

# Computação Além das Máquinas de Turing

Procurando métodos apropriados para modelar a computação e o pensamento humano\*

Peter Wegner e Dina Goldin

Alan Turing foi um matemático brilhante que mostrou que os computadores não podem provar asserções matemáticas completamente, estendendo a prova de Gödel que a lógica não pode modelar a verdade matemática completamente. Esta conexão entre computadores e matemática foi usada mais tarde para desenvolver um fundamento matemático para a ciência da computação, comparável às bases matemáticas para a física e outras ciências [1].

Esta coluna mostra que as máquinas de Turing não são adequadas como uma base universal para a resolução de problemas computacionais e que a ciência da computação é uma disciplina fundamentalmente não-matemática. Embora a **interação**<sup>1</sup> não seja a única maneira de estender a computação além das máquinas de Turing, nós mostramos que Turing, Milner e outros usam a interação para este propósito.

Nascido em 1912, Turing foi aceito pela Universidade de *Cambridge* em 1930 para estudar matemática. Tornou-se um membro do *Kings College* em 1934 aos 22 anos de idade ao completar uma dissertação que estendia os modelos de teoria dos grupos de Von Neumann. Seu artigo de 1936, *On Computable Numbers with an Application to the Entscheidungsproblem*<sup>2</sup>, provou que a matemática não pode ser completamente modelada por computadores. No início da década de 1940 ele desenvolveu um modelo de computador do código cifrado alemão

\*Traduzido por Marcelo Cezar Pinto a partir do original publicado em COMMUNICATIONS OF THE ACM, Vol. 46, No. 4, pág. 100-102, Abril/2003.

<sup>1</sup>Grifo do tradutor.

<sup>2</sup>Números computáveis com uma aplicação para o Problema de Decisão (N.T.)

que ajudou os Aliados a vencer a Segunda Guerra Mundial. No final desta década ele desenvolveu modelos computacionais de inteligência artificial, xadrez e da mente humana, sugerindo que os computadores poderiam modelar completamente o pensamento humano e que jogariam xadrez melhor do que os humanos antes do final do século.

Embora o artigo de 1936 dissesse respeito a inabilidade de máquinas de Turing para resolver problemas matemáticos, estas máquinas foram adotadas pelos cientistas teóricos da computação como um modo de resolver todos os problemas de computação na década de 1960. Aqui nós examinamos a evolução histórica do modelo de Turing da fragilidade matemática na década de 1930 para o poder computacional na década de 1960 e, então, para a deficiência computacional na década de 1990 à medida que o aumento da aplicação da computação expandiu nossa noção de “problema computacional” e revelou as limitações das máquinas de Turing para resolver problemas.

## Hilbert, Gödel e Church

Em 1900, Hilbert propôs que a lógica poderia provar completamente a verdade ou falsidade de asserções matemáticas e listou 25 asserções que os matemáticos deveriam tentar provar. *Principia Mathematica* de Russell e Whitehead aceitou o postulado de Hilbert e estabeleceu a lógica matemática como um modelo universal de demonstração matemática. O insucesso deles para realizar este objetivo levou a prova de Gödel em 1931 de que a lógica não pode demonstrar todos os teoremas matemáticos [3]. Gödel mostrou que o *Entscheidungsproblem* (“problema de decisão”) era, em princípio, insolúvel pela lógica e isto

levou muitos matemáticos a trabalhar na explicação da teoria e filosofia da insolubilidade matemática em termos de lógica ou outros modelos matemáticos.

As idéias de Gödel foram levadas em conta por Church, que provou em 1935 que o *Entscheidungsproblem* não poderia ser resolvido por  $\lambda$ -cálculo<sup>3</sup>. Por outro lado, Turing mostrou por computadores que o *Entscheidungsproblem* não poderia ser resolvido porque o “Problema da Parada” das máquinas de Turing era insolúvel. O resultado de Turing foi aceito por Gödel e Church como um argumento de insolubilidade melhor e mais simples. Turing foi convidado para trabalhar em Princeton em 1937 com Church no que foi chamado posteriormente de Tese de Church-Turing<sup>4</sup>. Esta tese iguala lógica,  $\lambda$ -cálculo, máquinas de Turing e computação de funções efetivas como mecanismos equivalentes de solução de problemas. Esta tese foi reinterpretada mais tarde como um mecanismo uniforme completo para resolver todos os problemas computacionais.

