonalização) que é 0 nas posições k em que skk é 1, e que é 1 nas posições k em que skk é 0:

d1, d2, d3, d4, d5, d6, d7, d8, d9, …

Esta sucessão diagonal foi *definida*. Como as definições foram todas enumeradas, esta definição é, digamos, a i-ésima definição Di. Assim, a sucessão acima é

si1, si2, si3, si4, si5, si6, si7, si8, si9, …

ou seja, dk = sik para todo o k. Em particular, di = sii. Isto é um absurdo, pois d foi definida de tal modo que di é 0 se sii é 1 e di é 1 se sii é 0.

Qual é a conclusão a tirar? O que se passa é que não se pode pressupor que a definição da sucessão diagonal ainda possa ser feita na linguagem (formal) originariamente dada. Sempre que fixamos uma linguagem para fazer definições, há uma nova definição (a da sucessão diagonal) que *não pode* fazer parte da linguagem original. Por isso a noção de definibilidade é sempre relativa a uma linguagem.

No caso da computabilidade efetiva, os programas (listas de instruções) mantêm-se sempre na *mesma* linguagem. Este é o “milagre”. Então, perguntará o leitor, o que é que impede que se utilize um argumento de diagonalização, à semelhança do caso da definibilidade, e se chegue a uma contradição? O que o impede é a indecidibilidade do problema da paragem pois essa indecidibilidade impossibilita a existência duma enumeração *efetiva* de todas as máquinas de Turing que calculam sucessões binárias infinitas.

Apontamentos complementares

A comunicação de Hilbert ao congresso de Bolonha encontra-se traduzida para português em (Hilbert 2003: 276-284) com o título “Problemas na fundamentação da matemática”. O ensaio (Ferreira 1992) é um “divertimento” sobre paradoxos e, em particular, sobre o paradoxo de Russell. A enciclopédia organizada por João Branquinho e Desidério Murcho (Branquinho e Murcho 2001) contém várias entradas sobre temas mencionados no corpo deste artigo: paradoxo de Russell, logicismo, formalismo, intuicionismo, programa de Hilbert, fundamentos da matemática, teoria dos conjuntos, máquina de Turing, funções recursivas, etc. As lutas dos fundamentos da matemática dos anos vinte da século passado não eram encaradas de ânimo leve pelos seus protagonistas, como se pode constatar pelo artigo (Ferreira 2008). Neste artigo também se descreve o programa de Brouwer. Trata-se de um programa especialmente interessante pois, se bem que tenha um cariz tradicionalista, é o mais revolucionário de todos.

A nossa descrição do programa de Hilbert está longe de fazer justiça à subtileza e sofisticação da posição hilbertiana. Roça, por vezes, a caricatura. Apenas descrevemos a faceta do programa que nos interessava para apresentar o *Entscheidungsproblem*. Insistimos na parte simbólica e formal, ignorando a componente finitista e o denominado método dos elementos ideais. Pode consultar-se (Ferreira 1995) para uma visão mais completa do programa de Hilbert. Os artigos de (Kahle 2006) e (Pereira 2006) também são informativos. A comparação do formalismo com o xadrez aparece num artigo de Hermann Weyl dos anos vinte. Esse artigo encontra-se traduzido para o inglês em (Mancosu 1998: 123-142) sob o título "The current epistemological situation in mathematics". A citação de Hilbert sobre a condição de consistência encontra-se no seu artigo "Sobre o infinito", traduzido para português em (Hilbert 2003: 234-255) assim como a alusão à honra do conhecimento humano. Esta alusão faz parte da seguinte frase (os destaques em itálico são do original): "queria deixar patente que a clarificação definitiva da *natureza do infinito* se tornou necessária, não apenas por interesse especial das diversas ciências particulares, mas antes para a *honra do próprio conhecimento humano*". O *Entscheidungsproblem* foi enunciado por Hilbert e Ackermann em termos semânticos: mostrar que existe um processo efetivo para determinar se uma fórmula do cálculo de predicados é uma verdade lógica (i.e., verdadeira sob todas as interpretações). Esta formulação é equivalente à do texto. A afirmação de que o *Entscheidungsproblem* é o principal problema da lógica matemática aparece em (Hilbert e Ackermann 1928: 77). Se é verdade que a visão hilbertiana da matemática repousava nas soluções positivas para os problemas da consistência e da completude, já se podem levantar dúvidas se também repousaria na solução positiva do *Entscheidungsproblem* pois Hilbert descreve este problema como um problema da lógica matemática e não como um problema dos fundamentos da matemática.

A locução *procedimento efetivo* vem duma terminologia estabelecida no inglês: "effective procedure". Neste artigo não estamos a utilizar a palavra 'efetivo' no seu sentido comum em português, mas sim num sentido técnico (discutido no texto). Os quatro pontos da caracterização de processo efetivo foram adaptados de (Copeland 2017). Uma solução positiva para o *Entscheidungsproblem* seria, não só magnífica, mas também aterradora para alguns. O célebre matemático inglês G. H. Hardy comenta em (Hardy 1929) que "se [o *Entscheidungsproblem* tivesse uma solução positiva] então teríamos um conjunto de regras mecânicas para a solução de todos os problemas da matemática e as nossas atividades - como matemáticos - terminariam". O estatuto dos matemáticos (e da matemática) ficaria bastante abalado já que o seu trabalho seria, em princípio, substituível por uma máquina. Acrescente-se que uma solução positiva para o problema da completude num determinado domínio da matemática (p. ex. a aritmética) teria como consequência a possibilidade de decidir mecanicamente se as asserções formais desse domínio são, ou não, teoremas. Esta observação (simples de justificar) parece ter escapado a Hilbert, tendo sido explicitamente feita por Turing na secção 11 do seu artigo de 1936.

A comunicação de Hilbert ao encontro de Königsberg encontra-se traduzida para o inglês em (Ewald 1996: 1157-1165) sob o título "Logic and the knowledge of nature". Os detalhes das vidas de Hilbert, Gödel e Turing mencionados no corpo do artigo podem encontrar-se em (Reid 1986), (Dawson 1997) e (Hodges 1992). Em português, vale a pena consultar (Goldstein 2009) que, apesar de alguma falta de rigor, é interessante. (Davis 2004) é uma muito boa obra de divulgação mas, infelizmente, a tradução para português é muito distraída. O resultado de Gödel sobre a incompletude da aritmética (o sistema que Gödel trata no seu artigo é um sistema de classes que contém a aritmética; NB, do ponto de vista matemático isso é irrelevante) tem uma hipótese mais forte do que a consistência (a denominada ω-consistência). Em 1936, J. B. Rosser mostra que o requisito da ω-consistência pode ser enfraquecido para a consistência. Os artigos de Gödel e Rosser encontram-se traduzidos para português em (Gödel et al. 2009). Os artigos originais não são o melhor lugar para quem se queira iniciar nestes assuntos. Já as lições de Gödel "Acerca de proposições indecidíveis de sistemas matemáticos formais" e a exposição de Rosser "Uma exposição informal das demonstrações dos teoremas de Gödel e de Church", ambas em (Gödel et al. 2009), são bons pontos de partida. Há também manuais que explicam estes temas. Em português existem (Oliveira 2010) e, mais avançados, (Sernadas e Sernadas 2012) e (Carnielli e Epstein 2008). Este último contém discussões sobre questões dos fundamentos da matemática.

Os excertos das entrevistas de Newman foram citados a partir de (Copeland 2004: 206). Este livro, como o título indica, junta as principais publicações de Turing (incluindo, é claro, o "On computable numbers with an application to the Entscheidungsproblem" e a correção de 1937) e complementa-as com introduções e comentários. É claro que Turing não usou a expressão "máquinas de Turing", usou antes a expressão "a--máquinas" (para máquinas automáticas). A nossa descrição do artigo de Turing de 1936 é-lhe, em traços gerais, fiel mas modifica e simplifica (e omite) algumas discussões. Em primeiro lugar, as a-máquinas têm uma fita que é apenas infinita numa das direções tendo, portanto, um começo. As instruções para as a-máquinas são também ligeiramente diferentes daquelas que apresentámos e Turing adota convenções que não seguimos (como é o caso da divisão das casas em casas E, para *erasable*, e casas F, para *fixed*). Por outro lado, os números computáveis a que o título do artigo se refere são números reais, não são inteiros positivos. As listas de instruções para estes números têm como objetivo "imprimir" a sua representação decimal (infinita). Em conformidade com esta visão, a noção de paragem de Turing é diferente: é, essencialmente, a noção de entrar em círculo, caso em que a máquina fica presa na impressão duma casa decimal, não conseguindo prosseguir para a casa decimal seguinte. Turing usa a terminologia (pouco feliz) de máquinas "circulares" e máquinas "sem círculos". Ao problema de decidir se uma máquina não tem círculos, Turing chama-lhe o problema da satisfatoriedade. A expressão "halting problem" foi introduzida bastante mais tarde em (Davis 1958: 70). Turing apresenta duas demonstrações para a indecidibilidade. Sobre a primeira, Turing é duma grande candura. Escreve que *apesar da demonstração ser perfeitamente correta, tem a desvantagem de poder deixar no leitor a sensação de que "algo deve estar errado"* (destaque no original). Tanta simplicidade poderia deixar o leitor desconfiado… A demonstração que apresentamos no texto é uma versão da primeira demonstração de Turing. O leitor que queira saber *exatamente* o que Turing escreveu deve, é claro, ler o artigo original.

Há uma ocasião no nosso texto em que, para não quebrar a fluidez da exposição, comprometemos um pouco o rigor: foi na descrição das entradas de máquinas de Turing para a máquina universal. A máquina universal, sendo ela própria uma máquina de Turing, é dada por uma lista finita de instruções na qual constam apenas um número finito de estados e de símbolos. Por outro lado, as máquinas de Turing - como entradas para a máquina universal - consistem num número finito de estados e de símbolos que podem ser tão numerosos quanto se queira (o número finito depende da máquina). A máquina universal deve estar preparada para aceitar e trabalhar sobre entradas (listas de instruções) com um número tão grande quanto se queira de estados e símbolos. É fácil de tornear este problema por meio de codificações apropriadas. P. ex., podem-se codificar os estados q_1, q_1, q_3,... por EI, EII, EIII,... Bastariam dois símbolos (E e I) para descrever qualquer um deste número infinito de estados (não é necessário um símbolo novo para cada estado). *Mutatis mutandis* para a codificação de símbolos.

A citação de Davies foi extraída do terceiro parágrafo do artigo "Corrections to Turing's universal computing machine" publicado em (Copeland 2004). O comentário da entrevista de Davies é citado também a partir de (Copeland 2004). Esta obra, assim como o capítulo 8 de (Davis 2004), são bons lugares de consulta para quem estiver interessado na história da construção dos primeiros computadores eletrónicos de propósito geral. A citação de Aiken está no capítulo 7 do livro de Davis.

Os dois artigos de Church de 1936 podem encontrar-se na colectânea (Davis 1965). O segundo artigo que mencionámos, onde é formulada a tese de Church, intitula-se "An unsolvable problem of elementary number theory". A designação de "tese" para a proposta de Church é devida a Kleene num artigo de 1943. Veja-se (Davis 1965: 274) e (Davis 1982: 13). Por vezes, a tese de Church também é chamada de tese de Church-Turing. A classificação, por Gödel, da proposta de Church como sendo "completamente insatisfatória" aparece numa carta de 1935 de Church a Kleene, citada em (Davis 1982: 9). A carta de Gödel a Davis vem mencionada em (Gödel et al. 2009: 51-52), assim como os dois parágrafos de Gödel de 1965 (Gödel et al. 2009: 113). Gödel afirma que a noção de computabilidade é *absoluta*: o leitor deve comparar esta noção com a noção de definibilidade (discutida no Apêndice II) ou de dedutibilidade formal aritmética (não é absoluta pois depende da axiomática de base). A locução "uma espécie de milagre" e a citação final também se encontram em (Gödel et al. 2009: 883). Estamos a referir-nos ao artigo "Reflexões sobre problemas em matemática apresentados à conferência do bicentenário de Princeton". Apesar do contraste que Gödel faz da noção de computabilidade com as noções de demonstrabilidade e definibilidade o artigo é, realmente, uma tentativa de salvar estas noções do seu relativismo à linguagem. A noção epistemológica a que Gödel se refere é, presumivelmente, o conhecimento que é obtido apenas por meio dum cálculo - o que é identificado com o conceito matemático de computabilidade à Turing.

Na internet encontra-se disponível bastante informação sobre Turing. Em http://www.turingarchive.org está o "The Turing Digital Archive". Há também a página http://www.turingcentenary.eu/ do "2012 The Alan Turing Year". É possível encontrar o artigo de Turing de 1936 na internet. Também é possível encontrar na internet os meus artigos mencionados nestes apontamentos.

Agradecimentos

Quero agradecer a leitura atenta e os comentários do José Carlos Espírito Santo, Gilda Ferreira, Sérgio Fernandes, José Almeida Santos, Reinhard Kahle, Luís Moniz Pereira, Augusto Franco de Oliveira e da Malu. As suas observações e críticas foram muito importantes para melhorar este trabalho e torná-lo mais acessível mas, é claro, a responsabilidade da versão final e das suas falhas é inteiramente minha. Quero também agradecer à Olga Pombo por me ter dado a conhecer a citação de Leibniz sobre o *calculemus* (Leibniz 1960: 200) e tê-la traduzido do latim para português. Finalmente, devo uma palavra de agradecimento ao amável convite do organizador deste volume e à sua louvável iniciativa de, por este meio, celebrar o centenário do nascimento de Alan Turing. Um muito obrigado ao José Carlos Espírito Santo.

A FCT - Fundação para a Ciência e Tecnologia (através dos projetos PTDC/FIL-FCI/109991/2009 e PEst-OE/MAT/UI0209/2013) e a Faculdade de Ciências da Universidade de Lisboa (por meio da concessão duma licença sabática no ano letivo de 2012/2013) também apoiaram a feitura deste trabalho.

Bibliografia

Branquinho, J. e Murcho, D. 2001. *Enciclopédia de Termos Lógico-Filosóficos*. Publicações Gradiva.

Carnielli, W. e Epstein, R. L. 2008. *Computabilidade, Funções Computáveis, Lógica e os Fundamentos da Matemática*. Editora UNESP, São Paulo.

Copeland, B. J. 2004. *The Essential Turing*. Oxford University Press.

Copeland, B.J. 2017. "The Church-Turing Thesis". *The Stanford Encyclopedia of Philosophy*, Edward. N. Zalta (principal editor), URL = <https://plato.stanford.edu/entries/church-turing/>.

Davis, M. 1958. *Computability and Unsolvability*. McGraw-Hill. Reimpresso com um novo prefácio e apêndice por Dover Publications, Inc. em 1982.

Davis, M. 1965. *The Undecidable*. Raven Press.

Davis, M. 1982. "Why Gödel didn't have Church's thesis". *Information and Control* 54: 3-24.

Davis, M. 2004. *O Computador Universal*. Editorial Bizâncio.

Dawson, J. W. 1997. *Logical Dilemmas*. A K Peters.

Ewald, W. 1996. *From Kant to Hilbert*. Volume II. Oxford University Press.

Ferreira, F. 1992. "Como ser sério com palavras cruzadas". Em *Matemática e Cultura I*, organização de J. Furtado Coelho, 37-53. Centro Nacional de Cultura e Edições Cosmos.

Ferreira, F. 1995. "No paraíso sem convicção... (uma explicação do programa de Hilbert)". Em *Matemática e Cultura II*, organização de J. Furtado Coelho, 87-121. Centro Nacional de Cultura e SPB Editores.

Ferreira, F. 2008. "*Grundlagenstreit* e o intuicionismo brouweriano". *Boletim da Sociedade Portuguesa de Matemática* 58: 1-23.

Gödel, K. e al. 2009. *O Teorema de Gödel e a Hipótese do Contínuo* (2ª edição). Colectânea organizada e traduzida por M. S. Lourenço. Fundação Calouste Gulbenkian.

Goldstein, R. 2009. *Incompletude. A Demonstração e o Paradoxo de Kurt Gödel.* Gradiva.

Hardy, G. H. 1929. "Mathematical proof". *Mind* 149:1-25.

Hilbert, D. 2003. *Fundamentos da Geometria*. Edição revista e coordenada por A. J. Franco de Oliveira. Publicações Gradiva.

Hilbert, D. e Ackermann, W. 1928. *Grundzüge der theoretischen Logik*. Springer.

Hodges, A. 1992. *Alan Turing: the Enigma*. Vintage.

Kahle, R. 2006. "Os teoremas da incompletude de Kurt Gödel". *Boletim da Sociedade Portuguesa de Matemática* 55: 63-76.

Leibniz, G. W. 1960. *Die philosophischen Schriften von Gottfried Wilhelm Leibniz*. Volume VII. Edição de C. I. Gerhardt. Hildesheim: Olms, 1960.

Mancosu, P. 1998. *From Brouwer to Hilbert. The Debate on the Foundations of Mathematics in the 1920s*. Oxford University Press.

Oliveira, A. J. F. 2010. *Lógica e Aritmética* (3.ª edição revista e aumentada). Gradiva.

Pereira, L. M. 2006. "Gödel e a computabilidade". *Boletim da Sociedade Portuguesa de Matemática* 55: 77-90.

Reid, C. 1986. *Hilbert. Courant*. Springer Verlag.

Sernadas, A. e Sernadas, C. 2012. *Fundamentos de Lógica e Teoria da Computação*. College Publications.

3. Aplicações do processo diagonal

José Carlos Espírito Santo

No seu artigo de 1936 [T36], Turing deu uma resposta negativa ao *Entscheidungproblem*, o Problema da Decisão colocado por Hilbert e Ackermann, o qual perguntava, em termos modernos, se existia um algoritmo para decidir se uma dada fórmula da Lógica de 1ª Ordem é ou não é um teorema (ver [F19]). Este importante resultado para a Lógica assentava noutro resultado, a resposta negativa ao Problema da Paragem, não menos importante para disciplinas ainda não inventadas à altura: a teoria e a prática da programação de computadores. Com efeito, a resposta negativa ao Problema da Paragem, que neste artigo chamaremos de Teorema de Turing, significa que não existe um algoritmo para decidir se, dado um programa, a execução desse programa desencadeada por um dado *input* termina ou não.

O Problema da Paragem é tratado na secção 8 do artigo de Turing de 1936, justamente intitulada "*Application of the diagonal process*". O processo diagonal, ou diagonalização, a que este título se refere é um método de demonstração inventado e aplicado pelo matemático Georg Cantor no último quartel do Séc. XIX, no contexto de investigações sobre a teoria de conjuntos, e a que Turing recorre para provar o seu resultado sobre o Problema da Paragem. O Teorema de Turing seguia-se a outros feitos notáveis (por exemplo, o Paradoxo de Russell, os Teoremas da Incompletude de Gödel), todos com o mesmo espírito de revelar paradoxos ou limitações dentro da Lógica, todos recorrendo à diagonalização na sua demonstração. Em contraste, Cantor havia aplicado a diagonalização para obter aquilo que hoje se designa por Teorema de Cantor, uma prova de que, na teoria de conjuntos, se pode passar sucessivamente de um conjunto infinito a outro ainda maior – um teorema, portanto, que não poderia ter um espírito mais expansivo.

Não surpreende que o "processo diagonal" tenha desde sempre despertado um fascínio que também guia este artigo. Uma coisa que se fez várias vezes (ver, por exemplo, [Y03]) foi identificar um teorema subjacente ao Teorema de Cantor e a todos os resultados limitativos da Lógica: a diagonalização seria utilizada uma única vez para demonstrar esse teorema, e cada um dos outros resultados sairia como consequência desse teorema comum, em virtude de razões adicionais específicas. Neste artigo, vamos identificar um teorema subjacente aos Teoremas de Cantor e Turing, para poder ver aquilo que Turing teve de acrescentar ao processo diagonal para obter o seu resultado – que foi sobretudo a ideia de máquina universal.

Por sua vez, a máquina universal depende da ideia de codificação: uma máquina pode ser reduzida a um número e codificada como *input* para outra máquina. Esse código é uma entidade que é duas coisas ao mesmo tempo: a máquina que ele codifica (e assim é "desmaterializada"), e o *input* de outra máquina. A importância da codificação no trabalho de Turing tem certamente inspiração no trabalho de Gödel [G31], com que Turing tomou contacto pouco antes de escrever o seu artigo: aí vemos que um numeral, dentro de um sistema formal para a aritmética dos números naturais, pode representar um número que vai substituir uma variável numa fórmula, e pode simultaneamente codificar outra fórmula.

Imaginemos um tabuleiro de xadrez ou batalha naval onde, quer as linhas, quer as colunas desse tabuleiro estão numeradas. A diagonal principal é constituída pelas casas de coordenadas (1,1), (2,2), etc. O mesmo número identifica a linha e a coluna, e avançamos em duas dimensões em simultâneo, quando damos um passo nessa diagonal. No teorema subjacente que vamos propor, identificamos um conceito de codificação e sublinhamos o seu papel nas demonstrações através do processo diagonal. Veremos como a codificação permite a diagonalização, e como a diagonalização subentende uma codificação.

O que se segue, portanto, é a exposição de alguns resultados matemáticos. Mas espera-se que o leitor acompanhe o desenrolar dos acontecimentos. Tudo o que precisa de compreender à partida é pouco mais que os conceitos matemáticos de conjunto e função. Em três pontos, far-se-ão afirmações matemáticas cuja justificação não é possível nem oportuno dar em todo o detalhe: essas afirmações serão claramente isoladas. No resto do texto, o convite é para uma viagem ao jeito de Cantor e Turing e na sua peugada: munidos de matemática elementar, mas com a coragem e o engenho para desvendar os segredos do infinito e da computação.

Notação e terminologia matemáticas

Valores lógicos: $\mathfrak{v}$ (verdade) e $\mathfrak{f}$ (falsidade).

O conjunto $\{\mathfrak{v},\mathfrak{f}\}$ dos valores lógicos designa-se por $\mathfrak{L}$.

O conjunto $\{1,2,3,\ldots\}$ designa-se por $\mathbb{N}$ e os seus elementos dizem-se números naturais.

Sejam A um conjunto e a um objeto. A simbologia $a \in A$ significa que a é *elemento* ou *membro* de A. Seja ainda B outro conjunto. A simbologia $A \subseteq B$ significa que A é um *subconjunto* de B, e isto, por sua vez, quer dizer que cada elemento de A é também elemento de B. Por exemplo, o conjunto dos números naturais pares é um subconjunto de $\mathbb{N}$, mas não vice-versa, pois cada número natural par é um número natural, mas há pelo menos um número natural que não é par. Por outro lado, tem-se sempre $A \subseteq A$.

Uma *função* do conjunto A no conjunto B é uma correspondência que associa, a cada elemento de A, um e um só elemento de B. As funções são habitualmente designadas pelas letras *f*, *g*, ou *h*. A simbologia $f{:}A \rightarrow B$ significa que f é uma função do conjunto A no conjunto B; neste caso, dizemos que A é o *domínio (de definição)* de f, que f *está definida* em A, e que f é de *tipo* $A \rightarrow B$. Se $f{:}A \rightarrow B$ e $a \in A$, então a diz-se um *argumento* de f; o único elemento de B que corresponde a a diz-se o *valor* de f em a (ou a *imagem* por f de a) e representa-se por $f(a)$; e diz-se que o valor $f(a)$ se obtém por *aplicação* de f a a. Por exemplo, se f for a função que associa, a cada número natural, o seu dobro, então $f{:}\mathbb{N} \rightarrow \mathbb{N}$, 3 é um argumento de f, $f\ (3) = 6$ e 6 é o valor de f em 3.

O conceito de função exclui, portanto, a situação em que um argumento tem duas imagens. Mas é perfeitamente possível que dois argumentos tenham a mesma imagem. Uma função onde esta situação não ocorre diz-se *injetiva*. Uma função injetiva associa imagens diferentes a argumentos diferentes. Por exemplo, a função $f{:}\mathbb{N} \rightarrow \mathbb{N}$ que associa, a cada número natural, o seu dobro é injetiva. Por outro lado, o conceito de função de A em B, digamos, não exige que cada elemento do conjunto B seja imagem de algum argumento da função. A uma função onde este requisito seja satisfeito vamos chamar *exaustiva* (em vez de sobrejetiva, que é o termo técnico usualmente empregue). Voltando ao mesmo exemplo, f não é exaustiva, porque as imagens por f são necessariamente números pares.

Se $f{:}\mathbb{N} \rightarrow B$, então f é uma *enumeração* de elementos de B. A terminologia justifica-se se imaginarmos a listagem $f(1), f(2), f(3), \ldots$ Nesta listagem $f(1)$ é o primeiro elemento, $f(2)$ o segundo, etc. Em geral, se $n \in \mathbb{N}$, então dizemos que $f(n)$ é o n-ésimo elemento, ou elemento de ordem n, da enumeração. Se a enumeração f for exaustiva, então cada $b \in B$ ocorre pelo menos uma vez na listagem $f(1), f(2), f(3), \ldots$; e se for injetiva, então cada $b \in B$ ocorre no máximo uma vez na mesma listagem, pelo que a listagem não tem repetições. Por exemplo, seja de novo

$f\colon\mathbb{N} \to \mathbb{N}$ a função tal que $f(n) = 2n$, para cada $n \in \mathbb{N}$. Esta função é uma enumeração não exaustiva e sem repetições de números naturais.

Uma *função parcial* do conjunto A no conjunto B é uma função de tipo $A' \to B$, onde A' é algum subconjunto de A, subconjunto esse que se diz o domínio da função parcial. A simbologia $f\colon A \hookrightarrow B$ significa que f é uma tal função (mas não especifica o domínio de f). Por exemplo, se f é a correspondência que associa, a cada número natural par, a sua metade, então f é uma função parcial de $\mathbb{N}$ em $\mathbb{N}$, ou, também vamos dizer, uma função de *tipo* $\mathbb{N} \hookrightarrow \mathbb{N}$. Naturalmente, na definição anterior de função parcial, é permitido o caso em que A' é afinal A; nesse caso, a função diz-se *total*. Isto significa que o conceito de função parcial engloba o conceito de função: uma função é uma função parcial cujo domínio de definição A' afinal compreende à totalidade de A; ou seja, uma função é uma função parcial que afinal é total.

Um *predicado* é uma função de tipo $A \to \mathfrak{L}$. Se o domínio do predicado for $\mathbb{N}$, o predicado diz-se *numérico*. Se nada for dito em contrário, os predicados neste artigo serão numéricos. Um exemplo de um predicado numérico é a função $\mathfrak{p}$ definida por: $\mathfrak{p}(n) = \mathfrak{v}$, se $n \leq 5$; e $\mathfrak{p}(n) = \mathfrak{f}$, caso contrário. Este é o predicado que representaríamos simplesmente pela fórmula $n \leq 5$.

Cantor

Em 1874 Cantor introduziu o seu método da diagonalização, para provar que os números transcendentes são incontáveis [C74]. Nas décadas seguintes desenvolveu a teoria dos conjuntos abstratos. Depois, aplicou o método da diagonalização para obter um resultado conhecido como teorema de Cantor [C91]. A nós, este teorema interessa na seguinte versão:

Teorema (Cantor): Não existe uma enumeração exaustiva do conjunto de todos os predicados numéricos.

Uma enumeração exaustiva de todos os predicados numéricos é, recorde-se, uma função cujo domínio é o conjunto $\mathbb{N}$, cujas imagens são predicados numéricos, de tal modo que cada predicado numérico é imagem de algum elemento do domínio. Em vez de predicados numéricos, consideremos predicados com domínio num conjunto A arbitrário. Seja B o conjunto de todos esses predicados. De facto, o argumento de Cantor prova um teorema mais geral, que afirma não existir uma função $f{:}A \to B$ com a propriedade de exaustividade, isto é, tal que cada predicado em B seja imagem, pela função f, de algum elemento do domínio A. Agora repetimos o processo, com C o conjunto dos predicados com domínio B: não existe função $f{:}B \to C$ com a propriedade de exaustividade. E assim sucessivamente.

Qual o significado de tudo isto? A existência de uma função $f{:}X \to Y$ com a propriedade de exaustividade caracteriza precisamente a situação que, em termos informais, se descreveria como "o número de elementos de Y não excede o número de elementos de X". Mas a caracterização em termos da existência de uma certa função tem a vantagem de se aplicar a conjuntos arbitrários, possivelmente infinitos, e dispensar a definição de "número de elementos". Voltando aos conjuntos A e B do parágrafo anterior, a inexistência de uma função $f{:}A \to B$ com a propriedade de exaustividade tem a leitura de que o número de elementos de B *excede* o número de elementos de A. Ao iteramos o processo, concluímos que o número de elementos de C excede o número de elementos de B, e assim sucessivamente. Ora, na base desta cadeia está um conjunto A arbitrário. Se A for um conjunto infinito, obtemos que, apesar da infinitude de A, o número de elementos de B excede o número de elementos de A. Certamente, B é um conjunto infinito, mas, no sentido preciso dado pelo teorema de Cantor, "B é mais infinito do que A". E, a seguir, vem o conjunto C, que é "mais infinito do que B", e assim sucessivamente.

A demonstração do Teorema de Cantor é curta mas engenhosa. E estas observações que acabámos de fazer são matematicamente simples. Espanta que estejam ao alcance da razão humana, a uma distância de aparência tão curta, consequências tão dramáticas para a teoria matemática do infinito.

Turing

Para dar uma solução negativa ao Problema da Decisão, Turing precisava de sustentar a inexistência de um algoritmo para determinada tarefa. Milénios de História haviam produzido inúmeros *exemplos* de algoritmos. Mas a tarefa de Turing tinha uma exigência inédita, nada menos que dar uma definição de algoritmo e assim delimitar matematicamente a totalidade dos algoritmos e determinar os limites da computabilidade, para poder argumentar que o Problema da Decisão estava para lá desses limites. O ainda estudante Turing enfrentou corajosamente o desafio e deu a sua definição em termos de um conceito de máquina – aquilo que hoje se designa por máquina de Turing.

O conceito de máquina de Turing determina o conceito de *função computável* de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$. Como não vamos repetir aqui os detalhes da definição de máquina de Turing, a definição de função computável, na medida em que depende do conceito de máquina de Turing, não vai ser dada em todo o detalhe. Intuitivamente, uma função computável é uma função para a qual existe um algoritmo (máquina de Turing) que calcula o valor da função para qualquer argumento. Sejamos um pouco mais precisos.

Uma máquina de Turing M determina (ou calcula) uma função, designada por f_M, de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$, do seguinte modo. Dado $n \in \mathbb{N}$, fornecemos a M uma representação convencionada de n como *input*. A máquina executa a partir desse *input* e uma de duas situações podem acontecer: (i) a execução termina, e nesse caso lê-se, de uma forma previamente convencionada, um *output* $\mathfrak{l} \in \mathfrak{L}$ a partir da configuração em que a máquina parou, e define-se $f_M(n) = \mathfrak{l}$; (ii) a execução não termina, e nesse caso deixa-se $f_M(n)$ indefinido.

Uma função computável é a função calculada por alguma máquina de Turing. Portanto, uma função f de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$ é computável se existir uma máquina de Turing M tal que $f = f_M$.

Por exemplo, o predicado “n é par?” é computável. Se a convenção for representar na máquina os números naturais em notação binária, consideremos M a máquina que procura o dígito menos significativo e produz um *output* consoante esse dígito é 0 ou 1. A função f_M é o desejado predicado – a função que, dado $n \in \mathbb{N}$, devolve $\mathfrak{p}$ ou $\mathfrak{f}$ conforme n é par ou não.

Um predicado numérico pode não ser apenas um predicado sobre números, se os números *codificarem* outras entidades – por exemplo as próprias máquinas de Turing.

Para a determinação de cada máquina de Turing, precisamos de especificar, grosso modo, o conjunto finito de instruções que a máquina executa, sendo que cada instrução da máquina de Turing é também finitamente especificável. Assim, a especificação de uma máquina de Turing M requer uma quantidade finita de informação, que pode ser codificada num número natural n_M – de modo que a máquinas diferentes correspondam códigos diferentes.

Será importante, no que se segue, formar, para cada máquina M, o par ordenado (n_M, f_M), e por fim colecionar todos estes pares num conjunto, que vamos designar por $\#$. Se $(n, f) \in \#$, vamos habitualmente escrever $n = \#f$.

A codificação de máquinas de Turing pode ser feita de tal modo que o predicado "n é código de alguma máquina de Turing?" é computável. Um predicado muito mais difícil é: "n é código de uma máquina de Turing cuja execução a partir do *input* 1 termina?". Será computável este predicado? Uma resposta afirmativa implicaria exibir um algoritmo (máquina de Turing) que, a partir apenas do código $n = n_M$ de uma máquina M (o qual é apenas uma representação das instruções de M), respondesse $\mathfrak{v}$ ou $\mathfrak{f}$ a uma questão acerca de uma determinada execução dessa máquina.

Vejamos outro exemplo. Seja u a função de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$ definida por:

- $u(n) = f(n)$, se $n = \#f$ *e* $f(n)$ *está definido*
- $u(n)$ *está definido, caso contrário*

A condição "$n = \#f$ *e* $f(n)$ *está definido*" é equivalente à condição "existe máquina de Turing M tal que $n = n_M$ e a execução de M a partir do *input* n termina". Um dos resultados mais importantes do artigo de 1936 é o seguinte:

Facto 1: u é computável.

Este facto decorre da existência da chamada máquina de Turing *universal*. Designemos esta função u por *função universal*.

