

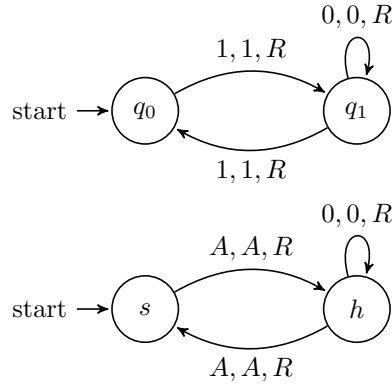


Figura 1: Variantes da máquina *Par*

tur:und:enu:
fig:variants

und.1 Enumerando Máquinas de Turing

tur:und:enu:
sec

Podemos mostrar que o conjunto de todas as máquinas de Turing é **enumerável**. Isso decorre do fato de que cada máquina de Turing pode ser finitamente descrita. O conjunto de estados e o vocabulário da fita são conjuntos finitos. A função de transição é uma função parcial de $Q \times \Sigma$ a $Q \times \Sigma \times \{L, R, N\}$ e, portanto, também pode ser especificada listando-se seus valores para os finitos pares de argumentos para os quais está definida.

explanation

Isso é verdade até certo ponto, mas há uma diferença sutil. A definição de máquinas de Turing não fez restrição alguma sobre quais **membros** o conjunto de estados e o alfabeto da fita podem ter. Assim, por exemplo, para todo número real, há tecnicamente uma máquina de Turing que usa esse número como um estado. Contudo, o *comportamento* da máquina de Turing é independente de quais objetos servem como estados e vocabulário. Considere as duas máquinas de Turing na **Figura 1**. Esses dois diagramas correspondem a duas máquinas, M com o alfabeto de fita $\Sigma = \{\triangleright, 0, 1\}$ e conjunto de estados $\{q_0, q_1\}$, e M' com alfabeto $\Sigma' = \{\triangleright, 0, A\}$ e estados $\{s, h\}$. Mas, quanto ao resto, suas instruções são as mesmas: M para em uma sequência de n 1 se, e somente se, n é par, e M' para em uma sequência de n A se, e somente se, n é par. Tudo o que fizemos foi renomear 1 para A , q_0 para s e q_1 para h . Esse exemplo se generaliza: podemos considerar máquinas de Turing como sendo a mesma desde que uma resulte da outra por tal renomeação de símbolos e estados. Na verdade, podemos simplesmente pensar nos símbolos e estados de uma máquina de Turing como inteiros positivos: em vez de σ_0 pense 1, em vez de σ_1 pense 2, etc.; $\triangleright$ é 1, 0 é 2, etc. Dessa forma, a máquina *Par* torna-se a máquina representada na **Figura 2**. Poderíamos chamar de máquina *padrão* uma máquina de Turing cujos estados e símbolos são inteiros positivos,

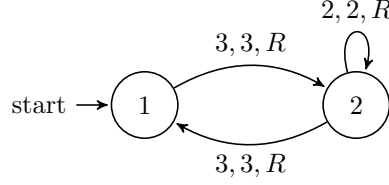


Figura 2: Uma máquina *Par* padrão

tur:und:enu:
fig:standard-even

e considerar apenas máquinas padrão de agora em diante.¹

Queríamos mostrar que o conjunto de máquinas de Turing é **enumerável** e, tendo em mente as considerações acima, basta mostrar que o conjunto de máquinas de Turing padrão é **enumerável**. Suponha que nos seja dada uma máquina de Turing padrão $M = \langle Q, \Sigma, q_0, \delta \rangle$. Como poderíamos descrevê-la usando uma cadeia finita de inteiros positivos? Primeiro listaremos o número de estados, os próprios estados, o número de símbolos, os próprios símbolos e o estado inicial. (Lembre-se, todos esses são inteiros positivos, já que M é uma máquina padrão.) E quanto a δ ? O conjunto de argumentos possíveis, isto é, pares $\langle q, \sigma \rangle$, é finito, já que Q e Σ são finitos. Assim, a informação em δ é simplesmente a lista finita de todas as 5-uplas $\langle q, \sigma, q', \sigma', d \rangle$ em que $\delta(q, \sigma) = \langle q', \sigma', D \rangle$, e d é um número que codifica a direção D (digamos, 1 para L , 2 para R e 3 para N).

Dessa forma, toda máquina de Turing padrão pode ser descrita por uma lista finita de inteiros positivos, isto é, como uma sequência $s_M \in (\mathbb{Z}^+)^*$. Por exemplo, a máquina *Par* padrão é codificada pela sequência

$$\underbrace{2, 1, 2, 3}_{Q}, \underbrace{1, 2, 3, 1}_{\Sigma}, \underbrace{1, 3, 2, 3, 2}_{\delta(1,3)=\langle 2,3,R \rangle}, \underbrace{2, 2, 2, 2, 2}_{\delta(2,2)=\langle 2,2,R \rangle}, \underbrace{2, 3, 1, 3, 2}_{\delta(2,3)=\langle 1,3,R \rangle} .$$

Teorema und.1. *Há funções de $\mathbb{N}$ a $\mathbb{N}$ que não são Turing-computáveis.*

Prova. Sabemos que o conjunto de sequências finitas de inteiros positivos $(\mathbb{Z}^+)^*$ é **enumerável** (??). Isso nos dá que o conjunto de descrições de máquinas de Turing padrão, como um subconjunto de $(\mathbb{Z}^+)^*$, é ele próprio enumerável. Toda função Turing-computável de $\mathbb{N}$ a $\mathbb{N}$ é computada por alguma (na verdade, muitas) máquina de Turing. Renomeando seus estados e símbolos para inteiros positivos (em particular, $\triangleright$ como 1, 0 como 2 e 1 como 3), podemos ver que toda função Turing-computável é computada por uma máquina de Turing padrão. Isso significa que o conjunto de todas as funções Turing-computáveis de $\mathbb{N}$ a $\mathbb{N}$ é também enumerável.

¹A terminologia “máquina padrão” não é padrão.

Por outro lado, o conjunto de todas as funções de $\mathbb{N}$ a $\mathbb{N}$ não é **enumerável** (??). Se todas as funções fossem computáveis por alguma máquina de Turing, poderíamos enumerar o conjunto de todas as funções listando todas as descrições de máquinas de Turing que as computam. Logo, há algumas funções que não são Turing-computáveis. $\square$

Problema und.1. Você consegue pensar em uma maneira de descrever máquinas de Turing que não exija que os estados e os símbolos do alfabeto sejam listados explicitamente? Você pode definir sua própria noção de máquina “padrão”, mas diga algo sobre por que toda máquina de Turing pode ser computada por uma máquina “padrão” no seu novo sentido.

Créditos das Imagens

Referências Bibliográficas