1 Introdução

No texto[1] que se segue vão ser apresentados resultados sobre não decidibilidade de alguns predicados (sobre os naturais), ou seja, para certos predicados vai ser apresentada uma prova de que não é possível escrever programas URM que permitam determinar, para cada tuplo adequado de argumentos, se sim ou não os predicados são verdadeiros.

Descrições informais de alguns destes predicados são

- problema da paragem: dado um programa arbitrário e valores de entrada para esse programa, o problema de determinar se sim ou não o programa termina (se sim ou não o programa tem um ciclo infinito) quando é executado com esses valores;
- *printing problem*: dado um programa e um valor arbitrários, o problema de determinar se sim ou não se vai obter como *output* do programa esse valor;
- fixada uma dada função unária computável θ, o problema de determinar se sim ou não um programa arbitário calcula essa função;
- dados dois programas arbitrários, o problema de determinar se sim ou não eles calculam a mesma função (unária).

Estes resultados de não decidibilidade, envolvendo programas URM e funções naturais, são relevantes pois, assumindo o postulado de Church-Turing, permitem avaliar limites[2] das capacidades computacionais dos computadores digitais (arquitectura de von-Neumann). Um programa P executável num computador digital calcula uma certa função (não necessariamente total[3]) em $[\mathbb{N}_0 \to \mathbb{N}_0]$. Com efeito, P calcula uma função que faz corresponder a cada conjunto de valores de entrada (*input*) um certo conjunto (eventualmente vazio) de valores de saída (*output*). Os valores de entrada, tal como os de saída, constituem uma sequência finita de caracteres e portanto P calcula uma determinada função que transforma sequências finitas de caracteres em sequências finitas de caracteres. Cada caracter pode ser representado por um sequência finita de 0's e 1's o que significa que o programa P calcula uma dada função de $\{0,1\}^*$ em $\{0,1\}^*$. O conjunto $\{0,1\}^*$ é efectivamente numerável[4] o que permite então concluir que P calcula uma função natural de variável natural, isto é, uma função em $[\mathbb{N}_0 \to \mathbb{N}_0]$. Esta função é efectivamente computável pois existe um procedimento efectivo que a calcula (P e as codifcações envolvidas). Conclui-se

[1] Estas notas são baseadas em Sintaxe e Semântica de Linguagens I, J-F Costa, DMIST,2000 e *Computability-an introduction to recursive function theory*, N. Cutland, Cambridge University Press, 1980

[2] assumindo que as computações têm de ser executadas em tempo finito, mas não limitado e que a memória utilizada é também finita, mas não limitada

[3] em particular, programas cuja execução possa não terminar correspondem a funções não totais

[4] basta pensar na bijecção $f : \{0,1\}^* \to \mathbb{N}_0$ tal que $f(\omega) = \pi(w_1, w_2)$ onde, sendo $\omega = \omega_1\omega_2$ tal que $\omega_1 \in \{0\}^*$ e $\omega_2 \in \{0\} \cup \{1x : x \in \{0,1\}^*\}$, $w_1 = |\omega_1|$ e w_2 é a representação em base 10 do natural cuja representação em base 2 é ω_2

assim que a cada programa executável num computador digital corresponde uma função em $[\mathbb{N}_0 \to \mathbb{N}_0]$ efectivamente computável. Tendo em conta o postulado de Church-Turing, esta função é, em particular, URM-computável. Assim, uma função natural de variável natural que não seja URM-computável não pode ser calculada por um programa executável num computador digital.

Deste modo, a não decidibilidade dos predicados acima mencionados tem como consequência que, em particular, a solução do problema da paragem está para além das capacidades computacionais de máquinas digitais (arquitectura de von-Neumann).