Vejamos uma variação, aparentemente inocente, da função anterior. O *predicado da paragem* é o predicado $\mathfrak{t}:\mathbb{N} \hookrightarrow \mathfrak{L}$ definido por

- $\mathfrak{t}(n) = \mathfrak{v}$, se $n = \#f$ *e* $f(n)$ *está definido*
- $\mathfrak{t}(n) = \mathfrak{f}$, *caso contrário*

Se este predicado fosse computável, teria de existir um algoritmo (máquina de Turing) para *decidir* (responder $\mathfrak{v}$ ou $\mathfrak{f}$) sobre a terminação da execução de M a partir do *input* $n = n_M$. Ora, o Teorema de Turing diz que tal algoritmo não existe.

Teorema (Turing): O predicado da paragem não é computável.

Teorema da diagonalização

Vamos agora identificar um teorema que subjaz aos Teoremas de Cantor e Turing e é demonstrado por diagonalização.

Seja $\mathfrak{D}$ o conjunto de todas as funções parciais de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$. Fixemos um subconjunto $\mathfrak{F}$ de $\mathfrak{D}$.

Uma *codificação* de $\mathfrak{F}$ é um conjunto – habitualmente designado por $\#$ – de pares ordenados (n, f), em que $n \in \mathbb{N}$ e $f \in \mathfrak{F}$, conjunto esse satisfazendo dois requisitos:

i) Toda a função $f \in \mathfrak{F}$ tem um código; ou seja, para toda a função $f \in \mathfrak{F}$, existe $n \in \mathbb{N}$ tal que $(n, f) \in \#$.
ii) Se n é código de f e f', então f e f' concordam no argumento n; ou seja, se $(n, f) \in \#$ e $(n, f') \in \#$, então f e f' estão ambas definidas em n e o valor de ambas as funções é igual nesse argumento, ou f e f' estão ambas indefinidas em n.

Em vez de $(n, f) \in \#$, vamos habitualmente escrever $n = \#f$ (ler: n é código de f).

Seja $\mathfrak{n}:\mathfrak{L} \hookrightarrow \mathfrak{L}$ a função que troca os valores lógicos, isto é, $\mathfrak{v}$ envia $\mathfrak{f}$ em e vice-versa. O nome $\mathfrak{n}$ é para sugerir "negação".

Seja $f \in \mathfrak{F}$. A partir desta função, pode fabricar-se uma outra função g, do mesmo tipo, a que vamos chamar a *negação* de f. Dado $n \in \mathbb{N}$, aplica-se f a este argumento. Se $f(n)$ não está definido, também $g(n)$ não está; caso contrário, $f(n)$ é um valor lógico, e o valor lógico $g(n)$ é definido como sendo $\mathfrak{n}(f(n))$.

Ora pode dar-se o seguinte caso notável: sempre que fabricamos g a partir de $f \in \mathfrak{F}$ segundo o modo descrito, o resultado é ainda uma função de $\mathfrak{F}$. Nesse caso, diz-se que $\mathfrak{F}$ é *fechado* para a operação de negação.

Seja g função parcial de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$. Dizemos que g *respeita* uma codificação # se g satisfaz as duas condições seguintes, para cada $f \in \mathfrak{F}$:

(A) $g(n) = \mathfrak{v}$, se $n = \#f$ e $f(n) = \mathfrak{v}$,
(B) $g(n) = \mathfrak{f}$, se $n = \#f$ e $f(n) = \mathfrak{f}$.

Estas duas condições podem ser combinadas numa só: se $n = \#f$ e $f(n)$ está definido, então $g(n) = f(n)$. Talvez assim fique mais clara a razão de g "respeitar" #: perante um argumento n, e caso n codifique a função f, o valor de g para esse argumento não é, por assim dizer, escolhido livremente, antes é o valor de f para esse argumento, quando este último valor está definido.

É simples definir funções satisfazendo (A) e (B). Um exemplo importante é o predicado $\mathfrak{a}$ definido por:

- $\mathfrak{a}(n) = \mathfrak{v}$, se $n = \#f$ e $f(n) = \mathfrak{v}$
- $\mathfrak{a}(n) = \mathfrak{f}$, caso contrário

Vamos designar este predicado por *predicado de auto-aceitação* (relativo à codificação #). A palavra "aceitação" advém de podermos dizer que *f aceita n* quando $f(n) = \mathfrak{v}$. Assim, satisfazem o predicado de auto-aceitação aqueles números n que codificam uma função f que aceita o próprio código $n = \#f$.

Isto é tudo quanto precisamos para obter o desejado teorema:

Teorema 1: Seja $\mathfrak{D}$ o conjunto de todas a funções parciais de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$. Seja $\mathfrak{F}$ um subconjunto de $\mathfrak{D}$ que admite uma codificação # e é fechado para a operação de negação. Então, nenhuma função parcial g de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$ satisfaz simultaneamente as três condições:

1. $g \in \mathfrak{F}$,
2. *g respeita* #,
3. g é total.

Por outras palavras: se uma função $g \in \mathfrak{F}$ *respeita* #, então g não é total (existe $n \in \mathbb{N}$ tal que $g(n)$ não é $\mathfrak{v}$ nem $\mathfrak{f}$).

Uma consequência imediata é a seguinte: um predicado numérico que respeite # não pertence a $\mathfrak{F}$. Por exemplo:

Corolário: O predicado de auto-aceitação não pertence a $\mathfrak{F}$.

Demonstração do Teorema: Seja g: $\mathbb{N} \hookrightarrow \mathfrak{L}$ em $\mathfrak{F}$ satisfazendo (A) e (B). Seja h a negação de g. Isto significa que:

i) $h(n) = \mathfrak{f}$, se $g(n) = \mathfrak{v}$,
ii) $h(n) = \mathfrak{v}$, se $g(n) = \mathfrak{f}$,
iii) $h(n)$ *está indefinido, se* $g(n)$ *está indefinido*.

Como $\mathfrak{F}$ é fechado para a operação de negação, $h \in \mathfrak{F}$, pelo que h tem direito a um código $k = \#h$. Este código é um número natural onde g não está definida. Vejamos porquê.

De i, ii, (A) e (B) segue, para cada $f \in \mathfrak{F}$:

(C) $h(n) = \mathfrak{f}$, se $n = \#f$ e $f(n) = \mathfrak{v}$,
(D) $h(n) = \mathfrak{v}$, se $n = \#f$ e $f(n) = \mathfrak{f}$.

Qual o valor de h para o argumento k? Consideremos os casos particulares de (C) e (D) resultantes de escolher f como sendo *a própria função h*. Por um lado, se $h(k) = \mathfrak{v}$, concluímos, pelo referido caso particular de (C), que $h(k) = \mathfrak{f}$; donde, $h(k)$ não é $\mathfrak{v}$. Por outro lado, se $h(k) = \mathfrak{f}$, concluímos, pelo referido caso particular de (D), que $h(k) = \mathfrak{v}$; donde $h(k)$ não é $\mathfrak{f}$. Portanto, $h(k)$ não é $\mathfrak{v}$ nem $\mathfrak{f}$. Mas, então, devido a i e ii, também $g(k)$ não é $\mathfrak{v}$ nem $\mathfrak{f}$. **QED**

Processo de diagonalização

A demonstração do Teorema 1 usa o “processo diagonal”, a técnica de diagonalização inventada por Cantor. Porquê “diagonalização”? Imaginemos uma tabela infinita com uma linha e uma coluna por cada número natural.

	1	2	3	…	□	…	□	…	□	…
1										
2										
3										
…				…						
□					$\mathfrak{I}$		$\mathfrak{I}$		$\mathfrak{n}(\mathfrak{I})$	
…						…				
□										
…								…		
□							?		?	
…										

Cada entrada desta tabela é identificada por um par de coordenadas (i, j), com i, (respetivamente j) o número natural que identifica a linha (respetivamente coluna) em que a entrada está. A *diagonal* da tabela (ver as entradas sombreadas) são as entradas com coordenadas (i, j) em que $i = j$.

Se $j = \#f$, a entrada (i, j) define-se assim: se $f(i)$ está definido, $(i, j) = f(i)$; caso contrário, pomos "?" na entrada (i, j). Portanto, se $j = \#f$, a j-ésima coluna representa a função f. Se j não é código de uma função, a j-ésima coluna é deixada em branco.

Seja $g\colon \mathbb{N} \hookrightarrow \mathfrak{L}$ em $\mathfrak{F}$ satisfazendo (A) e (B) e seja $m = \#g$. Decorre das condições (A) e (B), e de $m = \#g$, que: dado um número natural n, se a entrada (n, n) contiver um valor lógico, a entrada (n, m) contém o mesmo valor lógico (se a entrada (n, n) for "?", nada é exigido à entrada (n, m)). Então, para o argumento $n = k$, temos:

> (*) se a entrada (k, k) contiver um valor lógico, a entrada (k, m) contém o mesmo valor lógico.

Sejam h a negação de g e $k = \#h$. Agora, por momentos, k desempenha o papel de código de h. Dado um número natural n, decorre da definição de h que cada uma das entradas (n, m) e (n, k) contém um valor lógico se e só se a outra contiver; e, nesse caso, o valor lógico contido em (n, k) resulta de aplicar a operação de negação ao valor lógico contido em (n, m). Então, para o argumento $n = k$ (k volta ao papel de argumento), segue facilmente:

> (**) se a entrada (k, k) contiver um valor lógico, a entrada (k, m) contém a negação desse valor lógico.

De (*) e (**) segue que (k, k) não contém um valor lógico.

Grosso modo, a m-ésima coluna contém uma cópia dos valores lógicos da diagonal, enquanto a k-ésima coluna contém a negação desses valores lógicos. Ora a diagonal e a k-ésima coluna intersetam-se na entrada (k, k) pelo que esta entrada não pode conter valor lógico (caso contrário, esse valor lógico teria de ser igual à sua negação).

Demonstração dos teoremas de Cantor e Turing

Como prometido, vamos ver agora que os Teoremas de Cantor e Turing são consequências do Teorema 1.

Demonstração do Teorema de Cantor: Seja $\mathfrak{F}$ o conjunto de todos os predicados. Suponhamos que havia uma enumeração exaustiva φ de $\mathfrak{F}$. Consideremos o gráfico de φ, o conjunto dos pares $(n, \varphi(n))$. Este conjunto, designemo-lo por #, definiria uma codificação de $\mathfrak{F}$: o requisito I seria satisfeito porque φ é exaustiva; o requisito II seria satisfeito porque φ é uma função (se $(n, f) \in \#$ e $(n, f') \in \#$, então $f' = f$''). Por outro lado, $\mathfrak{F}$ é fechado para a operação de negação, porque a negação de um predicado é um novo predicado. Do Teorema 1 seguiria que o predicado $\mathfrak{a}$ de auto-aceitação não pertence a $\mathfrak{F}$, o que é absurdo. A hipótese da existência de uma enumeração exaustiva de $\mathfrak{F}$ conduziu-nos a uma contradição. Logo, não existe tal enumeração. **QED**

Com vista a provar o Teorema de Turing, seja agora $\mathfrak{F}$ o conjunto de todas as funções computáveis de tipo $\mathbb{N} \hookrightarrow \mathfrak{L}$. Um resultado técnico simples é o seguinte:

Facto 2: $\mathfrak{F}$ é fechado para a operação de negação.

Recordemos o conjunto #, formado por todos os pares ordenados (n_M, f_M). Este conjunto é uma codificação de $\mathfrak{F}$: o requisito I é satisfeito porque toda a função computável é da forma f_M; o requisito II é satisfeito por que, se $n = \#f$ e $n = \#f'$, então $n = n_M$, para uma máquina M bem definida, e segue que $f = f_M = f'$.

Recordem-se a função universal u e o predicado da paragem $\mathfrak{t}$, referido no Teorema de Turing. Seja o predicado $\mathfrak{b}$ definido por:

- $\mathfrak{b}(n) = u(n)$, *se* $\mathfrak{t}(n) = \mathfrak{v}$
- $\mathfrak{b}(n) = \mathfrak{f}$, *nos restantes casos*

Note-se que $\mathfrak{b}$ é uma variante de u, onde o predicado $\mathfrak{t}$ é usado para antecipar a não-terminação.

Facto 3: Se $\mathfrak{t}$ for computável, $\mathfrak{b}$ também o é.

Um esboço da prova deste facto é o seguinte: o predicado $\mathfrak{b}$ resulta de uma espécie de "composição" das funções $\mathfrak{t}$ e u. Essa operação de "composição" preserva computabilidade: se aplicada a funções computáveis, produz uma função computável. Sabendo que u é computável, concluímos: se $\mathfrak{t}$ fosse computável, o resultado dessa "composição" também o seria.

Por fim, é fácil verificar que $\mathfrak{a}$ e $\mathfrak{b}$ são, afinal, a mesma função: usando sucessivamente as definições de $\mathfrak{b}$, u, $\mathfrak{t}$ e $\mathfrak{a}$, prova-se que $\mathfrak{b}(n) = \mathfrak{v}$ se, e só se, $\mathfrak{a}(n) = \mathfrak{v}$. Assim, vemos que, através da "composição" com o predicado da paragem, a função universal é transformada no predicado de auto-aceitação.

Demonstração do Teorema de Turing: Do Teorema 1 segue que o predicado de auto-aceitação $\mathfrak{a}$ não pertence a $\mathfrak{F}$. Então $\mathfrak{b}$ não é computável (porque $\mathfrak{a}$ e $\mathfrak{b}$ são a mesma função e $\mathfrak{F}$ é o conjunto das funções computáveis). Logo, $\mathfrak{t}$ também não é computável, devido ao Facto 3. **QED**

Referências bibliográficas

[C74] G. Cantor, *Ueber eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen*. Journal für die Reine und Angewandte Mathematik, 77 (77): 258-262, 1874

[C91] G. Cantor, *Ueber eine elementare Frage der Mannigfaltigkeitslehre*. Jahresbericht der Deutschen Mathematiker-Vereinigung. 1: 75-78, 1891

[G31] K. Gödel. Über *formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*. Monatshefte für Mathematik und Physik, 38: 173-198, 1931. Tradução: *On Formally Undecidable Propositions of Principia Mathematica and Related Systems I*. Dover 1992.

[F19] F. Ferreira. *O problema da decisão e a máquina universal de Turing*. Neste volume.

[T36] A. Turing. *On computable numbers, with an application to the Entscheidungproblem*. Proceedings of the London Mathematical Society, 42(2):230-265, 1936.

[Y03] N. S. Yanofsky, *A universal approach to self-referencial paradoxes, incompleteness and fixed points*, Bulletin of Symbolic Logic, 9(3), 362-386, 2003.

Parte III –

Computabilidades

4. Alan Turing e a Filosofia da Mente

Sofia Miguens

«Sandy: Talking about consciousness really does call for a certain amount of restraint.

Otherwise you might as well just jump on either the solipsim bandwagon – "I am the only conscious being in the universe" – or the panpsychism bandwagon – "Everything in the universe is conscious!"
(…) I maintain that people give other people credit for being conscious simply because of their continual external monitoring of them – which is itself something like a Turing test.»
The Turing Test – a Coffeehouse Conversation,
in D. Hofstadter and D. Dennett,
The Mind's I – fantasies and reflections on self and soul

Talvez cause alguma surpresa saber que o nome de Alan Turing é um nome muito presente no campo da filosofia da mente. Isso acontece sobretudo devido ao artigo “Computing Machinery and Intelligence”, que Turing publicou em 1950 numa das mais célebres revistas de filosofia, a revista *Mind*. Nesse artigo explora (como filósofo, poderíamos dizer) as implicações do seu trabalho como matemático e cientista e propõe o célebre Teste de Turing. Mas Turing está também no horizonte de qualquer curso de filosofia da mente através dos conceitos de Máquinas de Turing e Máquina de Turing Universal. Todos estes conceitos fazem inevitavelmente parte da ‘bateria de instrumentos’ dos filósofos da mente.

A minha intenção neste artigo é simplesmente contextualizar a presença de Turing no campo da filosofia da mente, começando por dizer um pouco acerca daquilo que se faz em filosofia da mente. Como muitos leitores nunca terão ouvido falar de filosofia da mente, começo por caracterizá-la através daqueles a que chamarei ‘o problema central’ e ‘os problemas específicos’. Analisarei em seguida uma resposta de referência a esses problemas, a chamada Teoria Representacional-Computacional da mente. Em seguida, descreverei alguns passos da história da disciplina nos últimos sessenta anos, situando aí o Teste de Turing e o Quarto Chinês, uma experiência mental do filósofo americano John Searle, elaborada com o Teste de Turing em mente e com o propósito de contestá-lo. Entre o Teste de Turing e o Quarto Chinês fica ‘encenado’ muito do que está em causa quando pensamos sobre a natureza do pensamento, da inteligência e da consciência (e isto é afinal, basicamente, o que se faz em filosofia da mente).

1. O problema central e os problemas específicos

A filosofia da mente é uma disciplina da filosofia cujo problema nuclear é o problema das relações entre pensamento e matéria. Podemos formulá-lo assim: *Como é possível que os meus pensamentos e os neurónios no meu cérebro façam parte do mesmo mundo?* Como é possível que estas células eletricamente excitáveis feitas de proteínas tenham alguma coisa a ver com a ocorrência de crenças aritméticas como ‘2+2=4’ ou crenças históricas como ‘D. Afonso Henriques foi o primeiro

Rei de Portugal'? Como é que matéria biológica encerrada dentro do crânio pode representar formas de as coisas serem cá fora no mundo?

'2+2 = 4'
'D. Afonso Henriques foi o primeiro Rei de Portugal'
'Aquela parede é vermelha'
'Estamos em Lisboa'

Figura 1 Neurónios e pensamentos

É claro que nesta formulação ela própria ('*Como é possível que os meus pensamentos e os neurónios no meu cérebro façam parte do mesmo mundo?')* está já presente uma suposição: a suposição segundo a qual os meus pensamentos e os neurónios no meu cérebro fazem de facto parte de um mundo que é o mesmo mundo. Aceitar a suposição, ou pressuposição, é ser de alguma forma materialista ou fisicalista, e é importante notar que quase todos (mas não todos) os filósofos contemporâneos que trabalham em filosofia da mente são materialistas ou fisicalistas.

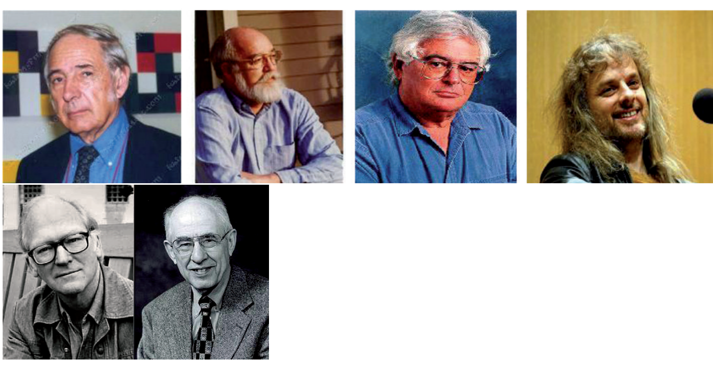

Figura 2 Alguns filósofos da mente (respetivamente: John Searle, Daniel Dennett, Jerry Fodor, David Chalmers, Donald Davidson, Hilary Putnam)

Pelo menos, muitos filósofos da mente contemporâneos partem da seguinte ideia: *Temos uma conceção de nós próprios a partir de dentro, em 1.ª pessoa como seres conscientes, racionais e livres / Temos uma conceção em 3.ª pessoa da nossa arquitetura cognitiva e do funcionamento dos nossos cérebros / Deve ser possível conjugar essas perspetivas de 1.ª e 3.ª pessoa numa mesma conceção metafísica do mundo.* Esse mundo é um, inclui matéria e pensamento, logo é necessário desenvolver uma metafísica, uma teoria geral da natureza última da realidade, que permita conceber tal facto. Esta tentativa de responder à questão metafísica acerca do 'lugar' para o pensamento num mundo basicamente físico está no núcleo da filosofia da mente. A filosofia da mente não visa apenas este problema metafísico – tem por exemplo fortes relações com a epistemologia e com ética, uma vez que se algum ser é mental e consciente, isso tem implicações relativamente à forma como sabemos tal facto e também relativamente à forma como o tratamos. Mas a verdade é que é a interrogação metafísica acerca de pensamento e mundo (físico) que está no centro de tudo.

Antes de passar aos problemas específicos gostaria de fazer uma observação breve acerca de método. Na filosofia da mente contemporânea, os problemas são caracteristicamente abordados de forma *naturalist*a**.** Em termos práticos isso significa que a investigação acerca do mental é feita relacionando e levando em conta aquilo que vai sendo conhecido acerca do suporte físico (neuronal ou outro) da cognição. De que forma? Pensemos num agente cognitivo como cada um de nós. À 'matéria' a partir da qual trabalham os filósofos da mente podemos chamar o nível pessoal ou fenomenológico. Chama-se nível pessoal ou fenomenológico ao 'como é ser a partir de dentro', sentir-se ser e pensar (*what it is like to be*, na célebre expressão do filósofo americano Thomas Nagel) – é a isso que temos acesso quando damos por nós próprios pensando e sentido e quando descrevemos a nossa vida mental. A referência dos cientistas cognitivos, por seu lado (sejam eles psicólogos, neurocientistas, etc) é o nível sub-pessoal, aquilo que se passa nos nossos cérebros, e que é algo a que não temos acesso direto mas que é causalmente responsável pelo que somos enquanto mentais (por exemplo o que se passa numa área determinada do nosso córtex visual quando vemos vermelho). Devido à postura metodológica naturalista, grande parte da filosofia da mente é hoje filosofia da ciência cognitiva e muitos dos problemas específicos que vou identificar são tratados em diálogo com a ciência cognitiva. Esta é mais uma razão para a importância de Turing: as suas ideias transcenderam campos disciplinares – e isso parece ser necessário para fazer ciência da cognição.

Reparemos então agora que coisas muito inesperadas ocorrem num mundo que se presume ser fundamentalmente físico (um mundo de matéria, energia, partículas, forças). Nomeadamente, deparamo-nos com:

1. Intencionalidade. A intencionalidade é representação mental, o ser acerca de (*aboutness*) que caracteriza o mental: quando temos por exemplo a crença 'esta parede é amarela' ou 'estas mãos são minhas' representamos mentalmente que esta parede é amarela e que estas mãos são minhas.

2. Consciência. Podemos definir consciência como sentir-se ser, sentir-se sentir e pensar, *what it's like to be*, *qualia* (estados qualitativos, sentidos, da mentalidade), ou estados mentais acerca de outros estados mentais (sei que sinto, sei que quero, sei que penso, etc.)

3. Ação (intuitivamente: o pensamento causa coisas). Pensemos num humano e nos seus movimentos intencionados e com propósito: ir daqui para ali para ir buscar um livro, atirar um vaso da janela de forma a acertar na cabeça da pessoa que passa. O que é que o distingue de um robô sem interior, ou de um sonâmbulo, que por hipótese executasse os mesmos movimentos físicos? Parece que algo de mental causa e dá sentido aos movimentos corporais do humano. O que o distingue de um ser 'sem interior mental' (a que os filósofos da mente tenderão a chamar *zombie*) é um apercebimento, que não acontece por observação, mas como se fosse a partir de dentro, dos movimentos de um corpo físico, o seu próprio corpo. É-lhe assim possível dar as razões dos movimentos do corpo. Há um corpo que se move, é este, é o meu, eu sei por que se move - são as minhas razões ('coisas mentais') que causam os movimentos. É a isso que chamamos *ação*, auto-intencionar-se de uma forma que envolve entidades mentais (crença, desejo, propósito). São estas coisas mentais que aparentemente causam, e portanto poderão também explicar, os movimentos de um corpo físico. O pensamento causa (aliás, pense-se: sem isto não seria possível por exemplo alguém ser livre; só a existência de ação pode dar apoio ao contraste entre o que eventualmente chamaremos liberdade e os movimentos não impedidos, até mesmo aparentemente propositados, de um corpo).

4. Pessoas. Já J. Locke, no século XVII, no *Essay Concerning Human Understanding* (1689) dizia que não bastava existir o corpo de um indivíduo de espécie humana para haver uma pessoa. E – embora ele fosse religioso e não estivesse a afastar as 'almas' – disse também que pessoa e alma não são a mesma coisa. Falamos de "Pessoas" quando há seres que se apercebem a si mesmos como um e o mesmo ser, único e unificado ao longo do tempo, e que apercebem de forma mentalista os movimentos do corpo próprio, podendo dar razões das suas ações. Nos termos de Locke, 'pessoa' é uma 'noção forense'. Na ausência de seres deste tipo não teria sentido termos moral, direito, política. (Note-se que não temos que pensar que só os humanos são pessoas, ou que todo e qualquer humano é sempre uma pessoa.)

5. Racionalidade. Vemos no mundo seres (animais, humanos) agindo de forma que recruta meios adequados à prossecução de fins, e seres que escolhem, quando postos perante um conjunto de opções com determinada desejabilidade e probabilidade de obtenção, de forma a maximizar a utilidade esperada (o produto da desejabilidade ou utilidade e da probabilidade). Isto não seria (em princípio) possível sem a existência de representações (crenças, desejos, intenções) nesses agentes.

6. Poderia enumerar ainda outros problemas específicos, tais como emoções, raciocínio e decisão, auto-conhecimento, atribuição de interior mental a outras coisas no mundo. Esta é uma enumeração minha, informal, de alguns problemas específicos da filosofia da mente.

2. Uma resposta de referência aos problemas da filosofia da mente

Como disse, os filósofos da mente procuram perceber como é que estas coisas ao mesmo tempo 'estranhas' e fundamentais para a humanidade dos humanos (intencionalidade, consciência, ação, racionalidade, etc) são possíveis no mundo se esse mundo é basicamente físico. E desenvolvem respostas específicas. Vou agora dar a palavra àquele que é para muitas pessoas o autor central da filosofia da mente, o filósofo americano Jerry Fodor. Na obra de Fodor encontramos uma solução de referência nas últimas décadas, aquilo a que se chama a Teoria Representacional-Computacional da mente. Um *slogan* resume a resposta de Fodor à pergunta como pode o mental existir num mundo basicamente físico: ele diz '*No representations, no computations, no computations, no mind*'.

Noutras palavras, Fodor defende que o mundo físico implementa o mental pela ocorrência de símbolos em agentes cognitivos ao nível sub-pessoal (implementar ou realizar não é identificar-se com – é por isso que a teoria de Fodor é num certo sentido – não noutros – anti-reducionista). Noutras palavras ainda, há aí coisas que são representações dentro das nossas cabeças, por isso é que somos mentais, por isso é que as coisas 'estranhas' acima são possíveis. A ideia-chave é a chamada Hipótese da Linguagem do Pensamento, que é o núcleo da Teoria Representacional-Computacional da Mente. De acordo com a Teoria Representational-Computacional da Mente, Descartes tinha razão quando falava do pensamento em termos de ideias: há representações, representações reais, nas nossas mentes, só que não são ideias imateriais numa *res cogitans* imaterial mas sim símbolos, símbolos da Linguagem do Pensamento, cuja sintaxe ainda não compreendemos mas que está no nossos cérebros. Mas para haver pensamento, diz Fodor, tem que haver algo mais do que representações, tem que haver processos. Nós não nos limitamos a representar – nós movemo-nos em pensamento, i.e. inferimos, de forma que preserva a verdade. Que processos são esses? Segundo Fodor, foi Alan Turing quem teve a ideia genial, a ideia a cuja necessidade as abordagens behavioristas e associacionistas do mental e da inteligência pretendem não dar o peso devido: a ideia foi que este processo consiste em computações, i.e. em transformação de representações internas, movida apenas pela pura forma (sintaxe), sem considerações de significado. Muito do trabalho de Turing como lógico e matemático foi precisamente sobre a natureza da computação e da computabilidade.

A tarefa da filosofia da mente consistirá então, pelo menos em parte, em explicitar de um ponto de vista epistemológico, semântico e metafísico aquilo com que o estudo científico da cognição em termos de representações e computações nos compromete.

3. Um pouco de história da filosofia da mente

Se Turing é incontornável na história recente da filosofia da mente desde os anos 50 do século XX é porque foram suas algumas ideias fundamentais na definição de uma certa 'ortodoxia' de discussão na disciplina (estou a chamar ortodoxia a algo como um património a partir do qual a polémica é possível). Como vimos, devemos reconduzir ao seu trabalho, segundo Fodor, a ideia mais importante da história acerca da natureza dos processos mentais, a única ideia que nos permite pensar no mental como consistindo não apenas em representação mas em processo de representações: a ideia de computação. O seu trabalho como lógico e matemático conduziu à criação dos conceitos de Máquina de Turing e Máquina de Turing Universal e é devido a esse trabalho que Turing é considerado um dos pais do computador.

Mas Turing fez mais: ele é relevante na história da filosofia da mente, como notei no início deste artigo, também como autor do artigo *Computing Machinery and Intelligence,* publicado na revista *Mind* em 1950. É aí que é proposto o célebre Teste de Turing que está sempre na mente de todos os filósofos da mente.

A pergunta inicial do artigo é: podem as máquinas pensar? Cada passagem do artigo é importante, até mesmo no que respeita à forma de colocar questões. A primeira coisa que Turing faz é recusar um certo tipo de discussão: uma discussão concetual acerca de *o que é máquina? O que é pensar?* seria interminável. Em vez disso, propõe um teste. O Teste de Turing é hoje uma das experiências de pensamento standard da filosofia da mente e é proposto como uma forma prática de substituir discussões aprioristas sem saída acerca da definição de inteligência. O teste é um jogo de imitação. Na situação original há um interrogador e dois interatores, cuja natureza de homem ou mulher o interrogador desconhece. O objetivo dos jogadores é enganar o interrogador acerca do seu estatuto. Só pode haver interação verbal e tudo o que pode ser feito para descobrir essa natureza é formular questões. Na situação que nos interessa não há um homem e uma mulher mas um homem e uma máquina. De novo, só há interação verbal, só se pode fazer perguntas. O objetivo continua a ser enganar o interrogador. Claro que têm estatutos diferentes questões relativas a cálculo matemático, interpretação de um poema, de uma observação irónica ou de uma metáfora. 'Passar no teste de Turing', respondendo a estas questões, é ser considerado humano pelo interrogador. O desafio é se alguma vez uma máquina passa o Teste de Turing. O ponto de Turing com o Teste é pensar no pensamento de uma forma neutra e despida de preconceitos, o que interessa aos filósofos da mente: o teste assume que o que se comporta como inteligente é inteligente, a sugestão implícita (holista e pragmática) é que inteligência é comportamento inteligente e não uma especial matéria (por exemplo necessariamente biológica, ou neuronal) ou um ingrediente a mais (por exemplo uma alma).

Figura 3 Teste de Turing

Turing passa o resto do artigo a responder a objeções contra a possibilidade da Inteligência Artificial[1]. É muito revelador notar que estas são as objeções que ainda hoje surgem espontaneamente quando se discute a possibilidade de Inteligência Artificial: Turing chama-lhes a objeção teológica (a inteligência estaria ligada a uma alma, que só os humanos possuem), a objeção 'cabeças na areia' (espere-se que não venha a existir Inteligência Artificial ou seria terrível), a objeção matemática, evocando o teorema de Gödel (segundo a qual os humanos têm capacidades mentais que transcendem o que é computável), as objeções ligadas ao sentimento e à consciência (de acordo com as quais uma máquina não poderia ter estados como depressão, amor, emoção, etc), a objeção ligada às incapacidades (de acordo com a qual uma máquina nunca seria capaz de humor, aprendizagem, moralidade, paixão), o argumento 'a máquina só faz o que lhe mandamos, não origina o novo,' o argumento da continuidade do sistema nervoso, em contraste com computadores digitais, etc. Turing analisa-os um a um e continua

1 Note-se que o artigo foi escrito antes da data de nascimento oficial da Inteligência Artificial, 1956.

a propor o seu teste contra qualquer veredicto apriorista acerca do que é pensar e acerca das entidades que podem pensar.

Do ponto de vista da abordagem filosófica do mental podemos então dizer que pelo menos duas contribuições extremamente importantes vieram de Alan Turing: uma ideia acerca da natureza do pensamento como processo, e uma postura anti-apriorista quanto à definição de inteligência, um descolamento da ideia de inteligência relativamente a substratos materiais específicos, e portanto a ideia segundo a qual o mental não tem que ter necessariamente *hardware* biológico.

Voltando à história, e porque a história da filosofia da mente é também a história das muitas formas, muito diferentes entre si, de se ser materialista ou fisicalista acerca do mental, foi pelas mãos de um outro filósofo americano, Hilary Putnam, que o *Turing-machine functionalism* veio também a opor-se, nos anos 60 do século XX, ao materialismo cru da teoria da identidade mente-cérebro. A Teoria Representacional-Computacional da Mente de Jerry Fodor que vimos atrás, é, embora eu não tenha enfatizado esse aspeto, uma teoria funcionalista (aliás Fodor é um aluno e discípulo de Putnam). O funcionalismo é uma forma de nos ajudar a pensar nas razões pelas quais a mente não é idêntica ao cérebro, nas razões por que é demasiado fácil dizer que os estados mentais são simplesmente idênticos a estados cerebrais. A ideia básica é que os estados mentais não são (ao contrário do que propõe o 'materialismo simplista da Teoria da Identidade' desenvolvida por filósofos australianos como J. J. C. Smart e U. T. Place) estados cerebrais, mas sim estados funcionais realizados por estados cerebrais (mas poderiam sê-lo por outro *hardware* – a ideia de realizabilidade múltipla acompanha o funcionalismo). O que Putnam propõe é portanto que os estados mentais de seres como nós estão para os estados neurofisiológicos da mesma forma que os estados lógicos de uma máquina estão para os estados físicos dessa máquina ('*A mente está para o cérebro como o software para o hardware*' é uma conhecida súmula do funcionalismo). Com esta posição pretende dissolver o problema mente-corpo, mostrar que este tem uma natureza meramente lógico-linguística, e que se coloca em relação a nós humanos como se colocaria relativamente a qualquer sistema cognitivo capaz de se auto-monitorizar e de produzir auto-descrições, se nesse sistema existisse, como existe em nós, uma assimetria entre o acesso ao nível lógico, o nível do programa (relativamente ao qual o sistema é incorrigível – sei que vejo vermelho, que penso que estou aqui) e o acesso ao nível físico (cada um de nós tem que aprender quanto ao seu cérebro: não temos acesso direto ao nosso cérebro, não é o facto de o nosso cérebro ser causalmente responsável pela nossa mente que faz de nós automaticamente neurocientistas).

