MAC 4722 - Linguagens, Autômatos e Computabilidade

Rodrigo Augusto Dias Faria - NUSP 9374992
Departamento de Ciência da Computação - IME/USP

8 de junho de 2016

Lista 6

L6.1 (Sipser 3.06) No Teorema 3.21 mostramos que uma linguagem é Turing-reconhecível sse algum enumerador a enumera. Por que não usamos o seguinte algoritmo mais simples para a direção de ida da prova? Tal qual anteriormente, $s_1, s_2, \ldots$ é uma lista de todas as cadeias em Σ^*.

$E =$ "Ignore a entrada.

1. Repita o que se segue para $i = 1, 2, 3, \ldots$

2. Rode M sobre s_i.

3. Se ela aceita, imprima s_i."

Resposta: O algoritmo indica que devemos rodar M sobre todas as cadeias possíveis de Σ^*. A mudança sugerida no enunciado faz com que M execute em uma cadeia s_i por vez e, caso M aceite s_i, o enumerador E imprime-na. Ocorre que M pode entrar em *loop* para uma cadeia s_k qualquer e, por conseguinte, nenhuma outra cadeia subsequente a s_k será impressa por E e, portanto, a linguagem de M será diferente do conjunto de cadeias listadas por E.

L6.2 (Sipser 3.08) Dê descrições a nível de implementação de máquinas de Turing que decidem as linguagens abaixo sobre o alfabeto $\{0, 1\}$.

a. $\{w \mid w$ contém o mesmo número de 0s e 1s$\}$

Resposta: Vamos chamar de M_a a MT que decide a linguagem em **a.** Note que o primeiro passo deve aceitar a cadeia vazia pois, neste caso, o número de 0s e 1s é igual a zero.

$M_a =$ "Sobre a cadeia de entrada w:

1. Se w é a cadeia vazia, *aceite*, caso contrário, vá para o passo 2.

2. Faça uma varredura na fita e marque o primeiro 0 que ainda não esteja marcado. Se nenhum 0 desmarcado foi encontrado, *rejeite*. Retorne a cabeça para a extremidade esquerda da fita.

3. Faça uma varredura na fita e marque o primeiro 1 que ainda não esteja marcado. Se a varredura não encontrou nenhum 1 desmarcado, *rejeite*. Retorne a cabeça para a extremidade esquerda da fita.
4. Faça uma nova varredura na fita. Se um 0 ou um 1 desmarcado for encontrado, mova a cabeça para a extremidade esquerda da fita e retorne ao passo 2, caso contrário, *aceite*.

Referência na resposta do livro [1].

b. $\{w \mid w$ contém duas vezes mais 0s que 1s$\}$

Resposta: Vamos chamar de M_b a MT que decide a linguagem em **b.** A estratégia utilizada aqui é similar à linguagem do item **a.**

$M_b =$ "Sobre a cadeia de entrada w:

1. Repita por duas vezes o próximo estágio:
2. Faça uma varredura na fita e marque o primeiro 0 que ainda não esteja marcado. Se nenhum 0 desmarcado foi encontrado, *rejeite*. Retorne a cabeça para a extremidade esquerda da fita.
3. Faça uma varredura na fita e marque o primeiro 1 que ainda não esteja marcado. Se a varredura não encontrou nenhum 1 desmarcado, *rejeite*. Retorne a cabeça para a extremidade esquerda da fita.
4. Faça uma nova varredura na fita. Se um 0 ou um 1 desmarcado for encontrado, mova a cabeça para a extremidade esquerda da fita e retorne ao passo 1, caso contrário, *aceite*.

c. $\{w \mid w$ **não** contém duas vezes mais 0s que 1s$\}$

Resposta: Vamos chamar de M_c a MT que decide a linguagem em **c.** A estratégia utilizada aqui é aplicar a MT obtida em **b.** como uma subrotina.

$M_c =$ "Sobre a cadeia de entrada w:

1. Rode w na máquina M_b.
2. Se M_b aceita w, *rejeite*, senão, *aceite*.

L6.3 (Sipser 3.14) Um ***autômato com fila*** é como um autômato com pilha, exceto que a pilha é substituída por uma fila. Uma ***fila*** é uma fita que permite que símbolos sejam escritos somente na extremidade esquerda e lidos somente da extremidade direita. Cada operação de escrita (denominá-la-emos *empurrar*) adiciona um símbolo na extremidade esquerda da fila e cada operação de leitura (denominá-la-emos *puxar*) lê e remove um símbolo na extremidade direita. Como com um `AP`, a entrada é colocada numa fita de entrada de somente-leitura separada, e a cabeça sobre a fita de entrada pode mover somente da esquerda para a direita. A fita de entrada contem uma célula com um símbolo em branco após a entrada, de modo que essa extremidade da entrada possa ser detectada. Um autômato com fila aceita sua entrada entrando num estado especial de aceitação em qualquer momento. Mostre que uma linguagem pode ser reconhecida por um autômato com fila determinístico **sse** a linguagem é Turing-reconhecível.

