Exercício Extra para prova valendo 1.0 ponto.

Principais diferenças do modelo apresentado para o de sipser

- A Máquina de Turing do Divério (DTM) é uma 8-tupla onde:
  $M = (\Sigma, Q, \Pi, q_0, F, V, \beta, \odot)$

- O alfabeto da fita é a união de Σ(Entrada), V(auxiliar), β(Branco) e $\odot$(marcador de início)

- A fita possui um símbolo para marcar explicitamente o início da fita e a entrada só é escrita após esse marcador.

- Π é uma função parcial. Uma transição não definida é uma das formas de rejeição. Enquanto $\delta: Q \times \Gamma \to Q \times \Gamma \times \{E, D\}$ é uma função total

- A MT de sipser só para com dois estados (q_{aceita}, $q_{rejeita}$). Já a MT de Divério possui os 3 cenários do enunciado.

Teorema: A classe de linguagem reconhecidas pelas MT de Sipser é equivalente à classe de linguagens reconhecidas pelas MT de Divério.

Prova $\boxed{\rightarrow}$

Objetivo: Mostrar que para toda MTS (Máquina de Turing de Sipser) existe uma MTD (Máquina de Turing de Divério) tal que L(MTS)=L(MTD)

/ /

Construção:

Dada uma MTS:

$M_S = (Q_S, \Sigma_S, \Gamma_S, \delta_S, q_{0S}, q_{aceita_S}, q_{rejeita_S})$.

Constrói-se uma MTD:

$M_d = (Q_d, \Sigma_d, \Pi_d, q_{0d}, F_d, V_d, \beta_d, \odot_d)$ com:

Alfabetos:

$\Sigma_d = \Sigma_S$

$\beta_d = \sqcup$

$V_d = \Gamma_S \setminus (\Sigma_S \cup \{\sqcup\})$

$\odot$ marcador de início

Estados:

$Q_d = Q_S$

$q_{0d} = q_{0S}$

$F_d = \{q_{aceita_S}\}$

Função de transição:

. Para cada $\delta_S(q, a) = (p, b, D)$ com $q \neq q_{aceita_S}$ e $q \neq q_{rejeita_S}$ defino $\Pi_d(q, a) = (p, b, D)$.

. ACEITAÇÃO: se δ_S leva a q_{aceita_S}, é definido Π_d de forma que ele leve a este estado final em M_d.

. REJEIÇÃO: Se δ_S leva a $q_{rejeita_S}$, é omitida a transição em Π_d. Assim, M_d rejeita pela indefinição pois já foi dito explicitamente no enunciado que a M_d se comporta de tal maneira.

. Borda Esquerda (Simulação):

No sipser ao tentar mover o cabeçote para a esquerda da célula 0, a cabeça fica parada. Na M_d, marca-se o início com $\odot$ e para ficar parado adiciona-se a regra $\Pi_d(p, \odot) = (p, \odot, D)$. Assim ao tentar mover para a esquerda o cabeçote volta para $\odot$.

Conclusão ida: A Md aceita e rejeita exatamente nas mesmas condições que M_S.

Logo, $L(Md) = L(M_s)$

PROVA $\boxed{\Leftarrow}$

Ideia: dada uma MTd, construimos uma MTs equivalente

Construção

Dada uma MTd:

$Md = (\Sigma d, Qd, \Pi d, q_{0}d, Fd, Vd, \beta d, \odot d)$

Constrói-se uma MTs:

$M_S = (Q_S, \Sigma_S, \Gamma_S, \delta_S, q_{0S}, q_{aceita_S}, q_{rejeita_S})$ com

Alfabeto:

$\Sigma_S = \Sigma d$

$\Gamma_S = \Sigma d \cup Vd \cup \{\beta d, \odot d\}$

Estados:

$Q_S = Qd \cup \{q_{setup}, q_{aceita_S}, q_{rejeita_S}$, estados auxiliares$\}$

$q_{0_S} = q_{setup}$

q_{aceita_S} e $q_{rejeita_S}$ são estados especiais de parada

Fase de Preparação da Fita:

- A entrada chega como w e precisa se tornar $\odot w$.

Passo a Passo:

. A maquina percorre a fita até encontrar o branco após o último símbolo de w.

Paula Branca

/ /

- A partir daí desloca-se os símbolos da direita para a esquerda:
  1. lê o último símbolo, guarda no estado,
  2. move à direita, escreve o símbolo guardado,
  3. volta à esquerda e repete tudo

- Após deslocar tudo a célula $\odot$ está livre, assim a máquina retorna ao início e escreve $\odot$

- Por fim move a cabeça para o primeiro símbolo de entrada e vai para o estado q_0d

SIMULAÇÃO (ajustação):

- Para cada transição $\pi d(q,a)=(p,d,M)$
  - Se $p \in F_d \rightarrow \delta_s(q,a)=(q_{aceita_s}, b, M)$
  - Se $p \notin F_d \rightarrow \delta_s(q,a)=(p,b,M)$
  - Se $\pi d(q,a)$ é indefinida $\rightarrow \delta_s(q,a)=(q_{rejeita_s}, a, R)$
  - Se a máquina tenta mover para $\odot \rightarrow \delta_s(p,\odot)=(q_{rejeitas}, \odot, R)$

Conclusão Volta: Após a preparação, a M_s simula M_d, fazendo com que quando uma aceita/rejeita a outra também. Logo, $L(M_s)=L(M_d)$

Conclusão: Mostramos que toda MTS pode ser simulada por uma MTD e que toda MTD pode ser simulada por uma MTS.

Portanto: $\{L(M_s) \mid M_s \text{ é MTS}\} = \{L(M_d) \mid M_d \text{ é MTD}\}$

Ambas as definições são computacionalmente equivalentes, reconhecendo exatamente as linguagens Turing-reconhecíveis.