6 MÁQUINAS UNIVERSAIS E A HIPÓTESE DE CHURCH

6.1 Equivalência entre as Máquinas de Turing e Norma

6.2 Modificações sobre a Máquina de Turing

6.2.1 Máquina de turing não-deterministica
6.2.2 Máquina de turing com fita infinita à esquerda e à direita
6.2.3 Máquina de turing com múltiplas fitas
6.2.4 Outras modificações sobre a máquina de turing

6.3 Hipótese de Church

6.4 Conclusões

6.5 Exercícios

6 MÁQUINAS UNIVERSAIS E HIPÓTESE DE CHURCH

6.1 Equivalência entre as Máquinas de Turing e Norma

Prova-se que a Máquina de Turing é equivalente à Máquina Norma.

- Reforçam-se as evidências de que ambas são Máquinas Universais
- *Lembre-se de que, no conceito de simulação, é necessário considerar funções de codificação e decodificação para permitir comparar máquinas com diferentes conjuntos de entrada e saída.*
- Resumidamente, a prova é como segue:

a) *Turing ≤ Norma.*

- A estrutura de fita da Máquina de Turing é simulada em Norma usando uma estrutura de arranjo unidimensional;

b) *Norma ≤ Turing.*

- Conforme observado anteriormente os registradores X e Y são suficientes para realizar qualquer processamento em Norma.
- A Máquina de Turing pode simular os dois registradores X e Y
  - O conteúdo de cada registrador (valor natural) é implementado de forma unária em Turing;
  - O registrador X ocupa as células pares da fita,
  - O registrador Y ocupa as impares maiores de 1.

Teorema 6.1

Máquina de Turing ≤ Máquina Norma.

- ***O formalismo Máquina de Turing pode ser simulado pelo formalismo Máquina Norma.***

PROVA

- Suponha uma Máquina de Turing $M = (\Sigma, Q, \Pi, q0, F, V, ß, ©)$.
- Então, a simulação de M por um programa P em Norma pode ser definida como segue:

Fita

- A fita é codificada como um arranjo unidimensional em X, sendo que cada célula da fita corresponde a uma posição do arranjo.
- O símbolo de cada célula é codificado como um número natural como segue: para um alfabeto $\Sigma = \{a1, a2, \ldots, an\}$, o símbolo ai é codificado como o natural i, e os símbolos especiais ß e © como zero e n+1, respectivamente.

Estados

- Para os estados de $Q = \{q_0, q_1, \ldots, q_n\}$ em M, o programa P possui correspondentes instruções rotuladas por $2^{0+1}, 2^{1+1}, \ldots, 2^{n+1}$;
- O rótulo inicial de P é $2 = 2^{0+1}$ (pois q_0 é o estado inicial de M), e, para qualquer $q_f \in F$, 2^{f+1} é rótulo que antecede ao rótulo final, preparando os dados para a função de saída em Norma;

Estado Corrente

- O estado corrente de M é simulado em Norma usando o registrador Q, o qual assume valores em $\{ 2^{0+1}, 2^{1+1}, \ldots, 2^{n+1} \}$ (correspondendo aos estados q0, q1, ..., qn);

Cabeça da Fita

- A posição corrente da cabeça da fita de M é simulada usando o registrador C de Norma, o qual contém a posição corrente do arranjo em X e é inicializado com o valor 1;

Função Programa

- A função programa de M pode ser simulada por um programa P de Norma
- Uma transição de M da forma: $\Pi(qu, ar) = (qv, as, m)$

onde m assume valores em { E, D } (esquerda e direita) é simulada pelo trecho de programa:

2^{u+1} : faça $A := 3^{Q}\square 5^{X(C)}$ vá_para End_A
...
a: faça X(C):= s vá_para a+1 grava na fita
a+1: faça adC vá_para a+2 move a cabeça
a+2: faça $Q:= 2^{v+1}$ vá_para End_Q novo estado

Observe que:

1) *No programa, é suposto que o movimento da cabeça da fita é para a direita, e, portanto é adicionado 1 ao registrador C; caso o movimento seja para a esquerda, é necessário subtrair 1;*

2) *A transição depende do estado corrente q_u e do símbolo lido a_r. Assim, na instrução rotulada por 2^{u+1}, é especificado um desvio incondicional para uma instrução rotulada pelo par (2^{u+1}, r), usando uma codificação de primos, similar a introduzida no estudo da máquina Norma;*