O computacionalismo de Turing e o funcionalismo de Putnam, conjuntamente com a ideia segundo a qual o nível dos estados funcionais é um nível simbólico, representacional-computacional autónomo, são precisamente os precedentes sobre os quais Jerry Fodor constrói a sua teoria da mente em termos de Linguagem do Pensamento. Já a sintetizei atrás também com o slogan *No representations, no computations, no computations, no mind*. Ele resume o chamado modelo cognitivista do mental. Este é um muito forte paradigma de investigação sobre cognição, e que incorpora muito de Turing – é claro que Turing talvez tivesse sido mais subtil e mais cheio de dúvidas, mas deixou essa herança. Quero agora terminar dizendo um pouco acerca das razões pelas quais desde há pelo menos três décadas esse paradigma é submetido a ataques dentro da própria filosofia da mente.

4. Um desafio: e se o mental não for apenas representacional-computacional?

A célebre experiência mental do Quarto Chinês pretende não apenas contestar o funcionalismo cognitivista mas argumentar que a Inteligência Artificial Forte é impossível (a Inteligência Artificial Forte é a ideia segundo a qual na Inteligência Artificial não se trata apenas de simulação de processos cognitivos mas eventualmente da coisa mesma: um sistema que implemente o programa correto será realmente mental).

A experiência mental do Quarto Chinês foi o primeiro grande ataque lançado por Searle contra o cognitivismo. O artigo *Minds, Brains and Programs*, onde o Argumento do Quarto Chinês é defendido, apareceu na revista *Behavioral and Brain Sciences* (1980), uma revista muito importante nestas áreas. A experiência mental consiste no seguinte: alguém, que não fala chinês, está fechado dentro de um quarto onde há símbolos chineses em caixas. Tem um livro de instruções em inglês, que explica como combinar os símbolos chineses e como enviar sequências de símbolos chineses para fora do quarto, quando são introduzidos no quarto, outros símbolos chineses, através de uma pequena janela. A pessoa que está dentro do quarto não sabe nada acerca disso, mas as pessoas que estão fora do quarto chamam aos símbolos que introduzem 'perguntas' e aos símbolos que saem 'respostas'. O sistema fala portanto chinês, na perspetiva das pessoas que estão fora ('passa o Teste de Turing') embora a pessoa lá dentro saiba que não percebe uma palavra de chinês. Searle afirma que a experiência mental do Quarto Chinês mostra claramente que é possível existir 'intencionalidade atribuída' sem 'intencionalidade intrínseca'.

Figura 4 O Quarto Chinês

Não é fácil saber exatamente o que é que o argumento de Searle prova. Antes de mais, é preciso ver que o Quarto Chinês não é exatamente um argumento, é mais propriamente uma parábola. Posto sob a forma de argumento seria basicamente a tentativa de obter a partir das premissas *Os programas são sintáticos / A sintaxe não é suficiente para a semântica / As mentes têm semântica* as conclusões *Implementar um programa é insuficiente para haver mente / A Inteligência Artificial Forte é uma pretensão injustificada*. O Quarto Chinês mostraria então que a mente não é um programa e que por isso nunca uma programação apropriada poderia dar mente a um sistema, já que as propriedades formais não constituiriam 'intencionalidade genuína'. Searle sublinha sempre que o seu argumento não tem nada a ver com um estádio evolutivo particular da tecnologia, mas antes diz respeito a princípios concetuais: o erro do cognitivismo é considerar que propriedades formais são suficientes para a mentalidade (esta posição é essencial à defesa da Inteligência Artificial Forte). Para Searle, pelo contrário, a essência da mente é consciência e é isso que falta aqui.

Será a consciência importante para a mente? Mais do que a inteligência? Serão consciência e inteligência duas coisas diferentes? Entre o Teste de Turing e o Quarto Chinês como ficam as relações interior/exterior quando se trata de pensar sobre o mental? Deixo as questões em suspenso – o meu propósito aqui era apenas contextualizar algumas contribuições fundamentais de Turing para o campo da filosofia da mente.

Conclusão

O que é que esta breve incursão pelas relações de Alan Turing com a filosofia da mente nos pode fazer pensar? Antes de mais que pensarmos, sermos inteligentes e conscientes, sermos mentais não é o mesmo que sabermos do ponto de vista científico e filosófico o que é a mente (da mesma forma que não é o facto de sermos feitos de materiais genéticos que faz de cada um de nós especialistas em genética). Por isso não devemos ser aprioristas (i.e. pensar que sabemos intuitivamente, antes de investigar e experimentar) acerca de mente e de inteligência; a nossa intuição sobre nós próprios como seres mentais parece ser muito íntima e muito segura mas a verdade é que ela não é um guia infalível para pensar sobre a natureza do mental. Talvez não seja mais do que o ponto de partida. O trabalho científico, no caso de Turing o trabalho lógico e matemático sobre computação, tem uma palavra decisiva no que respeita à nossa mentalidade e interioridade, e uma disciplina muito inesperada (sobretudo para aqueles que têm uma conceção mais literária e cultural da filosofia) mostrou ser fundamental nas últimas décadas para pensar sobre o mental: a Inteligência Artificial. Ou, pelo menos, Alan Turing fez-nos considerá-la assim.

Mas se Turing deu corpo aos desafios filosóficos trazidos pela Inteligência Artificial, a verdade é que os desafios filosóficos trazidos pela Inteligência Artificial, são uma versão do nosso tempo da preocupação que vêm dos primórdios da filosofia: pensar sobre o pensamento, pensar sobre as relações pensamento/mundo. E pensando sobre pensamento e mundo, sobre mente e corpo, as coisas podem ser mais complicadas do que parecem: simplesmente opor dualismo espiritualista (a ideia segundo a qual há matéria e há espírito, nós somos corpo e espírito) a um materialismo monista (a ideia segundo a qual tudo é matéria) revela ser uma simplificação demasiado crua.

Referências

Hodges, Andrew, *Alan Turing: the Enigma*, New York, Simon and Schuster, 1983.

Hodges, Andrew, *Turing*, London, Phoenix, 1997

Filosofia da Mente – algumas referências

Block, Ned, Ed., *Readings in Philosophy of Psychology - Vol. 1,* Cambridge MA, Harvard U. Press, 1980.

Block, Ned; Flanagan, Owen; Guzeldere, Guven, Ed., *The Nature of Consciousness - Philosophical Debates*, Cambridge MA, MIT Press, 1997.

Bradley, Raymond; Swartz, Norman, *Possible Worlds – An Introduction to Logic and Its Philosophy*, Indianapolis, Hackett/Blackwell, 1979.

Chalmers, David, *The Conscious Mind – In Search of a Fundamental Theory*, NY/Oxford, Oxford U. Press, 1996.

D. Hofstadter and D. Dennett, *The Mind's I – Fantasies and Reflections on Self and Soul,* New York, Bantam Books, 1981.

Fodor, Jerry, *Psychosemantics – The Problem of Meaning in the Philosophy of Mind*, Cambridge MA, MIT Press, 1987.

Guttenplan, Samuel, Ed., *A Companion to the Philosophy of Mind,* Oxford, Blackwell, 1994.

Kim, Jaegwon, *Philosophy of Mind*, Oxford, Westview, 1996.

Kim, Jaegwon, *Mind in a Physical World – An Essay on the Mind-Body Problem and Mental Causation*, Cambridge MA, MIT Press, 1998.

Lycan, William, Ed., *Mind and Cognition - A Reader*, Oxford, Blackwell, 1990.

Miguens, Sofia, *Uma Teoria Fisicalista do Conteúdo e da Consciência – D. Dennett e os debates da filosofia da mente,* Porto, Campo das Letras, 2002.

Moser, Paul K.; Trout, J. D., Ed., *Contemporary Materialism – A Reader,* London/NY, Routledge, 1995.

Pinto, João Alberto, *Superveniência, Materialismo e Experiência - Uma perspectiva sobre o problema da consciência em filosofia da mente,* Porto, Campo das Letras, 2007.

Wilson, Robert; Keil, Frank, Ed., *The MIT Encyclopedia of the Cognitive Sciences*, Cambridge MA, MIT Press, 1999.

Recursos eletrónicos

David Chalmers, Ed., David Bourget, Mind Papers – A Bibliography of the Philosophy of Mind and the Science of Consciousness [An annotated bibliography]: http://consc.net/mindpapers

Edward N. Zalta, Ed., *The Stanford Encyclopedia of Philosophy*: http://www.plato.stanford.edu

5. Razão, cálculo e computação. Três raízes da conceção computacional da razão em Leibniz[1]

Olga Pombo

1 Tradução port. do estudo "Three Roots for Leibniz's Contribution to the Computational Conception of Reason", publicado *in* F. Ferreira, B. Löwe, E. Mayordomo e L. M. Gomes (eds.) (2010), *Programs, Proofs, Processes*. *6th Conference on Computability in Europe*, Berlim: Springer, pp. 352--361, posteriormente incluído in Pombo, Olga (2010), *Palavra e Esplendor do Mundo*, Lisboa: Fim de Século, pp. 69-86).

O objetivo deste estudo é mostrar de que modo o contributo de Leibniz para aquilo que hoje se designa por conceção computacional da razão deve ser inscrito em três raízes fundamentais: 1. a tradição combinatória do século XIII, 2. os projetos seiscentistas para a construção de uma língua artificial universal e filosófica, e 3. a conceção hobbesiana de razão como cálculo.

1. Lull e Leibniz

A primeira grande inspiração é a velha *Ars Magna*, de Ramón Lull (1232-1315). A ideia central de Lull é que, pela combinação de um conjunto de termos simples, seria possível estabelecer todas as proposições possíveis, descobrir todos os enunciados válidos e demonstrar todas as verdades a que o conhecimento humano pode aspirar.

Para a realização deste projeto, Lull desenvolve um programa complexo que envolve: *a*) o estabelecimento de um conjunto de categorias organizadas em seis grandes classes de nove subcategorias cada, *b*) a definição de um conjunto de regras sintáticas dos elementos combinatórios, *c*) a conceção de um sistema de procedimentos técnicos da aplicação automática, capazes de efetuar a combinação destes elementos. O objetivo é determinar automaticamente todos os sujeitos de um atributo dado, assim como a conclusão e o termo médio de todos os silogismos incompletamente conhecidos[2]. Com base em categorias de aplicação universal e operando com um sistema de notações simbólicas e de diagramas combinatórios, Lull estabelece, pois, os princípios de um procedimento sintético e inventivo que, ao contrário da lógica demonstrativa de Aristóteles, pretende não se limitar à análise das verdades já conhecidas, mas apontar para a descoberta de novas verdades. A *Ars Magna*, de Lull exerceu uma influência profunda no Renascimento e nos tempos modernos[3] e constitui uma das mais remotas e prestigiadas propostas de mecanização das operações lógicas.

2 Entre o complexo sistema de sinais, tabelas e figuras construído por Lull há um conjunto de círculos materiais, girando em movimento de sobreposição concêntrica, que visa permitir a combinação dos símbolos marcados nos seus limites. Cf. Carreras y Artau (1939: 345) e também Couturat, *Sur L'Ars Magna de Raymond Lulle et Athanase Kircher* (1981: 541-543).

3 A saber, o lullismo enciclopedista de Johann Heinrich Alsted (1588-1638) ou de Sebastián Izquierdo (1601-1681) e, sobretudo, o trabalho de Athanasius Kircher, *Porta Scientiarum do seu do sciendi do Magna de Ars*, publicado em 1669, que se afirma como o desenvolvimento da *Ars Magna* de Lull.

Leibniz conhecia bem a *Ars* de Lull. De facto, desde os seus primeiros escritos, nomeadamente, desde o *De Arte Combinatória* (1660), Leibniz faz o balanço crítico da *Ars* de Lull, manifestando a sua discordância face *a*) à arbitrariedade do seu sistema de signos, *b*) ao conjunto das suas categorias e *c*) aos seus aspetos metodológicos. São críticas decisivas. Leibniz não pode aceitar o alfabeto da *Ars* Lulliana. Não pode admitir a lista das categorias que Lull define e que Leibniz considera "vagas", arbitrariamente escolhidas e organizadas em classes de 9 categorias cada, um número escolhido artificialmente, para razões de pura simetria (cf. GP 4: 63). Além disso, não aceita os procedimentos combinatórios de natureza mecânica, propostos por Lull, que considera insuficientes e rudimentares. Em suas vezes, Leibniz aponta para processos calculatórios de análise matemática (agrupamentos binários, ternários, quaternários, trocas, substituições, equivalências, etc.). E, com base na analogia entre a análise dos conceitos, a sua estrutura associativa e a decomposição de um número inteiro em fatores primos, propõe, pela primeira vez, no *De Arte*, atribuir um número inteiro a cada ideia primitiva, ficando assim cada ideia complexa a ser representada pelo produto que constituiria a sua definição.

No entanto, malgrado estas críticas, Leibniz reconhece o valor epistemológico do projeto de Lull, em dois pontos importantes: *a*) o estatuto da combinatória como fundamento da *ars inveniendi*; *b*) a possibilidade de aplicar os mesmos procedimentos lógicos à totalidade do conhecimento, incluindo o conhecimento novo. Quer dizer, Leibniz aceita a ideia de Lull segundo a qual o conhecimento das ideias primitivas e das leis de sua combinação poderia permitir encontrar todos os predicados possíveis de um sujeito dado e vice-versa, isto é, poderia permitir, não somente formular todas as verdades já conhecidas, mas também inventar ou descobrir novas verdades. Leibniz dará um passo em frente ao tomar a matemática como modelo e ao conceber o projeto de aritmetizar todos os mecanismos do pensamento, de submeter a totalidade da atividade intelectual humana a processos calculatórios capazes de reduzir a atividade racional a um cálculo.

A maior diferença de Leibniz face a Lull diz respeito ao tipo de sistema simbólico capaz de operar no interior do cálculo. Antecipando a sua experiência metodológica fundamental – a descoberta do algoritmo do cálculo diferencial, enquanto verdadeiro *organon* da investigação matemática[4] –, Leibniz está inteiramente consciente do valor não meramente representativo, mas prospetivo e heurístico do signo. Mas, para tal, Leibniz reivindica que os signos não sejam, como em Lull, escolhidos arbitrariamente. Quer isto dizer que o projeto de Leibniz não pode ser reduzido à constituição de uma mera linguagem formal, ou seja, a um sistema linguístico universal que permitiria o tratamento lógico da ciência e forneceria um conjunto de símbolos simples, necessários e rigorosos, para expressar todo o conhecimento real e possível. Leibniz não cai na ilusão – de alguma maneira presente no projeto de Lull –, de acordo com a qual, o funcionamento automático de um sistema simbólico e de um conjunto de regras operativas pode permitir o desenvolvimento da ciência. Para Leibniz, a ciência não pode ser reduzida a uma linguagem bem-feita. A heurística, em Leibniz, passa pela exigência de uma semântica.

2. Projetos de uma língua filosófica no século XVII

Sabemos que o século XVII foi um momento de intenso e profundo questionamento da natureza íntima da linguagem humana. As razões são muitas e diversas e não poderão ser aqui analisadas[5]. Permitam-me apenas indicar que a questão principal diz respeito ao papel que a linguagem desempenha no processo do conhecimento: pode a linguagem promover o conhecimento? Ou, inversamente, impede o seu progresso? É a linguagem fator perturbador ou elemento necessário à aquisição do conhecimento? São as línguas humanas meros meios para comunicar o conhecimento já adquirido ou um *medium* essencial para constituir novo conhecimento?

Duas posições principais podem ser identificadas nos tempos modernos: uma posição crítica, que atribui às línguas naturais funções meramente comunicativas e enfatiza as suas insuficiências e efeitos perturbadores, e uma posição positiva que, embora reconhecendo limites e imperfeições às línguas humanas, não obstante, reconhece o seu papel decisivo enquanto dispositivos cognitivos necessários à construção do conhecimento.

4 Como defende Cassirer (1923-1929.1: 76), Leibniz teria aplicado a sua experiência metodológica com o algoritmo do cálculo diferencial na construção da língua universal.

5 Para mais desenvolvimentos, cf. Pombo (1987 e 1997).

Ambas as posições foram defendidas por grandes pensadores. Na posição crítica, Bacon, Locke, Descartes, Arnauld, Malabranche e, em geral, todos aqueles que defenderam a construção de línguas novas e artificiais – dos pasígrafos ingleses aos construtores de línguas filosóficas (de Lodwick a Dalgano, de Seth Ward a Wilkins). Na posição positiva, somente dois nomes nos tempos modernos: Thomas Hobbes e Leibniz.

O que aqui gostaria de mostrar é que: *a*) estas duas posições foram ambas relevantes para o desenvolvimento de uma conceção computacional da razão; e que *b*) Leibniz foi o único que compreendeu inteiramente a possibilidade de fazer convergir aquelas duas posições opostas numa conceção linguística da razão inteiramente desenvolvida.

A posição crítica decorre da consciência aguda da inadequação das línguas naturais para representar, quer o universo que a ciência moderna está a construir, quer as categorias lógicas que estão na base desse progresso científico. Daí a necessidade de construção de uma língua artificial e filosófica que substituísse com vantagem as línguas naturais e as suas imperfeições. A grande inspiração é Francis Bacon (1561-1626), o primeiro a desenhar o projeto de construção de uma língua universal e filosófica que representaria, "não letras ou palavras, mas coisas e noções" (Bacon, 1605: 6.1: 439). A ideia era construir uma língua na base de um conjunto de caracteres (reais) que representassem, não os sons ou as vozes das línguas naturais, mas as coisas ou as noções. Língua que, assim, poderia ser lida e compreendida por todos os falantes das diferentes línguas. Como Bacon escreve: "Todo o livro escrito em caracteres deste tipo pode ser lido por cada nação na sua própria língua" (Bacon, 1605: 6.1: 439)[6].

Bacon aponta também para a construção de uma gramática filosófica, enquanto conjunto de regras comuns a todas as línguas naturais. Nas suas palavras: "Se alguém, bom conhecedor de um grande número de línguas, eruditas bem assim como vulgares, reunisse as várias propriedades dessas línguas, mostrando em que pontos cada uma é excelente e em que outros falha. Assim, não somente muitas línguas seriam enriquecidas por trocas mútuas, mas diversas belezas de cada uma poderiam ser combinadas (…) numa muito bela imagem e excelente modelo

6 "*Any book written in characters of this kind can be read of by each nation in their own language*" (Bacon, 1605: 6.1: 439).

de língua própria para a correta expressão dos significados da mente" (Bacon, 1605: 6.1: 421-422)[7].

As sugestões programáticas de Francis Bacon, em grande parte apresentadas em *The Advancement of Learning* (1605), estão na raiz dos numerosos projetos de construção de uma língua artificial universal que emergiram no século XVII. O objetivo de todos esses projetos, que ganharam grande popularidade no século XVII, é a construção de um instrumento comunicativo universal. Face às necessidades criadas pelo desenvolvimento dos contactos internacionais (económicos, políticos, científicos e religiosos), o objetivo é a construção de uma língua única, perfeitamente regular, capaz de substituir o latim, de superar as imperfeições e a diversidade de línguas naturais e de permitir uma adequada e plena comunicação entre homens. A ideia perseguida é a da pasigrafia, palavra definida, pelo *Larousse* do século XIX, como "a arte de escrever ou imprimir na única língua que uma pessoa sabe, de forma a poder ser lido e compreendido, sem necessidade da tradução, em qualquer outra língua que uma pessoa não sabe". O objetivo é a comunicação eficaz entre povos das línguas diferentes, por intermédio de um sistema de notações escritas, independentes da pronunciação, por meio do qual seria possível estabelecer correspondências diretas entre palavras diferentes de línguas diferentes. Qualquer indivíduo, independentemente da sua língua, seria capaz de ler e compreender qualquer texto assim codificado.

Dos muitos trabalhos empreendidos no século XVII para a construção de uma língua universal, são de referir os da escola inglesa de pasígrafos do barroco, Cave Beck, *The Universal Character, by which all the Nations in the World may Understand one Another's Conceptions* (1657), Henry Edmundson, *The Natural Languages in a Vocabulary Contrived and built upon Analogy* (1658), Edward Somerset, *A Century of the Names and Scantlings of such Inventions as at Present I can call to Mind to have Tried and Perfected* (1663). Além destes, devem ser mencionados os trabalhos dos alemães Johann J. Becher, *Character pro Notitia Linguarum Universali* (1661) e Athanasius Kircher, *Polygraphia Nova et Universalis ex Combinatoria Arte Delecta* (1663). Partindo da atribuição de um nú-

7 "*If some one well seen in a great number of tongues, learned as well as vulgar, would handle the various properties of languages, showing in what points each excelled, in what it failed. For so not only many languages be enriched by mutual exchanges, but the several beauties of each may be combined* (…) *into a most beautiful image and excellent model of speech itself, for the right expressing of the meanings of the mind*." (Bacon, 1605, 6.1: 421-422)

mero a cada palavra do dicionário latino e da organização de dicionários de diversas línguas, nos quais cada palavra devia ser seguida pelo mesmo número atribuído ao seu equivalente latino, Becher pretendia tornar possível a tradução, em qualquer língua, de um texto escrito, unicamente, com sinais numéricos. O leitor necessitaria, unicamente, de ter acesso ao dicionário-chave organizado para a sua própria língua. Mais elaborado, o projeto de Kircher consiste na construção de dois dicionários poliglotas (latim, italiano, francês e alemão), cada um organizado com vista ao seu futuro papel comunicativo, enquanto fonte e enquanto alvo[8]. Em ambos os casos, estamos perante sistemas escritos codificados, totalmente artificiais e arbitrários, do tipo que hoje designaríamos por línguas especiais ou técnicas (sistemas semióticos gráficos, limitados a domínios específicos, como códigos de sinais telegráficos ou marítimos). No entanto, os seus autores, diretamente inspirados pelas propostas programáticas de Bacon, procuravam construir línguas artificiais, capazes de evitar a ambiguidade e a irregularidade das línguas naturais e de, assim, permitir a divulgação do conhecimento e o desenvolvimento do comércio e da verdadeira religião. Como Cave Beck escreve, no prefácio a *The Universal Character* (1657: VIII), uma tal língua seria "um meio singular de propagar no mundo todos os tipos de conhecimento e a verdadeira religião"[9].

Os projetos filosóficos são mais ambiciosos. Para lá de objetivos meramente comunicacionais, o seu alvo é construir um sistema simbólico, capaz de expressar adequadamente o pensamento e as suas articulações e traduzir todo o conhecimento real e possível, isto é, de cumprir um papel essencialmente cognitivo. A hipótese de partida é que é possível reduzir todas nossas ideias e noções a um pequeno número de conceitos básicos, simples ou primitivos. A língua filosófica consistiria, então, em primeiro lugar, na atribuição de um carácter, não a cada uma das ideias possíveis, como Bacon havia defendido (procedimento que os construtores de línguas universais haviam adotado), mas a cada uma

8 No dicionário projetado para emitir a mensagem, as palavras são ordenadas alfabeticamente e a cada uma corresponde o número da sua página (em algarismos árabes). No dicionário projetado para a receção da mensagem, os sinónimos das cinco línguas são indicados em colunas, de acordo com a ordem alfabética das palavras latinas. Trata-se de um projeto que supõe um esboço da classificação dos conceitos, isto é, um projeto filosófico. Para a posição de Leibniz, no que diz respeito a Becher e a Kircher, cf. GP 4: 72, C: 536 e a sua *Carta a Oldenburg*, Julho 1670 (GP 7: 5).

9 "(...) *a singular means of propagating all sorts of learning and true religion in the world*" (Beck, 1657: VIII).

das ideias elementares. Em segundo lugar, a língua filosófica implicaria a definição de uma gramática ou conjunto de réguas fixas da combinação desses caracteres.

A inspiração é agora cartesiana. O objetivo é encontrar o pequeno número das ideias simples a que todas as outras podem ser reduzidas e, pela combinação de um número igualmente limitado de caracteres, cada qual correspondente a uma daquelas ideias simples, construir o sistema inteiro do conhecimento. Razão pela qual, antes da escolha dos sinais característicos, todos os projetos filosóficos começam por fazer uma elaborada classificação lógica e semântica dos conceitos.

Numa célebre *Carta a Mersenne*, de 20 Novembro 1629 (AT 1: 76-82), Descartes havia, de facto, apresentado uma tese muito relevante sobre a possibilidade de uma língua universal e filosófica. O argumento é o seguinte: se a totalidade da aritmética pode ser construída a partir de um pequeno número de axiomas, de maneira similar, deveria ser possível simbolizar exaustivamente a totalidade dos pensamentos humanos, com base num número limitado de sinais linguísticos. Como Descartes escreve:

> "Penso que se poderia acrescentar uma invenção, tanto para compor as palavras primitivas desta língua, como para os seus caracteres, de tal forma que essa língua poderia ser ensinada em muito pouco tempo, e isso por meio da ordem, quer dizer, estabelecendo uma ordem entre todos os pensamentos que podem entrar no espírito humano, tal como há uma, naturalmente, para os números; e, assim como é possível aprender num dia a dizer os nomes de todos números até ao infinito e a escrevê-los numa língua desconhecida, o que implica uma infinidade de palavras diferentes, o mesmo poderia ser feito com todas as outras palavras necessárias para exprimir todas as outras coisas que podem estar no espírito dos homens." (AT 1: 80-81)[10]

10 "*Je trouve qu'on pourroit ajouter à ceci une invention, tant pour composer les mots primitifs de cette langue que pour leurs caractères en sorte qu'elle pourroit être enseignée en fort peu de temps, et ce par le moyen de l'ordre, c'est-à-dire établissant un ordre entre toutes les pensées qui peuvent entrer en l'esprit humain de même qu'il y en a un naturellement établi entre les nombres; et comme on peut apprendre an un jour à nommer toutes les nombres jusques à l'infini et à les écrire en une langue inconnue, qui sont toutes fois une infinité de mots différents, qu'on pût faire le même de toutes les autres mots nécessaires pour exprimer toutes les autres choses qui tombent en l'esprit des hommes*." (AT 1: 80-81)

O paradigma é a matemática, como sistema simbólico da aplicação universal e, simultaneamente, ordem do mundo e estrutura da criação. Quer dizer, a matemática é, não somente um sistema dedutivo, fundado na evidência de seus axiomas ou num método rigoroso de demonstração, mas uma língua universal que pode representar o mundo porque é isomórfica com a sua estrutura, porque o mundo está, em si mesmo, matematicamente ordenado e estruturado. Neste sentido, é importante recordar que a matemática, que está na base da ciência moderna, está garantida pela conceção pitagórica da estrutura matemática da criação (vejam-se os exemplos de Galileu, Descartes, Kepler, Leibniz ou Newton). Assim, o ideal de uma língua filosófica universal pode ser pensado como a extensão e o aprofundamento dos procedimentos matemáticos capazes de operar com base em categorias universais adequadas à exposição e desenvolvimento do sistema integral do conhecimento, do mundo e das coisas.

Das muitas línguas filosóficas propostas no século XVII, a primeira parece ter sido a de Mersenne (1588-1648) que, em *Carta a Fabri de Peiresc*, de 1636 ou 1637[11], declara ter terminado a construção de uma língua filosófica, certamente de inspiração cartesiana[12]. Também em Inglaterra, é necessário mencionar nomes como o do matemático e astrónomo Seth Ward (1617-1698), *Vindiciae Academiarum* (1654). Em termos próximos dos usados por Descartes, tanto Mersenne como Seth Ward apontam para a decomposição dos conceitos em termos primitivos. Como, por exemplo, escreve Seth Ward: "porque todos os discursos se resolvem em frases e estas em palavras, e porque as palavras significam ou noções simples ou outras que podem ser resolvidas em noções simples, é manifesto que, se todos os tipos de noções simples fossem identificados e se tivessem um símbolo que lhes fosse atribuído, estes seriam muito poucos em relação aos outros." Ward (1954: 21)[13]

É ainda em Inglaterra que Georges Dalgarno e John Wilkins realizam as línguas filosóficas mais relevantes e mais completas. Dalgarno publica

11 Reproduzido parcialmente em Adam Tannery I, *Correspondance*, pp. 572-573.

12 Do qual não resta nenhum fragmento, exceto alguns proposições no *Traité de l'Harmonie Universelle* (1636).

13 "(...) *for all discourses being resolved in sentences, those into words, words signifying either simple notions or being resolvable into simple notions, it is manifest that if all the sorts of simple notions be found out, and have symbols assigned to them, those will be extremely few in respect of the other*", Ward (1954: 21).

Ars Signorum, vulgo Character Universalis et Lingua Philosophica, em 1661, e Wilkins, bispo de Chester e um dos *scholars* mais reputados deste período, publicará, em 1668, com o apoio da Royal Society, o seu célebre *An Essay towards a Real Character and a Philosophical Language, with an Alphabetical Dictionary*. Na base de uma organização hierárquica dos conceitos em classes e subclasses, Dalgarno aponta para um sistema lógico e simbólico de letras capazes de expressar a articulação normal do pensamento[14]. Similarmente, Wilkins parte de uma classificação dos conceitos que organiza num complexo sistema de s*umma genera, diferentia e species*. Estas categorias, com as quais Wilkins pretende cobrir a totalidade do conhecimento científico, são representados por um *character realis*, isto é, não por letras, como em Dalgarno, mas por caracteres escritos, de natureza ideográfica[15].

Leibniz conhecia bem todos estes projetos filosóficos, tendo elogiado, sobretudo, o projeto de Wilkins. Como escreve, em *Carta a Burnet*, de 24 de Augusto de 1697: "Considerei, com atenção, a grande obra do Carácter real e Linguagem Filosófica do Sr. Wilkins; penso que ele colocou aí uma infinidade de coisas belas e que nunca tivemos uma Tábua de predicamentos mais conseguida." (GP 3: 216)[16]

No entanto, Leibniz não evita a crítica. A maior oposição diz respeito às insuficiências analíticas que, a seu ver, subjazem às propostas de Dalgarno e Wilkins. Como Leibniz escreve, na mesma carta: "Considerei essa matéria antes do livro do Sr. Wilkins, quando era um jovem de 19 anos, no meu pequeno *De Arte Combinatoria*, e a minha opinião é que os caracteres verdadeiramente filosóficos devem responder à Análise dos pensamentos." (GP 3: 216)[17] Esta é, de facto, a tarefa mais difícil, mas também a mais fundamental. Somente a análise dos diversos conteúdos concetuais e das suas relações pode permitir constituir o

14 Para uma descrição detalhada da língua filosófica de Dalgarno, cf. Couturat (1901: 544-548), Cram (1980) e, também, Rossi (1960: 218 e segs.).

15 Para um estudo detalhado do livro de Wilkins, cf. Couturat, *Sur la Langue Philosophique de Wilkins* (1901: 548-552).

16 *J'ai consideré avec attention le grand ouvrage du Caractere reel et Langage Philosophique de Mons. Wilkins; je trouve qu'il y a mis une infinité de belles choses, et nous n'avons jamais eu une Table des predicateurs plus accomplie*", *Carta a Burnet*, de 24 de Agosto de 1697 (GP 7: 216).

17 *J'avois consideré cette matière avant le livre de Mr. Wilkins, quand j'estoit un jeune homme de 19 ans, dans mon petit livre de Arte Combinatoria, et mon opinion est que les Caracteres veritablement reels et philosophiques doivent repondre à l'Analyse des pensées*." (GP 3: 216)

conjunto dos termos primitivos, com base nos quais a língua filosófica deve ser construída. Só essa análise pode suportar a escolha do sistema não-arbitrário dos signos da nova língua filosófica. Como Leibniz diz: "o nome que, nesta língua, deve ser atribuído ao ouro será a chave do conhecimento que dele poderia ser obtido." (GP 7: 13)[18]

Em síntese, pode pois dizer-se que Leibniz conhecia muito bem todas as propostas dos seus predecessores e contemporâneos, em matéria de língua universal internacional e filosófica, que Leibniz compreendia perfeitamente o alcance científico e o carácter lógico dos seus projetos e que, além disso, Leibniz foi quem mais profundamente articulou os objetivos lógicos, semânticos e heurísticos desse empreendimento. Contra Bacon, defendeu a necessidade de começar a construção da língua universal pela atribuição de um carácter, não a cada uma de ideias possíveis, como Bacon tinha defendido, mas a cada uma das ideias elementares a que todas as outras poderiam ser reduzidas. Contra Descartes – que finalmente acabou por condenar o projeto porque, a seu ver, só a análise completa (enciclopédica) dos diversos conteúdos concetuais e das suas relações podia permitir encontrar os termos primitivos na base dos quais a língua filosófica poderia ser construída[19] –, Leibniz vai defender que essas duas tarefas – *Characteristica Universalis* e *Encyclopaedia* – podiam ser realizadas num processo de ziguezague, não como empreendimentos lineares e sequenciais, mas como esforços recíprocos e mutuamente condicionadores[20]. Cada vez que a análise avançasse e pudesse determinar uma nova unidade, esta unidade somente se tornaria clara e distinta quando fosse designada por um signo específico. Por outro lado, da atribuição desse signo decorreriam novas virtualidades heurísticas, em consequência da sua inserção na rede estrutural de todos os signos previamente constituídos.