Resposta: Vamos denominar de *cauda* a extremidade esquerda da fila onde a operação *empurrar* escreve símbolos. Analogamente, vamos denominar de *cabeça* a extremidade direita da fila onde a operação *puxar* lê e remove símbolos.

Demonstração. Para provar que uma linguagem pode ser reconhecida por um autômato com fila determinístico **sse** a linguagem é Turing-reconhecível, nós devemos mostrar que uma máquina de Turing e um autômato com fila são equivalentes e, para tanto, temos de demonstrar o seguinte:

1. Dada uma máquina de Turing M, podemos gerar um autômato com fila determinístico A que reconhece a mesma linguagem de M.

   Para provar que podemos gerar A a partir de M, temos que mostrar que é possível simular todas as operações de M com o autômato A. Seja $a, b \in \Gamma$ de M. Logo, temos que simular as transições de M para a direita (D) e esquerda (E) com as operações possíveis em A.

   Primeiro, vamos escrever um símbolo, digamos \$, para indicar onde está a cabeça da fita de M, de forma tal que em P temos $aw_1 \dots w_i \$ w_{i+1} \dots w_n$, onde a é o símbolo sob a cabeça da fita de M, $w_1 \dots w_i$ é o conteúdo da fita à direita de a e $w_{i+1} \dots w_n$ o conteúdo à esquerda de a.

   $a \to b, D$

   Nesse caso, basta *puxar* a da fila de A e *empurrar* o símbolo b na *cauda*. Essa operação simula a transição à direita de M, uma vez que a cabeça da fita se move para o próximo caractere da fita. Note que, ao encontrar o símbolo \$, nós devemos desfazer esse movimento.

   $a \to b, E$

   Esse movimento implica que devemos mover o caractere na *cauda* da fila para a *cabeça*. Seja o conteúdo da fila neste instante como $w_1 \dots w_n \$ a$ e a o símbolo que vamos movimentar. Primeiro, vamos *empurrar* um símbolo, digamos # para indicar a posição atual da fita, sendo assim, temos $w_1 \dots w_n \$ a$#. Depois, vamos *puxar* e *empurrar* todos os símbolos da fila de P até atingir # novamente e, então, nós descartamos este caractere pois todo o conteúdo da fila foi devidamente deslocado para a direita, o que resulta em $aw_1 \dots w_n \$$. Sendo assim, temos a simulação da transição à esquerda em M.

2. Dado um autômato com fila determinístico A, podemos gerar uma máquina de Turing M que reconhece a mesma linguagem de A.

   Nessa direção da prova nós devemos mostrar que M pode simular A ao produzir o mesmo efeito das operações de *empurrar* e *puxar*. Seja a o símbolo que vamos manipular.

   (a) No caso de *empurrar*, basta que a cabeça da fita de M percorra a entrada e escreva a ao encontrar o primeiro símbolo $\sqcup$.

(b) Para simular a operação de *puxar*, a cabeça da fita é movida até a extremidade esquerda, onde está o símbolo \$, e marca o símbolo a após \$ com uma marca especial, tal como $\ddot{a}$, indicando que ele foi removido.

Dessa forma nós podemos concluir que uma linguagem pode ser reconhecida por um autômato com fila determinístico **sse** a linguagem é Turing-reconhecível.

■

L6.4 (Sipser 3.16) Mostre que a coleção de linguagens Turing-reconhecíveis é fechada sob a operação de união, concatenação, estrela e intersecção.

Resposta: Sejam L_1 e L_2 linguagens Turing-reconhecíveis e M_1 e M_2 máquinas de Turing que reconhecem L_1 e L_2, respectivamente. Vamos mostrar que a classe de linguagens Turing-reconhecíveis é fechada sob as seguintes operações:

a. união

Vamos construir uma máquina de Turing M capaz de reconhecer a união $L_1 \cup L_2$ da seguinte forma:

$M =$ "Sobre a cadeia de entrada w:

1. Repita o seguinte para $i = 1, 2, 3, \ldots$

2. Rode as máquinas M_1 e M_2 sobre w por i passos. Se M_1 ou M_2 aceita w, *aceite*. Se M_1 e M_2 param e rejeitam w, então *rejeite*."

