

und.1 Representando Máquinas de Turing

tur:und:rep:
sec

A fim de representar máquinas de Turing e seu comportamento por meio de a **sentença** da lógica de primeira ordem, temos de definir uma linguagem adequada. A linguagem consiste em duas partes: **símbolos de predicado** para descrever configurações da máquina, e expressões para numerar os passos de execução (“momentos”) e as posições na fita.

explanation

Introduzimos dois tipos de **símbolos de predicado**, ambos de 2 lugares: para cada estado q , a **símbolo de predicado** Q_q , e para cada símbolo de fita σ , a **símbolo de predicado** S_σ . Os primeiros permitem-nos descrever o estado de M e a posição de seu cabeçote de fita; os segundos permitem-nos descrever o conteúdo da fita.

A fim de exprimir as posições do cabeçote de fita e o número de passos executados, precisamos de uma maneira de exprimir números. Isso é feito usando a **símbolo de constante** o e uma função de 1 lugar ι , a função sucessor. Por convenção, ela é escrita *depois* de seu argumento (e omitimos os parênteses).

Para cada número n há um termo canônico $\bar{n}$, o *numeral* de n , que o representa em $\mathcal{L}_M$. $\bar{0}$ é o , $\bar{1}$ é o' , $\bar{2}$ é o'' , e assim por diante. Mais formalmente:

$$\begin{aligned}\bar{0} &= o \\ \overline{n+1} &= \bar{n}'\end{aligned}$$

O termo $\bar{0}$, isto é, o , nomeia a posição mais à esquerda na fita, bem como o instante anterior ao primeiro passo de execução (a configuração inicial). O termo $\bar{1}$, isto é, o' , nomeia o quadrado à direita do quadrado mais à esquerda, e o instante posterior ao primeiro passo de execução, e assim por diante.

Também introduzimos a **símbolo de predicado** $<$ para exprimir tanto a ordenação das posições na fita (quando significa “à esquerda de”) quanto a dos passos de execução (caso em que significa “antes”).

Uma vez que tenhamos a linguagem montada, listamos os “axiomas” de $\tau(M, w)$, isto é, as **sentenças** que, tomadas em conjunto, descrevem o comportamento de M quando executada na entrada w . Haverá **sentenças** que estabelecem condições sobre o , ι e $<$, **sentenças** que descrevem a configuração de entrada, e **sentenças** que descrevem qual é a configuração de M após executar uma instrução particular.

tur:und:rep:
defn:tm-descr

Definição und.1. Dada uma máquina de Turing $M = \langle Q, \Sigma, q_0, \delta \rangle$, a linguagem $\mathcal{L}_M$ consiste em:

1. Um **símbolo de predicado** de dois lugares $Q_q(x, y)$ para todo estado $q \in Q$. Intuitivamente, $Q_q(\bar{m}, \bar{n})$ exprime “após n passos, M está no estado q examinando o m -ésimo quadrado.”
2. Um **símbolo de predicado** de dois lugares $S_\sigma(x, y)$ para todo símbolo $\sigma \in \Sigma$. Intuitivamente, $S_\sigma(\bar{m}, \bar{n})$ exprime “após n passos, o m -ésimo quadrado contém o símbolo σ .”
3. Um **símbolo de constante** o

4. Um **símbolo de função** de um lugar $'$
5. Um **símbolo de predicado** de dois lugares $<$

As **sentenças** que descrevem a operação da máquina de Turing M na entrada $w = \sigma_{i_1} \dots \sigma_{i_k}$ são as seguintes:

1. Axiomas que descrevem números e $<$:

- a) A **sentença** que diz que todo número é menor que seu sucessor:

$$\forall x \, x < x'$$

- b) A **sentença** que garante que $<$ é transitivo:

$$\forall x \, \forall y \, \forall z \, ((x < y \wedge y < z) \rightarrow x < z)$$

2. Axiomas que descrevem a configuração de entrada:

- a) Após 0 passos—antes de a máquina começar— M está no estado inicial q_0 , examinando o quadrado 1:

$$Q_{q_0}(\bar{1}, \bar{0})$$

- b) Os primeiros $k + 1$ quadrados contêm os símbolos $\triangleright, \sigma_{i_1}, \dots, \sigma_{i_k}$:

$$S_{\triangleright}(\bar{0}, \bar{0}) \wedge S_{\sigma_{i_1}}(\bar{1}, \bar{0}) \wedge \dots \wedge S_{\sigma_{i_k}}(\bar{k}, \bar{0})$$