2 Enumeração dos programas URM e das funções computáveis

Notação: No que se segue, para cada $n, m, k \in \mathbb{N}_0$

- ϕ_n^k é a função de aridade k calculada pelo programa cujo código (ou número de Gödel) é n; no caso de k ser 1 usa-se somente ϕ_n
- W_n^k é o domínio de ϕ_n^k; no caso de k ser 1 usa-se W_n
- E_n^k é o contradomínio de ϕ_n^k; no caso de k ser 1 usa-se E_n
- $\phi_n(m)\downarrow$ representa que a função ϕ_n está definida para o valor m
- $\phi_n(m)\uparrow$ representa que função ϕ_n não está definida para m

O facto de existir uma bijecção $\gamma : \mathbb{P} \to \mathbb{N}_0$ efectivamente computável permite construir uma enumeração de $\mathbb{P}$ que pode ser identificada com a sucessão

$$P_0, P_1, P_2, \ldots$$

isto é, a sucessão cujo i-ésimo termo é o programa URM cujo código (ou número de Gödel) é i. Como γ é uma bijecção, nesta enumeração não há repetições, ou seja, se $i \neq j$ então os programas P_i e P_j são diferentes.

A partir da enumeração acima referida pode construir-se, para cada $k \in \mathbb{N}$, uma enumeração de C_k (conjunto das funções URM-computáveis de aridade k)

$$\phi_0^k, \phi_1^k, \phi_2^k, \ldots$$

isto é, a sucessão cujo i-ésimo termo é a função de aridade k calculada pelo programa URM cujo código é i. Nesta enumeração há repetições, ou seja, existem $i, j \in \mathbb{N}_0$ tais que $i \neq j$ e $\phi_i^k = \phi_j^k$, pois a mesma função pode ser calculada por programas diferentes. No que se segue, tem particular interesse a enumeração

$$\phi_0, \phi_1, \phi_2, \ldots$$

de C_1.

Considere-se a seguinte tabela (infinita)

	0	1	2	...
ϕ_0	$\phi_0(0)$	$\phi_0(1)$	$\phi_0(2)$	...
ϕ_1	$\phi_1(0)$	$\phi_1(1)$	$\phi_1(2)$	...
ϕ_2	$\phi_2(0)$	$\phi_2(1)$	$\phi_2(2)$	...
...	...	...	...	...

Na coluna mais à esquerda da tabela encontra-se a referida enumeração de C_1. Para cada $n \in I\!N_0$ e cada $m \in I\!N_0$, $\phi_n(m)$ representa a imagem de m através da função ϕ_n, se m pertence ao domínio desta função, e nd (i.e., não definida) quando m não pertence a este domínio (i.e., a função ϕ_n não está definida para m).

Muitos resultados em computabilidade são provados com base em funções construídas à custa desta tabela, mais precisamente, à custa dos valores $\phi_0(0), \phi_1(1), \phi_2(2), \ldots$ da diagonal.

Proposição: Existe uma função total unária que não é URM-computável

Prova: A ideia é construir uma função total unária que seja diferente de todas as funções em C_1 (o conjunto das funções unárias URM-computáveis) e que portanto não pode ser computável. Mais precisamente, vai considerar-se uma função f que, para cada $k \in I\!N_0$, é diferente de ϕ_k no ponto k, ou seja, $f(n) \neq \phi_k(k)$.

Considere-se então a função $f : I\!N_0 \to I\!N_0$ tal que

$$f(n) = \begin{cases} \phi_n(n) + 1 & \text{se } \phi_n(n)\downarrow \\ 0 & \text{caso contrário} \end{cases}$$

Seja $k \in I\!N_0$. Tem-se que $f(k) \neq \phi_k(k)$ pois: (i) se $\phi_k(k)\downarrow$ então $f(k) = \phi_k(k) + 1$ e portanto $f(k) \neq \phi_k(k)$; (ii) se $\phi_k(k)\uparrow$ então $f(k)\downarrow$ (pois $f(k) = 0$) e portanto, de novo, $f(k) \neq \phi_k(k)$.

Conclui-se então que, para cada $k \in I\!N_0$, $f \neq \phi_k$ (dado que f e ϕ_k atribuem diferentes valores a k). A função f é assim distinta de todas as funções unárias URM-computáveis, como se pretendia, logo, naturalmente, f não é URM-computável.

