LFA - PARTE 5

Variedades Adicionais das Máquinas de Turing

1

Máquinas de Turing com uma Fita Infinita de um Sentido

- A fita da máquina é infinita apenas à direita
- O quadrado da fita mais à esquerda contém o marcador de final esquerda $
- Computação de função em um sentido
- Aceitação de linguagem em um sentido (reconhecimento)

2

M de um Sentido Computa $f(n_1, n_2, \ldots n_k)$

Inicialmente

$$\$\boldsymbol{q_0}\underbrace{1111\ldots11111}_{n_1+1\ 1s}B\underbrace{11111\ldots1111}_{n_2+1\ 1s}B\ldots\ldots B\underbrace{1111\ldots11111}_{n_k+1\ 1s}B$$

Por último

$$\$\boldsymbol{q}_{\textbf{pára}}\underbrace{1111\ldots1111111}_{f(n_1,\, n_2,\, \ldots\, n_k)\,+\,1\ 1s}B$$

3

M de um Sentido Aceita (Rejeita) *abb*

Inicialmente

$\$q_0abb$

Por último

$\$q_{pára}1$ $(\$q_{pára}0)$

- M de um sentido aceita L apenas no caso de M de um sentido aceitar todas e apenas as palavras em L
- M de um sentido reconhece L apenas no caso de (1) M de um sentido aceitar todas as palavras em L e (2) M de um sentido rejeitar todas as palavras em L^c

4

Resultados de Equivalência (⇔)

- **Teorema.** Seja f uma k-ésima função numérico-teorética. Então existe uma máquina de Turing com uma fita infinita de dois sentidos que computa f nos dois sentidos se e somente se existir uma máquina de Turing com uma fita infinita de um sentido que computa f em um sentido.
- **Teorema.** Seja L uma linguagem sobre o alfabeto X. Então L é aceita (reconhecida) por uma máquina de Turing com fita infinita de dois sentidos se e somente se L for aceita (reconhecida) por uma máquina de Turing com uma fita infinita de um sentido.

5

Aceitação de Linguagem pelo Estado Terminal

- Nenhum conceito de computação de função
- Um único estado terminal q_t (⇔ "um ou mais")
- M aceita w se pára em q_t
- *M aceita a linguagem L pelo estado terminal* desde que M aceite, pelo estado terminal, todas e apenas as palavras de L.
- "Aceitação pelo *1*"

6

Exemplos de Linguagens Turing-Aceitáveis pelo Estado Terminal

- Palíndromos
- Parênteses Casados

7

Resultado de Equivalência

- **Teorema.** Seja *L* uma linguagem sobre o alfabeto Σ. Então existe uma máquina de Turing que aceita *L* "pelo *1*" se e somente se existir uma máquina de Turing que aceita *L* pelo estado terminal.
- Ambos ⇐ e ⇒ mantêm os novos marcadores de final $ e ¢.

8

Máquinas de Turing Multi-fitas

- Fita de entrada superior (possivelmente *read-only*)
- Fita de saída inferior (possivelmente *write-only*)
- 0 ou mais fitas de trabalho entre elas

9

n div 2 (sem fitas de trabalho)

Inicialmente

fita de entrada: `H11111111`

fita de saída: `HB`

Por último pára na configuração

fita de entrada: `H11111111` (não mudada)

fita de saída: `H1111`

10

Esclarecimento

- **Lembrança.** Seja *M* uma máquina de Turing multi-fitas. Suponha que a fita *t* é uma das $k + 2$ fitas de *M*. Em geral, a determinação de qual ação será tomada pela cabeça de leitura/escrita na fita *t* dependerá dos símbolos correntemente percorridos nas *outras* fitas e não meramente no símbolo correntemente percorrido na própria fita *t*.

11

Resultados de Equivalência

- **Teorema.** Uma linguagem *L* é aceita por alguma máquina de Turing multi-fitas se e somente se *L* for aceita por alguma máquina de Turing de fita única.
- **Teorema.** Uma linguagem *L* é reconhecida por alguma máquina de Turing multi-fitas se e somente se *L* for reconhecida por alguma máquina de Turing de fita única.

12

Um Outro Resultado de Equivalência

- **Teorema.** Similarmente, uma função *f* é computada por alguma máquina de Turing multi-fitas se e somente se for computada por alguma máquina de Turing de única fita.
- ⇐ sempre trivial
- Idéia de prova para ⇒: O alfabeto de fita do simulador de fita única *M* será grande; um símbolo *a* representará uma “seção” inteira através das fitas da máquina multi-fitas *M´*

13

Codificação de Máquinas de Turing

- Qualquer máquina de Turing representada por um código de número natural
- Codificação estilo ASCII
- Codificação Euler-Gödel

14

ASCII

- Cada máquina terá vários códigos ASCII (ordem das quádruplas de instruções)
- O *alfabeto de descrição da máquina de Turing* Ψ
- A maioria dos números naturais não são códigos ASCII
- Se *n* é um código, então é o código de uma máquina única (propriedade de recuperação).