- c) Caso contrário, a fita está vazia:

$$\forall x \, (\bar{k} < x \rightarrow S_0(x, \bar{0}))$$

3. Axiomas que descrevem a transição de uma configuração para a seguinte:

Para o que segue, seja $\varphi(x, y)$ a conjunção de todas as **sentenças** da forma

$$\forall z \, (((z < x \vee x < z) \wedge S_{\sigma}(z, y)) \rightarrow S_{\sigma}(z, y'))$$

em que $\sigma \in \Sigma$. Usamos $\varphi(\bar{m}, \bar{n})$ para exprimir “exceto no quadrado m , a fita após $n + 1$ passos é a mesma que após n passos.”

- a) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', R \rangle$, a **sentença**:

$$\begin{aligned} \forall x \, \forall y \, ((Q_{q_i}(x, y) \wedge S_{\sigma}(x, y)) \rightarrow \\ (Q_{q_j}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y))) \end{aligned}$$

tur:und:rep:
rep-right

Isto diz que se, após y passos, a máquina está no estado q_i examinando o quadrado x que contém o símbolo σ , então após $y + 1$ passos ela está examinando o quadrado $x + 1$, está no estado q_j , o quadrado x agora contém σ' , e todo quadrado distinto de x contém o mesmo símbolo que continha após y passos.

tur:und:rep:
rep-left

b) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', L \rangle$, a **sentença**:

$$\begin{aligned} & \forall x \forall y ((Q_{q_i}(x', y) \wedge S_\sigma(x', y)) \rightarrow \\ & \quad (Q_{q_j}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ & \forall y ((Q_{q_i}(\bar{0}, y) \wedge S_\sigma(\bar{0}, y)) \rightarrow \\ & \quad (Q_{q_j}(\bar{0}, y') \wedge S_{\sigma'}(\bar{0}, y') \wedge \varphi(\bar{0}, y))) \end{aligned}$$

Reserve um momento para pensar em como isso funciona: agora não começamos com “se examinando o quadrado $x \dots$ ”, mas: “se examinando o quadrado $x + 1 \dots$ ”. Um movimento para a esquerda significa que no próximo passo a máquina está examinando o quadrado x . Mas o quadrado em que se escreve é $x + 1$. Fazemos assim porque não temos subtração nem uma função predecessor.

Note que os números da forma $x + 1$ são 1, 2, $\dots$, isto é, isso não cobre o caso em que a máquina está examinando o quadrado 0 e deveria mover-se para a esquerda (o que, é claro, não pode fazer—ela simplesmente permanece onde está). Esse caso especial é coberto pela segunda conjunção: ela diz que se, após y passos, a máquina está examinando o quadrado 0 no estado q_i e o quadrado 0 contém o símbolo σ , então após $y + 1$ passos ela ainda está examinando o quadrado 0, está agora no estado q_j , o símbolo no quadrado 0 é σ' , e os quadrados distintos do quadrado 0 contêm os mesmos símbolos que continham após y passos.

tur:und:rep:
rep-stay

c) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', N \rangle$, a **sentença**:

$$\begin{aligned} & \forall x \forall y ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \rightarrow \\ & \quad (Q_{q_j}(x, y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y))) \end{aligned}$$

Seja $\tau(M, w)$ a conjunção de todas as **sentenças** acima para a máquina de Turing M e a entrada w .

A fim de exprimir que M eventualmente para, temos de encontrar uma **sentença** que diga “após algum número de passos, a função de transição será indefinida.” Seja X o conjunto de todos os pares $\langle q, \sigma \rangle$ tais que $\delta(q, \sigma)$ é indefinida. Seja então $\alpha(M, w)$ a **sentença**

$$\exists x \exists y \left(\bigvee_{\langle q, \sigma \rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y)) \right)$$

Se usarmos uma máquina de Turing com um estado de parada designado h , é ainda mais fácil: então a **sentença** $\alpha(M, w)$

$$\exists x \exists y Q_h(x, y)$$

exprime que a máquina eventualmente para.

tur:und:rep:
prop:mlessk

Proposição und.2. Se $m < k$, então $\tau(M, w) \models \bar{m} < \bar{k}$

Prova. Exercício.

□

Problema und.1. Prove **Proposição und.2.** (Dica: use indução em $k - m$).

Créditos das Imagens

Referências Bibliográficas