Turing e a normalização

José Carlos Espírito Santo
Centro de Matemática
Universidade do Minho
4715-057 Braga, Portugal
e-mail: jes@math.uminho.pt

Resumo: Algumas contribuições de Turing em Lógica dizem respeito à Teoria de Tipos e foram desenvolvidas durante a Segunda Guerra Mundial. Um dos resultados desse trabalho é uma demonstração do teorema da normalização para tipos simples, que permaneceu ignorada durante 40 anos. O resultado de Turing é, portanto, um dos episódios da complicada história da (re)descoberta da normalização.

Abstract: The part of Turing's work in Logic concerning Type Theory was developed during Second World War. He obtained a proof of the normalization theorem for simple types, but his result was ignored for 40 years. Turing's result is therefore one of the episodes in the complex history of (re)discovery of normalization.

palavras-chave: Normalização; Teoria de Tipos; Cálculo lambda.

keywords: Normalization; Type Theory, Lambda-calculus.

1 Introdução

O primeiro e principal contributo de Turing em Lógica é o seu artigo de 1936 [27] sobre computabilidade e o *Entscheidungproblem*, a que se seguiram dois artigos sobre o Cálculo λ e as funções recursivas [28, 29]. O interesse de Turing pela Lógica manteve-se por alguns anos: na sua tese de doutoramento de 1938 estudou as consequências dos Teoremas da Incompletude de Gödel [30]; e, de 1941 a 1945, investigou a Teoria de Tipos, tendo publicado três artigos sobre o assunto [19, 31, 32].

Num manuscrito sobre Teoria de Tipos datável de 1941 ou 1942, Turing produziu uma das primeiras demonstrações de um teorema de normalização. A normalização tornou-se um tópico central na Teoria da Demonstração a partir dos anos 1960, mas a demonstração de Turing, depositada na biblioteca do King's College em Cambridge, e entretanto re-descoberta, esperou quase 40 anos para ser publicada [7, 8].

Neste artigo, a demonstração de Turing é explicada e posta lado a lado com os mais recentes desenvolvimentos, que aperfeiçoaram uma prova da

chamada normalização forte. Para tal, faz-se um resumo do trabalho de Turing em Teoria de Tipos, precedido de um mínimo de detalhes técnicos sobre o Cálculo λ e a Teoria de Tipos, para que o leitor interessado, mas não familiarizado com estas áreas, possa fazer sentido do contexto em que Turing trabalhou e seguir as demonstrações apresentadas. Finalmente, a complicada história da (re)descoberta da normalização é contada e o lugar da demonstração de Turing nessa história determinado.

2 Cálculo λ

O Cálculo λ é uma formalismo de funções inventado nos anos 1930 por Church [4], onde as funções são entendidas como regras de correspondência entre argumentos e respectivos valores. Nesta secção introduzimos o cálculo puro, *i.e.* sem tipos. Uma versão do sistema com tipos surgirá na secção seguinte.

Cálculo λ puro O Cálculo λ puro [4, 2, 13] será denotado por Λ. O seu alfabeto é constituído pelos três símbolos em $\{(,),\lambda\}$ mais os símbolos em $\mathcal{X}$ - em que $\mathcal{X}$ é um conjunto numerável de *variáveis*, denotadas por x, y, z, etc. O conjunto dos *termos* de Λ é definido indutivamente por

$$M, N, P, Q ::= x \mid (\lambda x\, M) \mid (MN)$$

Por abuso, o conjunto dos termos também é designado por Λ[1].

Um termo da forma (MN) diz-se uma *aplicação* - onde M é a função e N o argumento. Um termo da forma $(\lambda x\, M)$ diz-se uma *abstracção* - e M diz-se o *âmbito* dessa abstracção. $(\lambda x\, M)$ pode ser entendido como a definição de uma função, cuja correspondência entre argumentos e valores é dada por M, e onde x é o argumento formal. O valor dessa função para um argumento real N é obtido por substituição em M de x por N. Apesar destas intuições em termos de "função" e "argumento", sintacticamente há apenas uma categoria - a dos termos.

As convenções para omitir parênteses em termos são: (i) os parênteses mais externos podem omitir-se; (ii) a aplicação associa à esquerda (exemplo: MN_1N_2 representa $(MN_1)N_2$ e não $M(N_1N_2)$); (iii) os parênteses em abstrações iteradas são dispensáveis (exemplo: escrever $\lambda xy\, M$ em vez de $\lambda x(\lambda y\, M)$); (iv) um ponto "." é usado para marcar o limite esquerdo

[1] Para aumentar a expressividade, iremos usar ocasionalmente outras meta-variáveis, tais como F e R (resp. f) para representar termos (resp. variáveis).

do âmbito de uma abstração, convencionando-se que esse âmbito estende para a direita o mais possível (exemplo: $\lambda x.MN$ representa $\lambda x(MN)$ e não $(\lambda x\, M)N$).[2]

As variáveis podem ocorrer livres ou ligadas em termos, tal como acontece nas fórmulas da lógica de 1ª ordem, e o operador de abstracção funciona como um quantificador a este respeito: uma ocorrência de x em M diz-se *ligada* se estiver no âmbito de uma abstracção $(\lambda x.\_)$, diz-se *livre* caso contrário. Se x não ocorre livre em M, escrevemos $x \notin M$. Há igualmente a necessidade de cuidados especiais com a operação $[N/x]M$ - a substituição em M das ocorrências livres de x por N. Em lógica de 1ª ordem é habitual permitir a operação apenas se a substituição não der origem a captura de variáveis; aqui adoptamos a convenção de variáveis de Barendregt [2] - o que implica identificar termos a menos de mudança de variáveis ligadas.

