

und.1 O Problema da Parada

tur:und:hal:
sec

Suponha que tenhamos fixado alguma enumeração das descrições de máquinas de Turing. Cada máquina de Turing recebe assim um *índice*: seu lugar na enumeração $M_1, M_2, M_3, \dots$ das descrições de máquinas de Turing. explanation

Sabemos que devem existir funções não-Turing-computáveis: o conjunto de descrições de máquinas de Turing—e, portanto, o conjunto de máquinas de Turing—é **enumerável**, mas o conjunto de todas as funções de $\mathbb{N}$ a $\mathbb{N}$ não é. Mas também podemos encontrar exemplos específicos de funções não computáveis. Uma dessas funções é a função de parada.

Definição und.1 (Função de parada). A *função de parada* h é definida como

$$h(e, n) = \begin{cases} 0 & \text{se a máquina } M_e \text{ não para na entrada } n \\ 1 & \text{se a máquina } M_e \text{ para na entrada } n \end{cases}$$

Definição und.2 (Problema da parada). O *Problema da Parada* é o problema de determinar (para quaisquer e, n) se a máquina de Turing M_e para com uma entrada de n traços.

Mostramos que h não é Turing-computável mostrando que uma função relacionada, s , não é Turing-computável. Esta prova baseia-se no fato de que tudo o que pode ser computado por uma máquina de Turing pode ser computado por uma máquina de Turing disciplinada (??), e no fato de que duas máquinas de Turing podem ser conectadas para criar uma única máquina (??). explanation

Definição und.3. A função s é definida como

$$s(e) = \begin{cases} 0 & \text{se a máquina } M_e \text{ não para na entrada } e \\ 1 & \text{se a máquina } M_e \text{ para na entrada } e \end{cases}$$

Lema und.4. A função s não é Turing-computável.

Prova. Suponha, por contradição, que a função s seja Turing-computável. Então haveria uma máquina de Turing S que computa s . Podemos supor, sem perda de generalidade, que quando S para, ela o faz examinando o primeiro quadrado (isto é, que ela é disciplinada). Esta máquina pode ser “conectada” a outra máquina J , que para se for iniciada na entrada 0 (isto é, se ela lê 0 no estado inicial enquanto examina o quadrado à direita do símbolo de fim de fita), e caso contrário vagueia para a direita, sem nunca parar. $S \frown J$, a máquina criada conectando-se S a J , é uma máquina de Turing, logo é M_e para algum e (isto é, aparece em algum lugar da enumeração). Inicie M_e em uma entrada de e . Há duas possibilidades: ou M_e para ou não para.

1. Suponha que M_e pare para uma entrada de e . Então $s(e) = 1$. Assim, S , quando iniciada em e , para com um único 1 como saída na fita. Então J começa com um 1 na fita. Nesse caso, J não para. Mas M_e é a máquina

$S \cup J$, de modo que ela deveria fazer exatamente o que S seguida de J faria (isto é, neste caso, vaguear para a direita e nunca parar). Logo, M_e não pode parar para uma entrada de e 1.

2. Agora suponha que M_e não pare para uma entrada de e 1. Então $s(e) = 0$, e S , quando iniciada na entrada e , para com uma fita em branco. J , quando iniciada em uma fita em branco, para imediatamente. Novamente, M_e faz o que S seguida de J faria, de modo que M_e deve parar para uma entrada de e 1.

Em cada caso chegamos a uma contradição com nossa suposição. Isso mostra que não pode haver uma máquina de Turing S : s não é Turing-computável. $\square$

Teorema und.5 (Insolubilidade do Problema da Parada). *O problema da parada é insolúvel, isto é, a função h não é Turing-computável.*

*tur:und:hal:
thm:halting-problem*

Prova. Suponha que h fosse Turing-computável, digamos, por uma máquina de Turing H . Poderíamos usar H para construir uma máquina de Turing que computa s : primeiro, fazer uma cópia da entrada (separada por um símbolo 0). Depois, voltar ao início e executar H . Claramente podemos fazer uma máquina que realiza a primeira tarefa (veja ??), e se H existisse, poderíamos “conectá-la” a tal máquina copiadora para obter uma nova máquina que determinaria se M_e para na entrada e , isto é, computa s . Mas já mostramos que não pode existir tal máquina. Portanto, h também não é Turing-computável. $\square$

Problema und.1. O problema da Parada com Três (3-Parada) é o problema de dar um procedimento de decisão para determinar se uma máquina de Turing escolhida arbitrariamente para ou não para uma entrada de três 1 em uma fita que, fora isso, está em branco. Prove que o problema 3-Parada é insolúvel.

Problema und.2. Mostre que se o problema da parada é solúvel para pares de máquina de Turing e entrada M_e e n em que $e \neq n$, então ele também é solúvel para os casos em que $e = n$.

Problema und.3. Provamos que o problema da parada é insolúvel se a entrada é um número e , que identifica uma máquina de Turing M_e por meio de uma enumeração de todas as máquinas de Turing. E se permitirmos a descrição de máquinas de Turing de ?? diretamente como entrada? Pode haver uma máquina de Turing que decide o problema da parada, mas que recebe como entrada descrições de máquinas de Turing em vez de índices? Explique por que sim ou por que não.

Problema und.4. Mostre que a função *parcial* s' definida como

$$s'(e) = \begin{cases} 1 & \text{se a máquina } M_e \text{ para na entrada } e \\ \text{indefinida} & \text{se a máquina } M_e \text{ não para na entrada } e \end{cases}$$

é Turing-computável.

Créditos das Imagens

Referências Bibliográficas