3 O problema da paragem (*halting problem*) e o *printing problem*

Informalmente, o problema da paragem pode ser formulado do seguinte modo: dado um programa P arbitrário e um valor de entrada para P, determinar se P termina para esse valor de entrada. De um modo rigoroso, o problema da paragem pode ser formulado de diferentes formas, como se segue.

Definição: O PROBLEMA DA PARAGEM
O problema da paragem pode ser formulado do seguinte modo:

"dados $m, n \in I\!N_0$, determinar se $\phi_n(m)$ está definido".

Uma outra forma de formular o problema é:

$$\text{“dados } m, n \in \mathbb{N}_0\text{, determinar se } m \in W_n\text{”.}$$

Podem apresentar-se diferentes provas para o facto de o problema da paragem não ser decidível, isto é, para o facto de não ser possível encontrar um programa URM que, dados $n, m \in \mathbb{N}_0$, devolva, por exemplo, 1 se a computação do programa P_n a partir da configuração $< m >$ é finita e 0 caso contrário.

Proposição: O problema da paragem não é decidível, ou seja, o predicado " $\phi_n(m)$ está definido" não é decidível.

Prova: Vai fazer-se uma prova por absurdo. Suponha-se que o predicado é decidível. Então a sua função característica $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n,m) = \begin{cases} 1 & \text{se } \phi_n(m)\downarrow \\ 0 & \text{caso contrário} \end{cases}$$

é URM-computável. Considerando a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n) = \begin{cases} 0 & \text{se } c(n,n) = 0 \\ \text{não definida} & \text{caso contrário} \end{cases}$$

tem-se que se c é URM-computável então f também é.

Chega-se a uma contradição porque se consegue provar, por outro lado, que $f \neq \phi_k$ para todo $k \in \mathbb{N}_0$ (i. e., f é distinta de todas as funções unárias URM-computáveis), mostrando que, para cada $k \in \mathbb{N}_0$, $f(k) \neq \phi_k(k)$: (i) se $\phi_k(k)\downarrow$ então $c(k,k) = 1$, logo $f(k)\uparrow$ e portanto $f(k) \neq \phi_k(k)$; (ii) se $\phi_k(k)\uparrow$ então $c(k,k) = 0$, logo $f(k) = 0$ e portanto, de novo, $f(k) \neq \phi_k(k)$.

Conclui-se assim que c não pode ser URM-computável.

Como se referiu anteriormente, o problema da paragem pode também ser formulado em termos de W_n. Neste caso, para provar que o problema da paragem não é decidível, pode recorrer-se ao facto de o predicado "$n \in W_n$" não ser decidível. A não decidibilidade deste predicado é importante pois existem muitos outros predicados cuja não decidibilidade é provada pelo facto de "$n \in W_n$" não ser decidível.

Proposição: O predicado " $n \in W_n$ " não é decidível.

Prova: Vai fazer-se uma prova por absurdo. Suponha-se que o predicado é decidível. Então a sua função característica $c : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n) = \begin{cases} 1 & \text{se } n \in W_n \\ 0 & \text{se } n \notin W_n \end{cases}$$

é URM-computável. Considerando a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n) = \begin{cases} 0 & \text{se } c(n) = 0 \\ \text{não definida} & c(n) = 1 \end{cases}$$

tem-se que se c é URM-computável então f também é.

Chega-se a uma contradição porque se consegue provar, por outro lado, que $f \neq \phi_k$ para todo $k \in \mathbb{N}_0$ (i. e., f é distinta de todas as funções unárias URM-computáveis), mostrando que, para cada $k \in \mathbb{N}_0$, $f(k) \neq \phi_k(k)$: (i) se $\phi_k(k)\downarrow$ então $k \in W_k$, logo $c(k) = 1$ pelo que $f(k)\uparrow$ e portanto $f(k) \neq \phi_k(k)$; (ii) se $\phi_k(k)\uparrow$ então $k \notin W_k$ logo $c(k) = 0$ pelo que $f(k) = 0$ e portanto, de novo, $f(k) \neq \phi_k(k)$.