3) *As macros End_A e End_Q referem-se ao endereçamento indireto definido anteriormente. O conteúdo do registrador A é denotado por a.*

Rótulo final

- a cada estado final q_f associa-se um rótulo 2^{f+1}, antecessor do rótulo final, corresponde o seguinte trecho de programa em P, o qual especifica que o conteúdo de X ("fita") é atribuído a Y (pois em Norma a função de saída retorna o valor do registrador Y), preparando os dados para a função de saída de Norma:

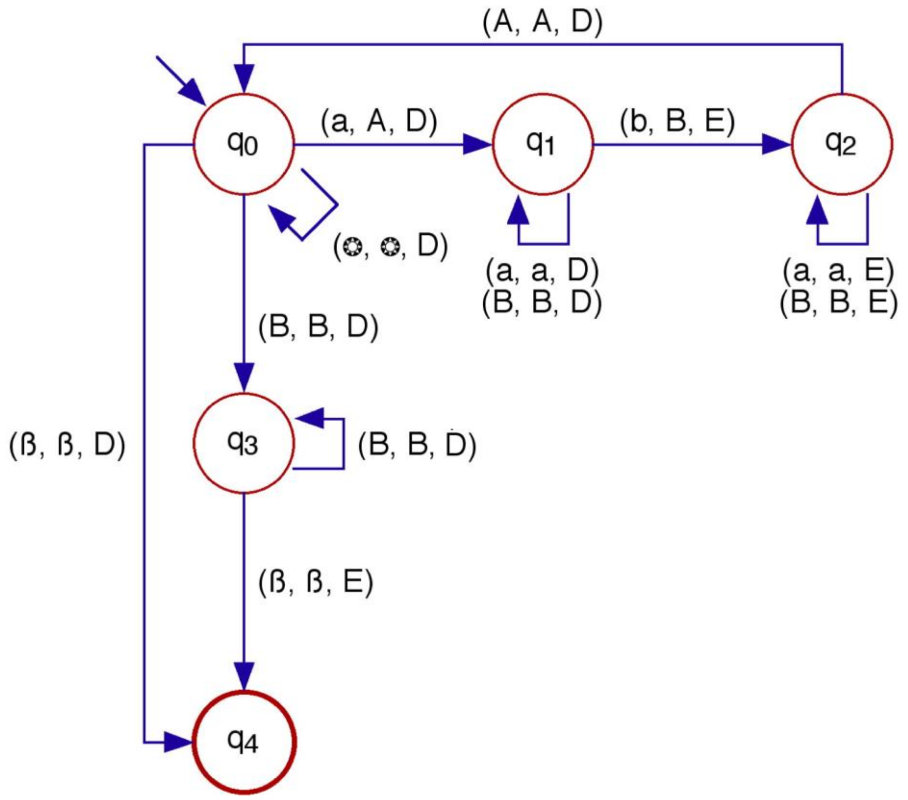
2^{f+1} : faça Y:= X vá_para fim

(`fim` é um rotulo final.)

Decodificação.

É o inverso da codificação acima.

Exemplo 6.1 Simulação do Programa Duplo-Bal



Π	©	a	b	A	B	ß
q0	(q0, ©, D)	(q1, A, D)			(q3, B, D)	(q4, ß, D)
q1		(q1, a, D)	(q2, B, E)		(q1, B, D)	
q2		(q2, a, E)		(q0, A, D)	(q2, B, E)	
q3					(q3, B, D)	(q4, ß, E)
q4						

figura 6.1 grafo e tabela de transições da Máquina de Turing Duplo-Bal

Para a palavra de entrada aabb, o valor inicial armazenado da fita da máquina de Turing é: ©aabbβββ...

correspondendo a seguinte representação no registrador X de Norma:

5	1	1	2	2	0	0

Também:

o conteúdo do registrador Q = 2 (estado inicial q_0, codificado como 2^{0+1});

o conteúdo do registrador C = 1 (denota a cabeça da fita posicionada no início).

