MT e Problemas de decisão

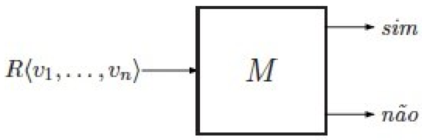
Como vimos no início do curso, a solução de um problema de decisão é um algoritmo que, para toda instância do problema, fornece a resposta correta. Pela tese de Church-Turing, traduzimos essa afirmação como: existe uma máquina de Turing que para em estado final quando a resposta for sim, e em estado não final quando a resposta for não. Ou seja, um problema de decisão é solucionável se a linguagem de todas as suas instâncias para as quais a resposta é sim for uma linguagem recursiva.

A implicação disso, como discutimos anteriormente, é que precisamos definir uma representação para um problema de decisão, usando algum alfabeto. Essa representação deve ser tal que:

- Toda instância do problema deve ser representada por ao menos uma palavra de Σ^*.
- Toda palavra de Σ^* representa uma única instância do problema.
- Sempre é possível verificar se uma palavra representa ou não uma instância.

Usaremos a notação $R\langle i\rangle$ para denotar a representação de uma instância i de um problema de decisão P. Se essa instância consistir em vários valores de parâmetros, então esse serão separados por vírgula como $R\langle v_1, \quad v_2, \ldots, \quad v_n\rangle$.

Dessa forma, a MT que solucione um problema de decisão será denotada esquematicamente por:



As saídas, nesse caso, representam a parada da máquina em um estado final ou não para a palavra de entrada.

Exemplo: Considere o problema de decisão de determinar se uma palavra $w \in L(G)$ para uma GLC G.

Nesse caso, precisamos representar tanto a palavra quanto a própria gramática para serem entregues ao algoritmo solucionador. A representação da gramática requer a representação das variáveis, terminais e regras. A mesma representação dos terminais pode ser usada na representação da palavra.

Assim, representamos uma variável X_i em unário 1^i. Naturalmente, escolhemos 1 para representar a variável de partida. Cada terminal a_j é representado por $1^{|V|+j}$. As regras, por sua vez, são traduções diretas dos respectivos códigos usando o 0 como separador. Ou seja, uma regra $r: \quad X_i \to a_j X_i$ é representada por $R\langle r\rangle = 1^i 01^{|V|+j} 01^i$.

Como as variáveis e terminais são representados em unário, só precisamos saber quantos elementos pertencem a cada um desses conjuntos na representação da gramática. As regras podem ser separadas por 00 e colocadas em sequência. Logo, a representação de uma gramática seria: $R\langle (V, \Sigma, R, P)\rangle = 1^{|V|} 01^{|\Sigma|} 0R\langle r_1\rangle 00R\langle r_2\rangle \ldots 00R\langle r_{|R|}\rangle$.

Por fim, a representação de uma instância $R\langle G, w\rangle = R\langle G\rangle 000R\langle w\rangle$. Sendo que a representação da palavra será a sequência dos terminais que a compõe separados por 0s.

Agora considere a instância desse problema "determinar se $aba \in L(H)$ " em que a (regras da) gramática são mostradas abaixo.

$$A \to aAa \mid B$$
$$B \to aB \mid CC$$
$$C \to b \mid \lambda$$

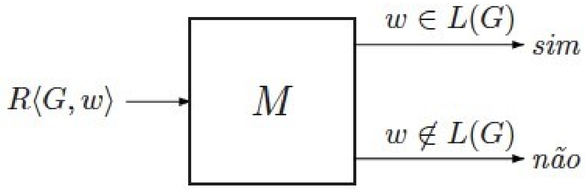
As variáveis seriam representadas por $A{:}1, \quad B{:}11, \quad C{:}111$; os terminais $a{:}1^4, \quad b{:}1^5$; e a regra

$A \to aAa$ representada por $R\langle r_1 \rangle = 101^4 0101^4$.

Seguindo essa lógica, a instância seria representada por:

$R\langle H, aba \rangle = 1^3 01^2 0R\langle r_1 \rangle 00R\langle r_2 \rangle 00R\langle r_3 \rangle 00R\langle r_4 \rangle 00R\langle r_5 \rangle 00R\langle r_6 \rangle 0001^4 01^5 01^4$.

Uma MT (algoritmo) que solucione o problema seria como:



Logo, a MT M, se existir, seria capaz de reconhecer a linguagem $\{R\langle G, w \rangle | \quad w \in L(G)\}$.

■

Similarmente ao posto acima, poderíamos questionar se uma palavra pertencia pode ser reconhecida por uma máquina de Turing; ou seja, se ela pertence à sua linguagem. O problema de decisão, no caso, seria, dadas uma MT M e uma palavra w, $w \in L(M)$? Assim como criamos uma representação para uma GLC e uma palavra, teríamos que definir também uma representação para MTs. Isso nos leva a um conceito importante: o de máquinas de Turing universais.

O poder das máquinas de Turing é tamanho que elas são capazes de simular quaisquer outras MTs, inclusive elas próprias. Esse poder computacional ficou bastante evidente quando demonstramos que as diferentes variantes eram equivalentes à MT padrão.