---

18 "*Nomen tamen quod in hac lingua imponetur, clavis erit eorum omnium quae de auro humanitus, id est ratione at que ordine scriri possunt, cum ex eo etiam illud appariturum sit, quaenam experimenta de eo cum ratione institui debeant.*" (GP 7: 13)

19 Como Descartes diz em *Carta a Mersenne*, de 20 de Novembro de 1629, "*L'invention de cette langue dépend de la vrai philosophie; car il est impossible autrement de dénombrer toutes les pensées des hommes, et de les mettre par ordre, ni seulement de les distinguer en sorte qu'elles soient claires et simples.*" (AT 1: 82)

20 De facto, na mole dos manuscritos de Leibniz, há uma cópia da carta de Descartes a Mersenne, em que, por sua própria mão, Leibniz escreveu: "*Il est vray que ces caractères pressupposeroient la veritable philosophie* (…) *Cependant, quoique cette langue depende de la vraye philosophie, elle ne depend pas de sa perfection. C'est à dire, cette langue peut estre* établie, *quoyque la philosophie ne soit pas parfaite: et à mesure que la science des hommes croistra, cette langue croistra aussi.*" (C: 28)

Por fim, Leibniz estava perfeitamente consciente de que, para construir um *organon* de investigação racional bem fundado, deveriam ser desenvolvidas uma nova conceção de simbolismo e uma nova conceção de razão. A primeira, que não posso apresentar aqui[21], foi, em grande parte, construída contra Descartes, no contexto das suas divergentes posições face ao sucesso das matemáticas e do estatuto gnosiológico do seu simbolismo. A segunda provém, principalmente, de Hobbes.

3. Hobbes e Leibniz

No *Leviathan*, Hobbes declara: "raciocinar é calcular" (L: 99)[22]. E acrescenta: "Quando um homem Raciocina, não fez nada mais que conceber uma soma total a partir da Adição das parcelas; ou conceber um resto, a partir de uma Subtração de uma soma a partir de outra" (L: 110)[23]. Mais adiante, explica: "aquelas operações não incidem apenas sobre números, mas sobre todos os tipos de coisas que podem ser adicionadas juntas e tiradas umas das outras" (L: 110)[24], isto é, a aritmética, a geometria, a lógica, e também a política, a lei, "em suma, em tudo onde há matéria para a adição e subtração, também há lugar para a Razão, e onde estas não têm lugar, lá a Razão não tem nada a fazer." (L: 110-111)[25]

Naturalmente, este objetivo de estender o mesmo tipo de procedimentos racionais à totalidade dos domínios encontra em Leibniz eco imediato. No entanto, se é indiscutível que Hobbes foi o predecessor de uma teoria linguística e computacional da cognição, a verdade é que não há em Hobbes nenhuma solução metodológica efetiva para a realização dessa tarefa racional. E, dada a pobre preparação matemática de Hobbes[26], é bem provável que a sua definição da razão não

21 Para uma apresentação deste debate, cf. Pombo (1990).

22 "*Reasoning is nothing but reckoning.*" (L: 99)

23 "*When a man Reasoneth, hee does nothing else but conceive a summe totall, from Addition of parcels; or conceive a Remainder, from Substraction of one summe from another.*" (L: 110)

24 "*Those operations are not incident to Numbers only, but to all manner of things that can be added together and taken one of another.*" (L: 110)

25 "*In summe, in what matter so ever there is place for addition and substraction, there also is place for Reason, and where these have no place, there Reason has nothing at all to do.*" (L: 110-101)

26 Para uma análise das polémicas entre Hobbes e alguns grandes matemáticos de seu tempo, como Wallis, Boyle e Huygens, cf. Breidert (1979).

seja muito mais do que um *dictum* vazio, isto é, que Hobbes não tenha penetrado as ricas e profundas consequências da sua própria afirmação. Como Couturat defende (1901: 461), Hobbes não teria penetrado todo o significado da sua própria fórmula. Pelo contrário, mercê da sua profunda capacidade matemática e do seu enorme interesse pelos processos combinatórios, Leibniz estava completamente preparado para desenvolver essa conceção computacional da razão cuja fórmula vai, incontestavelmente, buscar a Hobbes.

Mas há ainda um outro aspeto em que Hobbes é uma importante raiz para a conceção computacional da razão em Leibniz. Refiro-me à conceção hobbesiana de razão como atividade linguística, como faculdade que opera exclusivamente com base nos nomes[27]. Segundo Hobbes, somente a língua permite "transferir o nosso discurso mental em discurso verbal" (L: 100)[28], isto é, somente a linguagem estabiliza o fluido do discurso mental, estabelecendo pontos de orientação em torno dos quais as representações se tornam fixas e isoladas. Como Hobbes escreve: "é pelos próprios nomes que nós podemos estabilizar as múltiplas conceções de uma representação" (HNV, § 4)[29]. Quer dizer, somente a linguagem fornece os elementos simbólicos sobre os quais se pode realizar a atividade do cálculo: "Pois a Razão, neste sentido, nada mais é que contar (quer dizer, Adicionar e Subtrair) as consequências dos *nomes* gerais que foram acordados, para marcar e significar os nossos pensamentos" (L: 111, nossa ênfase)[30]. Quer dizer: sem linguagem não haveria razão. O que significa que para Hobbes as ideias não são imagens, mas palavras.

Estamos aqui face a uma contribuição muito importante de Hobbes para a filosofia da linguagem de Leibniz e para a conceção computacional da razão em geral. Se a linguagem é a *translatio* verbal de um *Discourso Mental previo*[31] –, esse fluido imagético e instável dos pen-

27 Como Hobbes escreve: "*Children therefore are not endued with Reason at all, till they have attained the use of Speech*." (L: 116)

28 "(…) transfer *our mental discourse into verbal discourse*." (L: 100, sublinhados nossos)

29 "*It is by the very names that we are able to stabilize the multiple conceptions of one representation*." (H. N. V, § 4)

30 "*For Reason, in this sense, is nothing but Reckoning (that is Adding and Substacting) of the Consequences of generall* names *agreed upon, for the marking and signifying of our thoughts*." (L: 111, *our emphasis*)

31 "*The general use of Speech is to transferre our Mentall Discourse, into verbal*." (L: 100)

samentos[32], no qual apenas a linguagem pode estabelecer pontos de referência, áreas de segurança, marcas, notas, nomes –, então, a linguagem é o suporte sensível do pensamento, aquilo que fornece as condições materiais e significantes requeridas para o desenvolvimento do cálculo. Somente a linguagem permite a substituição do cálculo das ideias pelo cálculo dos nomes. Somente a linguagem preserva a possibilidade de recordar etapas precedentes, de revisitar cadeias dedutivas, de progredir, gradual e firmemente, de uma consequência para outra. É por isso que Hobbes pode escrever: "O uso e Fim da razão, não é encontrar a *summe*, e verdade de uma, ou algumas consequências distantes das primeiras definições e das significações estabelecidas dos nomes, mas começar nestes e prosseguir de uma consequência para outra." (L: 112)[33]

Com esta tese, Hobbes está a dar uma contribuição significativa que Leibniz adotará inteiramente. E Leibniz reconhece a sua herança face a Hobbes, precisamente neste ponto. Como diz, os "nomes não são somente sinais dos meus pensamentos presentes para os outros, mas notas dos meus pensamentos passados para mim próprio, como Thomas Hobbes demonstrou."[34]

Desta importante tese, irá Leibniz extrair consequências epistémicas e heurísticas de que Hobbes nunca suspeitou. Mas, para isso, Leibniz terá que superar os limites internos da filosofia da linguagem de Hobbes. Porque Hobbes está preso a uma conceção evocativa do signo, isto é, porque reivindica que os nomes devem ser sempre acompanhados pelas conceções que significam, Hobbes recusará veementemente a hipótese de manipulação dos signos. Sem o preenchimento evocativo dos signos pelas ideias que lhes correspondem, a *ratio*, diz Hobbes, seria reduzida à *oratio* (cf. HN V, § 14), ao discurso mecânico, repetitivo, como os "pedintes, quando dizem o seu *paternoster*, pondo em conjunto

32 "*By consequence, or Trayne of Thoughts I understand that sucession of one thought to another which is called* (…) *Mentall Discourse*." (L: 94)

33 "*The use and End of reason, is not the finding of the summe, and truth of one, or a few consequences, remote from the first definitions, and settled significations of names; but to begin at these; and proceed from one consequence to another*." (L: 112). Para mais desenvolvimentos sobre esta posição de Hobbes, cf. Pombo (1985).

34 "*Verba do enim os alios do anúncio dos praensentis dos meae dos cogitationis do sunt do signa do tantum non, sed et* o anúncio dos *praeteritae dos meae dos cogitationes dos notae mim ipsum, demonstravit Thomas Hobbes do ut*." (A. VI, 1: 278)

muitas palavras, tal como aprenderam na sua instrução com as suas educadoras, os seus companheiros ou os seus professores, sem terem as imagens ou conceções nas suas mentes que respondem às palavras ditas" (HN V, § 14)[35]. Ora, pelo contrário, querer constantemente pensar o significado dos signos que se manipulam, é não somente uma impossibilidade d*e facto*, como *de jure* constitui uma obstrução ao rigor e à invenção, a qual supõe o abandono do espírito aos mecanismos formais por ele criados.

Este é o ponto crucial da célebre teoria leibniziana da *cogitatio caecae*[36]. Mas essa será uma das maiores descobertas da filosofia da linguagem de Leibniz.

4. Observações conclusivas

A contribuição de Leibniz para a conceção computacional da razão tem, assim, três importantes raízes que Leibniz foi, simultaneamente, capaz de integrar e superar.

1. Leibniz aceita o projeto combinatório de Lull, embora substituindo os procedimentos mecânicos, propostos por Lull, por processos calculatórios de análise matemática (neste ponto, antecipando Boole).

2. Na linha de Bacon, Leibniz conserva os objetivos comunicativos dos projetos de línguas universais, embora tentando dar-lhes uma fundamentação cognitiva, de forma a fazer deles verdadeiras línguas filosóficas. Nesse sentido, Leibniz secunda o regime metodológico, proposto por Descartes, para a construção de uma língua filosófica, embora recuse a relação linear que Descartes havia estabelecido entre a língua filosófica e a enciclopédia (contra, portanto, a separação que Frege, mais tarde, estabelecerá entre aquelas duas tarefas[37]).

35 "(…) *with beggars, when they say their paternoster putting together much words, and in such manner, as in their education they have learned from their nurses, from their companions, or from their teachers, having no images or conceptions in their minds answering to the words they speak*." (HN V, § 14)

36 Que estudámos em Pombo (1998).

37 Como vimos já, à parte as similaridades entre o projeto leibniziano de uma *Characteristica Universalis* e o projeto ideográfico de Frege, os dois projetos são distintos, justamente, pelo facto de Frege rejeitar a articulação estabelecida por Leibniz entre a *characterística* e o *encyclopaedia*, deixando às ciências o trabalho de definir seus próprios conceitos.

3. Leibniz desenvolve a conceção computacional da razão, formulada por Thomas Hobbes, embora com base numa nova teoria do signo que torna possível explorar todas as suas potencialidades cognitivas e heurísticas.

Referências

BACON, Francis (1605), *The Advancement of Learning* in *The Works of Francis Bacon. Faksimile--Neudruck der Ausgabe von Spedding. Ellis und Heath*. Londres: Stuttgart-Bad, Cannstatt: Frommann, 1963. Vol. III, 260-491.

BECHER, Johann (1661), *Character, pro notítia linguarum universali, inventum steganographicum hactenus inauditam quo quilibet suam legendo vernaculam diversas imo omnes linguas, unius etiam diet informatione, explicare ac intelligere potest*. Francofurti: sumtibus Johanis Wilh. Amonii et Wilhelmi Serlini, typis Johannis Georgii Spörlin.

BECK, Cave (1657), *The Universal Character, by which all the Nations in the World may understand one anothers Conceptions, Reading out on one Common Writing their own Mother Tongues*. Londres: T. Maxey, for W. Weekly.

CASSIRER, Ernst (1923/29), *Philosophie der symbolischen Formen. Erster teil. Die Spache* (Berlim, 1923), trad. franc. de Ole Hansen-Love, Jean Lacoste. Paris: Minuit, 1972.

BREIDERT, Wolfgang (1979), "Les Mathématiques et la Méthode Mathématique chez Hobbes" *in Revue Internationale de Philosophie*, 129: 415-431.

CARRERAS Y ARTAU, Tomás e CARRERAS Y ARTAU, Joaquín (1939), *Historia de la filosofía española. Filosofía cristiana de los siglos XII a XV*, tomo I. Madrid: Real Academia de Ciencias Exactas, Físicas y Naturales.

COUTURAT, Louis (1901), *La Logique de Leibniz d'après des Documents Inédits*. Paris: Alcan.

CRAM, David (1980), "George Dalgarno on 'Ars Signorum and Wilkins' Essay" *in* KOERNER, Ernest (ed.), *Progress in Linguistic Historiography*. Amesterdão: John Benjamins Publishing Company, 113-121.

DALGARNO, George (1661), *Ars Signorum, vulgo Character Universalis et Lingua Philophica*. In *The Works of George Dalgarno of Aberdeen*. Edimburgo: Constable, 1834, 1-82.

DESCARTES, René

—AT – *Oeuvres de Descartes. Ed. Charles Adam Paul Tannery*. Paris: Vrin, 1964-1976.

HOBBES, Thomas

— L – *Leviathan*, in Molesworth, *The English Works of Thomas Hobbes*, III, Aalen, Scientia Verlag (reed.), 1966 (para a edição inglesa, usámos a edição de C. B. Macpherson. London:

Penguin Classics), para a versão latina (LT), utilizámos a tradução francesa de François Tricaud, Paris, Sirey, 1971).

— HN – *Human Nature. Or the Fundamental Elements of Policy* in *Thomas Hobbes*, *Opera Philosophica quae Latina scripsit. Ed. by Sir William Moleswort*. Vol. 1. Londres, 1939.

LEIBNIZ

A – *Gottfried Wilhelm Leibniz Sämtliche Schriften und Briefe*. Akademie der Wissenschaften zu Berlin. Reihe I-VI. Darmstadt: Reichl (1923 e segs.).

C – *Opuscules et fragments inédits de Leibniz. Extraits des manuscrits de la Bibliothèque royale de Hannover par Louis Couturat*. Paris: Alcan, 1903.

GP – *Die philosophischen Schriften von Gottfried Wilhelm Leibniz. Hrsg v. Carl Immanuel Gerhardt*. 1-7. Hildesheim: Olms, 1960.

LlULL, Ramón (1308), *Ars Brevis* (tradução, introdução e notas de Armand Llinarès), Paris: La éditions du Cerf. (1991)

MERSENNE, Marin (1636), *Harmonie universelle contenant la théorie et la pratique de la musique, 1636. Ed. facs. de l'exemplaire conservé à la Bibliothèque des Arts et Métiers et annoté par l'auteur. Introduction par François Lesure*. Vols. 1-3. Paris: Editions du Centre National de la Recherche Scientifique, 1963.

Pombo, Olga (1985), Linguagem e verdade em Hobbes, *Filosofia*. 1: 45-61.

— (1987), *Leibniz and the Problem of a Universal Language*, Münster: Nodus Publikationen.

— (1990), *The Leibnizian theory of representativity of the sign*, in Hans-Joseph Niederehe and Konrad Koerner (eds), *History and Historiography of linguistics*, II, Amsterdam/Philadelphia: John Benjamins Publishing Company, 447-459.

— (1997), *Leibniz e o Problema de uma Língua Universal*, Lisboa: JNICT

— (1998), *La Théorie Leibnizienne de la Pensée Aveugle en tant que Perspective sur quelques--unes des Apories Linguistiques de la Modernité*, Cahiers Ferdinand Saussure, nº 51, pp. 63-75.

ROSSI, Paolo (1960), *Clavis Universalis, Arti Mnemoniche e Logica Combinatoria da Lullo a Leibniz*. Milão/Napoli: Riccardo Ricciardi.

SOMERSET, Edward Worcester (1663), *A Century of the Names and Scantlings of such Inventions, as at Present I can call to Mind to Have Tried and Perfected*. London: Printed by J. Grismond and reprinted by J. Adlard (1813).

WARD, Seth (1654), *Vindicae academiarum. Containing Some Briefe Animadversions Upon Mr. Websters Book. Stiled. The Examination of Academies. Together with an Appendix Concerning what Mr. Hobbes, and Mr. Dell have Published on the Argument*. Oxford: Leonard Lichfield for Thomas Robinson.

6. As vidas paralelas de Alan Turing e Edmundo Curvelo e a matematização da mente

José António Alves

Introdução

A mente é matematizável? Poderá a mente, um dos objetos do mundo que mais parece escapar aos poderes explicativos da ciência, ser traduzida e representada pelo rigor da linguagem lógico-matemática? Alan Turing e Edmundo Curvelo acreditaram que sim. Aliás, não só acreditaram como o trabalho intelectual que desenvolveram deixou pistas sobre o modo de matematizar a mente. Turing através da proposta de uma máquina de computação. Curvelo através da proposta da logificação da psicologia.

A mente parece estar nos antípodas do tipo de objeto analisável pela lógica e pela matemática. A lógica e a matemática procuram regularidades que possam ser formalizadas e a mente parece escapar a qualquer regularidade. Durante séculos e ainda hoje é fácil descobrir quem conceba a mente como o campo do singular, o reino do subjetivo, querendo com isto significar a irrepetibilidade e singularidade da mente. Nomeadamente quando o conceito de mente se dilui no conceito de consciência. Pois pode-se conceber que a linguagem lógico-matemática será capaz de formalizar a percepção, a memória, o raciocínio, a imaginação, mas não que será capaz de formalizar matematicamente a consciência. A este respeito afirmou o matemático Roger Penrose: «A consciência parece-me um fenómeno tão importante que pura e simplesmente não consigo acreditar que seja uma coisa 'acidentalmente' fabricada por uma computação complexa».[1] e um pouco mais à frente «… é na verdade 'óbvio' que a mente *consciente* não pode funcionar como um computador, ainda que grande parte dos processos envolvidos na atividade metal o possam fazer».[2] A mente consciente, assim entendida, sugere a conotação religiosa que a mente consciente adquire ao se aproximar do conceito de alma enquanto princípio individuador do ser humano. Contudo, este modo de pensar tem hoje bastantes opositores dentro da comunidade científica. Isto, apesar de se continuar sem saber como responder às seguintes perguntas difíceis: o que é a mente?; será a mente causalmente eficiente?; qual o papel da mente no mundo natural?; qual é a relação da mente com o cérebro? A comunidade científica não oferece ainda respostas satisfatórias a estas perguntas, mas desconfia que a estratégia para as obter aproxima-se mais do modelo de trabalho científico desenvolvido em física do que deixar-se enredar em discursos metafísicos. Turing e Curvelo também subscreveram este modo de pensar. Ambos conceberam que a via para solucionar o problema *mente* seria ensaiando estratégias de investigação alheias à substancialização tradicional. Ambos apostaram

1 Roger Penrose, *The Emperor's new mind. Concerning computers, minds, and the laws os phisics*. (Oxford: Oxford University Press, 1990 [1989]), p. 447.

2 *Ibidem*, p. 448.

que as perguntas difíceis se dissolveriam mediante a construção de uma nova abordagem do problema. Ambos tinham a convicção de que a ciência não só descobre velhos mundos, mas também constrói novos mundos onde os anteriores deixam de fazer sentido.

É certo que o leitor e eu poderemos ter mentes conscientes que pensam ou sentem coisas diferentes, mas a forma desse pensar e sentir tem uma *estrutura*, no mínimo, semelhante e, nesse sentido, conseguimos comunicar um com o outro. Os nossos corpos são estruturalmente semelhantes não só em termos físico-químicos e biológicos, mas também em termos psicológicos. Por exemplo, usamos o mesmo idioma, respeitando a mesma gramática; somos capazes de concordar em relação à percepção de determinados objetos; não teremos dificuldade em perceber as emoções e sentimentos um do outro por aproximação ao que nós próprios sentimos; e somos capazes de fazer previsões bastante razoáveis acerca do comportamento de cada um de nós e dos outros. Todas estas situações ocorrem porque as mentes humanas têm estruturas semelhantes. Acreditamos, pelos sintomas manifestados pela mente consciente, que as nossas mentes não serão assim tão diferentes umas da outras ou, quando muito, serão tão diferentes quanto são diferentes o fígado ou coração do leitor do meu fígado e coração. Tanto assim é que, por incrível que possa parecer, o cientista consegue descobrir ou construir regularidades na mente humana do mesmo modo que consegue descobrir ou construir regularidades no fígado ou no coração. Seja através da investigação das correlações entre o cérebro e a mente, seja através da investigação dos fenómenos mentais, seja através da reformulação dos conceitos usados na elaboração das teorias… A mente humana é estudada como se estuda qualquer outro objeto físico. E, nesta tarefa de investigação, a lógica e a matemática são instrumentos descritivos poderosos. Turing e Curvelo também não duvidaram desta afirmação. Porém, ambos foram ainda mais ambiciosos na afirmação do poder da lógica e da matemática. O pensamento dos dois autores revela a intenção de traduzir e representar integralmente a mente humana com o auxílio da lógica e da matemática. Turing, como acima se disse, através da construção de modelos, algoritmos, lógico-matemáticos aplicados a máquinas artificiais capazes de simular o funcionamento da mente humana. O cientista inglês acreditou ser possível materializar numa máquina todas as funções da mente humana. A tarefa consistiria em criar um algoritmo lógico-matemático capaz de replicar, num artefacto, a máquina que é o ser humano. Curvelo, como também acima se disse, através da logificação da psicologia, ou seja, através da construção de uma estrutura lógico-matemática capaz de descrever toda a vida psicológica: estados mentais e comportamentos.

A descodificação lógico-matemática da mente humana, incluindo a parte mais difícil, a consciência, é um programa intelectual e científico em curso. Na época em que viveram Alan Turing e Edmundo Curvelo o assunto não estava ainda tão claramente delineado. Só alguns anos após a morte de ambos a temática seria um dos núcleos importantes das hoje amplamente divulgadas ciências cognitivas. Esse núcleo conquistou o direito de disciplina com designação própria: Inteligência Artificial (IA). Em pouco mais de meio século de atividade, a Inteligência Artificial foi capaz de oferecer à comunidade científica resultados extraordinários para a compreensão da mente humana. Por exemplo, recentemente o *Blue Brain Project* foi capaz de apresentar as bases lógico-matemáticas do funcionamento dos neurónios do cérebro humano e de simular esse funcionamento em computador.

É convicção generalizada de que existe uma relação entre o funcionamento cerebral e a mente. Provavelmente a descodificação do cérebro humano será um passo necessário para a descodificação da mente humana. Ainda não dispomos da fórmula lógico-matemática da consciência, mas a agenda está definida. O passo seguinte do *Blue Brain Project* é o *Human Brain Project*. Este projeto propõe, até 2030, descodificar integralmente a mente humana de modo a produzir uma máquina indiscernível de um ser humano.[3]

A matematização da mente é indubitavelmente o tema que une as vidas intelectuais de Alan Turing e Edmundo Curvelo. E, por isso, a matematização da mente será o assunto da segunda parte do desenvolvimento do presente texto e a porta de entrada para percorrermos alguns passos da obra inspiradora do matemáticos inglês e do lógico português. Contudo, antes do percurso pela obra será interessante revisitar a vida de cada um dos autores e salientar os traços biográficos que fazem das suas vidas a razão do nosso título, vidas paralelas.

3 Sobre o *Human Brain Project* pode consultar-se o sítio http://www.humanbrainproject.eu/ .

Se a lógica e a matemática são capazes de descobrir e construir regularidades na mente humana, então será a vida que cada um de nós vive assim tão singular como habitualmente a julgamos? E se a vida nos parecer singular apenas porque na infinidade de conjugações físicas, biológicas e psicológicas ainda não nos foi possível encontrar vidas iguais à nossa? Será que a descoberta e construção de regularidades se poderá estender à totalidade de uma vida física, biológica e psicológica? Os seres humanos têm doenças físicas e biológicas que se assemelham de tal modo que são tratadas com procedimentos semelhantes. Os seres humanos têm doenças psicológicas que se assemelham de tal modo que são tratadas com procedimentos semelhantes. Os seres humanos têm e desenvolvem comportamentos semelhantes de tal modo que é possível categorizá-los e tipificá-los… Se descobrimos semelhanças no que se refere às partes, por que razão há de ser impossível descobrir semelhanças em relação ao conjunto? A existência de vidas que se assemelham ou repetem.

Se ninguém se banha duas vezes na mesma água do rio, poderemos suspeitar que muito menos as vidas humanas se igualam. Nem será necessário recorrer à mente humana consciente para assegurar a identidade dos indivíduos humanos. A *Biometria* oferece muitos elementos para distinguir os seres humanos. No entanto, as coincidências acontecem. De modo que, por exemplo, no século I encontramos exercícios de curiosidade semelhante ao que aqui desenvolvemos. Plutarco escreveu *As vidas paralelas*, colocando a par histórias de nobres gregos e romanos com moralidades semelhantes. As vidas das personalidades biografadas pelo historiador grego não se assemelhavam necessariamente nos factos vividos por cada uma delas. Péricles foi um chefe político democrático da majestosa Antenas e Fábio Máximo foi eleito ditador e cinco vezes cônsul num tempo em que Roma fraquejava. Todavia, apesar da eventual disparidade de vidas, Plutarco explora os traços comuns aos dois estadistas.[4] Ora, no presente texto propomos um exercício semelhante, colocando a par, não dois estadistas, mas dois cientistas. Curiosamente as vidas de Alan Turing e Edmundo Curvelo oferecem a possibilidade de um exercício semelhante ao desenvolvido por Plutarco. Alan Turing viveu em Inglaterra. Edmundo Curvelo viveu em

4 Para mais cf., por exemplo, Plutarco, *Vidas paralelas. Péricles e Fábio Máximo*. (Trad., Intr. e Notas de Ana Maria Guedes Ferreira e Ália Rosa Conceição Rodrigues, 3.ª edição, Coimbra: Imprensa da Universidade de Coimbra, 2013).

Portugal. Duas vidas separadas por um mar de água, cultura, contextos sociopolíticos que nada faria imaginar o paralelismo que entre elas se poderá descobrir. Contudo, duas vidas próximas, não só nos interesses intelectuais, mas também em alguns traços biográficos.

Sintetizando. Porquê Alan Turing e Edmundo Curvelo? Que relação existe entre os dois autores que motiva e justifica juntá-los? A resposta a estas perguntas é o primeiro andamento da exposição. Segue-se a apresentação de algumas das ideias principais de cada um dos autores para a descodificação lógico-matemática da mente humana. Procuramos deste modo salientar o génio internacionalmente reconhecido de Alan Turing e com ele iluminar o génio menos conhecido de Edmundo Curvelo.

1. Dois autores (des)iguais

Devemos reconhecer que a pretensão de identificar um paralelo entre Alan Turing e Edmundo Curvelo poderá causar estranheza. Primeiro, porque Alan Turing é um autor internacionalmente conhecido. Segundo, porque Edmundo Curvelo é um autor muito pouco conhecido, quer internacionalmente, quer mesmo a nível nacional. Por estas duas razões, compreendemos que poderá causar estranheza a associação de duas figuras aparentemente tão desiguais. Contudo, esperamos que a estranheza rapidamente se converta em curiosidade. Como veremos, apesar da desigualdade do reconhecimento pelo público leitor, existem entre os dois autores várias proximidades biográficas e intelectuais. A proximidade na eleição do assunto intelectual a que dedicaram a vida e o trabalho científico ficou explícita na introdução. Portanto, aqui, serão exploradas sobretudo as proximidades biográficas. Sem esquecer, obviamente, que o interesse pela proximidade biográfica surge da proximidade entre os dois na eleição do assunto intelectual.

Antes de avançar na descrição das proximidades biográficas entre o matemático inglês e o lógico português, convém clarificar que não pretendemos comparar como quem iguala, mas comparar como quem soma. Ou seja, aumentar a cada elemento comparado o seu domínio próprio. A intenção é a de nos deixarmos espantar com as coincidências da vida e aumentarmos o nosso conhecimento e interesse por duas figuras da História da Ciência do Século XX, que se dedicaram à resolução de um dos maiores problemas de sempre da ciência, a mente humana. Talvez, como na vidas comparadas por Plutarco, seja possível indagar qual a direção correta, aqui não das decisões políticas, mas da agenda intelectual para o estudo da mente humana.

Por que razão não queremos nem devemos comparar os dois autores? Porque decerto cairíamos em injustiças de apreciação. Quando no início dos anos noventa frequentámos a Universidade de Aveiro costumávamos ouvir que Aveiro era a Veneza portuguesa. Anos mais tarde tivemos a oportunidade de visitar Veneza e verificar, por nós próprios, o enorme exagero daquele modo de caracterizar Aveiro. Quando os aveirenses pretendem salientar a beleza da cidade pela comparação a Veneza só a podem diminuir. A razão da comparação não só diminui Aveiro como destaca, na mesma proporção, o quão mais bela é Veneza. A cidade dos ovos moles tem efetivamente um canal de água que rodeia um ou outro edifício da cidade, mas nada que se possa comparar aos inúmeros canais que desenham Veneza e fazem com que o barco seja o melhor meio de transporte na cidade. Percebe-se que os aveirenses façam a comparação tentando aproximar a beleza de Aveiro à beleza internacionalmente reconhecida de Veneza. Porém, ao fazê-lo, como dissemos, diminuem a beleza de Aveiro porque, na verdade, em quase nada ou mesmo muito pouco as duas cidades se podem comparar. Portanto, à comparação seria preferível a contemplação dos traços semelhantes. Se Aveiro é uma cidade bela e que merece a visita dos turistas, é pelas belezas que lhe são próprias e não pelas belezas que lhe pudessem vir da comparação. No entanto, também é verdade que, apesar das desigualdades entre as duas cidades, há qualquer coisa na menos conhecida, Aveiro, que quando a visitamos nos faz recordar a mais conhecida, Veneza. Esse *qualquer coisa* é a razão do que deve motivar a inteligência a confrontar e deixar-se espantar com o paralelo entre as duas cidades em vez de as comparar com um sinal de igualdade. Ora, por analogia, é o que sucede entre Alan Turing e Edmundo Curvelo. Pretendermos comparar Alan Turing e Edmundo Curvelo seria incorrer em erro semelhante ao que ocorre quando se compara Aveiro a Veneza. Portanto, procurar na obra de Edmundo Curvelo o Alan Turing português seria tarefa despicienda e comparação injusta para o autor português, pelo tamanho desnível existente entre as duas figuras. Alan Turing é mundialmente conhecido e estudado, enquanto Edmundo Curvelo nem em Portugal é largamente conhecido e estudado.

Não deixa de ser um mistério por que razão alguns autores são estudados e difundidos e outros não o são tanto quanto o poderiam merecer. Poder-se-á pensar que há aqui uma razão do foro linguístico. Turing escreveu em inglês e Curvelo em português. E a comunidade de leitores em inglês é maior do que a comunidades de leitores em português. Contudo não é uma razão que resolva o mistério. Primeiro, porque Edmundo Curvelo, mesmo escrevendo quase sempre em português, não deixou de ser lido por autores internacionais. Por exemplo, Alan Church escreveu e publicou no *Journal of Symbolic Logic* uma resenha ao livro, de Curvelo, *Introdução à Lógica*. No espólio do autor português há sinais da correspondência trocada com vários autores internacionais, entre eles, Willard Quine, Stephen Kiss, Réné Poirier. Segundo, porque Edmundo Curvelo, mesmo em português, não foi especialmente lido e comentado entre os leitores portugueses. Ainda hoje, o número de estudos na academia portuguesa sobre a obra curveliana é reduzido. Terceiro, porque, por seu turno, o génio de Alan Turing também não foi imediatamente reconhecido, mesmo escrevendo ele num idioma acessível a um número maior de elementos da comunidade científica. Na verdade, só depois da morte do matemático inglês a sua obra foi realmente conhecida e difundida. Por fim, a principal linguagem usada pelos dois autores foi a linguagem lógico-matemática, uma linguagem que ultrapassa qualquer dificuldade de tradução da linguagem natural.

Poder-se-á pensar, então, que a resolução do mistério tem que ver com o conteúdo das obras. Esta razão é mais plausível, mas não elimina todas as perplexidades. De novo, como salientámos, apenas depois da morte do autor inglês a obra de Turing foi verdadeiramente difundida e estudada. Por outro lado, repare-se que o conteúdo da obra só pode ser razão de difusão depois de estudado e criticado. Ora, porque é que tardou tanto a surgir um estudo aprofundado do conteúdo da obra de Curvelo? Não havendo ainda solução para todas as interrogações sobre a mente humana e sendo a mente humana um dos assuntos principais de Edmundo Curvelo, ainda maior é a perplexidade para a demora do estudo da sua obra. Quem sabe na obra quase desconhecida e pouco estudada do lógico português há resposta para alguns dos enigmas da mente humana? Recorrendo de novo à analogia da relação entre as duas cidades, poderemos afirmar que não é pela maior difusão turística de Veneza que Aveiro deixa de merecer a visita e a apreciação. Assim não o impeça o preconceito, a moda e os agentes turísticos.