Para cada estado qu ∈ { q0, q1, q2, q3, q4 }, se terá uma instrução rotulada, onde o rótulo é calculado por 2^{u+1}. Os rótulos calculados são: 2, 4, 8, 16 e 32. A instruções rotuladas correspondentes aos estados, são:

```
2:    faça A := 3^Q □ 5^X(C) vá para End_A       estado q0
4:    faça A := 3^Q □ 5^X(C) vá para End_A       estado q1
8:    faça A := 3^Q □ 5^X(C) vá para End_A       estado q2
16:   faça A := 3^Q □ 5^X(C)  vá para End_A      estado q3
32:   faça Y := X  vá_para fim                   estado q4
```

- A expressão $3^Q \square 5^{X(C)}$ que determina o valor do registrador A, serve para distinguir as diferentes transições definidas em função do estado e do símbolo lido, garantindo rótulos diferentes.
- O rótulo **fim** é um rótulo final.
- O Registrador X é o registrador de entrada o qual contem o conteúdo da fita de Turing. Ao final da computação transfere o conteúdo do registrador X para o registrador de saída de Y.
- Os símbolos da fita são codificados por números naturais, ou seja, os símbolos ©, a, b, A, B e ß serão codificados pelos números: 5, 1, 2, 3, 4 e 0, respectivamente.
- São usados dois registradores:
  - Q (indica estado corrente, começando por 2)
  - C (posição da cabeça da fita).
- A expressão $3^Q \square 5^{X(C)}$ indica as transições da função programa da Máquina de Turing. Neste caso, existem 12 transições as quais correspondem 12 conjuntos de três instruções rotuladas, que vão indicar: novo estado, novo símbolo e movimento na fita.

a) Para o estado inicial q0:

Π (q0, ©) = (q0, ©, D) e portanto:

$Q = 2^{u+1} = 2^{0+1} = 2$ $\quad A = 3^Q \square 5^{X(c)} = 3^2 \square 5^5 = 28125$

```
28125:    faça X(C) = 5 vá_para 28126
28126:    faça C := C+1 vá_para 28127
28127:    faça Q := 2 vá_para End_Q          (2^(u+1) = 2^(0+1) = 2)
```

Π(q0, a) = (q1, A, D) e portanto:

$Q = 2$ $\quad A = 3^Q \square 5^{X(C)} = 3^2 \square 5^1 = 45$

```
45:   faça X(C)= 3 vá_para 46
46:   faça C = C+1 vá_para 47
47:   faça Q = 4 vá_para End_Q               (2^(u+1) = 2^(1+1) = 4)
```

$\Pi(q_0, B) = (q_3, B, D)$ e portanto:

$Q = 2$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^2 \square 5^4 = 5625$

```
5625:faça X(C) = 4 vá_para 5626
5626:faça C = C+1 vá_para 5627
5627:faça Q = 16 vá_para End_Q                (2u+1 = 23+1 = 16)
```

$\Pi(q_0, \beta) = (q_4, \beta, E)$ e portanto:

$Q = 2$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^2 \square 5^0 = 9$

```
9:   faça X(C) = 0 vá_para 10
10:  faça C = C-1 vá_para 11
11:  faça Q = 32 vá_para End_Q                (2^(u+1) = 2^(4+1) = 32)
```

b) Para o estado q_1:

$\Pi(q_1, a) = (q_1, a, D)$ e portanto:

$Q = 2^{u+1} = 2^{1+1} = 4$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^4 \square 5^1 = 405$

```
405: faça X(C) = 1 vá_para 406
406: faça C = C+1 vá_para 407
407: faça Q = 4 vá_para End_Q                (2^(u+1) = 2^(1+1) = 4)
```

$\Pi(q_1, b) = (q_2, B, E)$ e portanto:

$Q = 4$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^4 \square 5^2 = 2025$

```
2025:faça X(C) = 4 vá_para 2026
2026:faça C = C-1 vá_para 2027
2027:faça Q = 8 vá_para End_Q                (2^(u+1) = 2^(2+1) = 8)
```

$\Pi(q_1, B) = (q_1, B, D)$ e portanto:

$Q = 4$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^4 \square 5^4 = 50625$

```
50625:     faça X(C) = 4 vá_para 50626
50626:     faça C = C+1 vá_para 50627
50627:     faça Q = 4 vá_para End_Q        (2^(u+1) = 2^(1+1) = 4)
```

c) Para o estado q_2:

$\Pi(q_2, a) = (q_2, a, E)$ e portanto:

$Q = 2^{u+1} = 2^{2+1} = 8$ $\qquad$ $A = 3^Q \square 5^{X(C)} = 3^8 \square 5^1 = 32805$

```
32805:     faça X(C) = 1 vá_para 32806
32806:     faça C = C-1 vá_para 32807
32807:     faça Q = 8 vá_para End_Q        (2^(u+1) = 2^(2+1) = 8)
```