Um termo sem variáveis livres diz-se *fechado* (ou um *combinador*). Há toda uma teoria de combinadores paralela ao Cálculo λ e chamada Lógica Combinatória, desenvolvida por Curry e colaboradores [5, 13]. Alguns exemplos de combinadores são:

$$\begin{array}{rcl} \mathbf{I} & := & \lambda x.x \\ \mathbf{K} & := & \lambda yx.y \\ \overline{0} & := & \lambda fx.x \\ \overline{1} & := & \lambda fx.fx \\ \Omega & := & \Delta\Delta, \text{ onde } \Delta = \lambda x.xx \end{array}$$

Definem-se três relações binárias em Λ: $\rightarrow_\beta$ (*redução* β em 1 passo), $\rightarrow_\beta^*$ (*redução* β) e $=_\beta$ (*conversão* β[3]) a partir da seguinte relação binária, designada β:

$$(\lambda x.M)N \rightarrow [N/x]M \ .$$

$\rightarrow_\beta$ é a menor relação congruente contendo β (em que *congruente* significa satisfazer $M \rightarrow M' \Rightarrow (MN \rightarrow M'N \wedge NM \rightarrow NM' \wedge \lambda x.M \rightarrow \lambda x.M')$. $\rightarrow_\beta^*$ é o fecho reflexivo e transitivo de $\rightarrow_\beta$ (é, portanto, a relação de redução em 0, 1 ou mais passos); $=_\beta$ é a menor relação de equivalência contendo $\rightarrow_\beta$. Quer $\rightarrow_\beta$, quer $=_\beta$ são congruentes: $=_\beta$ é a teoria equacional gerada pelo axioma β; $\rightarrow_\beta^*$ é uma relação de reescrita associada.[4]

[2] Uma indicação da seriedade da matéria em (iv) é que o próprio Turing escreveu sobre o uso de pontos para omitir parênteses, como veremos adiante.

[3] Church preferia o nome "conversão λ".

[4] Também usamos a notação usual $\rightarrow_\beta^n$ para denotar a composição iterada de $\rightarrow_\beta$.

A conversão é um conceito de igualdade de termos mais grosseiro que a igualdade de termos como sequências de símbolos a menos de mudança de variáveis ligadas; mas mais fino que a igualdade *extensional* de funções usada em Matemática: se $x \notin F$, então $FN =_\beta (\lambda x.Fx)N$, para todo N; mas não se tem, em geral, $F =_\beta \lambda x.Fx$.[5]

Por seu turno, a redução pode ser vista como uma ferramenta teórica para analisar a conversão; ou como uma relação interessante em si mesma, na medida em que modela o processo computacional de cálculo do valor das funções. Para o primeiro ponto de vista, é fundamental o teorema da *confluência*, ou de Church-Rosser, que pode ser enunciado como a relação básica entre conversão e redução[6]: $M =_\beta N$ se e só se $M, N \to^*_\beta P$ para algum P.

Alguns exemplos de redução:

$$\begin{array}{l}
\mathbf{I}N = (\lambda x.x)N \to_\beta [N/x]x = N \\
\mathbf{K}MN = (\lambda yx.y)MN \to_\beta (\lambda x.M)N \to_\beta M \\
\overline{0}FN = (\lambda fx.x)FN \to^2_\beta N \\
\overline{1}FN = (\lambda fx.fx)FN \to^2_\beta FN \\
\Omega = (\lambda x.xx)\Delta \to_\beta [\Delta/x](xx) = \Delta\Delta = \Omega \to_\beta \Omega \to_\beta \cdots
\end{array}$$

$\mathbf{I}$ é o combinador identidade. $\mathbf{K}M$ converte-se em $\lambda x.M$, a função constante que devolve sempre M; note-se que a convenção de variáveis garante que $x \notin M$, e $\lambda x.M$ é uma abstracção vacuosa. Há uma versão do Cálculo λ onde tais abstracções não contam como termos. Essa versão restrita é designada por Cálculo $\lambda\mathbf{I}$ e o cálculo irrestrito, tratado aqui é, por vezes, chamado de Cálculo $\lambda\mathbf{K}$. Curiosamente, foi a versão $\lambda\mathbf{I}$ que Church e Kleene usaram inicialmente no estudo da computabilidade.

Os combinadores $\overline{0}$ e $\overline{1}$ dizem-se *numerais de Church*. O caso geral é $\overline{n} := \lambda fx.f \cdots fx$ (n aplicações de f, $n \geq 0$). Estes numerais comportam-se como iteradores: $\overline{n}FN \to^*_\beta F \cdots FN$ (n applicações de F). Rosser propôs o seguinte combinador para a adição:

$$\overline{+} := \lambda nmfx.nf(mfx) \ .$$

Por exemplo,

$$\begin{array}{rcl}
\overline{+}\ \overline{1}\ \overline{1} & \to^*_\beta & \lambda fx.\overline{1}f(\overline{1}fx) \\
& \to^*_\beta & \lambda fx.\overline{1}f(fx) \\
& \to^*_\beta & \lambda fx.f(fx) \\
& = & \overline{2}
\end{array} \qquad (1)$$

[5] O axioma $F = \lambda x.Fx$ ($x \notin F$), usualmente denominado η, é muitas vezes adicionado ao conceito de igualdade de termos.

[6] Para além da relação trivial $\to^*_\beta \subset =_\beta$.

Um termo P diz-se uma *forma normal* se nem P nem outro sub-termo de P[7] são um redex. Um *redex* (abreviatura de "*reducible expression*") é um termo da forma $(\lambda x.M)N$. P diz-se uma forma normal de Q se P é uma forma normal e $Q \rightarrow_\beta^* P$. Todos os combinadores vistos acima são formas normais, excepto Ω, que não é nem tem forma normal. Uma consequência imediata do teorema da confluência é que um termo tem, no máximo, uma forma normal.