Conclui-se assim que c não pode ser URM-computável.

Proposição: O problema da paragem não é decidível, ou seja, o predicado " $n \in W_m$ " não é decidível.

Prova: Vai fazer-se uma prova por absurdo. Suponha-se que o predicado é decidível. Então a sua função característica $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n,m) = \begin{cases} 1 & \text{se } n \in W_m \\ 0 & \text{se } n \notin W_m \end{cases}$$

é URM-computável. Considerando a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n) = \begin{cases} 1 & \text{se } n \in W_n \\ 0 & n \notin W_n \end{cases}$$

tem-se que se c é computável então f também é pois $f(n) = c(n,n)$. Mas, pela proposição anterior, f não é computável, logo c também não o é.

Um outro resultado também importante é o que se segue. Este resultado estabelece que não é possível determinar se o valor n vai ser ou não obtido como *output* do programa P_n, isto é, mais precisamente, se existe algum $x \in \mathbb{N}_0$ tal que $\phi_n(x) = n$.

Proposição: O predicado " $n \in E_n$ " não é decidível.

Prova: Vai fazer-se uma prova por absurdo. Suponha-se que o predicado é decidível. Então a sua função característica $c : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n) = \begin{cases} 1 & \text{se } n \in E_n \\ 0 & \text{se } n \notin E_n \end{cases}$$

é URM-computável. Considerando a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n) = \begin{cases} n & \text{se } c(n) = 0 \\ \text{não definida} & c(n) = 1 \end{cases}$$

tem-se que se c é computável então f também é. Então $f = \phi_k$ para algum $k \in \mathbb{N}_0$. Tem-se que $k \in E_k$ sse k pertence ao contradomínio de f sse $c(k) = 0$ sse $k \notin E_k$, ou seja, chega-se a um absurdo. Conclui-se assim que c não pode ser computável.

A partir da proposição anterior pode provar-se um resultado mais geral: o predicado " $n \in E_m$ " não é decidível. Este resultado estabelece que não é possível determinar se um valor arbitrário n vai ser ou não obtido como *output* de um programa arbitário P_m, isto é, mais precisamente, se existe algum $x \in \mathbb{N}_0$ tal que $\phi_m(x) = n$. Este resultado é usualmente conhecido como a não decidibilidade do *printing problem*.

Proposição: O predicado " $n \in E_m$ " não é decidível.

Prova: Suponha-se que o predicado é decidível. Então a sua função característica $c : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n, m) = \begin{cases} 1 & \text{se } n \in E_m \\ 0 & \text{se } n \notin E_m \end{cases}$$

é URM-computável. Considerando a função $f : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n) = \begin{cases} 1 & \text{se } n \in E_n \\ 0 & n \notin E_n \end{cases}$$

tem-se que se c é computável então f também é pois $f(n) = c(n, n)$. Mas, pela proposição anterior, f não é computável, logo c também não o é.

4 O teorema da parametrização (para funções binárias) e as funções universais

Proposição: Teorema da parametrização (para funções binárias)
Seja f uma função binária URM-computável. Existe uma função total URM-computável $s : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(x, y) = \phi_{s(x)}(y)$$

Prova: Há que mostrar que é URM-computável a função $s : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$s(x) = c$$

onde c é o código do programa Q (ie, $\gamma^{-1}(c) = Q$) que calcula a função unária (ϕ_c) que a cada y faz corresponder $f(x, y)$, ie,

$$x \xrightarrow{s} c \quad \text{tal que} \quad y \xrightarrow{\phi_c} f(x, y)$$

Uma vez que f é URM-computável existe um programa F que calcula f. A ideia para calcular s é construir, a partir de F, o programa Q e a partir deste calcular o respectivo código c (usando γ).

O programa Q que se pretende é

$$\begin{array}{c} T(1,2) \\ Z(1) \\ S(1) \end{array}$$

$$\begin{array}{l} \ldots \\ S(1) \\ F \end{array}$$

onde após os dois primeiros comandos se tem x comandos $S(1)$.