Aproximemo-nos, portanto, das vidas paralelas de Alan Turing e Edmundo Curvelo. Ambos estiveram próximos nos dois momentos definidores da vida, o nascimento e a morte. Os dois autores nasceram com um ano de diferença, Alan Turing em 23 de junho de 1912 e Edmundo Curvelo em 18 de outubro de 1913. Os dois morreram no mesmo ano, em 1954. O lógico inglês a 7 de junho, o lógico português a 13 de janeiro. Tanto a morte do matemático inglês como a do lógico português estão imersas em algum mistério. Nos dois casos a opinião dominante é a de que se tratou de suicídio. No entanto, esta opinião não é corroborada integralmente pelas famílias. Para a mãe de Turing, a morte do filho foi um acidente fruto dos hábitos desleixados em que ele vivia.[5] Do lado de Curvelo, os testemunhos que nos transmitiram os sobrinhos sugerem ter-se tratado de um acidente ou um crime. No entanto, a tese de a morte do lógico português ter resultado de um crime não encontra justificação, nem nos autos da autópsia, nem nos autos da polícia, nem nos testemunhos de pessoas coevas. Já a tese de se ter tratado de um acidente, há quem também a partilhe para além da família. O escritor Jorge de Sena escreveu no seu diário, no dia 17 de janeiro de 1954, o seguinte:

> «Estive no café. Havia o Prof. Silveira, o Bandeira, com o qual combinei a operação do Pedro, o Casais, que acabou de mostrar-me as dúvidas do seu Hemingway, e o Chicó, que conversou comigo sobre o Ministério do Ultramar. Contei-lhe algumas histórias.
>
> Falou-se também da morte do Curvelo, que foi encontrado asfixiado pelo gás na sua casa de banho. Terá sido desastre. Teria chegado tarde a casa, um pouco tonto, e preparava-se para tomar banho. Não haveria motivos para outra hipótese».[6]

Os relatos dos familiares, quer em relação a um, quer em relação a outro, não descobrem qualquer comportamento que justificasse o suicídio. Para além da polémica e da dificuldade em aceitar um ou outro relato, certo é que a causa da morte de Alan Turing foi ter comido uma maçã que fora mergulhada em cianeto e a causa da morte de Edmundo

5 Cf. Sara Turing, *Alan M. Turing. Centenary Edtion*. (Cambridge: Cambridge University Press, 2012), pp. 114-121.

6 Jorge de Sena, *Diários*. (Porto: Caixotim, 2004), p. 78.

Curvelo foi intoxicação por monóxido de carbono. Os dois na casa dos quarenta anos e com uma obra prometedora.

A formação académica de cada um deles também revela semelhanças. É verdade que Alan Turing licenciou-se em Matemática e Edmundo Curvelo em Ciências Históricas e Filosóficas. Contudo, apesar das áreas de graduação terem sido tão dispares, a área de doutoramento e investigação situam-se no mesmo ramo, a lógica e a matemática. Alan Turing apresentou na Universidade de Princeton, em maio de 1938,[7] a tese *Systems of logic based on ordinals*. O grau foi-lhe concedido em 21 de junho de 1938.[8] Edmundo Curvelo apresentou na Universidade de Lisboa, em 20 de março de 1947, a tese *Multiplicidades lógicas discretas*. O grau foi-lhe conferido a 10 de dezembro de 1948. Os dois obtiveram o doutoramento em lógica.

No domínio das relações intelectuais, ambos partilharam um amigo comum. Turing e Curvelo relacionaram-se com o lógico americano Alonzo Church. A relação entre o lógico americano e o matemático inglês foi maior, porque os dois dedicaram-se à resolução de um problema matemático comum e porque no seguimento dessa circunstância Alonzo Church foi o orientador de doutoramento de Alan Turing, na Universidade de Princeton. Esta universidade americana, naquela época, com o início da II Guerra Mundial destronara Gotingan, na Alemanha, da qualidade de paraíso dos matemáticos. Por seu lado, Edmundo Curvelo relacionou-se com Alonzo Church à distância, através da troca de correspondência e partilha de textos publicados.

O lógico americano foi o diretor do *Journal of Symbolic Logic* durante vários anos e, também por causa dessa situação, muitos lógicos do mundo inteiro se relacionaram com ele. Alonzo Church desenvolveu um trabalho enorme em prol da promoção da lógica através da revista que dirigiu. Nesse esforço de promoção e divulgação do que se fazia em lógica no mundo inteiro encarregou-se ele próprio de escrever diversas recensões a trabalhos lógicos. Entre as muitas recensões que escreveu,

7 Andrew Hodges em *Alan Turing. The Enigma*. (Princeton: Princeton University Press, 2012 [1983]), p. 145, diz que a tese foi eventualmente submetida a 17 de maio; B. Jack Copeland em *The essencial Turing. The ideas that gave birth to the computer age*. (Oxford: Oxford University Press, 2010 [2004]), p. 125, diz que a tese foi aceite no dia 7 de maio, este autor cita uma carta de Turing enviada à mãe com data de 7 de maio e a consulta do processo da tese na Universidade de Princeton.

8 Cf. Andrew Hodges, *op. cit.*, p. 145.

dois dos autores que o lógico americano resenhou foram Alan Turing e, como referido acima, Edmundo Curvelo.

Além de Alonzo Church, há uma outra personalidade que marcou a vidas dos dois lógicos. Referimo-nos a Ludwig Wittgenstein. Alan Turing frequentou em Cambridge, na primavera de 1939, um curso sobre as bases filosóficas da matemática ministrado pelo filósofo austríaco.[9] Durante o curso, várias vezes, professor e aluno divergiram nas suas posições intelectuais. Por seu turno, Edmundo Curvelo não foi aluno de Wittgenstein, mas foi um leitor minucioso do *Tractatus*. Na biblioteca pessoal do lógico português dificilmente o investigador descobrirá um livro sublinhado. Porém, a obra filosófica do autor austríaco está minuciosamente sublinhada a lápis. A obra de Edmundo Curvelo foi claramente influenciada pela leitura daquele livro na definição da fronteira entre aquilo que a ciência pode afirmar e aquilo perante o qual a ciência só pode permanecer em silêncio.

As carreiras profissionais de Alan Turing e Edmundo Curvelo foram desenvolvidas num meio universitário e académico. Mas também os dois foram chamados pelos seus Governos a prestar alguns serviços à pátria. Alan Turing, durante a guerra, para colaborar na decifração de mensagem e códigos do exército alemão. Edmundo Curvelo para colaborar com o Instituto de Orientação Profissional e com a Comissão Especial para a Literatura Infantil e Juvenil, encarregada de emitir pareceres e recomendações sobre os livros e cinema apropriados a menores. Obviamente que não se pode colocar ao mesmo nível a colaboração de Curvelo com a importância e relevância do trabalho de Turing em Bletchley Park.[10] Apenas sublinhamos a chamada das entidades governamentais que cada um teve, porque significa, de certo modo, o reconhecimento institucional do trabalho que desenvolviam.

Por fim, também ambos terão partilhado na parte final das suas vidas dificuldades emocionais e sociais. Alan Turing era homossexual, comportamento socialmente inaceitável na Inglaterra da primeira metade do século XX e inclusive sancionado por lei. O matemático inglês foi uma vítima dessa lei que o expôs negativamente. Não terá sido uma

9 Sobre essas aulas veja-se por exemplo: David Leavitt, *El hombre que sabía demasiado. Alan Turing.* (Trad. Federico Corriente Basús, Barcelona: Antoni Bosch, 2006). pp. 138 e ss.

10 Cf. Andrew Hodges, *op. cit.*, pp. 260 e ss.

situação emocional fácil de integrar. O país que tanto ajudara durante a II Guerra Mundial, roubava-lhe a dignidade que lhe era devida.

As razões de Edmundo Curvelo foram de outra ordem. No início dos anos cinquenta o autor separou-se da mulher com quem casara em 1940. Essa separação não o terá favorecido emocionalmente. É curioso notar que a época mais produtiva do autor português corresponde aos anos em que partilhou a vida com a esposa. Além de dificuldades emocionais advindas da separação, juntava-se o ambiente pouco favorável na Faculdade de Letras, onde foi professor a partir de 1947, e o desalento que sempre manifestou em ver os seus trabalhos pouco lidos em Portugal.[11]

Apesar das proximidades biográficas e intelectuais os dois autores nunca se encontraram nem há indícios de que se tenham lido um ao outro. Porém, é muito provável que Edmundo Curvelo conhecesse o trabalho de Alan Turing. Se Curvelo não conheceu o trabalho de Turing diretamente, pelo menos, conheceu-o indiretamente. O lógico português faz menção, no artigo *Opucula Psychologica I*, ao livro *Cybernetics* de Nobert Wiener.[12] Neste livro, o autor americano considera o trabalho de Alan Turing pioneiro no estudo das possibilidades lógicas da máquina como experimento intelectual.[13] Além disso, Curvelo acompanhava a publicação da revista *Mind*, onde Turing publicou, em 1950, um dos seus artigos mais inspiradores e de maior projeção na reflexão sobre a mente humana.

11 A este respeito *vide*, por exemplo, a correspondência entre Edmundo Curvelo e Delfim Santos editada em Manuel Curado e José António Alves, *Um génio português: Edmundo Curvelo (1913-1954)*. (Coimbra: Imprensa da Universidade de Coimbra, 2013) pp. 331-344.

12 *Vide* Edmundo Curvelo, «Opuscula Psychologica I. Da teoria e da prática da psicotécnica». *Boletim do Instituto de Orientação Profissional*, 3.ª série, 1 (1950), pp. 104-105.

13 Norbet Wiener, *Cybernetics: or control and communication in the animal and the machine*, (2 ed., Cambridge: MIT Press, 1965) p. 13.

2. A matematização da mente

2.1. Da máquina como ser humano ao ser humano como máquina

Alan Turing escreveu cerca de 30 artigos ao longo da sua atividade académica e intelectual. A obra completa do matemático inglês foi editada em 4 volumes sob o título geral *The Collected Works of A. M. Turing*. Das três dezenas de artigos há dois que marcaram indubitavelmente a ciência do século XX. O primeiro intitulado «On Computable Numbers, with an Application to the Entscheidungsproblem» foi publicado, em 1936, na *Proceedings of the London Mathematical Society*; o segundo, com o título «Computing Machinary and Intelligence», foi publicado na revista *Mind* em 1950.

Qualquer problema difícil é ainda mais difícil quando não dispomos da linguagem adequada para investigar o problema de maneira a oferecer-lhe uma solução. Os problemas difíceis associados ao estudo da mente humana padecem da mesma dificuldade. Basta perguntar o que é a mente para nos apercebermos imediatamente da existência de definições divergentes e incompatibilizadas.[14] É difícil a ciência conseguir avançar sem um trabalho de investigação conceptual que permita operacionalizar o estudo dos problemas que a ciência pretende resolver. Os dois artigos maiores de Alan Turing ofereceram à comunidade científica dois conceitos extremamente relevantes para a abordagem do problema da mente humana.

O artigo de 1936 tinha por objetivo a resolução de um problema matemático. No entanto, mais do que a resolução do problema e da relevância do artigo para a matemática, «On Computable Numbers, with an Application to the Entscheidungsproblem» haveria de imortalizar-se pelo conceito de *Máquina de Turing* que dele seria extraído. O segundo conceito que colocaria definitivamente o matemático inglês na galeria dos pais da nova ciência cognitiva foi o de *Teste de Turing*. Este conceito foi estabelecido por Alan Turing no artigo de 1950. «[Os dois conceitos]

14 Sobre o problema conceptual no estudo da mente *vide*, por exemplo, Maxwell Bennett e Peter Hacker, *Philosophical foundations of neuroscience*. (Oxford: Wiley-Blackwell, 2003); e o debate publicado em Maxwell Bennett, et al., *Neuroscience & Philosophy. Brain, Mind & Language*. (New York: Columbia Universiy Press, 2007).

podem parecer como se tratassem de dois conceitos diferentes, mas não são. A Máquina de Turing é realmente uma tentativa para descrever de modo muito mecânico o que é que um ser humano faz levando a cabo um algoritmo matemático; o Teste de Turing é uma avaliação humana do funcionamento de um computador».[15] Os dois conceitos continuam a ser ainda hoje tópicos importantes de debate no âmbito das ciências cognitivas.

A fixação dos conceitos não se deveu propriamente a Turing. O cientista inglês definiu e descreveu os conceitos, mas a fixação da designação dos conceitos deveu-se aos leitores dos seus textos. Assim, *Máquina de Turing* é um conceito que o leitor não encontrará explicitamente no artigo do matemático inglês. No texto «On Computable Numbers, with an Application to the Entscheidungsproblem» o conceito utilizado é *computing machin*e (máquina de computação). *Máquina de Turing* deve-se a Alonzo Church. É na recensão que o lógico americano publicou em 1937 no *Journal of Symbolic Logic*, sobre o texto de Turing que o engenho criado pelo matemático inglês para computar números recebeu o nome de Máquina de Turing.

Também o conceito de *Teste de Turing* não foi a designação que o matemático inglês atribuiu à estratégia para aferir se uma máquina pensaria ou não. No texto «Computing Machinary and Intelligence» o autor refere-se a um *jogo de imitação* (*imitation game*). O jogo é jogado entre três elementos que não se veem uns aos outros e comunicam através de mensagens escritas. No jogo, o autor coloca em confronto uma máquina inteligente e dois seres humanos. O objetivo é indagar se pode uma máquina pensar, escapando à dificuldade de definição consensual e preconceito do que significa *máquina* e *pensar*. Nesse jogo um interrogador procura saber quem é a máquina e quem é o ser humano. Se a máquina vencer o jogo significa que a máquina pensa. *Teste de Turing* é, portanto, a designação que a comunidade científica acabou por fixar para designar o jogo de adivinhar quem é a máquina e quem é o ser humano? O objetivo entretanto passou a ser não só inquirir se a máquina pensa, mas se a máquina tem mente.

15 Charles Petzold, *The annotated Turing. A guided tour through Alan Turing's historic paper on computability and the Turing Machine* (Indianapolis: Wiley Publishing, 2008), p. X.

A ideia de um teste capaz de aferir a mentalidade de uma máquina tornou-se numa ideia de tal modo poderosa que aqueles que procuram argumentar contra a possibilidade da matematização da mente e a possibilidade de replicar a mente artificialmente sentiram necessidade de também conceberem um contra-teste. Nesse sentido, o filósofo americano John Searle contrapôs ao Teste de Turing a experiência de pensamento do Quarto Chinês.[16] Alguém dentro de uma sala sem conhecer nada de chinês comunica com um falante natural de chinês através de folhas escritas e passadas por baixo da porta. A pessoa que nada sabe de chinês comunica servindo-se de regras e modos de proceder escritos em inglês, mas sem compreender nada do que rabisca no papel em chinês e que o falante que está fora da sala compreende. Este conta-teste pretende salientar que existe uma diferença significativa entre o domínio da sintaxe e o domínio da semântica. No entanto, no argumento parece sobretudo existir a tentativa de salvaguardar um *algo mais* difícil de sustentar. John Searle diz haver um *algo mais* para além do comportamento pergunta-resposta. Porém, não é este o modo natural de relação entre os seres humanos? Também nós não temos acesso direto à mente do outro. As interpretações que fazemos baseiam-se em comportamentos e na conversação entre pergunta-resposta. Quer dizer, Turing responderia à objeção dizendo que o comportamento pergunta-resposta é já a definição do que se pretende explicar. O artigo do lógico inglês de 1950 dedica-se a responder a este tipo de objeções.

Mas voltemos à relação de Alan Turing com Alonzo Church. A aproximação de Alan Turing a Alonzo Church não ocorreu propriamente por causa da Máquina de Turing. O que motivou o encontro entre os dois foi o *Entscheidungsproblem,* o «problema da decisão». Este problema, habitualmente conhecido pelo seu nome em alemão, fora enunciado em 1928 por David Hilbert e Wilhelm Ackermann como o principal problema da lógica matemática e questionava se seria possível encontrar um processo geral, diríamos nós um «algoritmo», capaz de determinar a solução de qualquer teorema matemático. Tanto Alonzo Church como Alan Turing provaram que o problema é insolúvel.

Alan Turing deparou-se com o *Entscheidungsproblem* durante a frequência do curso sobre Fundamentos da Matemáticas lecionado pelo

16 *Vide* John Searle, «Minds, brains, and programs». *Behavioral and Brain Science*, 3 (1980), pp. 417-458.

professor Maxwell Newman na primavera de 1935. Em Abril de 1936, Alan Turing entregou ao professor um rascunho do artigo sobre números computáveis aplicados ao *Entscheidungsproblem*. Porém, pela mesma altura Maxwell Newman recebera uma separata do artigo «A Note on the *Entscheidungsproblem*» que Alonzo Church publicara, em Março de 1936, no *The Journal of Symbolic Logic* sobre o mesmo assunto. Segundo as regras da época a publicação de um artigo com resultados semelhantes a um outro escrito anteriormente era motivo para a não publicação do artigo posterior. Exatamente a situação que se verificava entre o texto de Alonzo Church e Alan Turing. Ou seja, nestas circunstâncias o artigo de Alan Turing ficaria por publicar. No entanto, Maxwell Newman entendeu que a solução apresentada pelo seu aluno era inovadora e consideravelmente diferente da proposta apresentada por Alonzo Church. Nesse sentido e apesar do artigo publicado pelo lógico americano, o professor inglês recomendou que Alan Turing submetesse o artigo à Sociedade Matemática de Londres. O artigo «On Computable Numbers, with an Application to the Entscheidungsproblem» foi publicado em Novembro de 1936 com a indicação de que fora recebido pelo editor em Maio desse mesmo ano. Na sequência do episódio da coincidência entre os dois artigos, Alan Turing viajaria para os Estados Unidos a fim de desenvolver a tese de doutoramento na Universidade de Princeton com a supervisão de Alonzo Church.

O objetivo inicial do lógico e matemático inglês na criação de uma Máquina de Turing era resolver um problema matemático específico, concretamente o *entscheidungsproblem*. Assim, a aplicação da Máquina de Turing a outras situações implicaria, obviamente, a reformulação das regras da máquina. Mas esta é uma novidade extraordinária. Alan Turing concebeu um mecanismo que respondendo a determinadas instruções imprimiria numa fita com intervalos quadrados um código binário de 0s e 1s que representavam determinados números. Por exemplo, o número 1/3 em forma binário seria (.010101…). Desta forma simples, alterando apenas as regras do jogo, a mesma máquina passava a poder processar qualquer computação.

Para os leitores contemporâneos habituados a lidar com computadores de aparência mais sofisticada e menos limitações, a Máquina de Turing pode parecer extremamente rudimentar. Todavia, os princípios básicos para a computação de qualquer tipo de objeto estavam delineados, sobretudo o matemático inglês sugeria a forma de uma máquina servir para diferentes atividades. Para tal bastava alterar as instruções em função do que se pretendia computar. A estas instruções, a partir dos anos 1960, na literatura sobre computadores, passámos a designar algoritmos. Com a Máquina de Turing, o cientista inglês estabelecia a separação importante para o desenvolvimento das máquinas de computação entre *hardware* e *software*. Estávamos ainda longe do desenvolvimento que a teoria da computação teria nas décadas subsequentes, mas o artigo de Alan Turing contribuía sobremaneira para o início desse desenvolvimento. A Máquina de Turing definia as potencialidades e limites da computação. O engenho teórico criado por Turing estabelecia os requisitos para uma efetiva computação através da definição de instruções e dos mecanismos necessários para uma máquina poder seguir as instruções. Simultaneamente o autor britânico identificava os limites da computação: «podemo-nos equipar com milhares de processadores para executar em paralelo, podemo-nos esforçar para criar computadores quânticos capazes de executarem cálculos paralelos massivamente, mas simplesmente não podemos trazer o infinito para mais perto deste mundo irremediavelmente finito em que vivemos».[17]

Não demorou muito a que o artigo de Alan Turing fosse explorado para além da área da matemática onde fora germinado. Sobretudo não demorou muito para a comunidade científica aplicar o conceito de Máquina de Turing ao estudo e compreensão da mente humana. Como dizíamos na introdução, um modo de avançar na resolução de perguntas difíceis é descobrir ou construir novas estratégias de as abordar. A Máquina de Turing oferecia à comunidade científica uma nova estratégia para investigar o problema *mente humana*.

A aplicação da Máquina de Turing ao estudo da mente humana é fácil de compreender. Na perspectiva de um lógico-matemático a mente humana, ainda que *natural*, é um mecanismo que executa operações lógicas ou computacionais. Neste sentido, Turing inventou uma máquina capaz de simular e executar atividades lógico-matemáticas semelhantes às desenvolvidas pela mente humana. O cientista inglês projetou uma máquina capaz de rivalizar com a inteligência do ser humano,

17 Charles Petzold, *op. cit.*, p. 330.

capaz de solucionar um problema matemático, por exemplo; e lançou as bases da investigação do ser humano como máquina.

Durante séculos e provavelmente ainda hoje a maioria dos matemáticos concebe a sua atividade de investigação auxiliada por um papel e um lápis. Os matemáticos de nada mais precisam para explorar as demonstrações lógico-matemáticas. Alan Turing, com o artigo publicado em 1936, abriu a porta a um novo método de trabalho: o recurso a uma máquina para explorar uma demonstração matemática. As demonstrações bidimensionais do papel e lápis ganhavam com a Máquina de Turing modelização tridimensional. Alan Turing fazia a matemática levantar-se do papel. Ora a potencialidade desta modelização foi rapidamente transposta para o estudo da mente humana.

Os primeiros investigadores a perceberem as potencialidades de aplicação da Maquina de Turing ao estudo da mente humana foram Warren McCulloch e Walter Pitts, em 1943. Os dois autores americanos conceberam, pela primeira vez, o funcionamento cerebral como uma Máquina de Turing. Também Norbet Wiener e John von Neumann consideraram o artigo sobre números computáveis de Alan Turing muito estimulante e aplicaram, respetivamente, as consequências do artigo à cibernética e à construção de computadores. Sem surpresas um dos autores que exploraria a noção de inteligência artificial seria o próprio Alan Turing, no artigo que publicou na revista *Mind* em 1950.

Alan Turing ao conceber uma máquina capaz de efetuar operações semelhantes às de um ser humano abriu um campo de investigação imenso com o objetivo de reproduzir artificialmente a mente humana. O lógico inglês levava o funcionalismo às últimas consequências, afirmando ser indiferente o tipo de suporte material para a operação e realização das funções mentais. Além disso, o autor inglês concebeu um modo de avaliar se o objetivo de replicar artificialmente a mente humana fora alcançado ou não. Deste modo, Turing definiu uma estratégia e agenda de investigação para o estudo da mente humana. As décadas seguintes à sua morte nunca mais deixariam de investigar se *pode uma máquina pensar?*

2.2. Da mente como estrutura lógica à estrutura lógica como mente

Os pressupostos teóricos da obra de Edmundo Curvelo não são diferentes dos pressupostos teóricos subjacentes à obra de Alan Turing. Ambos partilham da ideia de que a mente humana obedece a uma estrutura lógico-matemática e que, portanto, a linguagem lógico-matemática é a via mais rigorosa para representar a mente humana. O raciocínio é o seguinte: se a mente humana é estrutura lógico-matemática, então a linguagem lógico-matemática será capaz de representar a mente. Assim, o problema *mente humana* fica reduzido à investigação de qual a melhor linguagem lógico-matemática para representar a mente humana. Nesse sentido, os dois autores preocuparam-se com a descodificação lógico-matemática da mente humana. Situação que hoje provoca a interrogação sobre por que razão os dois jovens autores, se estiveram tão próximos no assunto de eleição intelectual, estiveram tão afastados na possibilidade de diálogo. Pois não existe qualquer referência à obra do lógico inglês na obra do lógico português, apesar da proximidade dos objetivos do trabalho de Edmundo Curvelo relativamente ao trabalho de Alan Turing. A guerra, o contexto sociopolítico de Portugal e as eventuais dificuldades de comunicação não explicam necessariamente o desconhecimento de Emundo Curvelo da obra do cientista britânico. O artigo de Alan Turing sobre os números computáveis foi resenhado por Alonzo Church no *The Jounal of Symbolic Logic*, revista conhecida e lida pelo autor português. O artigo sobre máquinas computacionais e inteligência foi publicado na revista *Mind*, igualmente um periódico conhecido e lido por Edmundo Curvelo. Além do mais, o autor português mostra conhecer um conjunto de trabalhos na área da lógica e da matemática que, segundo ele, estavam a contribuir para o conhecimento psicológico e refere-se explicitamente ao «conjunto de investigações conhecido pelo nome de *Cybernetics*, efetuadas durante a última década por um grupo de cientistas … e publicadas em 1948 por N. Wiener».[18] Como referido na Introdução, Nobert Wiener salienta no livro de 1948 o trabalho de Alan Turing. Portanto, o assunto relativo à simulação artificial da mente não era certamente estranho a Edmundo Curvelo. Mais, o assunto não era estranho à cultura portuguesa.

18 Edmundo Curvelo, «Opuscula Psychologica I. Da teoria e da prática da psicotécnica». *Boletim do Instituto de Orientação Profissional*, 3.ª série, n.º 1, 1950, pp. 104-105.

Em 1952, Sílvio Lima publicou, no número 4 da *Revista Filosófica*, o texto «Cérebros Electrónicos e Cérebros Humanos».[19] Curvelo foi leitor e colaborador da revista editada pelo Professor Joaquim de Carvalho. Será que as reticências e objecções colocadas por Sílvio Lima à possibilidade dos cérebros eletrónicos substituírem integralmente e desempenharem as mesmas funções dos cérebros humanos representam as dúvidas dos intelectuais portugueses perante o assunto? Será que o parco desenvolvimento científico português não favorecia o desenvolvimento de ideias tão arrojadas como as preconizadas pelo movimento cibernético que despontava no mundo anglo-saxónico? Será que podemos ler nas palavras de Sílvio Lima a descrença de Curvelo e o desinteresse em simular artificialmente a mente humana?

Como se disse, na obra curveliana não há nenhuma referência direta ao trabalho de Alan Turing, nem a alusão ao movimento cibernético é desenvolvida, pelo que poderá ser provável a resposta positiva às perguntas enunciadas. Tanto mais que Edmundo Curvelo escreveu: «é escusado procurar-se visualizar a estrutura lógico-matemática de tal sistema de convenções. Essa estrutura, que não é de tipo espacial, não é visualizável».[20] No pensamento do lógico português parece ter existido apenas a ideia de construir estruturas lógico-matemáticas teóricas e nunca concebeu que essas estruturas teóricas pudessem ser replicadas ou visualizadas através da construção de modelos mecânicos capazes de reproduzirem e simularem o universo psicológico humano. Aos olhos de hoje e conhecendo o desenvolvimento científico que se verificou na segunda metade do século XX, a desatenção do lógico português à investigação sobre a possibilidade de simulação artificial da mente humana é uma das debilidades que descobrimos na sua obra, sobretudo porque o problema da capacidade e possibilidade de representação dos símbolos lógicos é um assunto que deverá merecer a atenção de qualquer lógico.

De qualquer modo a desatenção muito provavelmente foi mais fruto de consequências contextuais do que de alguma eleição científica. Convirá, uma vez mais, ter presente que o trabalho de Alan Turing não teve inicialmente muito eco entre os colegas. O artigo que o lógico

19 Silvio Lima, «Cérebros electrónicos e cérebros humanos». *Revista Filosófica*, n.º 4, 1952, pp. 5-17.

20 Edmundo Curvelo, *Princípios da Logificação da Psicologia*, (Lisboa: Ática, 1947), p. 91.

inglês publicou em 1936 sobre os números computáveis foi escrito para resolver um problema matemático que, na verdade, fora resolvido previamente por Alonzo Church com uma nomenclatura mais apreciada pelos colegas. Também não se poderá olvidar que só a partir de 1950 a investigação em Inteligência Artificial obteria maiores desenvolvimentos e que os livros principais de Curvelo foram escritos até 1947. É verdade que o artigo seminal para a IA de Warren McCulloch e Walter Pitts, «A Logical Calculus of Ideas Immanent in Nervous Activity», foi publicado em 1943. Porém, o artigo foi publicado na *Bulletin of Mathemetical Biophysics*, uma revista que integrava matemática e biofísica e que não possuía uma divulgação muito vasta. Por fim, é um facto que os primeiros computadores mostravam-se longe daquilo que os mais entusiastas profetizavam como sendo possível, a simulação artificial da mente humana. E não podemos ignorar que o trabalho lógico-matemático tradicionalmente se desenvolvia exclusivamente com recurso a lápis e papel. A proposta de Turing em modelar essa estratégia era uma novidade absoluta e eventualmente encarada como extravagante, sobretudo para alguém, como Curvelo, empenhado essencialmente na expressão abstrata da formalização lógica da psicologia e «só aqui ou além emergindo a ilha de um sistema ou interpretação especial ou da constantificação de uma variável, como ilustrações ou por acidental conveniência de exposição».[21]

Edmundo Curvelo esteve próximo de Alan Turing na conceção de uma estrutura lógica capaz de unificar todo o universo psicológico humano, mas esteve longe do pensamento do autor português a ideia de que se poderia ou seria pertinente para a compreensão do fenómeno mental a construção de uma máquina capaz de reproduzir e simular a mente humana. Em todo o caso, o lógico português elegeu para epígrafe do seu livro *Os Princípios da Logificação da Psicologia* um extrato do livro de Gardner Murphy, *An Historical Introduction to de Modern Psychology*, onde se lê: «Talvez não seja fantástico sugerir que, à medida que o nosso conhecimento e a nossa linguagem se tornarem mais rigorosos, a resposta à questão: "Quanto dói?" poderá ser: "$42xy^3$ cos A"».[22] A crença na possibilidade de matematizar toda a vida mental é evidentemente o primeiro passo para a possibilidade de replicar informaticamente a

21 *Ibidem*, p. 11.

22 *Ibidem*.

vida mental. A concretização será em grande medida preocupação e esforço de engenharia.

Portanto, não se poderá asseverar que Curvelo não consideraria propriamente interessante ou relevante a investigação em ordem à possibilidade de simular e replicar numa máquina a mente humana, mas efetivamente não foi assunto com o qual e ao qual se tivesse ocupado e dedicado. Não descobrimos na obra curveliana a preocupação com a construção de uma máquina inteligente, nem com o critério de validação da máquina. Neste aspeto, o investigador português foi um lógico tradicional. Edmundo Curvelo preocupou-se sobretudo com a investigação de uma linguagem suficientemente rigorosa para representar toda a vida mental. Para ele era suficiente que a representação se verificasse no papel e não que se levantasse deste para se visualizar ou replicar numa máquina. Por isso, o lógico português julgou ser suficiente a exploração dos instrumentos teóricos da matemática e da lógica simbólica. Concretamente do algoritmo da lógica probabilista--estatística polivalente e dos grupos de transformações. Sendo assim, como se desenvolveu a proposta de Curvelo?

O primeiro momento é similar a Turing. Só será possível avançar na explicação e compreensão da mente humana deixando de lado preconceitos que impeçam a exploração de todas as vias de pesquisa por mais abstrusas que possam parecer. Responder à pergunta sobre se uma máquina pensa só será possível se primeiro eliminarmos os argumentos que impedem olhar a pergunta com imparcialidade científica. No caso da resposta à pergunta formulada no artigo «Computing machinery and intelligence», o cientista inglês dedicou várias páginas a responder às objeções que a tradição poderia colocar à apreciação da pergunta. Depois, se o passado não foi capaz de resolver determinadas dificuldades, parece razoável entender que sejam ensaiadas vias diferentes. Deste modo, também Curvelo entendeu que investigar a mente humana só seria possível alterando a estratégia tradicional na forma de abordar o assunto. Segundo a perspectiva curveliana nem a psicologia experimental, nem o senso comum são as vias apropriadas. Estas encaminham a ciência da mente para o reino da entificação, impossibilitando a pesquisa científica. As vias que o lógico português critica entendiam que a mente é uma coisa (ente) e Curvelo entendia que a mente é um acontecimento dinâmico (relação). Neste sentido, o professor lisboeta assegura: «não são as noções relacionais e extremamente abstratas e analíticas da psicologia logificada que hão de ser intuídas em termos de vivências globais da experiência laboratorial ou do senso comum, mas ambas estas que hão de ser analisadas e compreendidas em termos de aquelas».[23] Ao contrário do que é habitual pensar, a logificação da psicologia precede a experiência. É a logificação que oferece luz de entendimento à experiência. Mas o lógico português não hesita em ser ainda mais assertivo em relação à necessidade de alteração da perspectiva para levar a investigação da vida mental mais longe. Diz ele: «Nem sequer palavras ou frases da linguagem vulgar estão aptas a traduzir tais noções [da vida mental como relação dinâmica], que só podem ser expressas numa simbologia rigorosa, pragmaticamente adequada, altamente analítica e abstrata. E essa simbologia apenas tem significado e apenas é entendida em termos e através das leis de uso e das propriedades operatórias dos símbolos».[24] Sem sombra de dúvidas Edmundo Curvelo propõe o fim da predominância da observação, até porque a observação só produziu diversidade de teorias psicológicas. Cada teoria psicológica perspectivou sob ângulo diferente: do elemento (a psicologia atomística), do todo (a psicologia da *Gestalt*), da aprendizagem e motivação (o funcionalismo), do comportamento (o behaviorismo). As diferentes teorias originaram fragmentação teórica e perda de capacidade para unificarem a totalidade da vida mental. Para alterar esta situação, Curvelo pensou ser necessário que na psicologia passasse a dominar a linguagem lógico-matemática, por claramente ser a única linguagem capaz de unificar as diferentes teorias psicológicas e o assunto que estas pretendem explicar e compreender, a mente humana. Assim, reconhecendo que a própria experiência implica teoria lógico-matemática, a investigação sobre a mente humana deverá oferecer o ponto de partida à investigação lógico-matemática. E porquê?

