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 máqui-

nas 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)