Turing sugeriu em seu artigo de 1936 que as máquinas de Turing (as quais ele chamou máquinas automáticas ou a-máquinas) não poderiam produzir um modelo completo para todas as formas de computação assim como não produziram um modelo para todas as formas de matemática. Ele definiu c-máquinas (máquinas de escolha) como um modelo alternativo de computação, que adicionava a escolha interativa como uma forma de computação. Mais tarde ele também definiu u-máquinas (máquinas desorganizadas) como outra alternativa que modelava o cérebro. Mas Turing não as formalizou e, uma década após sua morte prematura, elas foram descartadas na década de 1960 como desnecessárias porque foi assumido que modelos de máquina de Turing poderiam descrever completamente todas as formas de computação.

Apesar desta interpretação limitada da Tese de Church-Turing contradizer a afirmação de Turing de que estas máquinas só poderiam formalizar a solução de problemas algorítmicos, ela foi aceita pela comunidade de Ciência da Computação na década de 1960

<sup>3</sup>cálculo lambda (N.T.)

<sup>4</sup>A melhor tradução seria Hipótese de Church-Turing, embora seja usual chamá-la de Tese de Church-Turing ou somente de Tese de Church (N.T.)

e se tornou um dogma<sup>5</sup> da teoria da computação. A ciência da computação foi moldada por um modelo matemático de máquinas de Turing cujo modelo científico encontra paralelos na física, química e biologia, proporcionando uma teoria da computação aceitável porém frágil.

## Dos Algoritmos para a Interação

Depois de seu início com o artigo pioneiro de Turing em 1936, a ciência da computação surgiu como uma disciplina desenvolvida na década de 1960 quando universidades de todo o país começaram a oferecê-la como um curso de graduação. Em 1968 havia um consenso geral sobre o que deveria ser ensinado como parte deste novo curso, que foi enunciado no documento *Curriculum '68* da ACM [1]. O novo curso de ciência da computação via a computação como processamento de informação, uma transformação da entrada em saída — onde a entrada é completamente definida antes do início da computação e a saída mostra uma solução para o problema em mãos. Tais transformações mecânicas eram há muito conhecidas na matemática como algoritmos. O modelo de computação adotado pela ciência da computação é, portanto, referida como **algorítmica**.

A área da computação foi muito expandida desde a década de 1960 e tem sido reconhecido cada vez mais que a inteligência artificial, os gráficos e a Internet não podem ser expressas por máquinas de Turing. Em cada caso a interação entre o programa e o mundo (ambiente) que ocorre durante a computação tem um papel fundamental que não pode ser substituído por nenhum conjunto de entradas determinado antes da computação. No caso da inteligência artificial, a interação pode ser vista como um pré-requisito para o comportamento de sistemas inteligentes como afirmou Brooks [2]:

*Sistemas computacionais reais não são agentes racionais que recebem entradas, e computam logicamente e produzem saídas...*

<sup>5</sup>No dicionário Michaelis: 1 ponto fundamental indiscutível de uma doutrina religiosa e de qualquer doutrina ou sistema. 2 doutrina. 3 máxima, axioma, proposição apresentada como indiscutível e incontestável, preceito. (N.T.)

*É difícil distinguir o que é inteligência e o que é interação ambiental. Em um certo sentido, não importa realmente saber qual é qual pois todos os sistemas inteligentes devem estar em algum mundo se eles pretendem ser entidades úteis.*

O cientista da computação britânico Robin Milner desenvolveu uma nova estrutura conceitual para modelos de computação baseada em Cálculo para Sistemas Comunicantes (CCS, ou seja, *Calculus for Communicating Systems*) e mais tarde em  $\pi$ -cálculo<sup>6</sup>. Em sua palestra pelo Prêmio Turing<sup>7</sup>, “Elementos de Interação” [5], Milner afirma que os modelos estabelecidos de computação são insuficientes:

*No decorrer da década de 1970 eu me convenci de que a teoria da concorrência e interação requer uma nova estrutura conceitual, não apenas um refinamento do que achamos natural para a computação (algorítmica) seqüencial.*

## Além das máquinas de Turing

A palestra de Milner pelo Prêmio Turing em 1991 apresentou modelos de interação como complementares a computação caixa preta<sup>8</sup> das máquinas de Turing. Contudo, ele evitou questionar se a computação do Cálculo para Sistemas Comunicantes e do  $\pi$ -cálculo iam além das máquinas de Turing e algoritmos. Máquinas de Turing tem sido aceitas como um paradigma fundamental de computação completa e era prematuro desafiar abertamente esta visão no final da década de 1970 e início da década de 1980. Nas últimas duas décadas a tecnologia da computação foi deslocada dos *mainframes* e estações para redes e dispositivos sem fio, com o deslocamento correspondente em aplicações numéricas e de processamento de dados para sistemas embarcados (ou embutidos) e interfaces gráficas de usuário. Nós acreditamos que já não é mais prematuro incluir a interação como parte da computação. Uma mudança de paradigma

<sup>6</sup>cálculo pi (N.T.)