Tem-se então que $\gamma(Q) = c$. Como γ é efectivamente computável, s é efectivamente computável. Pelo postulado de Church, s é URM-computável.

O teorema da parametrização para funções binárias assegura que, dada uma função binária URM-computável f, existe uma função URM-computável *unária* e *total* s que calcula as imagens $f(x, y)$ da função f da seguinte forma: dados os argumentos x e y, (i) $s(x)$ é o código de um certo programa Q e (ii) este programa Q calcula precisamente a função unária que a cada y faz corresponder $f(x, y)$ (ou seja, $Q(y) \downarrow f(x, y)$ se $f(x, y)$ está definido e $Q(y) \uparrow$ caso contrário).

O teorema da parametrização é por vezes também designado teorema *s-m-n* (por razões que não serão aqui detalhadas) e pode ser generalizado para funções URM-computáveis de aridade superior a 2 (ver Computability-an introduction to recursive function theory de N. Cutland, por exemplo).

Definição: Função universal para funções unárias
A função universal para funções unárias é a função $\Psi_U : \mathbb{N}_0^2 \to \mathbb{N}_0$ tal que

$$\Psi_U(x, y) = \phi_x(y).$$

De certo modo pode dizer-se que esta função incorpora todas as funções unárias computáveis $\phi_0, \phi_1, \ldots$ pois fixando o x, ao fazer variar y em $\Psi_U(x, y)$ está a calcular-se ϕ_x. Esta ideia pode ser generalizada como se segue.

Definição: Função universal de aridade n
A função universal para funções de aridade n, $n \in \mathbb{N}$, é a função de aridade $n + 1$

$$\Psi_U^n : \mathbb{N}_0^{n+1} \to \mathbb{N}_0$$

tal que

$$\Psi_U^n(x, x_1, \ldots, x_n)) = \phi_x^n(x_1, \ldots, x_n).$$

Usa-se Ψ_U para representar Ψ_U^1

Programa Universal: Será a função Ψ_U (ou, mais genericamente, Ψ_U^n) computável? Em caso afirmativo, qualquer programa P que calcule esta função pode ser visto como incorporando todos os outros programas: é um programa universal!

Proposição: Para cada $n \in \mathbb{N}_0$, Ψ_U^n é computável.

Prova: Mostra-se primeiro que Ψ_U^n é efectivamente computável. Dados $x \in \mathbb{N}_0$ e $x_1, \ldots, x_n$, calcula-se $\gamma^{-1}(x)$, isto é, o programa P cujo código é x. Assim, a computação $P(x_1, \ldots, x_n)$ permite encontrar o valor de $\phi_x^n(x_1, \ldots, x_n)$.

Conclui-se então que Ψ_U^n é efectivamente computável e, pelo postulado de Church, Ψ_U^n é computável[5].

5 Mais problemas não decidíveis

Proposição: O predicado "$\phi_n = \lambda x.0$" não é decidível.

Prova:

1. Considere-se a função $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$f(n,m) = \begin{cases} 0 & \text{se } n \in W_n \text{ (ie, } \phi_n(n)\downarrow) \\ \text{não definida} & \text{se } n \notin W_n \text{(ie, } \phi_n(n)\uparrow) \end{cases}$$

   A função f é computável pois $f(n,m) = z(\Psi_U(U_{2,1}(n,m), U_{2,1}(n,m)))$, com $z = \lambda x.0$, ou seja, f é obtida por composição de funções computáveis. Pelo teorema da parametrização para funções binárias, existe $s : \mathbb{N}_0 \to \mathbb{N}_0$ total e computável tal que

$$f(n,m) = \phi_{s(n)}(m)$$

   .