Na resposta sobressairá a teoria do conhecimento de Edmundo Curvelo. Tradicionalmente, concebe-se que a ciência descobre o mundo. O lógico português, pelo contrário, concebe que a ciência constrói o mundo. Não significa que não exista um mundo totalmente exterior ao ser humano, mas esse é incognoscível. Pois o ser humano só conhece o mundo que a sua mente constrói. Imaginemos um grupo abstrato

23 *Ibidem*, p. 10

24 *Ibidem*.

constituído por um conjunto de factos naturais, por uma operação e por uma relação fundamental. Chamemos a esse grupo, *N*. O trabalho da ciência consistirá em construir um grupo *N'* de tal modo que *N* e *N'* se correspondam biunivicamente Ou seja, *N'* será uma cópia de *N* na medida em que os dois grupos sejam equivalentes. No entanto, Curvelo afirma que este modo de conceber o trabalho científico é ilusório, porque não há modo de comparar os dois grupos. Para comparar os dois grupos seria necessário que *N* fosse conhecido *independentemente* de *N'*. Porém, se *N* fosse conhecido independentemente de *N'*, então *N'* seria inútil. Mas acontece que *N'* é precisamente logicamente construído para se conhecer *N*. E acontece ainda que *N'* é construído, como instrumento lógico para conhecer *N*, de acordo com a estrutura que o ser humano possui para conhecer (construir *N*). Assim, «nada podemos conhecer a não ser *N'*; o conhecimento de *N* está-nos vedado absolutamente. O conhecimento marcha de *N'* para *N*, e não de *N* para *N'*. *N* é, pura e simplesmente, postulado».[25] Este é um pressuposto importante a salientar, porque informa toda a teoria da mente de Edmundo Curvelo e o modo como desenvolverá a logificação da psicologia e, por conseguinte, também da mente humana. Se *N* for a mente humana que queremos conhecer e *N'* a estrutura lógico-matemática da mente que se quer conhecer, então a única mente humana que poderemos conhecer é a que irrompe da análise da estrutura lógico-matemática. Assim, a estrutura lógico-matemática que a ciência for capaz de construir de modo a que o sistema de relações em *N'* seja de tal forma que permita construir relações entre *N* e *N'*, acabará por se identificar com o que é a mente humana, *N*.[26]

Esta proposta poderá não parecer evidente. Contudo, Curvelo responde que a evidência sensorial e racional deverão ser destronadas pela «fecundidade de certos conjuntos ou sistemas de condições, para nós, critério de verdade».[27] Mais importante do que qualquer evidência sensorial ou racional é a possibilidade de se postularem, num determinado conjunto ou sistema de condições, certas operações que se revelem utilitariamente e pragmaticamente mais fecundas para a construção de um esquema científico. «O progresso da investigação científica tem

25 *Ibidem*, pp. 73-74

26 Cf. *Ibidem*, pp. 124 e ss.

27 *Ibidem*, p. 34.

historicamente dependido, precisamente, da nossa coragem para negarmos o que é evidente, aceite por toda a gente em todos os tempos e em todos os lugares, e substituí-lo pelo seu contraditório à luz dos esquemas tradicionais, ou até mesmo pelo não-imaginável».[28] Portanto, a lógica, que tem por objeto a análise das estruturas científicas, não tem que se submeter a evidências sensoriais ou racionais, mas procurar construir um sistema onde os postulados «permitam demonstrar os teoremas e não conduzam a contradição interna»,[29] que não afirmem e neguem simultaneamente uma determinada identidade.

O lógico português dedicou à logificação da mente três livros e uma série de três artigos com o título geral *Opuscula Psychologica*. Dos livros, os dois primeiros, *Fundamentos Lógicos da Psicologia* (1945) e *Relações Lógicas, Psicológicas e Sociais da Ética* (1946) abordam o tema de modo essencialmente teórico. É no último livro, do conjunto dos três livros dedicados à logificação da psicologia, intitulado *Os Princípios da Logificação da Psicologia*, que Edmundo Curvelo desenvolve os aspetos técnicos. Ao longo de todos estes textos fica clara a seguinte ideia. A mente humana é uma estrutura lógica, porque a mente é conhecimento e conhecimento é lógica. Portanto, o estudo da lógica é a via de acesso à compreensão da mente humana, porque as duas estruturas são equivalentes.

Conclusão

Neste texto procuramos estabelecer um paralelo entre Alan Turing e Edmundo Curvelo. Dois intelectuais que partilharam praticamente os mesmo anos de vida e que têm a particularidade de terem eleito para assunto principal de investigação o estudo da mente humana de um ponto de vista lógico-matemático. Ambos, autores de uma obra sugestiva para todo o interessado no desenvolvimento que haveria de ter a ciência cognitiva nas décadas subsequentes às suas mortes. Dois autores que comungaram da afirmação de que a via lógico-matemática é a melhor via para a compreensão da vida mental. Tanto Alan Turing como Edmundo Curvelo compreenderam a necessidade de reformular o problema da investigação sobre a mente humana com base numa

28 *Ibidem*.

29 *Ibidem*, pp. 34-35.

linguagem rigorosa capaz de anular a visão substancialista da mente. Tanto Alan Turing como Edmundo Curvelo compreenderam a importância da linguagem lógico-matemática para traduzir a mente humana consciente, mas não meramente como linguagem-instrumento, também como linguagem-representação. A linguagem lógico-matemática como instrumento auxiliar da ciência sempre foi bem entendida e utilizada pelos cientistas. Porém, os lógicos inglês e português cultivaram uma ideia muito mais radical. A ideia de que a linguagem lógico-matemática poderá substituir integralmente a realidade que procura estudar e compreender, no caso a mente humana. Porque estiveram tão próximos no objetivo último, resulta tão estranho a razão de Edmundo Curvelo não haver prestado mais atenção ao trabalho do lógico inglês. Indubitavelmente o encontro intelectual de Edmundo Curvelo com Alan Turing teria permitido ao lógico português alargar os horizontes de compreensão e aplicabilidade do projeto de logificação da psicologia que tinha entre mãos.

Juntamos agora quem nunca se juntou em vida. Dois cientistas que bem merecem a homenagem através da releitura e estudo da obra que deixaram às gerações vindouras. Felizmente um, Alan Turing, reconhecido internacionalmente. Infelizmente outro, Edmundo Curvelo, praticamente um ilustre desconhecido. Curiosamente ambos comungando da mesma agenda intelectual: a explicação da mente humana através da via da linguagem lógico-matemática.

Bibliografia

Bennett, Maxwell, Hacker, Peter, *Philosophical foundations of neuroscience*. (Oxford: Wiley--Blackwell, 2003)

Bennett, Maxwell, *et al.*, *Neuroscience & Philosophy. Brain, Mind & Language*. (New York: Columbia Universiy Press, 2007).

Church, Alonzo, «A note to the entscheidugsproblem». *The Journal of Symbolic Logic,* 1, 1(1936), pp. 40-41.

Church, Alonzo, «An unsolvable problem of elementary number theory». *The Journal of Symbolic Logic*, 58, 2(1936), pp. 345-363.

Church, Alonzo, «On computable cumbers, with an application to the entscheidungsproblem by A. M. Turing». *The Journal of Symbolic Logic*, 2, 1(1937), pp. 42-43.

Curado, Manuel, Alves, José António, *Um génio português: Edmundo Curvelo (1913-1954)*. (Coimbra: Imprensa da Universidade de Coimbra, 2013)

Curvelo, Edmundo, *Fundamentos lógicos da psicologia*. (Coimbra: Atlântida, 1946)

Curvelo, Edmundo, *Princípios da logificação da psicologia*. (Lisboa: Ática, 1947)

Curvelo, Edmundo, «Opuscula Psychologica I. Da teoria e da prática da psicotécnica». *Boletim do Instituto de Orientação Profissional*, 3.ª série, 1 (1950), pp. 85-122.

Copeland, J. B, ed., *The essential Turing. The ideas that gave birth to the computer age*. (Oxford: Oxford University Press, 2010 [2004]).

Hodges, Andrew, *Alan Turing. The enigma*. (Princeton: Princeton University Press, 2012 [1983]).

Leavitt, David, *El hombre que sabía demasiado. Alan Turing*. (Trad. Federico Corriente Basús, Barcelona: Antoni Bosch, 2006).

Penrose, Roger, *The Emperor's new mind. Concerning computers, minds, and the laws os phisics*. (Oxford: Oxford University Press, 1990 [1989]).

Petzold, Charles, *The annoted Turing. A guided tour through Alan Turing's historic paper on computability and Turing Machine*. (Idianapolis: Wiley Publishing, 2008).

Plutarco, *Vidas paralelas. Péricles e Fábio Máximo*. (Trad., Intr. e Notas de Ana Maria Guedes Ferreira e Ália Rosa Conceição Rodrigues, 3ª edição, Coimbra: Imprensa da Universidade de Coimbra, 2013).

Searle, John, «Minds, brains, and programs». *Behavioral and Brain Science*, 3 (1980), pp. 417-458.

Turing, Alan M., «On computable numbers, with an application to the entscheidungsproblem». *Proccedings of the London Mathematical Society*, 42 (1936), pp. 230-265.

Turing, Alan M., «On computable numbers, with an application to the entscheidungsproblem. A correction». *Proccedings of the London Mathematical Society*, 43 (1936), pp. 544-546.

Turing, Alan M., «Computing machinery and intelligence». *Mind*, 59, 236 (1950), pp 433-460.

Turing, Sara, *Alan M. Turing. Centenary edition*. (Cambridge: Cambridge University Press, 2012).

7. Ferramentas Inteligentes para a Geometria

Pedro Quaresma

Por *Ferramentas Inteligente para a Geometria* entende-se ferramentas computacionais capazes de uma *inteligência algorítmica*. Não vou aqui tentar definir formalmente inteligência algorítmica, antes vou descrever as ferramentas computacionais inteligentes, ficando ao cargo do leitor a compreensão do que o conceito de *inteligência algorítmica* poderá abarcar no que concerne a Geometria.

Pretende-se apresentar de que forma a Máquina de Turing se entrecruzou com a Geometria e de que forma é que a Geometria e a utilização que fazemos dela mudou.

Julgo que a primeira área da geometria a ter sido, de forma significativa, tocada pelo advento dos computadores foi a da demonstração automática de teoremas. A área da inteligência artificial [20, 46, 47] viu na teoria axiomática da geometria um conjunto de axiomas e de regras de inferência simples e manejável [25]. A geometria mostrou-se um campo muito interessante para quem, explorando os campos da inteligência artificial, pretendia automatizar as demonstrações de teoremas. É assim que surgem os primeiros demonstradores automáticos de teoremas para a geometria (GATP, do Inglês "Geometry Automated Theorem Provers").

As complexidades inerentes ao desenvolvimento automático das demonstrações, nomeadamente com a "explosão" dos diferentes casos a considerar à medida que se vai tentando desenvolver a demonstração, levou a que estas primeiras tentativas não tenham sido muito profícuas. Ver-se-á mais tarde como é que este ramo da história se desenvolveu.

Um outro campo da Geometria, a sua representação visual, teve que esperar um pouco mais. Teve que esperar pelo desenvolvimento das capacidades gráficas (anos 80 do século XX) nos sistemas computacionais.

É a partir das décadas de 1980/1990 que surgem os primeiros sistemas computacionais de geometria dinâmica (DGS, do Inglês "Dynamic Geometry Systems"). Não estamos perante programas de desenho que permitem o desenhar de figuras geométricas por colocação de pontos, retas e outros objetos geométricos mais complexos num dado plano (Cartesiano), mas sim de programas capazes de lidar com as figuras geométricas de uma forma construtiva. Uma dada figura geométrica deixa de ser desenhada para passar a ser construída de acordo com um conjunto de regras geométricas bem definidas [54]. Este tipo de programas tem tido uma evolução contínua e são já ferramentas maduras de larga utilização em diferentes ambientes.

O retomar das aproximações à demonstração automática de teoremas na geometria deu-se a partir dos fins dos anos 1970 e depois nos anos 1980 com o desenvolvimento de métodos algébricos tais como o método do conjunto característico de Wu-Ritt [7, 58] e os métodos baseados nas bases de Gröbner [3,4]. Estes métodos revelaram-se muito poderosos sendo capazes de demonstrar automaticamente centenas de teoremas geométricos complexos. Estes métodos recorrem à representação algébrica das construções geométricas, desenvolvendo as demonstrações por métodos puramente algébricos (resolução de sistemas de equações).

Mais recentemente, década de 90 do século XX, foram desenvolvidos métodos semialgébricos, por exemplo o método da área [27], em que uma dada teoria axiomática específica foi desenvolvida e em que os métodos de demonstração geométrica, com algumas manipulações algébricas elementares auxiliares, tornam possível a demonstração de centenas de teoremas geométricos e em que a demonstração formal é passível de ser seguida por um matemático.

Atualmente estas duas áreas das ferramentas inteligentes para a geometria começam a juntar-se em ferramentas computacionais que permitem, baseando-se na geometria construtiva, a representação visual dinâmica e a ligação à demonstração automática para a validação da própria construção e/ou a tentativa de demonstração de uma dada conjetura sobre a construção.

Uma outra aproximação também recente, tornada possível pelo desenvolvimento de ferramentas computacionais interativas de demonstração (ITP, do Inglês "Interactive Theorem Proving") tais como o *Coq* e o *Isabelle* [39, 53] é o da formalização de fragmentos da geometria [32, 35, 37]. De certa forma esta aproximação dá-nos o fechar de um ciclo, desde a máquina formal da computação, a máquina de Turing, até ao uso dessa máquina para formalizar áreas da matemática.

Vejamos agora em mais detalhe cada uma destas áreas.

1. Métodos da Inteligência Artificial na Geometria

Ao tentarem automatizar os métodos tradicionais de demonstração, muitos dos métodos da área da inteligência artificial introduzem pontos auxiliares de forma a que determinados postulados se possam aplicar à demonstração. Estas aproximações levam a uma explosão combinatória no espaço de procura. O desafio é então o de, utilizando heurísticas, minimizar o número de passos necessários, evitando, desse modo, a explosão combinatória no espaço das soluções possíveis. Exemplos de métodos deste tipo incluem aproximações por Gelernter [21], Nevins [38], Elcock [18], Greeno et al. [23], Coelho e Pereira [14], Chou Gao, e Zhang [9].

Os métodos sintéticos tentam automatizar os métodos usuais de demonstração em geometria de forma a conseguir demonstrações legíveis por matemáticos.

Em 1950 Gelernter criou um demonstrador automático de teoremas que era capaz de achar solução para problemas em geometria plana ao nível dos livros de estudo do ensino secundário. Era baseado na simulação dos métodos humanos tendo sido considerado, na altura, como um caso exemplar na área da inteligência artificial [21].

Vejamos a seguinte conjetura geométrica e como os métodos sintéticos, como o proposto por Gelernter, a tentam demonstrar.

Teorema 1 *Seja ABCD um paralelogramo (i.e. $AB \| CD$, $BC \| AD$) e E o ponto de interseção de AC e BD. Então tem-se que $\overline{AE} = \overline{EC}$.*

A conjetura expressa-se da seguinte forma:

$$\forall_{\text{elementos geométricos}}[(H_1 \wedge \cdots \wedge H_r) \Rightarrow G]$$

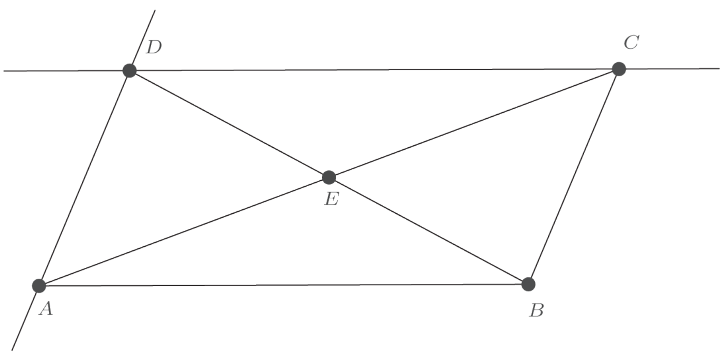

Figura 1. Propriedade das Diagonais de um Paralelogramo

Neste caso concreto, ter-se-ia (ver Figura 1):

pontos $(A, B, C) \wedge AB||CD \wedge AD||BC \wedge \text{col}(E, A, C) \wedge \text{col}(E, B, D) \Rightarrow AE = EC$,

com col(E, B, D) a significar que os pontos A, B e C são colineares. O facto de que os pontos A, B e C são objetos livres, isto é, universalmente quantificados, é representado por pontos (A, B, C).

A demonstração é efetuada por encadeamento retrógrado, isto é, partindo da conclusão, G, procura-se no conjunto de regras de inferência e axiomas uma regra da seguinte forma

$$[(G_1 \wedge \cdot\cdot\cdot \wedge G_r) \Rightarrow G].$$

Este processo repete-se para os novos sub-objetivos $(G_1 \wedge \cdot\cdot\cdot \wedge G_r)$ até que todos os sub-objetivos sejam axiomas ou hipóteses. É usual representar a demonstração através de uma "árvore de demonstração" conjunção-disjunção. Esta "árvore" terá como raiz a conclusão da conjetura, ramificando-se à medida que os novos sub-objetivos são encontrados e, se a conjetura for válida, terá como "folhas" somente axiomas ou as hipóteses da conjetura.

Lendo a árvore de demonstração (ver Figura 2) vê-se que da conclusão que se pretende obter se "sobe" por aplicação da regra $\Delta \text{ECD} \cong \Delta EAB \Rightarrow AE = EC$, para este sub-objetivo aplicar-se-ia a regra: $AB = \text{CD} \wedge \angle \text{AEB} = \angle CED \wedge \angle \text{ECD} = \angle \text{EAB} \Rightarrow \Delta ECD \cong \Delta EAB$. Olhando para a árvore de demonstração verifica-se que, neste caso concreto, é possível "fechar" esta árvore tendo como folhas da mesma somente axiomas ou hipóteses da conjetura (ou, como está representado na Figura 2, o valor de verdade V).

Para demonstrar a congruência dos triângulos recorre-se ao facto de que $\angle CAD = \angle ACB$, o que se deve ao facto de os dois ângulos em questão serem ângulos alternos internos das retas paralelas AB e CD, no entanto para tal assumiu-se o facto "trivial" que os pontos D e B estão em lados opostos da reta AC. Este último facto é mais difícil de estabelecer de forma automática do que a afirmação inicial.

Para poder desenvolver as demonstrações sintéticas de forma eficiente vários autores sugerem a utilização das construções geométricas como auxiliares ao processo demonstrativo. Duas utilizações possíveis de um modelo geométrico [14]:

- o diagrama como um filtro (um contra-exemplo);
- o diagrama como um caso exemplar (um exemplo sugerindo uma eventual conclusão).

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\mathscr{V}}{AC = CA} \qquad \dfrac{\mathscr{V}}{\angle ACD = \angle CAB}\,{\scriptstyle AB \parallel CD} \qquad \dfrac{\mathscr{V}}{\angle CAD = \angle ACB}\,{\scriptstyle AD \parallel BC}}{\Delta ABC \cong \Delta CDA}}{AB = CD}
\qquad
\dfrac{\mathscr{V}}{\angle AEB = \angle CED}
\qquad
\dfrac{\dfrac{\mathscr{V}}{\angle ACD = \angle CAB}\,{\scriptstyle AB \parallel CD}}{\angle ECD = \angle EAB}\,{\scriptstyle \mathrm{col}(E, A, C)}
}{\dfrac{\Delta ECD \cong \Delta EAB}{AE = EC}}
$$

Figura 2. Árvore de demostração do Teorema 1

A introdução de novos pontos é vista como uma forma de explicitar mais informação proveniente do modelo [14].

Por outro lado, é dito que um demonstrador deve poder combinar um encadeamento retrógrado com um encadeamento direto aquando da execução [14].

Embora tivessem havido muitos desenvolvimentos referentes a estratégias e heurísticas de demonstração, o problema da explosão combinatória no espaço das soluções continua a ser a maior dificuldade a enfrentar por estes tipos de métodos. É de notar que as aproximações da IA não são procedimentos de decisão, isto é, procedimentos capazes de determinar se uma fórmula arbitrária *P* é válida (ou não).

A despeito de alguns sucessos parciais e de melhoramentos significativos, os vários esforços nessa direção não conduziram ao desenvolvimento de nenhum demonstrador automático de teoremas, eficiente e capaz de demonstrar um grande número de conjeturas geométricas [1,13, 14, 22, 30, 40].

2. Sistemas de Geometria Dinâmica

O prémio Turing de 1988 foi atribuído a Ivan Sutherland, pelo seu trabalho pioneiro na área dos grafismos computacionais. O programa *Sketchpad* mudou a forma como as pessoas interagiam com os computadores, de uma forma não gráfica, para uma forma gráfica [51]. O programa *Sketchpad* pode ser considerado como o ponto de origem para os atuais programas de conceção gráfica assistida por computador, CAD, no acrónimo derivado do Inglês *Computer-Aided Drafting*.

Não desmerecendo dos programas de CAD, eles pouco têm de interessante para o praticante de geometria. Falta-lhes umas "gotas de inteligência geométrica" para nos serem úteis. Então o que são e como se distinguem dos CAD, os sistemas de geometria dinâmica?

A palavra "dinâmica" esconde muito mais do que aquilo que aparenta à primeira. Não se trata de um dinamismo no sentido de animação, movimento de objetos geométricos. Trata-se da movimentação de certos elementos de uma dada construção geométrica, preservando as propriedades da mesma.

Os programas de geometria dinâmica (DGS) estão baseados numa teoria construtiva da geometria [54] permitindo a realização de construções geométricas a partir de objetos livres e de construções elementares. A natureza dinâmica de tais programas permite aos seus utilizadores manipular as posições dos objetos livres (objetos quantificados universalmente), preservando as propriedades geométricas da construção. Vejamos o seguinte resultado, referente a uma propriedade dos triângulos isósceles retângulos.

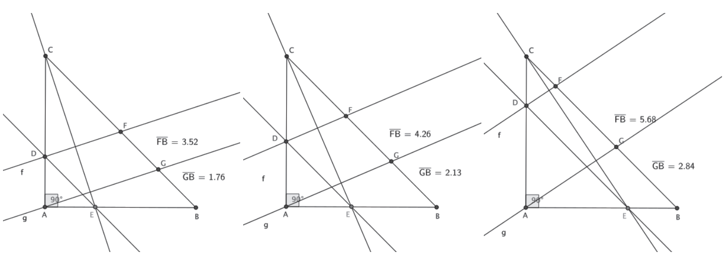

Figura 3. Propriedade de um Triângulo Isósceles Retângulo.

Teorema 2 *Seja ΔABC um triângulo isósceles retângulo com o ângulo reto em A. Seja E um ponto na aresta AB do triângulo. Seja D a interseção da linha que passa por E e é paralela à aresta BC e à linha AC. Sejam f e g duas retas perpendiculares com a linha CE e que passam pelos pontos D e A respetivamente, e sejam F e G as interseções das retas f e g com a hipotenusa BC. Então tem-se que o ponto G é o ponto médio do segmento FB.*

A construção de uma tal figura geométrica (ver Figura 3) é de muito fácil execução num programa como o GeoGebra[1]. Além disso podemos explorar o carácter dinâmico deste DGS para verificar que a manipulação dos objetos livres (pontos *A*, *B* e *E*) não altera a propriedade em estudo.

Temos de ter em atenção que esta manipulação dinâmica da figura, embora muito apelativa, não providencia uma demonstração da conjetura geométrica, trata-se somente de uma validação da mesma num número finito de casos, no modelo Cartesiano. É interessante notar uma outra situação: o *GeoGebra* possui uma opção de verificação da relação entre dois objetos "(a $\stackrel{?}{=}$ b)". Se a aplicarmos às retas *f* e *g* ele vai responder-nos que "as retas f e g são paralelas, verificado numericamente". Mas, como o próprio programa refere, este é um resultado verificado numericamente e não formalmente[2]. Uma recente evolução do GeoGebra introduz a demonstração formal de algumas propriedades geométricas, passa a ser possível pedir a validação formal de uma dada propriedade da construção [2].

Os DGS revelam-se excelentes substitutos para os "velhinhos" instrumentos: régua e compasso. O seu carácter dinâmico permite estabelecer uma primeira ponte entre os modelos e as teorias da Geometria.

Veja-se, por exemplo, o seguinte teorema geométrico (ver Figura 4):

Teorema 3 (Teorema de Ceva) *Seja ΔABC um triângulo e P um ponto arbitrário do plano. Seja D a interseção de AP e BC, E a interseção de BP e AC e F a interseção* CP e AB. *Então temos que*:

$$\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} = 1.$$

1 http://www.geogebra.org/cms/

2 Numa utilização anterior chegou a obter-se o resultado "as retas f e g não são paralelas".

Usemos agora o programa GCLC[3] para fazer a representação gráfica deste resultado. O programa GCLC é um programa de geometria dinâmica que incorpora vários demonstradores automáticos de teoremas para a geometria.

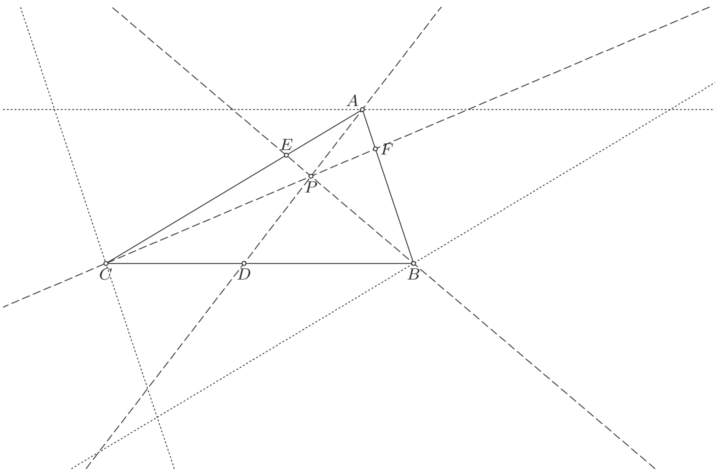

Figura 4. Teorema de Ceva.

Podemos, a exemplo do que já fizemos no exemplo anterior, manipular a construção movendo os objetos livres, isto é, aqueles que estão implicitamente quantificados universalmente, neste caso os pontos *A, B, C* e *P*. Se à construção adicionarmos os comprimentos dos segmentos atrás referidos formando a equação pretendida podemos, a menos de imprecisões numéricas, confirmar o resultado.

Mas, usando a sintaxe do GCLC, podemos acrescentar o seguinte à construção

3 http://www.emis.de/misc/software/gclc/

```
prove { equal { mult  { mult
                          { sratio A F F B }
                          { sratio B D D C }
                        }
                        { sratio C E E A }
                }
          1
        }
```

não é mais do que a equação $\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}}=1$ expressa na sintaxe do GCLC e o pedido para que a mesma seja formalmente demonstrada.

Este é um resultado que é facilmente demonstrado por qualquer um dos demonstradores automáticos de teoremas embutidos no GCLC, com tempos de execução abaixo das milésimas de segundo. Além da demonstração do resultado, o programa também nos indica que o resultado é válido, mas não no caso mais geral. É necessário ter em conta os casos degenerados, para este resultado verifica-se que o ponto *P* não pode pertencer a nenhuma das retas paralelas aos lados do triângulo e que passam pelo vértice oposto (as linhas a pontilhado na construção da Figura 4).

Como veremos na próxima secção, os GATP podem ser úteis, tanto na verificação da correção de uma dada construção, como na demonstração formal de uma dada conjetura geométrica. Se o método de demonstração usado pelo GATP for geométrico, e não algébrico, a própria demonstração pode ser um objeto de estudo.

3. Demonstração Automática de Teoremas Geométricos

Existem duas aproximações possíveis principais quando se trata da automação da demonstração em geometria. Os métodos sintéticos e os métodos algébricos.

Os métodos algébricos têm as suas raízes nos trabalhos de Descartes e na tradução de um problema geométrico num problema algébrico. A automação da demonstração de acordo com esta aproximação começou com o método da eliminação dos quantificadores de Tarski [52] tendo tido muitos melhoramentos desde então [15]. O método do conjunto característico, também designado por método de Wu-Ritt [5, 56], o método de eliminação [55], o método das bases de Gröbner [29] e a aproximação das álgebras de Clifford [31] são exemplos de métodos algébricos já implementados computacionalmente. Todos estes métodos têm em comum o facto de se basearem em manipulações algébricas, sem relação com os processos habituais de demonstração geométricas. As demonstrações são muito difíceis, se não impossíveis, de seguir dada a complexidade das manipulações sobre polinómios já de si complexos. Na secção 3.1 apresenta-se de forma breve o método de Wu-Ritt.

Uma segunda aproximação à automação da demonstração em geometria é dada pelos métodos sintéticos. Como já vimos na secção 1.1, esta foi uma área muito ativa, utilizando métodos da inteligência artificial, nas décadas de 1950 e 1960. Um tipo de métodos que estão, podemos assim dizer, a meio caminho entre estas duas aproximações são os, assim designados, métodos livres de coordenadas. Neste tipo de métodos as demonstrações combinam inferências geométricas com manipulações algébricas conseguindo demonstrar muitos teoremas complexos da geometria de forma eficiente e, em geral, com demonstrações sucintas e legíveis por um matemático. Na secção 3.2 apresenta-se o método da área, um método representativo deste tipo de aproximação.

3.1. Métodos Algébricos

O método de Wu-Ritt e o método das Bases de Gröbner [7] são dois dos métodos algébricos com um maior número de implementações atualmente (veja-se [28, 59], entre outros). O método de Wu-Ritt é um método capaz de demonstrar automaticamente um grande número de construções. O método é completo para proposições geométricas que envolvam somente igualdades em geometria métrica, um sub-conjunto da geometria Euclidiana, no qual as construções geométricas seguem um certo conjunto de passos construtivos [7, 57, 58].

Vejamos este método por meio de um exemplo. Relembrando o teorema 1 (ver Figura 1).

Teorema 4 *Seja ABCD um paralelogramo (i.e.* $AB \| CD$*,* $BC \| AD$*) e E o ponto de interseção de AC e BD. Então tem-se que* $\overline{AE} = \overline{EC}$*.*

Só três dos objetos da construção são objetos livres (implicitamente quantificados universalmente). Sejam eles *A*, *B* e *C*. Para simplificar podemos situar a reta *AB* sobre o eixo das abcissas com *A* na origem, isto dar-nos-ia, *A*(0, 0), *B*(u_1, 0), *C*(u_2, u_3), *D*(x_2, x_1) e E(x_3, x_4), em que a distinção entre variáveis livre e dependentes é-nos dada pela diferente designação, *u* e *x*. Pretende-se demonstrar que $g = 2u_2x_4 + 2u_3x_3 - u^2{}_3 - u^2{}_2 = 0$, isto é que $\overline{AE} = \overline{EC}$.

Usando as equações das retas *AD* e *CD* obtém-se as coordenadas do ponto *D*. Usando as equações das retas *BD* e *AC* obtém-se as coordenadas do ponto *E*. Podemos então substituir estas soluções no polinómio *g* verificando se o mesmo é, ou não, igual a zero.

Para x_1 e x_2 temos as seguintes duas equações:

$h_1 = u_1x_1 - u_1u_3 = 0$ $\qquad$ *AB*||*DC*

$h_2 = u_3x_2 - (u_2 - u_1)\, x_1 = 0$ $\qquad$ *DA*||*CB*

Para x_3 e x_4 temos também duas equações:

$h_3 = x_1x_4 - (x_2 - u_1)\, x_3 - u_1x_1 = 0$ $\qquad$ *E* está em *BD*

$h_4 = u_3x_4 - u_2x_3 = 0$ $\qquad$ *E* está em *AC*

Resolvendo estes sistemas de equações vai obter-se $x_1 = u_3$, $x_2 = u_2 - u_1$, $x_3 = u_3/2$ e $x_4 = u_2/2$. Substituindo estes valores em *g* tem-se que $g = 0$. Em consequência podemos afirmar que se demonstrou o teorema.

Esta solução não é isenta de problemas dado que não estabelece as condições de não degenerescência, isto é, não estabelece nenhuma condição sobre a qual a propriedade não é válida. Para este caso concreto temos que assumir que $u_1 \neq 0$ e que $u_3 \neq 0$, isto é, que os pontos *A*, *B* e *C* não são colineares.

O Método de Wu-Ritt

O método de Wu-Ritt é um método de demonstração automática de teoremas geométricos para os quais, na sua forma algébrica, as hipóteses e as conclusões podem ser expressas por equações polinomiais.

Para essas proposições geométricas e após adotar um sistema de coordenadas apropriado, as hipóteses e a conclusão podem ser expressas como um conjunto de equações polinomiais:

$$
\begin{array}{lcl}
h_1(u_1, \ldots, u_d, x_1, \ldots, x_i) & = & 0 \\
h_2(u_1, \ldots, u_d, x_1, \ldots, x_i) & = & 0 \\
\ldots & & \\
h_n(u_1, \ldots, u_d, x_1, \ldots, x_i) & = & 0 \\
g(u_1, \ldots, u_d, x_1, \ldots, x_i) & = & 0
\end{array}
$$

onde $h_1, \ldots h_n, g$ são equações polinomiais em $Q\,[u_1 \ldots, u_d, x_1, \ldots, x_i]$ (Q o corpo dos números racionais), $u_1 \ldots, u_d$ são parâmetros ou variáveis independentes e as variáveis $x_1, \ldots, x_i$ são algebricamente dependentes dos *u*'s.

Revendo a demonstração do teorema 1. A formulação inicial do problema é dada por:

$$(h_1 = 0 \wedge h_2 = 0 \wedge h_3 = 0 \wedge h_4 = 0) \Rightarrow g = 0.$$

No entanto, como já vimos acima, esta fórmula não é válida dado que as condições de não degenerescência não foram acrescentadas às hipóteses. Nem sempre é fácil conseguir descortinar qual é o conjunto completo de condições de não degenerescência (ndg), o método de Wu-Ritt vai permitir desenvolver a demonstração de tal forma que no fim obtém-se as condições de não degenerescência.