$\Pi(q_2, A) = (q_3, A, D)$ e portanto:

$Q = 8$ $\quad$ $A = 3^{Q} \square 5^{X(C)} = 3^{8} \square 5^{3} = 820125$

820125: faça $X(C) = 3$ vá_para 820126
820126: faça $C = C+1$ vá_para 820127
820127: faça $Q = 16$ vá_para End_Q $(2^{u+1} = 2^{3+1} = 16)$

$\Pi(q_2, B) = (q_2, B, E)$ e portanto:

$Q = 8$ $\quad$ $A = 3^{Q} \square 5^{X(C)} = 3^{8} \square 5^{4} = 4100625$

4100625: faça $X(C) = 4$ vá_para 4100626
4100626: faça $C = C-1$ vá_para 4100627
4100627: faça $Q = 8$ vá_para End_Q $(2^{u+1} = 2^{2+1} = 8)$

d) Para o estado q_3:

$\Pi(q_3, B) = (q_3, B, D)$ e portanto:

$Q = 2^{u+1} = 2^{3+1} = 16$ $\quad$ $A = 3^{Q} \square 5^{X(C)} = 3^{16} \square 5^{3} = 26904200625$

26904200625: faça $X(C) = 4$ vá_para 26904200626
26904200626: faça $C = C+1$ vá_para 26904200627
26904200627: faça $Q = 16$ vá_para End_Q $(2^{u+1} = 2^{3+1} = 16)$

$\Pi(q_3, \beta) = (q_4, \beta, E)$ e portanto:

$Q = 16$ $\quad$ $A = 3^{Q} \square 5^{X(C)} = 3^{16} \square 5^{0} = 43046721$

43046721: faça $X(C) = 0$ vá_para 43046722
43046722: faça $C = C-1$ vá_para 43046723
43046723: faça $Q = 32$ vá_para End_Q
$(2^{u+1} = 2^{4+1} = 32)$

Programa Monolítico Duplo Balanceamento

```
2:     faça A := 3^Q □ 5^X(C) vá_para End_A
4:     faça A := 3^Q □ 5^X(C) vá_para End_A
8:     faça A := 3^Q □ 5^X(C) vá_para End_A
9:     faça X(C) = 0 vá_para 10
10:    faça C = C-1 vá_para 11
11:    faça Q = 32 vá_para End_Q
16:    faça A := 3^Q □ 5^X(C)  vá_para End_A
32:    faça Y := X  vá_para fim
45:    faça X(C)= 3 vá_para 46
46:    faça C = C+1 vá_para 47
47:    faça Q = 4 vá_para End_Q
405:   faça X(C) = 1 vá_para 406
406:   faça C = C+1 vá_para 407
407:   faça Q = 4 vá_para End_Q
2025:  faça X(C) = 4 vá_para 2026
2026:  faça C = C-1 vá_para 2027
2027:  faça Q = 8 vá_para End_Q
5625:  faça X(C) = 4 vá_para 5626
5626:  faça C = C+1 vá_para 5627
5627:  faça Q = 16 vá_para End_Q
28125:faça X(C) = 5 vá_para 28126
28126:faça C := C+1 vá_para 28127
28127:faça Q := 2 vá_para End_Q
32805:faça X(C) = 1 vá_para 32806
32806:faça C = C-1 vá_para 32807
32807:faça Q = 8 vá_para End_Q
50625:faça X(C) = 4 vá_para 50626
50626:faça C = C+1 vá_para 50627
50627:faça Q = 4 vá_para End_Q
820125:faça X(C) = 3 vá_para 820126
820126:faça C = C+1 vá_para 820127
820127:faça Q = 16 vá_para End_Q
4100625:faça X(C) = 4 vá_para 4100626
4100626:faça C = C-1 vá_para 4100627
4100627:faça Q = 8 vá_para End_Q
43046721:faça X(C) = 0 vá_para 43046722
43046722: faça C = C-1 vá_para 43046723
43046723: faça Q = 32 vá_para End_Q
26904200625:faça X(C) = 4 vá_para 26904200626
26904200626:faça C = C+1 vá_para 26904200627
26904200627:faça Q = 16 vá_para End_Q
```

figura 6.2 Programa Monolítico Duplo-Bal

Teorema 6.2
Máquina Norma ≤ Máquina de Turing

- **O formalismo Máquina de Norma pode ser simulado pelo formalismo Máquina Turing.**

PROVA

- Seja o programa monolítico P de Norma com somente dois registradores X e Y.