2. Tendo em conta a definição de f, tem-se que

$$\begin{array}{c} n \in W_n \\ \text{sse} \\ f(n,m) = 0 \text{ para cada } m \in \mathbb{N}_0 \\ \text{sse} \\ \phi_{s(n)}(m) \text{ para cada } m \in \mathbb{N}_0 \\ \text{sse} \\ \phi_{s(n)}(m) = \lambda x.0. \end{array}$$

3. Considere-se a função característica do predicado em causa, $c : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$c(n) = \begin{cases} 1 & \text{se } \phi_n = \lambda x.0 \\ 0 & \text{se } \phi_n \neq \lambda x.0 \end{cases}$$

   Se c for computável então também é computável a função $h : \mathbb{N}_0 \to \mathbb{N}_0$ tal que

$$h(n) = \begin{cases} 1 & \text{se } \phi_{s(n)} = \lambda x.0 \text{ (ie, } n \in W_n) \\ 0 & \text{se } \phi_{s(n)} \neq \lambda x.0 \text{ (ie, } n \notin W_n) \end{cases}$$

   pois $h(n) = c(s(n))$. Mas, por uma proposição anterior (o predicado "$n \in W_n$" não é decidível) h não é computável e por isso c também não é.

[5] Em *Computability-an introduction to recursive function theory*, N. Cutland, Cambridge University Press, 1980, página 86 (e seguintes) apresenta-se uma prova axiomática de que Ψ_U^n é computável (ou seja, mostra-se que Ψ_U^n é parcial recursiva).

Proposição: Seja θ uma função computável. O predicado "$\phi_n = \theta$" não é decidível.

Prova: A prova é semelhante à prova relativa à não decidibilidade de "$\phi_n = \lambda x.0$" bastando substituir $\lambda x.0$ por θ. Mais precisamente, há que fazer seguintes alterações: (i) em 1., 0 é substituído por $\theta(m)$ na definição de f; (ii) em 1., $z(\Psi_U(U_{2,1}(n,m), U_{2,1}(n,m)))$ é substituído por

$$U_{2,2}(\Psi_U(U_{2,1}(n,m), U_{2,1}(n,m)), \theta((U_{2,2}(n,m)));$$

(iii) em 2. e 3., $\lambda x.0$ é substituído por θ.

Proposição: O predicado "$\phi_n = \phi_m$" não é decidível.

Prova: Considere-se a função característica do predicado $c : I\!N_0 \times I\!N_0 \to I\!N_0$ tal que

$$c(n,m) = \begin{cases} 1 & \text{se } \phi_n = \phi_m \\ 0 & \text{se } \phi_n \neq \phi_m \end{cases}$$

e seja $P =< Z(1) >$. Tem-se que $\gamma(P) = 0$ e $\phi_0 = \lambda x.0$. Se c é computável então também é computável a função $f : I\!N_0 \to I\!N_0$ tal que

$$f(n) = g(n,0) = \begin{cases} 1 & \text{se } \phi_n = \phi_0 (= \lambda x.0) \\ 0 & \text{se } \phi_n \neq \phi_0 (= \lambda x.0) \end{cases}$$

Mas por uma proposição anterior, f não é computável e portanto c também não o é.

6 O teorema de Rice

O teorema da parametrização foi utilizado (directa ou indirectamemente) na secção anterior para provar que certos predicados não são decidíveis (e.g.: (i) saber se a função unária calculada por um programa arbitrário é ou não a função nula; (ii) dada uma função computável θ, saber se a função unária calcula por um programa arbitário é ou não igual a θ; (iii) saber se dois programas arbitrários calculam ou não a mesma função unária).

Mas este teorema da parametrização pode ser utilizado para provar um resultado bastante mais geral $-$ o *teorema de Rice*$-$ sendo os resultados de não decidibilidade acima referidos, por exemplo, consequência simples deste novo teorema.

Proposição (Teorema de rice):
Seja B um conjunto de funções unárias URM-computáveis (i.e., $B \subseteq C_1$) tal que $B \neq \emptyset$ e $B \neq C_1$. O predicado "$\phi_n \in B$" não é decidível.