A demonstração prossegue por triangulação das hipóteses usando pseudo-divisões de polinómios. É necessário que cada sucessivo polinómio nas hipóteses introduza uma variável dependente. Verifica-se que o polinómio h_3 não está nessas condições dado que introduz duas variáveis dependentes, x_3 e x_4. Por um simples processo de eliminação obtemos o sistema de equações na forma triangular em que f_3 é dado pelo pseudo-resto (prm) da divisão de h_4 por h_3 em x_4, denotado por $f_3 = \text{prm}(h_4, h_3, x_4)$:

$$\begin{aligned} f_1 &= u_1x_1 - u_1u_3 = 0 \\ f_2 &= u_3x_2 - (u_2 - u_1)\, x_1 = 0 \\ f_3 &= (u_3x_2 - u_2x_1 - u_1 u_3)\, x_3 + u_1 u_3x_1 = 0 \\ f_4 &= u_3x_4 - u_2x_3 = 0 \end{aligned}$$

A demonstração desenvolve-se por sucessivas pseudo-divisões:

$$\begin{aligned} R_3 &= \text{prm}(g, f_4, x_4) = 2(u_3^2 + u_2^2)x_3 - u_3^3 - u_2^2u_3 \\ R_2 &= \text{prm}(R_3, f_3, x_3) = (-u_3^4 - u_2^2u_3^2)x_2 + ((u_2 - 2u_1)u_3^3 + (u_2^3 - 2u_1u_2^2)u_3)x_1 + \\ &\qquad + u_1u_3^4 + u_1u_2^2u_3^2 \\ R_1 &= \text{prm}(R_2, f_2, x_2) = (-u_1u_3^4 - u_1u_2^2u_3^2)x_1 + u_1u_3^5 + u_1u_2^2u_3^3 \\ R_0 &= \text{prm}(R_1, f_1, x_1) = 0 \end{aligned}$$

Dado que $R_0 = 0$, temos que o teorema é verdadeiro, sujeito às condições de não degenerescência que advêm da fórmula do resto para as sucessivas pseudo-divisões de g com respeito a uma fórmula triangular $f_1, f_2, \ldots, f_r$:

$$I_1^{s_1} \ldots I_1^{s_r} g = Q_1 f_1 + \cdots + Q_r f_r + R_0$$

na qual os I_k são os coeficientes iniciais de f_k em x_k.

Temos então que, dado que $R_0 = 0$, então $g = 0$ sujeito as condições adicionais $I_k \neq 0$ ($k = 1, \ldots, r$).

Para este exemplo ter-se-ia.

$$\begin{aligned} I_1 &= u_1 \neq 0 \\ I_2 &= u_3 \neq 0 \\ I_3 &= u_3x_2 - u_2x_1 - u_1u_3 \neq 0 \\ I_4 &= u_3 \neq 0 \end{aligned}$$

ou, geometricamente, *A*, *B* e *C* são não colineares e *AC* e *BD* intersectam-se.

A implementação deste método no sistema *GCLC* permite a demonstração deste resultado em menos de uma milésima de segundo de tempo de execução no processador central (CPU).

Os demonstradores automáticos de teoremas baseados neste método são em geral muito eficientes, provando centenas de teoremas geométricos com tempos de CPU da ordem dos poucos segundos [7,41]. As demonstrações desenvolvem-se através de manipulações algébricas não permitindo o estabelecer de ligações com a geometria que não sejam, no fim da demonstração, a validação, ou não, de uma dada conjetura geométrica.

3.2. Métodos Livres de Coordenadas (Semi-Sintéticos)

Os, assim designados, métodos livres de coordenadas são métodos em que as demonstrações combinam inferências geométricas com manipulações algébricas. As suas implementações são capazes de demonstrar muitos teoremas complexos da geometria de forma eficiente. Exemplos de métodos deste tipo são, por exemplo, o método da área, o método do ângulo pleno, o método para a geometria dos sólidos [8, 11, 12].

Nesta secção vamos focar o *método da área*, um método desenvolvido por Chou et al. [9, 10,12] para um fragmento da geometria Euclidiana.

As várias implementações deste método [12, 28, 36, 48] são capazes de provar, de forma eficiente, centenas de teoremas geométricos complexos. Este método é ainda capaz de gerar demonstrações tradicionais, concisas e fáceis de seguir por um matemático. As demonstrações são formadas por passos de inferências geométricas entrecruzadas com manipulações algébricas. Nem sempre as demonstrações são concisas e fáceis de seguir por um matemático, em alguns casos a complexidade das manipulações algébricas sobrepõe-se às restantes manipulações de carácter geométrico.

A ideia principal do método consiste no expressar das hipóteses do teorema através de um conjunto de "pontos livres", isto é, pontos que são implicitamente quantificados universalmente, e de um conjunto de passos construtivos, sendo que cada um destes passos vai introduzir um novo ponto. No fim a conclusão é expressa como uma igualdade entre dois polinómios em quantidades geométricas próprias do método da área, sendo que, para expressar estas quantidades geométricas, não será necessário usar coordenadas geométricas. A demonstração é então desenvolvida em sentido contrário, por eliminação dos pontos introduzidos utilizando para tal um conjunto de lemas próprios do método. Após a eliminação de todos os pontos introduzidos, a igualdade que expressa a conjetura a demonstrar colapsa numa igualdade entre duas quantidades racionais envolvendo unicamente os pontos livres. Se as expressões em ambos os lados da igualdade são iguais, a conjetura é um teorema, caso contrário, não é um teorema.

O método da área é um procedimento de decisão para um fragmento da geometria Euclidiana do plano. O método lida com problemas colocados como uma sequência de passos geométricos construtivos.

Vamos introduzir este método utilizando para tal o teorema de Ceva (Teorema 3) como exemplo, reproduzindo aqui de novo a figura, mas numa nova configuração feita utilizando o GeoGebra (ver Figura 5). Não se pretende aqui ser exaustivo, introduzindo somente o necessário para poder seguir a construção, o estabelecer da conjetura e o demonstrar da mesma, para este exemplo concreto. Para todos os detalhes ver o artigo [27].

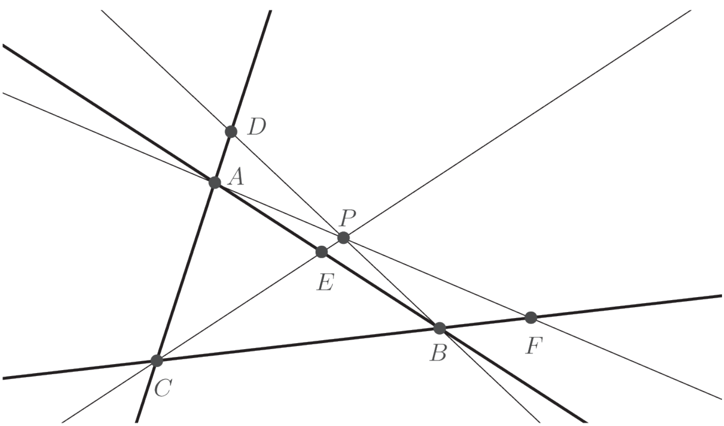

Figura 5. Teorema de Ceva.

A construção O fragmento da geometria Euclidiana determinado pelas entidades geométricas e correspondentes propriedades (ver mais à frente) permite expressar as noções de ponto, linha, círculo, assim como as diferentes relações entre estes elementos.

Para poder expressar uma conjetura no fragmento da geometria referido acima, as construções geométricas têm de ser expressas como uma sucessão de construções elementares, as quais introduzem os pontos livres, assim como os pontos obtidos através dos diferentes passos construtivos. Estes passos construtivos elementares vão constituir as hipóteses das conjeturas que pretendemos demonstrar. É de notar que as condições de não degenerescência que alguns destes passos construtivos introduzem foram omitidas (para mais detalhes, ver [27]).

CE1 construção de um ponto arbitrário (pontos livres);

CE2 construção de um ponto Y, interseção de duas retas UV e PQ.

condição-ndg: $UV \nparallel PQ$; $U \neq V$; $P \neq Q$.

graus de liberdade para Y: 0

No total tem-se cinco construções elementares, sendo que somente quatro é que introduzem novos pontos.

No caso do exemplo que estamos a descrever ter-se-ia: os pontos A, B, C e P são pontos livres (CE1), pontos quantificados universalmente de forma implícita. O ponto D é definido como sendo a interseção das retas BP e AC (CE2). Os pontos E e F são construídos de forma similar (CE2).

Para este problema um conjunto inicial de condições de não degenerescência é dado por $F \neq B$, $D \neq C$ e $E \neq A$. É de notar também que o ponto P não é um ponto completamente arbitrário do plano. Na verdade as condições de não degenerescência acima referidas são um caso particular das condições que estabelecem que o ponto P não pode pertencer a nenhuma das retas paralelas aos lados do triângulo e que passam pelo vértice oposto (ver Figura 4).

Estabelecendo a Conjetura Como já foi dito acima, um dos problemas chave na automação das demonstrações é o controlo da explosão combinatória no espaço das soluções. Este efeito ocorre devido ao número de configurações similares, mas diferentes, que têm de ser analisadas. Por exemplo, dados três pontos *A*, *B* e *C*, quantos triângulos são definidos por estes três pontos? Pode-se argumentar que a resposta é um único triângulo, no entanto, do ponto de vista sintático, e as máquinas de Turing são máquinas de manipulação de símbolos, os triângulos *ABC* e *ACB* são diferentes. Para reduzir a explosão combinatória, mas também para assegurar um raciocínio rigoroso, é necessário estabelecer relações de ordem, tais como *segmentos de reta orientados*, *dois triângulos com a mesma orientação*, etc. Estas relações não são absolutas, se se fala de um sentido positivo é só em contraposição a um sentido negativo. O efeito pretendido é somente o de estabelecer formas normais que nos permitam dizer que, por exemplo, um dado conjunto de três pontos define um, e um só, triângulo.

Para estabelecer e demonstrar conjeturas geométricas, o método da área define um conjunto de *quantidades geométricas*, as quais permitem tratar as relações de incidência em geometria.

Definição 1 (Razão entre Segmentos Orientados Paralelos) *Se os pontos A, B, C e D são colineares, a razão entre segmentos orientados paralelos, denotado* $\frac{\overline{AB}}{\overline{CD}}$ *é a razão entre os comprimentos dos segmentos orientados AB e CD. Se os pontos A, B, C e D são não colineares e verifica-se que AB||CD, existe um paralelogramo ABPQ tal que P, Q, C e D* são colineares e então $\frac{\overline{AB}}{\overline{CD}} = \frac{\overline{QP}}{\overline{CD}}$.

Definição 2 (Área com Sinal) *Dado o triângulo ABC, a área com sinal do triângulo, denotada por* S_{ABC} é a área do triângulo ABC, eventualmente com sinal negativo caso o triângulo ABC tenha a orientação negativa.

Definição 3 (Diferença Pitagórica) *A diferença Pitagórica denotada por* P_{ABC}*, para os pontos A, B e C, é definida por* $P_{ABC} = AB^2 + CB^2 - AC^2$.

A diferença Pitagórica é a generalização de igualdade de Pitágoras para triângulos retângulos, para uma expressão aplicável a um qualquer triângulo. Para um triângulo retângulo tem-se $P_{ABC} = 0$.

Um primeiro conjunto de lemas dá-nos as propriedades algébricas que as quantidades geométricas possuem, manipulações que permitem encarar estas quantidades geométricas como "formas normais" representativas das construções geométricas que queremos manipular. Apresentam-se de seguida algumas dessas propriedades [27, 44]:

- $\frac{\overline{PQ}}{\overline{AB}} = -\frac{\overline{QP}}{\overline{AB}} = \frac{\overline{QP}}{\overline{BA}} = -\frac{\overline{PQ}}{\overline{BA}}$.
- $\frac{\overline{PQ}}{\overline{AB}} = 0$ sse $P=Q$.
- $S_{ABC} = S_{CAB} = S_{BCA} = - S_{ACB} = - S_{BAC} = - S_{CBA}$.
- $P_{AAC} = 0$.
- $P_{ABC} = P_{CBA}$.

Um segundo conjunto de lemas, os assim designados *lemas de eliminação*, relacionam entre si as quantidades geométricas permitindo a eliminação dos pontos introduzidos aquando da construção.

Para demonstrar a conjetura 3 necessitamos somente do primeiro de um total de treze lemas de eliminação.

LE1 Seja M a interseção de duas retas paralelas AB e PQ com $Q \neq P$. Temos então que

$$\frac{\overline{PM}}{\overline{QM}} = S_{PAB} / S_{QAB}\,; \quad \frac{\overline{PM}}{\overline{PQ}} = S_{PAB} / S_{PAQB}\,; \quad \frac{\overline{QM}}{\overline{PQ}} = S_{QAB} / S_{PAQB}\,.$$

Através das quantidades geométricas, das suas propriedades e lemas, o método da área permite expressar muitas das propriedades usuais em geometria (ver Tabela 1).

No exemplo, a conjetura 3 é expressa usando razões entre segmentos orientados paralelos entre si.

Propriedade Geométrica	em termos das Quantidades Geométricas
os pontos A e B são idênticos	$P_{ABA} = 0$
os pontos A, B e C são colineares	$S_{ABC} = 0$
AB é perpendicular a CD	$P_{ABA} \neq 0 \wedge P_{CDC} \neq 0 \wedge P_{ACD} = P_{BCD}$
AB é paralela a CD	$P_{ABA} \neq 0 \wedge P_{CDC} \neq 0 \wedge S_{ACD} = S_{BCD}$
O é o ponto médio de AB	$S_{ABO} = 0 \wedge P_{ABA} \neq 0 \wedge \frac{\overline{AO}}{\overline{AB}} = \frac{1}{2}$
AB tem o mesmo comprimento de CD	$P_{ABA} = P_{CDC}$
A, B, C e D são pontos harmónicos	$S_{ABC} = 0 \wedge S_{ABD} = 0 \wedge P_{BCB} \neq 0 \wedge P_{BDB} \neq 0 \wedge \frac{\overline{AC}}{\overline{CB}} = \frac{\overline{DA}}{\overline{DB}}$
o ângulo ABC tem a mesma amplitude que DEF	$P_{ABA} \neq 0 \wedge P_{ACA} \neq 0 \wedge P_{BCB} \neq 0 \wedge P_{DED} \neq 0 \wedge P_{DFD} \neq 0 \wedge P_{EFE} \neq 0 \wedge S_{ABC} \cdot P_{DEF} = S_{DEF} \cdot P_{ABC}$
A e B pertencem ao mesmo arco de círculo CD	$S_{ACD} \neq 0 \wedge S_{BCD} \neq 0 \wedge S_{CAD} \cdot P_{CBD} = S_{CBD} \cdot P_{CAD}$

Tabela 1. Propriedades Geométricas – Método da área.

A demonstração. A demonstração da conjetura é baseada na eliminação de todos os pontos obtidos por construção, isto é, pretende-se expressar a igualdade a demonstrar em termos dos objetos livres, os quais, relembrando, estão implicitamente quantificados universalmente. Se a igualdade, expressa somente em termos dos pontos livres, é demonstrável, então a conjetura estabelecida inicialmente é um teorema.

Temos então que, para poder provar uma dada conjetura geométrica através do método da área, é necessário:

- Expressar a hipótese do teorema como um conjunto de passos construtivos. Cada passo construtivo introduz um novo ponto.
- A conclusão é expressa em termos de uma igualdade polinomial sobre as quantidades geométricas. A conclusão não contém nenhuma referência a um qualquer sistema de coordenadas.
- A demonstração decorre por eliminação dos pontos anteriormente introduzidos, por ordem inversa, usando para tal os lemas que expressam as propriedades das quantidades geométricas introduzidas.

Adicionalmente é necessário verificar a condições de não degenerescência das construções. Para este exemplo concreto, a demonstração pode decorrer desta forma:

$$
\begin{aligned}
\left(\left(\frac{\overline{AF}}{\overline{FB}} \cdot \frac{\overline{BD}}{\overline{DC}}\right) \cdot \frac{\overline{CE}}{\overline{EA}}\right) &= 1 \quad \text{conjectura} \\
\left(\left(\left(-1 \cdot \frac{\overline{AF}}{\overline{BF}}\right) \cdot \frac{\overline{BD}}{\overline{DC}}\right) \cdot \frac{\overline{CE}}{\overline{EA}}\right) &= 1 \quad \text{simplificação algébrica} \\
\left(-1 \cdot \left(\frac{\overline{AF}}{\overline{BF}} \cdot \left(\frac{\overline{BD}}{\overline{DC}} \cdot \frac{\overline{CE}}{\overline{EA}}\right)\right)\right) &= 1 \quad \text{simplificação algébrica} \\
\left(-1 \cdot \left(\frac{S_{APC}}{S_{BPC}} \cdot \left(\frac{\overline{BD}}{\overline{DC}} \cdot \frac{\overline{CE}}{\overline{EA}}\right)\right)\right) &= 1 \quad \text{o ponto } F \text{ é eliminado (lema LE1)} \\
\left(-1 \cdot \left(\frac{S_{APC}}{S_{BPC}} \cdot \left(\frac{\overline{BD}}{\overline{DC}} \cdot \left(-1 \cdot \frac{\overline{CE}}{\overline{AE}}\right)\right)\right)\right) &= 1 \quad \text{simplificação geométrica} \\
\frac{\left(S_{APC} \cdot \left(\frac{\overline{BD}}{\overline{DC}} \cdot \frac{\overline{CE}}{\overline{AE}}\right)\right)}{S_{BPC}} &= 1 \quad \text{simplificação algébrica} \\
\frac{\left(S_{APC} \cdot \left(\frac{\overline{BD}}{\overline{DC}} \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{S_{BPC}} &= 1 \quad \text{o ponto } E \text{ é eliminado (lema LE1)} \\
\frac{\left(S_{APC} \cdot \left(\left(-1 \cdot \frac{\overline{BD}}{\overline{CD}}\right) \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{(-1 \cdot S_{CPB})} &= 1 \quad \text{simplificação geométrica} \\
\frac{\left(S_{APC} \cdot \frac{\overline{BD}}{\overline{CD}}\right)}{S_{APB}} &= 1 \quad \text{simplificação algébrica} \\
\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{S_{CPA}}\right)}{S_{APB}} &= 1 \quad \text{o ponto } D \text{ é eliminado (lema LE1)} \\
\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{(-1 \cdot S_{APC})}\right)}{(-1 \cdot S_{BPA})} &= 1 \quad \text{simplificação geométrica} \\
1 &= 1 \quad \text{simplificação algébrica}
\end{aligned}
$$

Q.E.D.

O demonstrador automático de teoremas GCLC finalizaria com as condições de não degenerescência que são necessárias verificar para que a conjetura possa ser demonstrada, assim como alguns dados sobre como a demonstração decorreu (alguns detalhes foram omitidos).

```
NDG conditions are:

S BPA =  S CPA  i.e., lines BC  and P A  are not parallel
(construction based assumption) (...)
P  FBF  =  0    i.e., points F  and B  are not identical
(conjecture based assumption) (...)
Number of elimination proof steps:      3
Number of geometric proof steps:        6
Number of algebraic proof steps:       23
Total number of proof steps:           32
Time spent by the prover: 0.000 seconds
```

Este exemplo serve para ilustrar a forma de expressar um problema usando as quantidades geométricas próprias do método da área e como o demonstrar. Este é também um caso exemplar, mesmo que se mostrassem todos os detalhes da demonstração (alguns passos de manipulações algébricas foram omitidos) esta continuava muito fácil de seguir.

Implementação do método O programa GCLC possui uma implementação deste método [26] o qual, de forma eficiente, permite demonstrar centenas de conjeturas geométricas (ver os sistemas GeoThms [43] e TGTP [41]). No entanto nem sempre temos resultados satisfatórios.

Alguns dos pontos menos positivos deste método são:

- As quantidades geométricas definidas não são as usuais. É necessário estudar este método particular para poder desenvolver demonstrações através dele.

- Como é evidente dos passos construtivos elementares introduzidos para expressar as conjeturas que podem ser demonstradas por este método, nem todas as construções geométricas usuais podem ser expressas desta forma.

- Nem sempre as demonstrações são curtas e fáceis de seguir. Em alguns casos as manipulações algébricas envolvidas tornam a demonstração difícil de seguir.

Vejamos um outro exemplo (exemplo 84 [6, pg 145]). Neste caso temos uma formulação do problema "normal", isto é, sem ser necessário usar de forma explícita as quantidades próprias do método da área, e dentro do fragmento da geometria determinado pelo método da área. Pela negativa temos que a demonstração produzida está longe de ser legível.

Teorema 5 *Uma linha que passe pela ponto O, interseção das diagonais de um paralelogramo ABCD intersecta as retas definidas pelos quatro lados do paralelogramo em E, F, G, H. Mostre que* $\overline{EF} = \overline{GH}$ *(ver Figura 6).*

Utilizando a linguagem do sistema GCLC podemos expressar esta conjetura da seguinte forma:

```
% Definição dos Pontos Livres A, B e C
point A 30 30   point B 110 30   point C 90 80
% Construção do Ponto D
line ab A B   line bc B C
parallel cd C ab
parallel ad A bc
intersec D cd ad
% Definição (aleatória) do ponto H (na linha BC)
online H B C
% Construção do Ponto O
line bd B D
line ac A C
intersec O ac bd
% Construção dos Pontos F, G e E
line oh O H
intersec F oh cd
intersec G oh ab
intersec E oh ad
% A conjectura EF = GH
prove { equal { sratio E F G H } { 1 } }
```

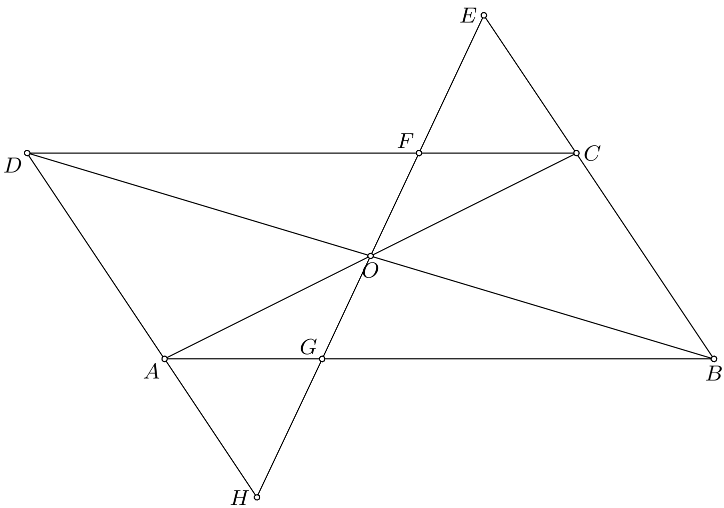

Figura 6. Exemplo 84 de [6].

É de notar que as coordenadas dos pontos *A*, *B* e *C*, os pontos livres da construção, são somente necessárias para a visualização da construção, não sendo usadas na demonstração.

O GCLC consegue demonstrar este teorema em 46,559s de tempo de CPU e em 7903 passos, dos quais 6333 são manipulações algébricas, 1127 manipulações geométricas e 443 aplicações dos lemas de eliminação, o documento contendo a prova tem 44 páginas. Estes números já não são compatíveis com uma demonstração que se possa considerar curta e legível.

Como já foi referido dentro desta categoria de métodos temos ainda o *método do ângulo pleno* [8], também para a geometria plana mas permitindo considerar problemas com que o método da área não lida, e o método para a geometria dos *sólidos* [11], que, como o próprio nome indica, aplica-se à geométrica dos sólidos Euclidianos.

4. Formalização da Geometria

Uma das áreas da ciência da computação muito ativa atualmente, centrando à sua volta o esforço de muitos investigadores, é a área da formalização da matemática (por exemplo, [16, 17, 24, 27, 32, 33, 34, 35, 36], entre outros). De certa forma esta área constitui um completar do círculo. Da máquina de Turing, formalização matemática do conceito de computação, até à utilização dos computadores para ajudar no esforço de formalização das várias áreas da matemática. A geometria não escapa a esse esforço de formalização. Por formalização da geometria entende-se a possibilidade de mecanização das demonstrações, de forma a que as mesmas possam ser executadas de uma forma sintática, desde as hipóteses até à conclusão por uma estrita aplicação das regras de inferência, sem recurso, em nenhuma parte da demonstração, à intuição.

De Euclides a Tarski, passando por Hilbert, a geometria teve um papel central na história da demonstração em matemática. Euclides é considerado como um dos precursores do método axiomático. Nos *Elementos* Euclides [19] estabelece um conjunto de postulados, proposições supostas verdadeiras à evidência e que como tal não passíveis de serem demonstradas. Através unicamente de regras lógicas, ele deduziu todos os resultados que entretanto tinham sido estabelecidos. Os *Elementos* constituíram, num certo sentido, o primeiro sistema formal matemático.

No entanto, estudando com todo o rigor as demonstrações feitas por Euclides, verifica-se que nem sempre as mesmas estão estritamente de acordo com o método axiomático. Em algumas das demonstrações certas etapas, mesmo que parecendo evidentes, não podem ser deduzidas a partir do sistema de axiomas definido. A razão para esta falha deve-se em muitos casos à utilização da intuição, em particular a posição relativa de pontos e retas que são implicitamente admitidas.

Por exemplo, nos *Elementos*, Euclides apresenta uma demonstração do teorema lado-ângulo-lado sobre a igualdade de triângulos [36].

Teorema 6. Se *os dois lados de um triângulo* ΔABC *e o ângulo formado por esses lados são iguais aos lados e ângulo de um outro triângulo* ΔDEF*, então os dois triângulos são iguais.*

A demonstração de Euclides é então: deslocar ΔABC de forma a que o ponto A coincida com o ponto D e a reta AB coincida com a reta DE. O ponto B coincidirá com o ponto E dado que $\overline{AB} = \overline{DE}$. De igual modo a reta AC coincidirá com a reta DF dado que $\angle BAC = \angle EDF$. O ponto C coincidirá com o ponto F dado que $\overline{AC} = \overline{DF}$. A reta BC coincidirá com a recta EF dado que se viu que os quatro pontos que as definem coincidem dois a dois. Finalmente $\overline{BC} = \overline{EF}$ dados que as extremidades dos segmentos coincidem. Temos então que $\angle ACB = \angle DFE$ e $\angle ABC = \angle DEF$.

Q.E.D.

A falha neste raciocínio reside na utilização do termo *deslocar*. Nada nos postulados de Euclides nos diz que podemos utilizar este método de sobreposição.

Em 1899 Hilbert propõe uma nova axiomática com a qual se pretendia um desenvolvimento perfeitamente rigoroso da matemática, no qual a intuição não tivesse nenhum papel a desempenhar no desenvolvimento das demonstrações. Se, para verificar uma demonstração, se pode dispensar de todo a intuição, passa então a ser possível verificar mecanicamente as demonstrações. Estando perante um teorema, por definição decidível, é então possível usar um programa de computador para verificar as demonstrações e fazê-lo de uma forma totalmente formal.

4.1. Formalização do Método da Área

O método da área (ver secção 3.2) foi objeto de formalização no programa *Coq* [53], um sistema genérico de demonstração assistida, que permite expressar asserções matemáticas e verificar mecanicamente as demonstrações dessas asserções.

Utilizando a formalização do método da área [27, 36, 37] todas as propriedades geométricas requeridas pelo método da área foram verificadas formalmente utilizando o programa *Coq*, demonstrando deste modo a correção do sistema e eliminando as eventuais dúvidas sobre a demonstrabilidade dos lemas do método, e, como consequência direta, também sobre as demonstrações efetuadas utilizando o método.

Vejamos como podíamos lidar com o teorema de Ceva (Teorema 3). O código *Coq* faz a importação do módulo referente ao método da área, especifica a conjetura assim como as condições de não degenerescência e finalmente invoca o método area_method para proceder à demonstração.

```
Require Import area_method.

Theorem Ceva :
   forall A B C D E F G P : Point,
   inter_ll D B C A P -> inter_ll E A C B P -> inter_ll F
   A B C P ->
   F <> B -> D <> C -> E <> A ->
   parallel A F F B -> parallel B D D C -> parallel C E E A ->
   (A ** F / F ** B) * (B ** D / D ** C) * (C ** E / E ** A) = 1.
Proof.
   area_method.
Qed.
```

Como o *Coq* é um sistema interativo de demonstração, e não um sistema para a geometria dinâmica, a componente da construção da figura geométrica está omissa. Por outro lado o método area_method é um, assim designado, *tactical*, isto é uma meta-regra de inferência que automatiza a aplicação de regras de inferência. Neste caso o método em questão automatiza a aplicação do método da área, aplicado ao teorema que se pretende demonstrar.

O resultado da demonstração é o seguinte.

```
Area method:
   initialisation...
   elimination...
   elimination of point : F
   we need to show that:
(- (S A C P / S B C P * (B ** D / D ** C) * (C ** E / E **
A)) = 1)
   elimination of point : E
   we need to show that:
(S A C P / S B C P * (B ** D / D ** C) * (S C B P / S A B P) = 1)
   elimination of point : D
   we need to show that:
```

```
(- (S A C P / S B C P * (S B A P / S C A P) * (S C B P / S A
B P)) = 1)
   uniformize areas...
   simplification...
   before field...
```

Sendo que o *Coq* necessitou de 3,64 segundos de CPU para completar a demonstração.

Os sistemas computacionais interativos de demonstração tais como o *Coq* e o *Isabelle* [39, 53] poderão em breve fazer parte da "caixa-de--ferramentas" dos matemáticos, permitindo-lhe validar as suas intuições, as suas demonstrações, através da verificação mecânica, totalmente formal, das mesmas.

5. Ferramentas Inteligentes para a Geometria

Embora a Geometria não tenha sido uma das primeiras áreas a ser tratada pelas primeiras "máquinas de Turing", ela esteve no entanto presente em duas (r)evoluções importantes: nos primórdios da área da "Inteligência Artificial" e na introdução da componente visual.

Apresento de seguida alguns exemplos[4] de sistemas computacionais que exploram a ligação entre o visual e o dedutivo.

O sistema *geometriagon* (polarprof-001-site1.smarterasp.net/geometriagon/) propõe aos seus utilizadores uma série de problemas geométricos (ver a Figura 7).

4 O portal do projeto *Interoperable Interactive Geometry for Europe* (http://i2geo.net/) é um local a explorar para procurar recursos computacionais na área da geometria.

Núm.	Enunciado do problema	Instrumentos	Dif.	Tipologia	Data	DD	Sol
1	Dados três pontos A, B e C, desenhar um círculo de centro em C e raio AB		2.1	construção de uma circunferência de raio dado	2004-08-31	3312	143
524	Livro I Prop. I: Sobre uma linha recta determinada descrever um triângulo equilátero		1.1	de **"Elementos"** de *Euclides*	2006-09-24	2558	223
527	Livro I Prop. X : Dividir em duas partes iguais uma linha recta de um comprimento dado		1.1	de **"Elementos"** de *Euclides*	2006-09-24	2558	99
528	Livro I Prop. XI : De um ponto dado em uma linha recta dada levantar uma perpendicular sobre a mesma recta dada		1.2	de **"Elementos"** de *Euclides*	2006-09-24	2558	90
529	Livro I Prop. XII : Conduzir uma perpendicular sobre uma linha recta dada indefinida de um ponto dado fora dela		1.1	de **"Elementos"** de *Euclides*	2006-09-24	2558	104
539	Livro III Prop. XXX : Dividir ao meio um dado arco.		1.1	de **"Elementos"** de *Euclides*	2006-10-02	2550	102
545	Livro IV Prop. V: Circunscrever un círculo a um dado triângulo.		1.2	de **"Elementos"** de *Euclides*	2006-10-18	2534	120
546	Livro IV Prop. VI: Inscrever um quadrado num círculo dado.		1.1	de **"Elementos"** de *Euclides*	2006-10-18	2534	87
553	Livro IV Prop. XIII: Inscrever um círculo num pentágono dado que seja equilátero e equiângulo.		1.2	de **"Elementos"** de *Euclides*	2006-10-21	2531	115
557	Livro VI Prop. IX: Cortar de um segmento dado uma terça parte.		1.1	de **"Elementos"** de *Euclides*	2006-10-24	2528	71

Figura 7. geometriagon, Lista de Problemas.

Cada um dos problemas propostos tem um conjunto de ferramentas geométricas possíveis de serem utilizadas para a sua resolução, sendo que o sistema tem a capacidade de verificar quando é que uma solução para o problema foi atingida (ver Figura 8).

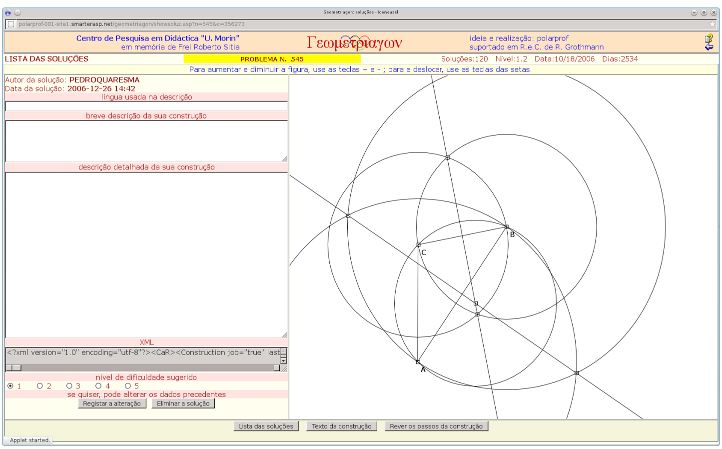

Figura 8. *geometriagon*, Bancada de Trabalho.

Os problemas são classificados por níveis de dificuldade sendo que é possível explorar algumas das construções já efetuadas por outros utilizadores. Este é um sistema para os entusiastas das construções geométricas com régua e compasso, pois são essas as ferramentas (computacionais) que estão disponíveis.