A máquina construída reconhece cadeias de $L_1 \cup L_2$, pois se M_1 ou M_2 aceitam w, em algum momento a máquina M aceitará w, já que a máquina vai, passo a passo, tentando reconhecer w em M_1 e M_2 simultaneamente. M poderá entrar em *loop* se M_1 ou M_2 entrar em *loop* e ambas rejeitarem w.

Referência na resposta do livro [1].

b. concatenação

Vamos construir uma máquina de Turing M capaz de reconhecer a concatenação $L_1 L_2$. Seja w uma cadeia de comprimento n, tal que pode ser segmentada em duas partes. Seja $p_i = x_i y_{n-i}$ cada uma dessas possíveis partições, onde $i = 0, 1, \ldots, n$ representa a quantidade de caracteres de w (a partir do início) no primeiro segmento da partição e $n - i$ representa a quantidade de caracteres de w (a partir de $i + 1$) no segundo segmento da partição. Vamos inserir na fita da máquina M cada uma das p_i partições separadas por um símbolo, digamos #, da forma $\#p_0\#p_1\#\ldots p_n\#$, e cada segmento da partição p_i separado por um outro caractere, digamos β, da forma $x_i \beta y_{n-i}$, ou seja, a fita ficaria da forma $\#x_0\beta y_n\#x_1\beta y_{n-1}\#\ldots\#x_n\beta y_0$. O que faremos é rodar a máquina M_1, simultaneamente, em todos os segmentos x_i de cada partição p_i e a máquina M_2 em todos os segmentos y_{n-i} de cada partição p_i. Se M_1 e M_2 aceitar alguma partição p_i, onde M_1 aceita o primeiro segmento e M_2 aceita o segundo, então M *aceita* w. Se para todo i, M_1 e M_2 rejeitar p_i, onde M_1 rejeita o primeiro segmento ou M_2 rejeita o segundo, então M *rejeita* w. A máquina M pode entrar em *loop* se $w \notin L_1 L_2$ e M_1 ou M_2 entrar em *loop* na tentativa de reconhecimento das partições.

c. estrela

Vamos construir uma máquina de Turing M capaz de reconhecer a operação estrela L_1^*. Seja w uma cadeia de comprimento n, tal que pode ser particionada em m partes. Seja $p_i = w'_{i1}w'_{i2}\ldots w'_{im}$, para $i = 0, 1, \ldots$ cada uma dessas possíveis partições. Vamos inserir na fita da máquina M cada uma das p_i partições separadas por um símbolo, digamos #, da forma #p_0#p_1#$\ldots p_n$#, e cada parte da partição p_i separada por um outro caractere, digamos β, da forma $w'_{i1}\beta w'_{i2}\ldots$, ou seja, a fita ficaria da forma #$w'_{01}\beta w'_{02}\beta \ldots w'_{0m}$#$w'_{11}\beta w'_{12}\beta \ldots$#. O que faremos é rodar a máquina M_1 simultaneamente em todas as partes $w'_{i1}w'_{i2}\ldots w'_{im}$ de cada partição p_i. Se M_1 aceitar alguma partição p_i, onde M_1 aceita cada uma das partes de p_i, então M *aceita* w. Se para todo i, M_1 rejeitar p_i, então M *rejeita* w. A máquina M pode entrar em *loop* se $w \notin L_1^*$ e M_1 entrar em *loop* na tentativa de reconhecimento das partições.

d. intersecção

Vamos construir uma máquina de Turing M capaz de reconhecer a intersecção $L_1 \cap L_2$ da seguinte forma:

$M =$ "Sobre a cadeia de entrada w:

1. Repita o seguinte para $i = 1, 2, 3, \ldots$

2. Rode as máquinas M_1 e M_2 sobre w por i passos. Se M_1 e M_2 aceitam w, *aceite*. Se M_1 ou M_2 para e rejeita w, então *rejeite*."

A máquina construída aceita cadeias de $L_1 \cap L_2$, pois se $w \in L_1 \cap L_2$, significa que $\exists w \mid w \in L_1$ e $w \in L_2$. Como a máquina testa w em M_1 e M_2, e só aceita caso w seja aceita por ambas, temos que a máquina M reconhece as cadeias de $L_1 \cap L_2$. M poderá entrar em *loop* se M_1 ou M_2 entrar em *loop* e rejeitar w.

Note que, no caso do fechamento sob concatenação e estrela, como existe apenas uma quantidade finita de fatorações de w, uma máquina de Turing pode tentar todas as possibilidades em um número finito de passos.

Referências

[1] Michael Sipser. *Introduction to the theory of computation*. Boston: Thomson Course Technology, 2006. ISBN: 0534950973.