<sup>7</sup>Turing Award lecture (N.T.)

<sup>8</sup>No original, *closed-box* (N.T.)

é necessária em nossa noção de solução de problemas computacionais de forma que ela possa fornecer um modelo completo para os serviços dos sistemas computacionais e agentes de *software* atuais.

O modelo de máquinas interativas como uma extensão das máquinas de Turing foi desenvolvido no final da década de 1990 [10]; a estrutura teórica foi aperfeiçoada [4]. Van Leeuwen, um especialista holandês em teoria da computação, escreveu um artigo estendendo os computadores para além das máquinas de Turing [9], o qual citou estes modelos de interação recentes admitindo que:

*O paradigma clássico de Turing pode não ser mais completamente apropriado para capturar todas as características da computação atual.*

Nosso conceito de modelos interativos foi questionado porque originalmente falhamos ao prover uma estrutura teórica comparável a das máquinas de Turing. Contudo, modelos completos de computação tem sido frequentemente desenvolvidos sem fundamentos teóricos ou modelos matemáticos. Até mesmo Turing apresentou c-máquinas [7] e u-máquinas [8] sem uma base formal.

Embora a matemática tenha sido adotada como um fim para modelar computadores na década de 1960 por analogia com modelos da física, Gödel mostrou em 1931 que a lógica não pode modelar a matemática [3] e Turing mostrou que nem a lógica nem os algoritmos podem modelar completamente a computação e o pensamento humano. Em adição à interação, outras maneiras de se estender a computação para além das máquinas de Turing foram consideradas, como a computação com números reais [6]. Contudo, a afirmação de que toda computação pode ser algorítmicamente especificada é ainda amplamente aceita. Máquinas interativas foram criticadas como uma mudança de paradigma Kuhiano. Mas Gödel, Church, Turing e mais recentemente Milner, Wegner e Van Leeuwen argumentaram que este não é o caso.

## Referências

- [1] Association for Computing Machinery. Curriculum '68: Recommendations for academic programs in computer science. In *ACM Curricula Recommendations for Computer Science*. ACM, NY, 1968.
- [2] Brooks, R.A. *Intelligence Without Reason*. MIT AI Lab Technical Report No. 1293, 1991.
- [3] Gödel, K. On formally undecidable propositions of principia mathematica and related systems. *Monatshefte für Mathematik und Physik*, 38, 1931 (in German); English translation in M. Davis, Ed., *The Undecidable*. Raven Press, 1965.
- [4] Goldin, D., Smolka, S. and Wegner, P. Turing machines, transition systems, and interaction. In *Proceedings of the 8th International Workshop on Expressiveness in Concurrency*, Aarlborg, Denmark, August 2001.
- [5] Milner, R. Elements of interaction. (Turing Award lecture). *Commun. ACM* 36, 1 (Jan. 1993).
- [6] Siegelmann, H. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhauser, 1999.
- [7] Turing, A. On computable numbers with an application to the Entscheidungsproblem. In *Proceedings of the London Math Society* 2, 42, 1936.
- [8] Turing, A. Intelligent Machinery. In D.C. Ince, Ed., *Mechanical Intelligence*, North-Holland, 1992.
- [9] van Leeuwen, J. and Wiedermann, J. The Turing machine paradigm in contemporary computing. In B. Enquist and W. Schmidt, Eds., *Mathematics Unlimited — 2001 and Beyond*. LNCS, Springer-Verlag, 2000.
- [10] Wegner, P. Why interaction is more powerful than algorithms. *Commun. ACM* 40, 5 (May 1997).

---

PETER WEGNER (pw@cs.brown.edu) é Professor Emérito no Departamento de Ciência da Computação da Universidade Brown, RI.

DINA GOLDIN (dqg@cse.uconn.edu) é Professora Assistente no Departamento de Ciência e Engenharia da Computação da Universidade de Connecticut.

---