A um nível mais sofisticado estão os programas que associam os sistemas de geometria dinâmica com os sistemas de cálculo algébrico e os demonstradores automáticos de teoremas (consultar en.wikipedia.org/wiki/List_of_interactive_geometry_software para uma lista de sistemas deste tipo) cujas capacidades variam entre a validação das construções que estão a ser efetuadas, à descoberta de propriedades geométricas (por exemplo a equação de um dado lugar geométrico), à demonstração de uma dada conjetura geométrica.

Um sistema muito interessante que faz esta ligação e que além disso tem um acervo muito interessante de demonstrações visuais, por exemplo do Teorema de Pitágoras (ver Figura 9), é o sistema JGEX (www.cs.wichita.edu/~ye/). A lista de exemplos de demonstrações visuais é bastante extensa (ver Figura 10).

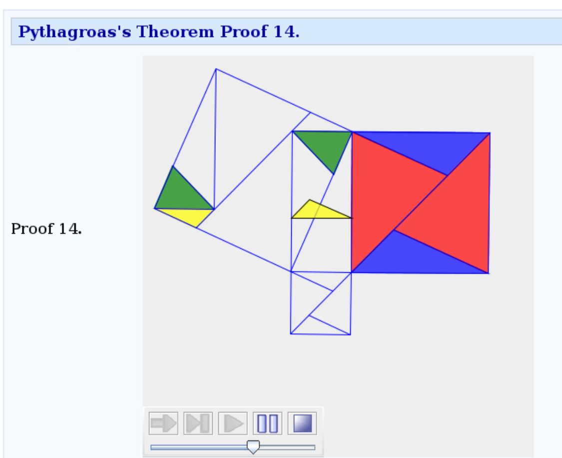

Figura 9. JGEX, Teorema de Pitágoras.

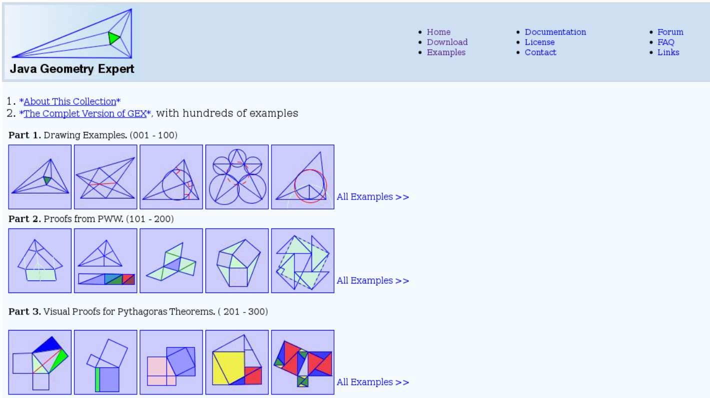

Figura 10. JGEX, Lista de Exemplos.

Este sistema além de ser um sistema de geometria dinâmica incorpora vários demonstradores automáticos de teoremas, os quais podem ser usados para demonstrar conjeturas sobre as construções efetuadas. Além destas componentes o sistema possui a componente das demonstrações visuais dinâmicas a qual permite a utilização das duas outras componentes para criar de forma manual ou (em alguns casos) automática, animações que, usando diferentes efeitos visuais, demonstram visualmente e dinamicamente propriedades geométricas. Por exemplo o sistema possui uma série de demonstrações visuais dinâmicas do Teorema de Pitágoras, na Figura 9 pode-se ver um "instante" da animação (Proof 14), ou acessível através do programa.

Atualmente os sistemas de geometria dinâmica, os demonstradores automáticos de teoremas, a formalização da geometria [16, 17, 37] e a obtenção de conteúdos geométricos de repositórios de informação geométrica [41, 42, 43] constituem recursos computacionais prontos a serem desfrutados pelos seus utilizadores. Recursos que permitem aos seus utilizadores a exploração do conhecimento geométrico.

Não menos importante é a aplicação destas ferramentas geométricas inteligentes em áreas como a educação. Também aqui (r)evoluções importantes estão prestes a acontecer permitindo o desenvolvimento de ambientes de aprendizagem desafiadores das capacidades dos seus utilizadores [45, 49, 50].

Referências Bibliográficas

[1] Philippe Balbiani and Luis del Cerro. Affine geometry of collinearity and conditional term rewriting. In Hubert Comon and Jean-Pierre Jounnaud, editors, Term Rewriting, volume 909 of *Lecture Notes in Computer Science*, pages 196–213. Springer Berlin / Heidelberg, 1995. 10.1007/3-540-59340-3 14.

[2] F. Botana, M. Hohenwarter, P. Janicic, Z. Kovács, I. Petrovic, T. Recio, S. Weitzhofer. Automated Theorem Proving in GeoGebra: Current Achievements, JAR, 55(1):39-59, 2015.

[3] B. Buchberger. An Algorithmic Criterion for the Solvability of a System of Algebraic Equations. In B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*, London Mathematical Society Lecture Notes Series 251, pages 535–545. Cambridge University Press, 1998.

[4] B. Buchberger. Introduction to Gröbner Bases. In B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*, number 251 in London Mathematical Society Lecture Notes Series, pages 3–31. Cambridge University Press, 1998.

[5] Shang-Ching Chou. *Proving and discovering geometry theorems using Wu's method*. PhD thesis, The University of Texas, Austin, 1985.

[6] Shang-Ching Chou. *Mechanical Geometry Theorem Proving*. D. Reidel Publishing Company, Dordrecht, 1987.

[7] Shang-Ching Chou. An Introduction to Wu's Method for Mechanical Theorem Proving in Geometry. *Journal of Automated Reasoning*, 4:237–267, 1988.

[8] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants, II. theorem proving with full-angles. *Journal of Automated Reasoning*, 17:349–370, 1996.

[9] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated production of traditional proofs for constructive geometry theorems. In Moshe Vardi, editor, Proceedings *of the Eighth Annual IEEE Symposium on Logic in Computer Science LICS*, pages 48–56. IEEE Computer Society Press, June 1993.

[10] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. *Machine Proofs in Geometry*. World Scientific, 1994.

[11] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated production of traditional proofs in solid geometry. *Journal of Automated Reasoning*, 14:257–291, 1995.

[12] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants, I. multiple and shortest proof generation. *Journal of Automated Reasoning*, 17:325–347, 1996.

[13] H. Coelho and L. M. Pereira. GEOM: A Prolog geometry theorem prover. Memorias 525, Laboratório Nacional de Engenharia Civil, Ministério de Habitação e Obras Públicas, Portugal, 1979.

[14] H. Coelho and L. M. Pereira. Automated reasoning in geometry theorem proving with Prolog. *Journal of Automated Reasoning*, 2(4):329–390, 1986.

[15] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decomposition: a synopsis. *SIGSAM Bull.*, 10(1):10–12, February 1976.

[16] Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck. Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry. In Dongming Wang Jürgen Richter-Gebert, editor, *Proceedings of Automated Deduction in Geometry (ADG00)*, volume 2061 of *Lecture Notes in Computer Science*, pages 306–324, 2000.

[17] Jean Duprat. Basis of the Euclid's plane geometry. Coq user contribution, 2010. http://coq.inria.fr/pylons/contribs/view/EuclideanGeometry/v8.3.

[18] E. W. Elcock. Representation of knowledge in geometry machine. *Machine Intelligence*, 8:11– 29, 1977.

[19] Euclides. Elements de Euclides dos seis primeiros livros, do undecimo, e duodecimo. Imprensa da Universidade de Coimbra, 1855. Tradução da versão latina de Federico Commandino addicionados e illustrados por Roberto Simson.

[20] Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors. Handbook of Logic in *Artificial Intelligence and Logic Programming*, volume 1. Oxford Science Publications, Oxford, 1993.

[21] H. Gelernter. Realization of a geometry theorem proving machine. In Proceedings of the *International Conference Information Processing*, pages 273–282, Paris, June 15-20, 1959.

[22] Paul C. Gilmore. An examination of the geometry theorem machine. *Artif. Intell.*, 1(3):171–187, 1970.

[23] J.G. Greeno, M. E. Magone, and S. Chaiklin. Theory of constructions and set in problem solving. *Memory and Cognition*, 7(6):445–461, 1979.

[24] Frédérique Guilhot. Formalisation en Coq d'un cours de géométrie pour le lycée. In *Journées Francophones des Langages Applicatifs*, Janvier 2004.

[25] David Hilbert. *Fundamentos da Geometria*. Gradiva, Lisboa, 2003. Edição revista e coordenada por A. J. Franco de Oliveira.

[26] Predrag Janicic. GCLC — A tool for constructive euclidean geometry and more than that. In Andrés Iglesias and Nobuki Takayama, editors, Mathematical Software – ICMS 2006, volume 4151 of *Lecture Notes in Computer Science*, pages 58–73. Springer, 2006.

[27] Predrag Janicic, Julien Narboux, and Pedro Quaresma. The Area Method: a recapitulation. *Journal of Automated Reasoning,* 48(4): 489 – 532, 2012

[28] Predrag Janicic and Pedro Quaresma. System description: GCLCprover + GeoThms. In Ulrich Furbach and Natarajan Shankar, editors, Automated Reasoning, volume 4130 of *Lecture Notes in Computer Science*, pages 145–150. Springer, 2006.

[29] Deepak Kapur. Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4):399–408, 1986.

[30] Kenneth R. Koedinger and John R. Anderson. Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, 14(4):511–550, 1990.

[31] H. Li. Clifford algebra approaches to mechanical geometry theorem proving. In X.-S. Gao and D. Wang, editors, *Mathematics Mechanization and Applications*, pages 205–299, San Diego, CA, 2000. Academic Press.

[32] Nicolas Magaud, Julien Narboux, and Pascal Schreck. Formalizing Desargues' Theorem in Coq using Ranks. In Sung Y. Shin and Sascha Ossowski, editors, SAC, pages 1110-1115. ACM, 2009.

[33] Nicolas Magaud, Julien Narboux, and Pascal Schreck. Formalizing Projective Plane Geometry in Coq. In Thomas Sturm and Christoph Zengler, editors, Automated Deduction in Geometry, volume 6301 of *Lecture Notes in Artifical Intelligence*, pages 141–162. Springer, 2011.

[34] Filip Maric, Ivan Petrovic, Danijela Petrovic, and Predrag Janicic. Formalization and implementation of algebraic methods in geometry. *In First Workshop on CTP Components for Educational Software (THedu'11)*, volume 79 of EPTCS, 2012.

[35] Laura Meikle and Jacques Fleuriot. Formalizing Hilbert's Grundlagen in Isabelle/Isar. In David A. Basin and Burkhart Wolff, editors, Theorem Proving in Higher Order Logics, volume 2758 *of Lecture Notes in Computer Science*, pages 319–334. Springer, 2003.

[36] Julien Narboux. *Formalisation et Automatisation du Raisonnement Géométrique en Coq*. PhD thesis, Université de Paris Sud, 2006.

[37] Julien Narboux. Formalization of the area method. Coq user contribution, 2009. http://dpt-info.u-strasbg.fr/~narboux/area_method.html.

[38] Arthur J. Nevins. Plane geometry theorem proving using forward chaining. *Artificial Intelligence*, 6(1):1-23, 1975.

[39] Lawrence C. Paulson. *The Isabelle Reference Manual*. Computer Laboratory, University of Cambridge, 1998.

[40] Art Quaife. Automated development of Tarski's geometry. *Journal of Automated Reasoning*, 5:97-118, 1989. 10.1007/BF00245024.

[41] Pedro Quaresma. Thousands of Geometric problems for geometric Theorem Provers (TGTP). In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, Automated Deduction in Geometry, volume 6877 of Lecture Notes in Computer Science, pages 169-181. Springer, 2011.

[42] Pedro Quaresma and Yannis Haralambous. Geometry Constructions Recognition by the Use of Semantic Graphs. In Proceedings of RecPad 2012, Coimbra, 2012.

[43] Pedro Quaresma and Predrag Janicic. GeoThms – a Web System for euclidean constructive geometry. *Electronic Notes in Theoretical Computer,* 174(2):35-48, 2007.

[44] Pedro Quaresma and Predrag Janicic. The area method, rigorous proofs of lemmas in Hilbert's style axiom system. Technical Report 2009/006, Centre for Informatics and Systems of the University of Coimbra, 2009.

[45] Pedro Quaresma, Vanda Santos, and Seifeddine Bouallegue. The Web Geometry Laboratory Project. In CICM 2013, volume 7961 of LNAI, pages 364-368. Springer, 2013.

[46] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning* (Vol 1). Elsevier and MIT Press, 2001.

[47] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning* (Vol 2). Elsevier and MIT Press, 2001.

[48] Judit Robu. *Geometry Theorem Proving in the Frame of the Theorema Project*. PhD thesis, Kepler Universitat, Linz, September 2002.

[49] Vanda Santos and Pedro Quaresma. *Adaptative Learning Environment for Geometry*, volume Advances in Learning Processes, chapter 5, pages 71–92. I-Tech Education and Publishing KG, Vienna, Austria, 2010.

[50] Vanda Santos and Pedro Quaresma. Integrating DGSs and GATPs in an adaptative and collaborative blended-learning Web-environment. In *First Workshop on CTP Components for Educational Software (THedu'11)*, volume 79 of EPTCS, 2012.

[51] Ivan Edward Sutherland. Sketchpad: A man-machine graphical communication system. Technical Report UCAM-CL-TR-574, University of Cambridge, Computer Laboratory, September 2003.

[52] Alfred Tarski. A decision method for elementary algebra and geometry. Technical report, RAND Corporation, 1951.

[53] The Coq Development Team. The Coq Proof Assistant, Reference Manual, Version 8.2. TypiCal Project, Lyon, France, 2009.

[54] Jan von Plato. The axioms of constructive geometry. In *Annals of Pure and Applied Logic*, volume 76, pages 169-200, 1995.

[55] D. Wang. Reasoning about geometric problems using an elimination method. In J. Pfalzgraf and D. Wang, editors, *Automated Practical Reasoning*, pages 147-185, New York, 1995. Springer.

[56] Wen-tsü Wu. *Automated Theorem Proving: After 25 Years*, volume 29, chapter On the decision problem and the mechanization of theorem proving in elementary geometry, pages 213-234. American Mathematical Society, 1984.

[57] Wen-tsü Wu. Basic principles of mechanical theorem proving in elementary geometries. *Journal of Automated Reasoning*, 2:221–252, 1986. 10.1007/BF02328447.

[58] Wen- tsü Wu. The characteristic set method and its application. In X.-S. Gao and D. Wang, editors, *Mathematics Mechanization and Applications*, pages 3{41, San Diego, CA, 2000. Academic Press.

[59] Zheng Ye, Shang-Ching Chou, and Xiao-Shan Gao. An introduction to Java geometry expert. In *Proceedings of the 7th international conference on Automated deduction in geometry*, ADG'08, pages 189-195. Springer, 2011.

Parte IV – Testemunhos

8. Memórias do período paleocomputacional[1]

E. R. de Arantes e Oliveira

O primeiro computador a que recorri foi o *Stantec Zebra* instalado em 1959 no LNEC, uma máquina da primeira geração com uma memória de 8k de 35 bits, que representou um salto tecnológico significativo relativamente às duas *IBM* que a precederam. Para dar uma ideia das possibilidades da primeira, posso revelar ter sido faturada à "Hidrotécnica Portuguesa" a importância de 600 escudos pela resolução de um

1 Comunicação apresentada à Academia das Ciências de Lisboa a 8 de Novembro de 2012.

sistema de 34 equações a 34 incógnitas que consumiu 30 minutos de tempo de máquina.

No entanto, foi o *Stantec Zebra* que tornou possível a implementação do primeiro método computacional adequado à análise de pontes suspensas. Concebido por Ferry Borges para a análise da ponte sobre o Tejo, o método em questão idealizava esta como uma estrutura plana constituída por duas torres, uma viga de rigidez, e um cabo solidarizado com a viga por intermédio dos pendurais. A estrutura assim constituída era analisada como hiper-estática, tomando para incógnitas as trações nos pendurais.

A capacidade de um tal computador depressa se revelou francamente baixa para as necessidades. As técnicas de programação eram, por outro lado, excessivamente complicadas. Em 1963, foi instalado um computador da segunda geração, mas só em 1968 passou a dispor-se de um computador da terceira, um *NCR-Elliott 4100*, do qual me recordo com uma mistura de amor e ódio. Se, por um lado, me serviu para a elaboração da tese com que me apresentei ao concurso para especialista do LNEC, fiquei, por outro lado, de tal modo traumatizado que, durante muito tempo, não consegui voltar a tocar num computador.

A máquina estava provida de dois tipos de memória: uma memória central, e uma memória periférica tendo como suporte físico três unidades de fita magnética. O recurso a esta última tinha que ser previsto pelo próprio programador: era de acordo com instruções incluídas no programa, que se transferiam da memória central para as fitas magnéticas, vetores que, oportunamente, seriam de novo chamados à memória central para efeitos de processamento.

Era altamente improvável que os programadores conseguissem acertar logo à primeira: as sucessivas versões dos programas continham em regra muitíssimos "gatos" e só lentamente se convergia para a versão final. Uma vez realizado cada ensaio, corrigiam-se os erros detetados através do exame dos resultados. Só com muita sorte seria possível fazer-se novo ensaio nesse mesmo dia. Considere-se, para mais, que nesses tempos heróicos era necessário marcar hora para aceder ao computador (o único que existia em todo o LNEC).

Recordo com particular horror as pesadas máquinas periféricas utilizadas para perfurar os cartões, e mais tarde as fitas de papel, com que se introduziam e extraiam dados. É justo mencionar, porém, que a tecnologia das fitas perfuradas representou um enorme progresso relativamente à dos cartões. De facto, embora tanto os cartões como as fitas produzissem imenso lixo, o lixo resultante das fitas perfuradas não era totalmente inútil, dado que as fitas já usadas podiam servir, por exemplo, como serpentinas. Foi o que concluí quando, na *Times Square* de Nova Iorque, assisti à entrada do ano de 1966. Ao soar a meia-noite, a multidão dispersou e o chão da praça ficou coberto das fitas perfuradas de várias cores que haviam sido lançadas das janelas dos arranha-céus.

Se a tecnologia dos computadores utilizados nos anos 60 podia ser considerada assaz primitiva, a da sua utilização na Engenharia não o era menos. Nomeadamente, tendo os computadores trazido consigo a possibilidade de automatizar o cálculo, parecia inevitável que tudo gravitasse à volta do conceito de "biblioteca de programas".

Ora foi justamente nessa década de 60, mais precisamente em 1963, que Charles Miller, presidente do Departamento de Engenharia Civil do *Massachusetts Institute of Technology – MIT*, publicou um notável artigo que alertava a comunidade dos engenheiros para o facto de os problemas da engenharia estarem a ser forçados a conformar-se, na sua formulação, com as bibliotecas de programas disponíveis. Precisamente, a metodologia da engenharia estava a ser sacrificada, no que tinha de essencial, aos novos instrumentos de que se dispunha para os tratar.

Segundo Miller, tornava-se cada vez mais evidente que as dispendiosas bibliotecas de programas com que os grandes "centros de cálculo" – outro conceito em voga – procuravam dotar-se representavam investimentos de tal maneira importantes que constituíam, uma vez realizados, verdadeiros coletes-de-forças que impediam o progresso da própria Engenharia.

A deficiente comunicação homem-máquina, com a sua base exclusivamente numérica, era outra das causas que tornavam a situação verdadeiramente grave. Um dos aspetos mais desagradáveis da utilização dos computadores estava por exemplo associado à penosa rotina de preencher as extensas folhas dos dados numéricos exigidos para a definição dos problemas.

Também os resultados apareciam, depois das muitas horas de cálculo impostas pela lentidão das máquinas, sob a forma de tabelas de milhares de números que enchiam longas tiras de papel dobradas em ziguezague. A síntese dos resultados era pois extremamente difícil de realizar. Limitada por atividades deste tipo, a criatividade dos engenheiros não podia deixar de ser altamente prejudicada.

Miller chamou lucidamente a atenção para o facto de o engenheiro ser sobretudo um decisor e para que uma das características do processo da tomada de decisões em engenharia se basear em informações imprecisas e incompletas ou, até, em fatores não quantificáveis e não exclusivamente de natureza técnica. Por outro lado, sem ser tão elevada como, por exemplo na Medicina, a urgência posta na resolução dos problemas é em geral muito maior para os engenheiros que para os cientistas puros. O que se esperava justamente dos computadores era que permitissem satisfazer a necessidade de tomar decisões típicas da Engenharia com um grau de urgência apropriado.

Em 1963, o que se tornara evidente era que, justamente porque os engenheiros não tinham sido aliviados, como se esperava, das tarefas rotineiras ligadas com o processamento dos dados e resultados, a parte do tempo dedicada ao processo da decisão não pudera tomar, no contexto da atividade global, uma posição tão importante como seria desejável. De uma maneira geral e, apesar das expetativas criadas, as vantagens devidas à utilização dos computadores eram pois muito mais reduzidas do que inicialmente se imaginara.

Miller concluiu que os computadores só se tornariam determinantes para o progresso da Engenharia se: i) o acesso às máquinas fosse consideravelmente facilitado; ii) a comunicação homem-máquina se tornasse tão eficiente como a comunicação homem-homem.

A primeira destas condições está hoje largamente satisfeita. Quanto à segunda, deram-se passos importantes para a sua satisfação. Progrediu-se, nomeadamente, na automatização de uma série de processos que antes careciam de intervenção humana. Não é de admirar, pois, que o estudo destes processos e a sua modelação – um dos objetivos da chamada “inteligência artificial” – se tenha tornado tão importante para o progresso da Engenharia.

9. Ciências da computação de Alan Turing: Uma viagem pessoal[1]

José Manuel Valença

O moderno cientista da computação, interessado na *correção* dos seus artefatos, navega num terreno construído sobre as respostas a algumas questões fundamentais. Primeiro ele/ela pode perguntar "O que é

1 Texto originalmente escrito em Inglês e traduzido pelo editor.

demonstrável, refutável ou decidível?", imediatamente seguido de "O que é inconsistente e o que pode ser validado?".

Mais ainda, é seguro dizer que, quando comparado com o âmbito do artefacto, ele/ela terá sobretudo que basear-se em respostas a outra questão "o que é que se pode aprender?". Finalmente, quando o interesse se estende à *segurança* do artefacto e à *privacidade* do utilizador, o cenário é complicado por questões adicionais, nomeadamente "O que é aleatório? O que é obscuro e à prova de fuga, no sentido da teoria da informação?".

Através de toda a paisagem da Ciência e Engenharia da Computação (CEC) podem detetar-se as pegadas de Turing, algumas desmaiadas, mas outras muito claras e incisivas. Nesta breve monografia tentarei delinear algumas dessas pegadas, escolhendo aquelas que eu acredito serem as mais significativas para a minha visão da CEC.

A questão que mais interessou Turing não foi nenhuma das anteriores mas antes

> O que é computável e, de entre essas 'coisas computáveis', o que é fazível?

É bem conhecido o papel de Turing (Turing 1936; Soare 2016) na construção de uma resposta pelo menos à primeira parte da questão. A fazibilidade das computações veio mais tarde e é o domínio da *Teoria da Complexidade*; a sua importância e o papel de Turing foram reconhecidos há algum tempo (Hartmanis 1994).

Algumas décadas antes, o início do Séc. XX viu o florescimento de novos fundamentos para o pensamento matemático: a axiomatização dos conjuntos de Zermelo-Fraenkel, construtivismo & intuicionismo e os problemas de decisão de Hilbert são alguns dos programas[2] desta era; para além do seu impacto na matemática como um todo, estes programas tiveram um efeito direto nos fundamentos da CEC; de facto, podemos dizer que estes "são" três quartos dos fundamentos da CEC, o restante quarto sendo, obviamente, a computabilidade.

2 Por "programa matemático" entendo uma linha de atividade que vai para além de estabelecer definições e teoremas e também inclui conjeturas, tentativas e erros (sobretudo estes) contribuindo para um conjunto coerente de crenças matemáticas.

No contexto da CEC, a linguagem dos quantificadores (LQ) tem um papel central. Em geral, a LQ é altamente relevante para qualquer programa matemático: como lógica de primeira ordem, segunda ordem ou ordem superior, em formalização clássica ou intuicionista, LQ é sempre uma candidata a linguagem matemática "standard".

Tome-se, por exemplo, ZFC, i.e. a axiomatização dos conjuntos de Zermelo-Fraenkel incluindo o axioma da escolha. ZFC é construída sobre a lógica de primeira ordem (LPO) e estende a Aritmética de Peano (AP). Portanto ZFC fornece um enquadramento muito genérico onde quer os objetos ("dados") quer as entidades de ordem superior (i.e. relações e algoritmos) da CEC podem ser formalizados.

Considere-se também o *Entscheidungsproblem* de Hilbert, o qual (em linguagem moderna) *procura construir um mecanismo universal capaz de transformar, com recursos finitos e incerteza negligenciável, uma frase f de LPO arbitrária numa resposta correta Sim/Não à questão "f é demonstrável em LPO?"*. Conjuntamente, estes dois programas fornecem a base inicial na discussão sobre "o que é demonstrável, refutável ou decidível".

A fazibilidade da indagação de Hilbert depende crucialmente do significado pretendido para o dito "mecanismo universal". Turing, ele próprio (Turing 1936), deu uma primeira resposta negativa: tomando "mecanismo universal" como sendo as suas "máquinas-a", equivalentes aos *procedimentos efetivos* de Gödel ou às *funções recursivas parciais* de Kleene, ele provou a existência de um máquina-a universal; a partir daí, e usando um argumento de diagonalização simples, ele construiu depois uma máquina-a cuja "propriedade de paragem" não pode ser decidida por nenhuma outra máquina-a.

Alguns anos antes, Gödel (Gödel 1931, Davis 2006) tinha demonstrado que nenhum sistema consistente de axiomas cujos teoremas possam ser listados por um procedimento efetivo (i.e. uma *axiomatização efectiva*) é capaz de demonstrar todas as verdades da aritmética dos números naturais; assim, Gödel não só refuta o programa de Hilbert, mas também nega a consistência de qualquer sistema formal efetivamente axiomatizado que fosse capaz de provar todos os teoremas da aritmética dos números naturais (uma hipotética extensão de ZFC, por exemplo).

No contexto dos Fundamentos da Matemática, estes são dois resultados negativos poderosos, ambos refutando o programa de Hilbert. No entanto, a CEC precisa, se não exatamente do programa de Hilbert, de algo muito semelhante: de facto, é essencial que se confie na correção dos artefactos da CEC (i.e. que se confie que eles fazem o que é suposto fazerem). Tradicionalmente, a confiança nos artefactos da CEC era procurada sobretudo através de testes; no entanto, recentemente, as provas formais estão a substituir os testes e a tornar-se o veículo preferencial para este propósito.

Com o construtivismo (Troelstra and van Dalen 1998), nomeadamente no contexto das *teorias de tipos construtivas* (Martin-Löf 1998), a correção dos artefactos da CEC tornou-se uma realidade: *assistentes de prova* como Isabelle, Coq e outros tornaram-se a norma em grandes projetos de CEC e a prova formal tornou-se um componente standard na Indústria do Software.

O papel de Turing aqui está bem documentado (Constable 2012). O aspeto central concerne a semântica da lógica intuicionista (LI), chamada de *semântica de Brower-Heyting-Kolmogorov (BHK)* mas mais conhecida na comunidade CEC como o *isomorfismo/analogia de Curry-Howard*. Essencialmente, este é um método que converte provas em programas e isto tem uma enorme importância, não só no nível fundamental, mas também no nível operacional. Primeiro, porque estabelece uma conexão direta entre "demonstrar" e "computar"[3]. Segundo, porque providencia uma forma consistente e rápida de assegurar construções computacionais corretas.

Nas últimas décadas podemos notar que "inconsistência", "validade" e "satisfazibilidade" se tornaram palavras familiares no vocabulário da CEC; de facto, elas estão na génese das ferramentas mais bem sucedidas ao dispor dos chamados *métodos formais* no desenvolvimento de software: os resolventes SAT, os resolventes SMT ("satisfazibilidade modulo teorias") e os "verificadores de modelos". Sistemas dinâmicos modernos muito grandes como controladores de tráfego aéreo, aviónica e dinâmica espacial altamente complexa, e, sobretudo, a microeletrónica altamente complexa que domina o nosso mundo, todos dependem destas ferramentas.

3 Isto teria agradado a Hilbert e Turing.

É claro, a *Teoria de Modelos*, tal como é usada hoje, é sobretudo uma criação de Gödel, Tarski (sobretudo) e Kripke; enquanto a *Teoria de Modelos "Finitos"* (TMF), da qual a maioria das ferramentas acima deriva, tem uma conexão muito mais forte com Turing. Um resultado standard de TMF (Libkin 2004) é o teorema de Trakhtenbrot que resumidamente diz que a validade de LPO na classe de modelos finitos é indecidível. Isto é crucial para a correção de resolventes SAT/SMT e este resultado vem essencialmente de Turing: a saber, o resultado é demonstrado mostrando que a classe de frases válidas em algum modelo finito não pode ser recursivamente enumerada.

O papel de Turing em Bletchley Park e na quebra do Código Enigma é evidentemente bem documentado; pelo menos tem sido objeto de vários livros e filmes de popularização. No entanto, afastando-nos da ficção, sabemos agora que, em meados dos anos 30, Jerzy Rozycki, Henryk Zygalski e Marian Rejewski usaram grupos simétricos[4] para cripto-analisar efetivamente o Código Enigma e propor a máquina (a "bomba") com a qual um ataque poderia ser levado a cabo. Também sabemos que Turing conhecia o seu trabalho e a "bomba" construída sob sua proposta foi um melhoramento significativo do design original.

O que é menos conhecido e mais interessante é o papel de Turing, não em criptografias antigas, mas nas dos dias modernos. A Criptografia Moderna requer *provas de segurança*[5] e, em consequência, existe investigação bem estabelecida no uso de assistentes de prova como fonte de confiança para esquemas e protocolos criptográficos. Também a *privacidade individual* se tornou uma preocupação criptográfica por fim tão relevante como a própria segurança do artefacto, e aqui é onde as máquinas-a de Turing se tornam relevantes.

A segurança demonstrável standard usa jogos para emular um esquema ou protocolo, e depois usa conceitos como "indistinguibilidade" ou "indiferenciabilidade" para comparar o comportamento do jogo como o de alguma forma de comportamento aleatório. Usualmente, há aqui um problema com uma mal definida noção de aleatoriedade. O uso da

4 Os serviços secretos britânicos, na altura, não acreditavam na matemática e preferiam a linguística.

5 Por exemplo, o corrente *NIST Post-Quantum Cryptography Standardization Process* requer, em cada submissão, uma prova completa de segurança PQ.

teoria completa da *aleatoriedade computacional* (Nies 2009, Downey e Hirschfeldt 2010) tornou-se um requisito moderno; ora tal teoria é um descendente direto das máquinas-a de Turing. De facto, sabemos que podemos modelar segurança e privacidade "à prova de fuga" com base, não só em aleatoriedade, mas também em "aleatoriedade indireta", os chamados oráculos "baixos" ou "K-triviais", os quais, de novo, são definidos com máquinas-a de Turing.

Poderíamos continuar nesta linha, citando problema atrás de problema, método atrás de método, até mesmo disciplinas inteiras de CEC, nos quais Turing deixou a sua inegável pegada. A sua importância não pode ser ignorada ou minimizada; graças a iniciativas como a deste volume, esperamos que ele nunca seja esquecido.

Referências

Constable, R. L. (2012). "On Building Constructive Formal Theories of Computation. Noting the Roles of Turing, Church, and Brouwer". In: 27th Annual IEEE Symposium on Logic in Computer Science, pp. 2–8. doi: 10.1109/LICS.2012.9.

Davis, Martin (2006). "The Incompleteness Theorem". In: Notices of the AMS 53.4, pp. 414–418.

Downey, R.G. and D.R. Hirschfeldt (2010). Algorithmic Randomness and Complexity. Theory and Applications of Computability. Springer New York. isbn:9780387684413. url: https://books.google.pt/books?id=FwIKhn4RYzYC.

Gödel, Kurt (1931). "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I." In: Kurt Gödel Collected works, Vol. I. (1986). Ed. by Solomon Feferman. ISBN 978-0195147209, pp. 144-195.

Hartmanis, Juris (1994). "Turing Award lecture: on computational complexity and the nature of computer science." In: Communications of the ACM.

Libkin, Leonid (2004). Elements of Finite Model Theory. Texts in Theoretical Computer Science. Springer-Verlag.

Martin-Löf, Per (1998). "An Intuitionistic Theory of Types". In: Twenty-Five Years of Construtive Type THeory. Ed. by Giovanni Sambin and Jan Smith.

Oxford Science Publications. Clarendon Press, Oxford, pp. 127-172.

Nies, Andre (2009). Computability and Randomness. Oxford University Press, Inc.

Soare, Robert Irving (2013). “Interactive Computing and Relativized Computability”. In: Computability: Turing, Godel, Church, and Beyond. Ed. By Jack Copeland, Carl Posy, and Oron Shagrir. Cambridge Massachusetts, London: The MIT Press, pp. 203–260.

—(2016). Turing Computability: Theory and Applications. Theory and Applications of Computability. Springer.

Troelstra, A.S. and D van Dalen (1998). Construtivism in Mathematics: an Introduction. Vol. 1. Studies in Logic and the Foundations of Mathematics 121. North-Holland.

Turing, AM (1936). “On computable numbers, with an application to the Entscheidungsproblem”. In: Proceedings of the London Mathematical Society 38.1931, pp. 173–198. url: http://classes.soe.ucsc.edu/cmps210/Winter11/Papers/turing-1936.pdf.

"If a machine is expected to be infallible,
it cannot also be intelligent."

Alan Turing, 1947

UMinho Editora

Universidade do Minho

ISBN 978-989-8974-02-0

9 789898 974020 >