Paradoxos No Cálculo puro, a nível sintactico, nada impede que um termo seja aplicado a si próprio. Durante muito tempo pensou-se que esta possibilidade levantaria problemas semânticos insolúveis: a construção de um modelo do cálculo na teoria de conjuntos chocaria com questões de cardinalidade (o domínio D do modelo teria de conter todas as funções de tipo $D \rightarrow D$), ou com o axioma da regularidade (teria de haver em D funções f em cujo domínio estaria f). No entanto, em 1969, Scott provou que o cálculo puro tem, de facto, modelos [2, 12]. Seja como for, a auto-aplicação dá origem a situações de aparência paradoxal.

Para cada $N \in \Lambda$, defina-se $R_N := \lambda x.N(xx)$, com $x \notin N$. Imaginemos que, no termo R_N, N representa a operação de negação e, em xx, a mesma variável desempenha o papel de função (um certo predicado) e argumento dessa função. A situação é reminiscente do paradoxo de Russell. Defina-se $\mathbf{auto}_N := R_N R_N$.[8] Então:

$$\mathbf{auto}_N = (\lambda x.N(xx))R_N \rightarrow_\beta N(\mathbf{auto}_N) \rightarrow_\beta N(N(\mathbf{auto}_N)) \rightarrow_\beta \cdots$$

O "paradoxo" produz aqui uma sequência de redução infinita. Tem-se também $N(\mathbf{auto}_N) =_\beta \mathbf{auto}_N$. Dizemos por isso que $\mathbf{auto}_N$ é um *ponto fixo* de N. Acabámos de ver que todo o termo tem um ponto fixo.

É simples agora encontrar $Y \in \Lambda$ tal que, para cada $F \in \Lambda$, YF seja um ponto fixo de F. Basta tomar $Y = \lambda f.\mathbf{auto}_f$. Note-se que $[F/f]\mathbf{auto}_f = \mathbf{auto}_F$, donde $YF \rightarrow_\beta \mathbf{auto}_F$. Deste último facto, juntamente com $\mathbf{auto}_F \rightarrow_\beta F(\mathbf{auto}_F)$, segue que YF é um ponto fixo de F:

$$YF \rightarrow_\beta \mathbf{auto}_F \rightarrow_\beta F(\mathbf{auto}_F) \leftarrow_\beta F(YF) \ .$$

Y é o combinador "paradoxal" de Curry. Em 1937, Turing [29] definiu outro combinador para calcular pontos fixos:

$$\Theta := UU, \text{ onde } U = \lambda uf.f(uuf) \ .$$

[7] O conceito de sub-termo fica claro com um exemplo: se $P = (\lambda x.xx)y$, os sub-termos de P, além de P, são x, xx, $\lambda x.xx$ e y.

[8] À parte: $\mathbf{auto}_I \rightarrow_\beta^2 \Omega$.

Este combinador tem a propriedade $\Theta F \to_\beta^2 F(\Theta F)$. Turing mostrou como ele poderia ser usado na representação das funções recursivas em Λ, simplificando a representação no Cálculo $\lambda\mathbf{I}$ obtida anteriormente por Kleene.

3 Teoria de tipos

A introdução de tipos no Cálculo λ, originalmente feita por Church [3] em 1940, pode ser vista como uma restrição do cálculo puro por forma a proibir a auto-aplicação e os "paradoxos" decorrentes. Esse não é, porém, o ponto de vista de Church. Nos anos 1930, Church assegurou-se de que o cálculo puro é consistente [4], significando isto que $=_\beta$ não identifica todos os termos (outra consequência simples do teorema da confluência). A perspectiva de Church em [3] é a de que o Cálculo λ empresta à Teoria de Tipos uma formulação vantajosa.

Historicamente, a Teoria de Tipos foi iniciada por Russell na primeira década do século XX em reacção à descoberta de paradoxos nos fundamentos da Matemática. As formulações iniciais da teoria, incluindo aquela usada nos *Principia Mathematica* [35], empregavam os chamados tipos ramificados. O objectivo era banir as definições "impredicativas", que Russell (e Poincaré) consideravam estar na raiz dos paradoxos. Porém, a exigência da predicatividade impede o sistema dos *Principia* de formalizar certas partes da Matemática, e este conflito só foi resolvido em [35] pela adopção de um axioma que, posteriormente, foi muito criticado[9]. Nos anos 1920, Chwistek e Ramsey sugerem os chamados tipos simples como modificação do sistema dos *Principia*, o que produz uma teoria simultaneamente mais simples (justamente) e forte (porque impredicativa).

O sistema de Church em [3] é uma teoria de tipos simples[10]. Turing tomou contacto com ele durante a sua estadia em Princeton (1936-1938), num curso dado pelo próprio Church. Em 1940 Turing mostrou-se satisfeito por Church ter decido publicar "*his form of Principia*" ([8], p.151).

Cálculo λ tipificado O sistema que vamos de seguida esboçar, denotado $\Lambda^\to$, constitui o coração do sistema introduzido por Church.

Seja $\mathcal{A}$ um conjunto de símbolos, entendidos como *tipos atómicos*. Um

[9] Trata-se do chamado axioma da reducibilidade, que foi demolido nos anos 1930 por Quine - ver [22].

[10] Para um tratamento moderno das teorias de tipos de Russell e Ramsey, e sua comparação com o sistema de Church, ver [17].

tipo (simples) é um elemento do conjunto gerado pela gramática

$$\rho, \sigma, \tau ::= a \,|\, (\sigma \to \tau)$$

onde $a \in \mathcal{A}$[11]. Para omitir parênteses em tipos, usamos a convenção de que $\to$ associa à direita. Por exemplo, $\rho \to \sigma \to \tau$ representa $\rho \to (\sigma \to \tau)$ e não $(\rho \to \sigma) \to \tau$.