- A simulação do programa P de Norma por uma Máquina de Turing M = (∑, Q, Π, q_0, F, V, ß, ©) onde o alfabeto ∑ é o conjunto unário { 1 } pode ser definida como segue:

Registrador X

- conteúdo inicial do registrador X é codificado em unário na células pares da fita de M.
- se o natural em X é x, então x células pares da fita possuem o símbolo 1.

Registrador Y

- o registrador Y é armazenado na fita em unário, mas nas células ímpares (excetuando-se a primeira, que contém o marcador de início de fita ©);

Rótulos

- A cada rótulo r de instrução de P corresponde um estado q_r de M. Aos rótulos: ***inicial*** *e* ***final*** (pode ser mais de um) correspondem os estados inicial e final, respectivamente;

Programa

➢ Uma instrução rotulada de P da seguinte forma

adição

```
r:   faça adK vá_para s
```

É simulada por um trecho da função programa de M, resumido:

1) no estado q_r move a cabeça, pesquisando as células:
   - pares (caso K = X)
   - ímpares (caso K = Y)
2) até encontrar o primeiro branco, o qual é substituído pelo símbolo 1;
3) reposiciona a cabeça no início da fita e assume o estado q_s;

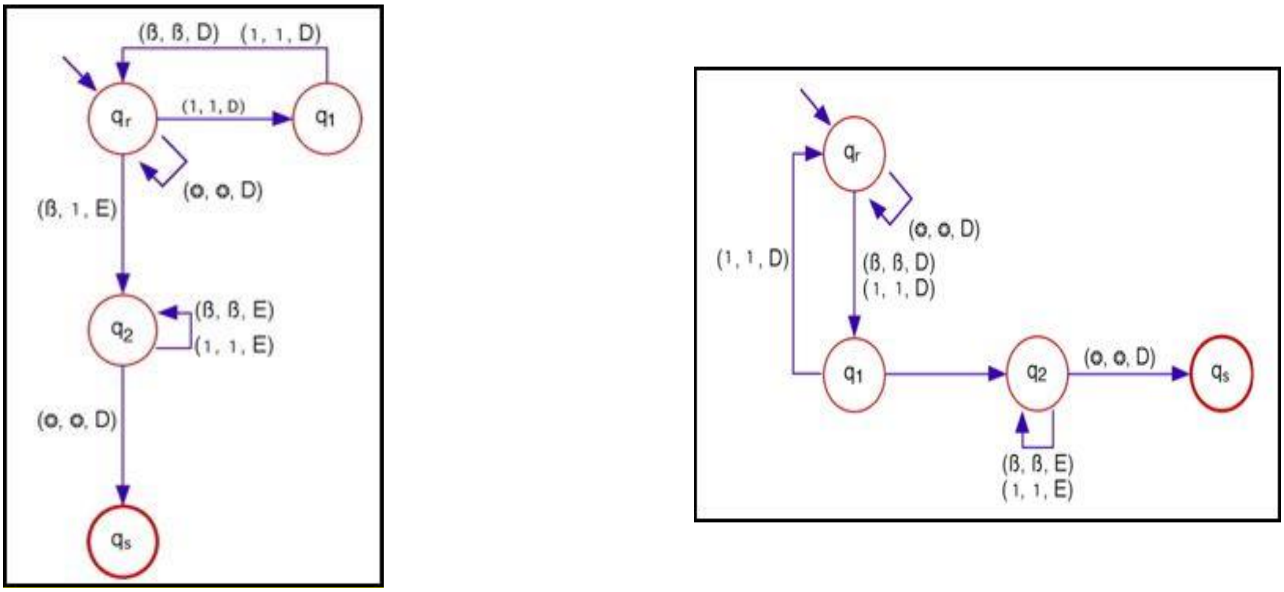


figura 6.3 Grafo da Máquina de Turing: X := X+1 e Y := Y+1

subtração

r:faça subK vá_para s

É simulada por um trecho da função programa de M, resumido

1) no estado q_r move a cabeça, pesquisando as células:
   - pares (caso K = X)
   - ímpares (caso K = Y)
2) até encontrar o último símbolo 1, o qual é substituído por um branco. Caso a primeira célula pesquisada já contenha o símbolo branco, nada é substituído;
3) reposiciona a cabeça no início da fita e assume o estado qs.