Para solucionar problemas de decisão envolvendo MTs, precisamos definir uma representação para elas. Somente dessa forma podemos simulá-las por outra máquina.
A simulação de uma MT requer essencialmente duas informações: quais são os estados finais; e suas transições. Logo, a representação de uma MT requer a representação dessas duas informações.

Seja $M = (Q, \Sigma, \Gamma, \langle, \sqcup, \delta, i, F)$ uma MT arbitrária. Podemos representar os estados de Q e os símbolos do alfabeto de fita como:

Estado	Representação
$e_1 = i$	1
e_2	11
$\vdots$	$\vdots$
e_n	1^n

Símbolo de Γ	Representação
$a_1 = \langle$	1
$a_2 = \sqcup$	11
$\vdots$	$\vdots$
a_k	1^k

Assim, podemos representar os estados finais por:

- $R\langle F \rangle = R\langle f_1 \rangle 0R\langle f_2 \rangle 0 \ldots 0R\langle f_{|F|} \rangle$

As transições, por sua vez, podem ser representadas por:

- $t_i : \delta(e, a) = [e', b, \quad d] \Rightarrow R\langle t_i \rangle = R\langle e \rangle 0R\langle a \rangle 0R\langle e' \rangle 0R\langle b \rangle 0R\langle d \rangle$

As movimentações do cabeçote podem ser representadas por $D{:}1 \quad e \quad E{:}11$.

Logo, a representação da máquina seria:

- $R\langle M \rangle = R\langle F \rangle 00R\langle t_1 \rangle 00R\langle t_2 \rangle 00 \ldots 00R\langle t_s \rangle$

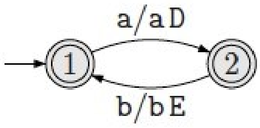
A representação da palavra faz uso da representação dos símbolos do alfabeto de fita, já que $\Sigma \subset \Gamma$.

Por fim, a representação de uma instância de um problema de decisão envolvendo uma MT e uma palavra seria:

- $R\langle M, w\rangle = R\langle M\rangle 000 R\langle w\rangle$

Segue um exemplo de representação de MT.

Exemplo: Obtenha uma representação para a MT abaixo e a palavra ab.



Estado	Representação
1	1
2	11

Símbolo	Representação
⟨	1
⊔	11
a	111
b	1111

$R\langle M\rangle = 101100101^{3}01001101^{4}011$

$R\langle M, w\rangle = 101100101^{3}01001101^{4}0110001^{3}01^{4}$

■

Uma **máquina de Turing universal** (MTU) é uma MT que simula qualquer outra MT. Ou seja, Uma MTU é aquela que recebe como entrada uma representação, $R\langle M, w\rangle$ de uma MT M e uma palavra w de entrada para essa máquina, e simula a computação de M com w.

Uma MTU U será especificada com 3 fitas. A primeira contém a entrada de U, ou seja, $R\langle M, w\rangle$. A segunda fita fará o papel da fita de M. E a terceira funcionará como o registrador de M, isto é, armazenará a representação do estado atual de M. O algoritmo abaixo descreve o funcionamento da MTU U.

1. copie $R\langle w\rangle$ na fita 2 e posicione cabeçote no início;
2. escreva $R\langle i\rangle$ na fita 3 e posicione cabeçote no início;
3. **ciclo**
   - 3.1 seja $R\langle a\rangle$ a representação sob o cabeçote da fita 2;
   - 3.2 seja $R\langle e\rangle$ a representação sob o cabeçote da fita 3;
   - 3.3 procure $R\langle e\rangle 0 R\langle a\rangle 0 R\langle e'\rangle 0 R\langle a'\rangle 0 R\langle d\rangle$ na fita 1;
   - 3.4 se encontrou **então**
     - 3.4.1 substitua $R\langle e\rangle$ por $R\langle e'\rangle$ na fita 3 e volte cabeçote da fita 3 ao seu início;
     - 3.4.2 substitua $R\langle a\rangle$ por $R\langle a'\rangle$ na fita 2;
     - 3.4.3 mova cabeçote da fita 2 na direção d
   - 3.5 **senão**
     - 3.5.1 se e é estado final **então**
       - pare em estado final
     - **senão**
       - pare em estado não final
     - **fimse**
   - **fimse**

   fimciclo.

Em resumo, inicialmente U copia a palavra de entrada de M na fita 2. Depois escreve o estado i de M na fita 3, e finalmente simula as transições M, olhando o estado na fita 3 e o símbolo (palavra) na fita 2.

A linguagem reconhecida pela máquina universal U é portanto $\{R\langle M,w\rangle \mid \; w \in L(M)\}$.

Se a máquina M reconhecesse por parada, então poderíamos simplificar sua representação, eliminando a representação dos estados finais. Nesse caso, a linguagem dessa MTU U_p (para diferenciá-la da máquina em que M é padrão) seria:

- $L(U_p) = \{R\langle M,w\rangle \mid \; M \;\; para \;\; ao \;\; computar \;\; w\}$

Em outras palavras, U_p para com $R\langle M,w\rangle$ se, e somente se, M para com w.

Vamos usar essa representação para identificar linguagens que não são LRec; ou seja, não são decidíveis, e posteriormente para definir a fronteira do que pode ou não ser computado pela MT.