A ideia do sistema é atribuir tipos a termos para assim os classificar; e a possibilidade de atribuir um tipo a um termo é uma condição forte de boa formação do termo. Um termo de tipo $\sigma \to \tau$ representa uma função com argumentos de tipo σ e valores de tipo τ. A classificação em tipos permite formular restricções gramaticais. Por exemplo, a aplicação de um termo de tipo $\sigma \to \tau$ a um argumento só receberá um tipo se esse argumento for do tipo esperado.

Os termos de $\Lambda^{\to}$ definem-se indutivamente por:

$$M, N, P, Q ::= x \,|\, (\lambda x_\sigma \, M) \,|\, (MN)$$

A novidade está em que a variável abstraída exibe o seu tipo (o que será essencial para garantir tipificação única). Por exemplo, os combinadores $\mathbf{I}$ e $\mathbf{K}$ passam a ser famílias de termos $\mathbf{I}^\sigma := \lambda x_\sigma.x$ e $\mathbf{K}^{\sigma,\tau} := \lambda y_\sigma x_\tau.y$, parametrizadas pelo tipo das variáveis abstraídas. Na prática, quando não resultar ambiguidade, omitem-se os tipos das variáveis abstraídas.

A atribuição de tipos a termos faz-se relativamente a um *contexto* Γ, um conjunto finito de atribuições de tipo a variáveis (escritas na forma $x : \sigma$) onde há no máximo uma atribuição para cada variável. A relação $\Gamma \vdash M : \sigma$ (M tem tipo σ, ou M *habita* o tipo σ, no contexto Γ) é definida indutivamente pelas seguintes regras (onde $\Gamma, x : \sigma$ denota a união $\Gamma \cup \{x : \sigma\}$, que se assume disjunta):

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma} \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x_\sigma.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \tag{2}$$

Por exemplo, tem-se, para todo σ, τ:

$$\vdash \mathbf{I}^\sigma : \sigma \to \sigma$$
$$\vdash \mathbf{K}^{\sigma,\tau} : \sigma \to \tau \to \sigma$$

onde a ausência de contexto significa que o contexto é vazio.

M diz-se *tipificável* se $\Gamma \vdash M : \sigma$ para algum Γ e algum σ. Fixados Γ e M, há no máximo uma derivação $\mathcal{D}$, construídas com as regras (2), com

[11] Note-se a sobrecarga do símbolo $\to$, antes usado para denotar redução.

conclusão da forma $\Gamma \vdash M : \sigma$; portanto, M tem, no máximo, um tipo num contexto Γ. Prova-se ainda que esse tipo, se existir, é também o tipo de todos os termos a que M se reduz. Não se pode ter $\Gamma \vdash M : \sigma \to \tau$ e $\Gamma \vdash M : \sigma$. Assim, MM não é tipificável. Logo, Ω também não o é.

Sistema de Church Uma vez definido $\Lambda^{\to}$, a metodologia de Church consiste em usar $\Lambda^{\to}$ como infra-estrutura, no topo da qual de define, com toda a simplicidade e elegância, a sintaxe da lógica de predicados de ordem superior. Para tal permitem-se tipos e termos *constantes*; cada um destes últimos tem associado um tipo único.

Os tipos constantes são o (o tipo das fórmulas) e ι (o tipo dos indivíduos no "universo de discurso"). Uma fórmula será um termo de tipo o; um indivíduo será um termo de tipo ι (um "termo", na terminologia da lógica de 1ª ordem).

Para construir fórmulas, podemos adoptar os seguintes termos constantes: $\neg : o \to o$ (negação), $\supset : o \to o \to o$ (implicação) e $\forall^{\sigma} : (\sigma \to o) \to o$ (quantificador universal). Por exemplo, se $M : o$ e $N : o$ então $\neg(\supset MN) : o$ - uma fórmula habitualmente escrita $\neg(M \supset N)$; se $x : \iota \vdash M : o$, então $\vdash \forall^{\iota}(\lambda x.M) : o$ - uma fórmula usualmente escrita $\forall x.M$ (mas, nesta metodologia, a quantificação surge decomposta em termos da abstracção); se $y : o \vdash M : o$, então $\vdash \forall^{o}(\lambda y.M) : o$ - uma fórmula definida de forma impredicativa, por quantificação sobre todas as fórmulas; finalmente, se $z : \iota \to o \vdash M : o$, então a fórmula $\forall^{\iota \to o}(\lambda z.M)$ é uma quantificação de 2ª ordem.

O cálculo $\Lambda^{\to}$ é uma infra-estrutura para fazer, não só Lógica, mas também Matemática. Por exemplo, o tipo dos números naturais pode definir-se como

$$\mathsf{Nat} := (\iota \to \iota) \to \iota \to \iota \ .$$

Os numerais de Church, na versão $\lambda f_{\iota \to \iota} x_{\iota}.f \cdots fx$, habitam o tipo Nat. O combinador de Rosser para a adição, na versão

$$\overline{+} := \lambda n_{\mathsf{Nat}} m_{\mathsf{Nat}} f_{\iota \to \iota} x_{\iota}.nf(mfx) \ ,$$

tem tipo $\mathsf{Nat} \to \mathsf{Nat}$. A redução (1) vista acima acontece no tipo Nat e prova o famoso teorema $\overline{+}\ \overline{1}\ \overline{1} =_{\beta} \overline{2}$.[12]

[12] O teorema $1 + 1 = 2$ dos *Principia Mathematica*, pela sua natureza e difícil demonstração, é motivo irresistível de paródia. Ver, por exemplo, a capa da *Notices of the American Mathematical Society*, 57(11), Dezembro 2010.