Prova: Seja *indf* : $I\!N_0 \to I\!N_0$ a função unária tal que, para cada $n \in I\!N_0$, *indf*(n) não está definido. Facilmente se conclui que *indf*(n) é URM-computável.

(A) Suponha-se que $indf \notin B$ e seja $g \in B$ (existe uma tal função porque B não é vazio). Considere-se a função binária $f : I\!N_0 \times I\!N_0 \to I\!N_0$ tal que

$$f(n,m) = \begin{cases} g(m) & \text{se } n \in W_n \\ \text{não definida} & \text{se } n \notin W_n \end{cases}$$

a qual, como facilmente se conclui, é efectivamente computável e portanto URM-computável. Pelo teorema da parametrização, existe uma função unária, total e URM-computável $s : I\!N_0 \to I\!N_0$ tal que $f(n,m) = \phi_{s(n)}(m)$. Assim tem-se que

– se $n \in W_n$ então $\phi_{s(n)}(m) = g(m)$ para cada $m \in I\!N_0$, isto é, $\phi_{s(n)} = g$ e portanto $\phi_{s(n)} \in B$

– se $n \notin W_n$ então $\phi_{s(n)}(m)$ é função unária que está indefinida para cada $m \in I\!N_0$, isto é, $\phi_{s(n)} = indf$ e portanto $\phi_{s(n)} \notin B$

e portanto, para cada $n \in I\!N_0$, $n \in W_n$ sse $\phi_{s(n)} \in B$.

Considere-se agora função característica do predicado "$\phi_n \in B$", ou seja, a função $c : I\!N_0 \to I\!N_0$ tal que

$$c(n) = \begin{cases} 1 & \text{se } \phi_n \in B \\ 0 & \text{se } \phi_n \notin B \end{cases}$$

e suponha-se que c é computável. Então é também computável a função $c' : I\!N_0 \to I\!N_0$ tal que

$$c'(n) = \begin{cases} 1 & \text{se } n \in W_n \\ 0 & \text{se } n \notin W_n \end{cases}$$

pois $c'(n) = c(s(n))$ para cada $n \in I\!N_0$. Mas, por uma proposição anterior, c' não é computável e portanto c também não o é. Conclui-se assim que, se $indf \notin B$, então o predicado "$\phi_n \in B$" não é decidível.

(B) Suponha-se agora $indf \in B$ e considere-se o conjunto $\overline{B} = C_1 \backslash B$. Tem-se que $\overline{B}$ não é vazio porque, por hipótese, $B \neq C_1$ e, naturalmente, $indf \notin \overline{B}$. Raciocinando como em (A), mas agora relativamente a $\overline{B}$, conclui-se que o predicado "$\phi_n \in \overline{B}$" não é decidível.

Como facilmente se conclui, se o predicado "$\phi_n \in B$" fosse decidível, a função c referida em (A) era computável e portanto seria também computável a função $c'' : I\!N_0 \to I\!N_0$ tal que

$$c''(n) = \begin{cases} 1 & \text{se } c(n) = 0 \\ 0 & \text{se } c(n) = 1 \end{cases}$$

que é a função característica do predicado "$\phi_n \in \overline{B}$". Mas concluiu-se acima que c'' não é computável e portanto c também não o é. Deste modo, se $indf \in B$, então também o predicado "$\phi_n \in B$" não é decidível.

Utilizando o teorema de Rice é possível fazer uma prova da não decidibilidade do predicado "$\phi_n = \lambda x.0$" mais simples do que a anteriormente apresentada.

Proposição: O predicado "$\phi_n = \lambda x.0$" não é decidível.

Prova: A função $\lambda x.0$ pertence a C_1. Sendo $B = \{\lambda x.0\}$, pelo teorema de Rice, o predicado "$\phi_n \in B$" não é decidível logo, o predicado "$\phi_n = \lambda x.0$" não é decidível.

Outros resultados de indecidibilidade apresentados na secção anterior podem também ter uma prova mais simples que a apresentada se se recorrer ao teorema de Rice. Deixa-se como exercício estas provas.