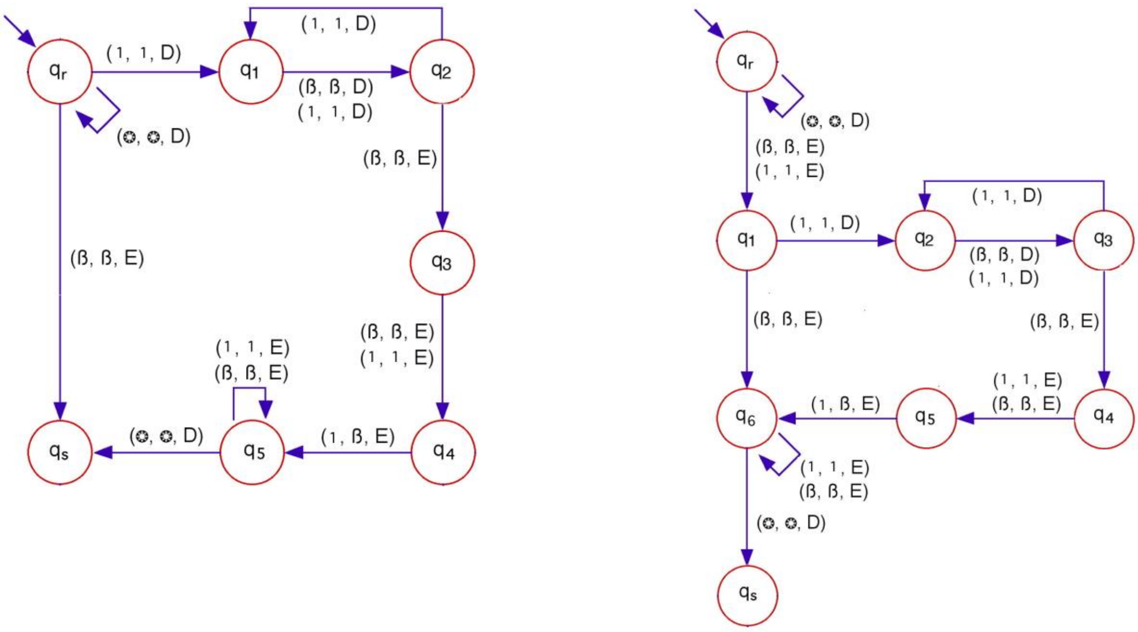


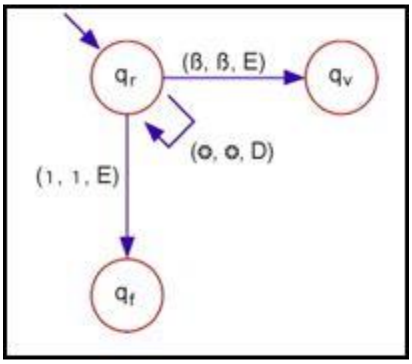
figura 6.4 Grafo da Máquina de Turing: X := X-1 e Y := Y-1

teste

```
r:     se zeroK vá_para v senão vá_para f
```

É simulada por um trecho da função programa de M resumido:

1) no estado q_r move a cabeça, pesquisando a primeira célula
   - par (caso K = X)
   - ímpar (caso K = Y);
2) Caso a célula pesquisada contenha:
   - o ***símbolo branco***, reposiciona a cabeça no início da fita e assume o estado q_v (verdadeiro);
   - caso contrário, reposiciona a cabeça no início da fita e assume o estado q_f (falso).



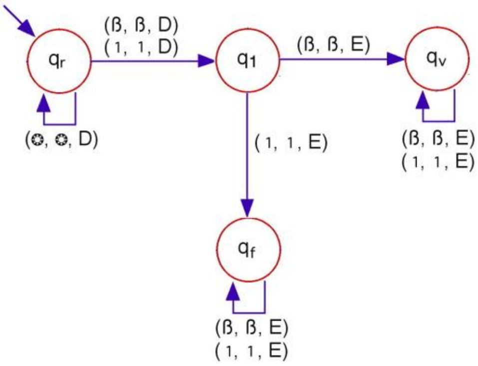


figura 6.5 Grafo da Máquina de Turing: X := 0 ? e Y := 0 ?