Turing e a Teoria de Tipos Turing publicou três artigos sobre Teoria de Tipos, dois em 1942 [19, 31] e um em 1948 [32], sendo que, de acordo com [8], os primeiros resultados são referidos em correspondência de fins de 1941 ou inícios de 1942 e [32] estaria essencialmente concluído no Verão de 1945. É, portanto, um trabalho contemporâneo da Segunda Guerra Mundial. Uma parte considerável do trabalho de Turing em Teoria de Tipos foi deixada inacabada e/ou não-publicada e é datável do mesmo período, tendo-se tornado acessível apenas através da edição dos *Collected Works* [8].

Turing manteve interesse na Teoria de Tipos até ao fim da vida - o assunto era tema de discussão com o seu aluno de doutoramento Robin Gandy nos anos 1950 e foi tópico de uma palestra em Oxford em 1953 ([8], pp. 154, 183); mas Turing não retomou as suas investigações após o fim da Guerra. É fácil sugerir explicações: o período 1945-1950 traz a Turing uma série de mudanças intelectuais e biográficas, com a envolvimento na construção dos primeiros computadores britânicos, o interesse na Inteligência Artificial e a mudança para Manchester [14].

O trabalho de Turing em Teoria de Tipos desenvolveu-se em duas direcções. A primeira consistiu na investigação de várias questões acerca do sistema de Church: a magna questão da omissão de parênteses [31] - uma questão com pedigree, que recua até aos *Principia Mathematica* e a Peano [8]; questões de consistência relativa e independência dos axiomas do sistema [19]; e a questão da normalização, a que voltaremos na secção seguinte.

A segunda direcção de investigação procurou formulações alternativas da Teoria de Tipos que fossem "práticas" [32]. Por trás do sistema específico introduzido no artigo citado, há uma intenção, timidamente descrita nas poucas linhas do primeiro parágrafo desse artigo, mas melhor compreendida lendo os manuscritos. No manuscrito (lido em [8], p. 183) que deu origem a [32], os objectivos do sistema introduzido são claramente descritos :

> *In the present paper a system will be described which takes account of type theory, but at the same time follows very closely the normal mathematical outlook. The type theory intrudes itself on the system only very slightly, and its effect may be summed up in the form of one or two simple and natural cautions, which are easily carried over to unformalised mathematics.*

As "*natural cautions*" são vistas noutro manuscrito de forma mais ambiciosa, como uma reforma da notação e fraseologia matemáticas - um projecto que não saiu da fase de esboço. Alguns dos primeiros passos dessa reforma são bem concretos e incluem: os conselhos "*There should be no danger of*

mistaking a real variable for a function taking real values" e "*The deduction theorem should be as well known as the rule for integration by parts*"; e a adopção de "*new notation suggested by symbolic logic*" - certamente a notação λ. Nos aspectos mais avançados dessa reforma, Turing especula sobre como implementar na linguagem comum (a linguagem usada pela Matemática não formalizada) o conceito de "classe substantiva" ("*noun class*") - um conceito de colecção do sistema de [32] com a propriedade de que é possível provar formalmente que todos os seus elementos habitam um tipo.

Subjacente a tudo isto está a opinião de Turing de que a formalização em Teoria de Tipos é o melhor método para obter uma descrição aproximada do raciocínio matemático. Mas Turing deixa claro que esta opinião não implica que a Matemática deva ser conduzida na prática, ou concebida filosoficamente, dentro de um sistema lógico ([8], pp. 184, 215).

4 Normalização

Tal como no Cálculo puro, em $\Lambda^{\rightarrow}$ a unicidade de forma normal é consequência imediata da confluência. A novidade no Cálculo tipificado é:

Teorema 1 *Se M é tipificável então M tem forma normal.*

Este é o teorema da normalização, que tem uma versão forte:

Teorema 2 *Se M é tipificável então toda a sequência de redução a partir de M é finita.*

Uma *sequência de redução a partir de M* é uma sequência finita ou infinita $M = M_0, M_1, M_2, \cdots$ de termos tais que $M_i \rightarrow_\beta M_{i+1}$. O teorema forte diz que o processo de redução - entendido como o processo de eliminação de todos os redexes -, a partir de um termo M tipificável, termina sempre, independentemente das sucessivas escolhas de redexes (e termina necessariamente com uma forma normal de M).

Estado da arte As demonstrações mais simples de normalização para $\Lambda^{\rightarrow}$ obtiveram-se no período 1998 - 2003 [1, 33, 6, 16]. David dá em [6] uma prova do Teorema 2 que ocupa meia página. De acordo com o autor, a prova foi obtida simplificando uma demonstração que ouvira a Matthes numa palestra em Janeiro de 1998. Esta última demonstração só pode ser aquela que surge

em [16][13] - a qual parte de uma caracterização indutiva do conjunto

$$SN := \{M \in \Lambda^{\rightarrow} \mid \text{toda a sequência de redução a partir de } M \text{ é finita}\} \ ,$$

estratégia também seguida em [33]. Mas, na prova de David, não há vestígio dessa estratégia: com efeito, parece mais justo *a posteriori* dizer que a prova de David é uma simplificação da demonstração de Amadio e Curien em [1]. Vejamos porquê.