15

Todo Número É Um Código

- Pode-se assumir que todo número natural é a codificação de alguma máquina de Turing única de acordo com o esquema de codificação estilo ASCII (ou qualquer outro esquema de codificação efetivo)
- Idéia: Use a posição do código de *M* dentro da seqüência de todos os códigos; a própria posição torna-se o novo código de *M*

16

Máquinas de Turing Universais

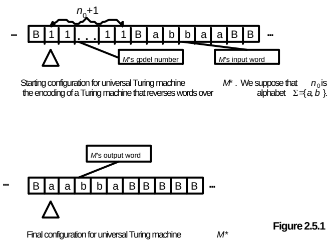


Figure 2.5.1

17

MTU como Computadores de Funções

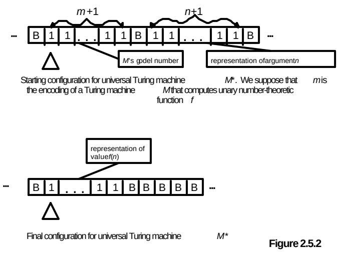


Figure 2.5.2

18

MTUs Existem

- Turing (1936)
- Computadores por programa armazenado (EDVAC/1946)
- Sistemas operacionais modernos
- Plasticidade do hardware

Máquina de Turing Não Determinística

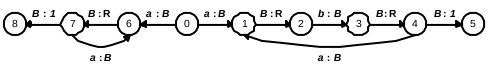


Figure 2.6.1

Aceitação de Linguagem

- Aceita-se *w* se ***alguma*** computação de *M* resulta em uma aceitação de *1*
- Em geral haverá computações de “não aceitação” mesmo no caso de um *w* aceito
- *M* aceita *L* se e somente se *M* aceitar todas e apenas as palavras de *L*

Exemplos

- $L = \{a^{2n} | n \geq 0\} \cup \{a^{3n} | n \geq 0\}$ contendo λ, *aa*, *aaa*, *aaaa*, a^6, a^8, a^9, …
- $L = \{a^{n^2} | n \geq 0\}$ contendo λ, *a*, *aaaa*, a^9, …

22

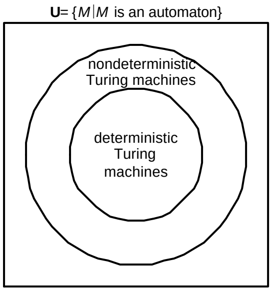
Funções de Transição

- No caso de uma máquina não determinística *M*, tem-se que d_M não é de valor único, e.g., $d_M(q_0, a) = (q_1, B, R)$ e $d_M(q_0, a) = (q_6, B, R)$.
- Fala-se de um *mapeamento* de transição.
- Máquinas determinísticas dentre as máquinas não determinísticas com mapeamentos de transição de valor único.
- Determinismo = um caso especial de não determinismo.

23

O Mundo dos Autômatos

U= {*M* | *M* is an automaton}



24

Item Teórico Importante

- Item teórico importante: a relação entre um conceito de computabilidade de Turing determinística e um conceito de computabilidade de Turing não determinística.
- A disponibilidade de escolhas não deveria aumentar a potência computacional?

25

A Resposta é Não

- Suponha que *M* não determinística aceita a linguagem *L*. É possível efetivamente enumerar todas as seqüências finitas de instruções possíveis de *M* para a palavra de entrada dada *w*.
- Isto seria enumerar todos os caminhos de comprimentos finitos possíveis a partir de q_0 para a palavra de entrada *w*.
- **Teorema.** Existe alguma máquina não determinística M_{nd} que aceita *L* se e somente se existir alguma determinística M_d que aceita *L*.
- $\Leftarrow$ trivial; para $\Rightarrow$ seja M_d seqüências de instruções enumeradas de M_{nd}

26

Reconhecimento de Linguagem

- Não determinística *M* reconhece *L*
- Se $w \in L$, então existe uma computação de aceitação de *M* para a entrada *w*
- Se $w \notin L$, então existe alguma computação de rejeição de *M* para a entrada *w*
- Para nenhuma palavra *w* existe uma computação de aceitação **e** uma de rejeição por parte de *M*. Meta: determinância.

27

Computação de Funções

- A máquina não determinística M computa f (unária)
- Existe computação de M tal que M começa percorrendo o símbolo mais a esquerda de $n + 1$ *1*s e pára percorrendo o símbolo mais a esquerda de $f(n) + 1$ *1*s
- Não há computação de M tal que M começa percorrendo o símbolo mais a esquerda de $n + 1$ *1*s e pára percorrendo o símbolo mais a esquerda de $m + 1$ *1*s para qualquer $m \neq f(n)$. Meta: Determinância.

28

Turing-Computabilidade

- **Teorema.** Seja f uma função numérico-teorética. Então f é computada por alguma máquina não determinística M_{nd} se e somente se f for computada por alguma máquina determinística M_d.
- $\Leftarrow$ trivial; para $\Rightarrow$ seja M_d seqüências de instruções enumeradas de M_{nd}

29

Computabilidade/Não Computabilidade

- Toda função numérico-teorética de Turing é computável?
- Não (veja abaixo).
- Isto significa que há funções não computáveis? (Depende)

30

MT_n (com $n \geq 0$)

MT_n = classe das máquinas de Turing determinísticas, de fita única, de (n + 1) estados, com alfabeto de fita {*1*}

Todas as máquinas começam percorrendo um branco em uma fita completamente em branco.

31

A Grande Questão

- Se começar percorrendo um quadrado em uma fita completamente em branco, *M* pára e, se acontecer, quantos *1*s estão na fita quando *M* parar?
- Chama-se este número de *produtividade* de *M*
- Se *M* nunca pára, então sua produtividade é 0.

32

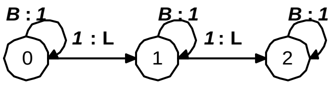
$\Theta(n)$

- $\Theta(n)$ = a produtividade máxima de qualquer elemento de MT_n
- $\Theta(n)$ não é Turing-Computável.

33

$\Theta(2) > 2$ e, em geral,
$\Theta(n) > n$

Figure 2.7.3



34