A demonstração de $M \in SN$, com M tipificável, é por indução em M. Os casos $M = x$ e $M = \lambda x.P$ são triviais. No caso final $M = PQ$, usa-se o artifício $M = [P/z](zQ)$, com $z \notin Q$ (o que constitui a primeira melhoria da prova de David relativamente à de Amadio e Curien). P e Q são tipificáveis e, por hipótese de indução, estão em SN. De $Q \in SN$ segue imediatamente $zQ \in SN$. Assim, toda a dificuldade do teorema está localizada na seguinte implicação:

$$M, N \in SN \Rightarrow [N/x]M \in SN \ . \tag{3}$$

Em geral, a implicação é falsa. Por exemplo, se $M = xx$ e $N = \Delta$, então $[N/x]M = \Omega \notin SN$. Amadio e Curien demonstram (3) adicionando a hipótese $\Gamma \vdash N : \tau$ e fazendo indução no trio ordenado $(\tau, ||M||, M)$, onde $||M|| \in \omega$ é o máximo dos comprimentos das sequências de redução a partir de M[14]. Estes trios estão ordenados lexicograficamente, com a componente τ (resp. $||M||$) a primeira (resp. segunda) a ser comparada. Amadio e Curien analisam de novo os casos $M = x$, $M = \lambda x.P$ (estes dois muito simples) e $M = PQ$. Este último é complicado, pois, a certa altura, é necessário compreender como é que $[N/x]P$ se reduz a uma abstracção. David prova (3), sob a hipótese $\Gamma \vdash N : \tau$, fazendo a mesma indução, mas com uma diferente análise de casos: $M = xN_1 \cdots N_m$, $M = \lambda x.P$, e $M = (\lambda x.P)QN_1 \cdots N_m$ ($m \geq 0$). Esta manobra contorna todas as dificuldades da prova de Amadio e Curien, de tal modo que todos os casos seguem agora da hipótese de indução, e podem ser deixados ao cuidado do leitor. O Teorema 1 segue como corolário.

A demonstração de Turing É uma prova directa do Teorema 1.

Suponhamos que $\Gamma \vdash M : \rho$ e seja $\mathcal{D}$ a única derivação deste facto pelas regras de tipificação (2). Dada uma ocorrência do redex $(\lambda x_\sigma.P)Q$ em M,

[13] Isto foi confirmado por Ralph Matthes em comunicação privada. O artigo [16], embora tenha aparecido apenas em 2003, foi submetido em 1998.

[14] $M \in SN$ garante a existência deste máximo. Prova: Primeiro, define-se, para M arbitrário, o conceito óbvio de *árvore de redução* de M. Uma tal árvore ramifica finitamente. Depois, $M \in SN$ e o lema de König garantem que a árvore de redução de M é finita.

seja τ o tipo dessa ocorrência determinado por $\mathcal{D}$ (τ é também o tipo da correspondente ocorrência de P). Define-se a *ordem* dessa ocorrência de redex como sendo o tamanho do tipo $\sigma \to \tau$, onde o *tamanho* de um tipo é o número de ocorrências de $\to$ nesse tipo[15]. A ideia de Turing é que, se o redex $(\lambda x_\sigma.P)Q$ a reduzir for convenientemente escolhido, o processo de normalização avança realmente: há apenas que garantir que não existem redexes de ordem máxima em Q (que, caso contrário, poderiam ser copiados pela substituição $[Q/x]P$):

> *Now when we perform a reduction on a formula, in which we reduce one of the unreduced parts of highest order, we necessarily decrease the number of unreduced parts of the highest order, for we destroy one and we do not create any more: this at any [rate] will be the case if we choose the unreduced part of highest order whose λ lies farthest to the right.* [7][16]

Tomemos como definição de redex *convenientemente escolhido* (abreviatura: redex c.e.) aquela que Turing dá nesta citação.

De seguida, Turing define uma boa ordem nos termos de modo que, se $M \to M'$ através da redução do redex c.e., então $M > M'$. Esta parte da demonstração pode ser ligeiramente simplificada do seguinte modo. Associamos a M o par ordenado $(m(M), n(M))$, onde $m(M)$ é o máximo das ordens de ocorrências de redexes em M, e $n(M)$ é o número de ocorrências de redexes em M de ordem $m(M)$. Convenção: $m(M) = n(M) = 0$, se M é normal. Consideremos os pares $(m(M), n(M))$ ordenados lexicograficamente (com a componente $m(M)$ a primeira a ser comparada). A redução do redex c. e. de M produz um termo M' tal que $(m(M), n(M)) > (m(M'), n(M'))$.

A justificação completa desta última afirmação (ou da afirmação $M > M'$ no caso de Turing) requer uma análise (que Turing e Gandy omitem [7], mas que é feita por exemplo em [10, 24]) dos redexes "novos" gerados por $[Q/x]P$. Há 3 casos:[17] (i) Q é uma abstracção e há em P um sub-termo da forma xN - o redex "novo" é QN; (ii) a ocorrência do redex c.e. em M é da forma $(\lambda x_\sigma.P)QQ'$ com $P = x$ e Q uma abstracção - o redex "novo" é QQ'; (iii) a ocorrência do redex c.e. em M é da forma $(\lambda x_\sigma.P)QQ'$ com $P = \lambda y.P'$ - o

[15] A ordem da ocorrência de um redex num termo tipificável depende, portanto, de Γ. A dependência de Γ vai, no entanto, ser omitida na notação.

[16] A inserção da palavra "rate" é sugerida por Gandy. As palavras "*formula*" e "*unreduced part*" significam termo e redex, respectivamente.

[17] Barendregt ([2], p. 382) atribui a identificação destes casos à tese de doutoramento de J.-J. Lévy, de 1978.

redex "novo" é $(\lambda y.[Q/x]P')Q'$. Em todos os casos se verifica que a ordem dos "novos" redexes será o tamanho de σ ou de τ, e será, portanto, inferior a $m(M)$.

5 A (re)descoberta da normalização

As primeiras demonstrações publicadas de teoremas de normalização, no contexto do Cálculo λ ou variantes, são as de Curry e Feys [5] e de Tait [26]. Curry e Feys trabalham em Lógica Combinatória, Tait estuda uma extensão do Cálculo λ chamada sistema T. Em ambos os casos, a motivação para o teorema da normalização é a de que esse teorema tem como consequência a consistência de algum sistema formal de interesse: versões tipificadas da Lógica Combinatória, no caso de Curry e Feys[18]; e a aritmética de Peano, no caso de Tait[19].

Foi, porém, Prawitz quem tornou evidente a centralidade da normalização, no seu estudo sobre a dedução natural [20]. O contexto aqui é a Teoria da Demonstração, e Prawitz escreve na continuação de Gentzen [9]. Gentzen havia introduzido dois sistemas formais de dedução, a dedução natural e o cálculo de sequentes[20], mas provou um "*Hauptsatz*" apenas para o último, o chamado teorema da eliminação do corte[21]. Prawitz demonstrou o teorema da normalização, quer para lógica clássica, quer para lógica intuicionista, classificando esta demonstração - em comparação com a demonstração da eliminação do corte - como "*simpler and more illuminating*". Em analogia com o que Gentzen fizera, Prawitz extraiu da normalização aplicações e consequências em Teoria da Demonstração, desde logo a consistência da lógica de 1^a ordem. Prawitz mostrou ainda como obter o próprio teorema da eliminação do corte como consequência do teorema da normalização: de facto, num certo sentido, os dois teoremas são equivalentes [21, 36]. Não é, por isso, difícil concordar com Gandy quando diz [7]:

[18] A história de como Curry foi levado à descoberta da normalização está muito bem explicada num recente artigo de Seldin [23].

[19] Gödel havia demonstrado, através da sua interpretação *Dialectica*, publicada em 1958 [11], que há uma redução da consistência da Aritmética à normalização do sistema T.

[20] Em rigor, a história da dedução natural recua até aos anos 1920, conforme explica Prawitz [20].

[21] Algumas palavras na sinopse de [9] dão a entender que Gentezen terá tentado provar o teorema da normalização para lógica clássica e terá falhado, e que esse seria o motivo pelo qual ele foi levado a desenvolver o cálculo de sequentes, onde uma só demonstração estabelece o *Hauptsatz* clássico e intuicionista.

> *In a very loose sense, one may say that Gentzen's Hauptsatz was the first normal form theorem.*

Os desenvolvimentos paralelos em Cálculo λ e dedução natural tornaram-se convergentes quando Howard, num manuscrito de 1969 [15][22], formalizou a correspondência entre os dois formalismos, complementando - e estendendo ao nível da lógica de 1ª ordem - as analogias entre Lógica Combinatória e sistemas axiomáticos proposicionais observadas por Curry desde os anos 1930 [24] e demonstradas em [5]. Usando essa *correspondência de Curry-Howard*, Prawitz [21] transferiu o método de Tait [26] para a dedução natural, e observou que esse método era afinal suficiente para estabelecer a normalização forte. Esta prova de normalização forte tem a desvantagem de não ser finitária, *i.e.* formalizável na Aritmética[23]. No entanto, como vimos na Secção 4, foram entretanto obtidas demonstrações finitárias da normalização forte para $\Lambda^{\rightarrow}$.

Por outro lado, se usarmos a correspondência de Curry-Howard no sentido inverso, e transferirmos para o Cálculo λ a demonstração de normalização de [20], no caso restrito da implicação intuicionista, obtemos o argumento de Turing. Por outras palavras, a demonstração de Prawitz de 1965 contém, a menos da correspondência de Curry-Howard, uma re-descoberta da demonstração de Turing[24]. Turing, por sua vez, não poderia ter percebido o significado do seu resultado para a dedução natural, pois não estava familiarizado com o sistema ([8], p.212), nem com a forma de conceber a relação entre Cálculo λ e Lógica própria da correspondência de Curry-Howard. O conceito de "proposições-como-tipos", subjacente à correspondência de Curry-Howard, é completamente estranho à metodologia de Church de representação de uma lógica (externa) sobre $\Lambda^{\rightarrow}$: o conceito exige reconhecer que as regras de tipificação (ver (2) acima) constituem uma lógica interna de $\Lambda^{\rightarrow}$, onde os tipos (não os termos) desempenham o papel de fórmulas.

O último volte-face desta história é a descoberta recente, relatada em [34], de que Gentzen, em finais de 1932, demonstrara o teorema da normalização

[22] Tal como a demonstração de normalização de Turing, este manuscrito só seria publicado no volume [12] dedicado a Curry. Porém, ao contrário do manuscrito de Turing, o manuscrito de Howard circulou e foi absorvido imediatamente após ser escrito - ver, por exemplo, [21].

[23] Se o método de Tait fosse finitário, ele poderia ser combinado com o resultado de Gödel referido na nota 19 - a redução de Gödel é finitária [21] -, produzindo assim uma demonstração finitária da consistência da Aritmética, o que é impossível pelo segundo Teorema da Incompletude de Gödel.

[24] Em rigor, referindo-nos à demonstração de Turing apresentada na Secção 4, a demonstração de Prawitz introduz a simplificação aí mencionada, mas também omite a análise de "fórmulas maximais novas", que seria correspondente à análise dos redexes "novos".

para lógica de 1ª ordem intuicionista, fazendo a mesma demonstração que Prawitz viria a publicar em 1965. Essa demonstração constitui um capítulo no manuscrito da versão inicial da tese de doutoramento de Gentzen, onde este projectava uma demonstração da consistência da Aritmética baseada na normalização. Quando Gentzen descobriu a impossibilidade do seu projecto, abandonou o plano inicial em favor da versão posteriormente publicada [9]. Do plano inicial sobreviveu o referido manuscrito (ignorado até há poucos anos) e um artigo onde Gentzen dá uma interpretação da Aritmética clássica na Aritmética intuicionista (artigo que Gentzen não conseguiu publicar em virtude de Gödel ter independentemente demonstrado o mesmo resultado)[25].

Gentzen morreu aos 35 anos de idade, em circunstâncias trágicas relacionadas com o fim da Segunda Guerra Mundial [18][26]. Se tivesse sobrevivido à Guerra, Gentzen teria certamente publicado o seu capítulo sobre normalização, que aliás chegara a classificar como "*ready for publication*" [34]. O fim da Guerra também pôs termo à investigação da Teoria de Tipos por parte de Turing, de modo que o artigo onde Turing planeara incluir o teorema da normalização, e que chegara a anunciar como "*forthcoming*" [31], nunca se materializou.

Agradecimentos: O autor agradece a Luís Cruz-Filipe e a Luís Pinto os comentários a uma versão preliminar deste artigo. O autor agradece ainda aos editores do Boletim o convite para participar neste número dedicado a Turing.

Referências

[1] R. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*. Cambridge University Press, 1998.

[2] H.P. Barendregt. *The Lambda Calculus*. North-Holland, 1984.

[3] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.

[25] Os *Collected Papers*, publicados em 1969 [25], incluem o referido artigo de Gentzen (em tradução inglesa), mas ignoram o manuscrito da tese, com a excepção de que uma página deste surge fotografada no preâmbulo da colectânea.

[26] Gentzen morreu numa prisão em Praga, a 4 de Agosto de 1945 [18]. Entre meados de Julho e meados de Agosto de 1945, Turing encontrava-se em Bayreuth, Alemanha, a 200 km de Praga, integrado numa missão anglo-americana com o seguinte objectivo: "*to report on German progress in communications*" ([14], pp. 311-312).

[4] A. Church. *The Calculi of Lambda-Conversion.* Princeton University Press, 1941.

[5] H. B. Curry and R. Feys. *Combinatory Logic.* North-Holland, 1958.

[6] R. David. Normalization without reducibility. *Annals of Pure and Applied Logic*, 107(1-3):121–130, 2001.

[7] R. O. Gandy. An early proof of normalization by A. M. Turing. In *Hindley and Seldin [12]*, pages 453–455. Academic Press, 1980.

[8] R. O. Gandy and C. E. M. Yates, editors. *Collected Works of A. M. Turing - Mathematical Logic.* Elsevier, 2001.

[9] G. Gentzen. Untersuchungen über das logische schliessen I, II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. Tradução inglesa em [25].

[10] J-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge University Press, 1989.

[11] K. Gödel. Über eine bischer noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958. *Acerca de uma extensão até agora não utilizada do ponto de vista finitista*, Tradução de M. S. Lourenço, Boletim da Sociedade Portuguesa de Matemática, número 55, Outubro de 2006.

[12] J. R. Hindley and J. P. Seldin, editors. *To H. B. Curry: Essays in Combinatory Logic, Lambda-calculus, and Formalism.* Academic Press, 1980.

[13] J. R. Hindley and J. P. Seldin, editors. *Lambda-Calculus and Combinators: an Introduction.* Cambridge University Press, 2008.

[14] A. Hodges. *Alan Turing: the enigma.* Vintage, 1992.

[15] W. A. Howard. The formulae-as-types notion of construction. In *Hindley and Seldin [12]*, pages 480–490. Academic Press, 1980.

[16] F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic*, 42:59–87, 2003.

[17] F. Kamareddine, T. Laan, and R. Nederpelt. *A Modern Perspective on Type Theory.* Kluwer, 2004.

[18] E. Menzler-Trott. *Logic's Lost Genious: The Life of Gerhard Gentzen.* American Mathematical Society, 2007.

[19] M. H. A. Newman and A. M. Turing. A formal theorem in Church's theory of types. *Journal of Symbolic Logic*, 7(1):28–33, 1942.

[20] D. Prawitz. *Natural Deduction. A Proof-Theoretical Study.* Almquist and Wiksell, Stockholm, 1965.

[21] D. Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavium Logic Colloquium.* North Holland, 1971.

[22] W. Quine. *Set Theory and Its Logic.* Harvard University Press, 1963.

[23] J. Seldin. The search for a reduction in combinatory logic equivalent to $\lambda\beta$-reduction. *Theoretical Computer Science*, 412:4905–4918, 2011.

[24] M. H. Sorensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism.* Elsevier, 2006.

[25] M. E. Szabo, editor. *The Collected Papers of Gerhard Gentzen.* North-Holland Publishing Company, 1969.

[26] W. Tait. Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic*, 32(2):198–212, 1967.

[27] A. Turing. On computable numbers, with an application to the *Entscheidungproblem. Proceedings of the London Mathematical Society*, 42(2):230–265, 1936. Correcções: *ibid*, vol. 43, pp. 544–546, 1937.

[28] A. Turing. Computability and λ-definability. *Journal of Symbolic Logic*, 2(4):153–163, 1937.

[29] A. Turing. The p function in $\lambda - K$-conversion. *Journal of Symbolic Logic*, 2(4):164, 1937.

[30] A. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, 45:161–228, 1938.

[31] A. M. Turing. The use of dots as brackets in Church's system. *Journal of Symbolic Logic*, 7(4):146–156, 1942.

[32] A. M. Turing. Pratical forms of type theory. *Journal of Symbolic Logic*, 13(2):80–94, 1948.

[33] F. van Raamsdonk, P. Severi, M. Sorensen, and H. Xi. Perpetual reductions in λ-calculus. *Information and Computation*, 149(2):173–225, 1999.

[34] J. von Plato. Gentzen's proof of normalization for natural deduction. *Bulletin of Symbolic Logic*, 12(2):240–257, 2008.

[35] A. N. Whitehead and B. Russell. Principia Mathematica. Cambridge University Press, 1910.

[36] J. Zucker. The correspondence between cut-elimination and normalization. *Annals of Mathematical Logic*, 7:1–112, 1974.