Codificação.

- conteúdo inicial do registrador X é codificado em unário nas células pares da fita de M;

Decodificação.

É o inverso da codificação acima.

6.2 Modificações sobre a Máquina de Turing

Máquinas Universais são equivalentes às diversas versões modificadas do modelo básico, com características que supostamente aumentariam o poder computacional.

6.2.1 Máquina de Turing não determinística

- ***não-determinismo** é uma importante generalização dos modelos de máquinas.*
- Na Máquina de Turing, para o mesmo estado corrente e símbolo lido, diversas alternativas são possíveis.
- Cada alternativa é percorrida de forma totalmente independente.
- *As alterações no conteúdo da fita realizadas em um caminho, não modificam o conteúdo da mesma nos demais caminhos alternativos.*

Genericamente, *não-determinismo* é interpretado como:

- a máquina, ao processar uma entrada, tem como resultado um conjunto de novos estados.
- ela assume um conjunto de estados alternativos, como se houvesse uma multiplicação da unidade de controle, uma para cada alternativa, processando independentemente, sem compartilhar recursos com as demais.
- processamento de um caminho não influi no estado geral, nem no símbolo lido dos demais caminhos alternativos.

Para uma máquina M não-determinística, uma palavra w pertence a:

- ACEITA(M) se existe pelo **menos um caminho** alternativo que aceita a palavra.
- REJEITA(M) se **todos** os caminhos alternativos rejeitam a entrada.
- LOOP(M) se **nenhum** caminho aceita a palavra e **pelo menos um** fica em *loop*.

6.2.2 Máquina de Turing com fita infinita à esquerda e à direita

➢ ***A modificação da definição básica da Máquina de Turing, permitindo que a fita seja infinita dos dois lados, não aumenta o poder computacional.***

Simulação:

➢ As células pares representam a parte direita da fita,

➢ As células ímpares representam a parte esquerda da fita.

➢ O símbolo © é usado para controlar a fronteira entre as partes esquerda e direita da fita.

...	a_{-3}	a_{-2}	a_{-1}	a_1	a_2	a_3	...

©	a_1	a_{-1}	a_2	a_{-2}	a_3	a_{-3}	...

figura 6.9 Simulação de uma fita infinita dos dois lados

6.2.3 Máquina de Turing com múltiplas fitas

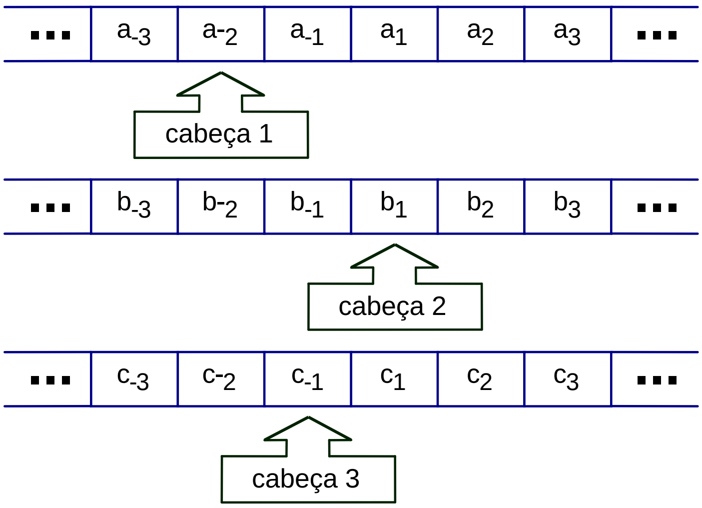
A Máquina de Turing com múltiplas fitas possui k fitas infinitas à esquerda e à direita e k correspondentes cabeças de fita.

O processamento realizado

- depende do estado corrente da máquina e do símbolo lido em cada uma das fitas;
- grava um novo símbolo em cada uma das fitas, move cada uma das cabeças independentemente para a esquerda ou direita, e a máquina assume um (único) novo estado.

Simulação:

- *Inicialmente, a palavra de entrada é armazenada na primeira fita, ficando as demais com valor branco.*
- *As três fitas são simuladas em uma única fita, modificando os alfabetos de entrada e auxiliar.*
- *Cada símbolo contido em uma célula é uma 6-upla, sendo 3 componentes para representar as células de cada uma das 3 fitas, e as demais 3 componentes para marcar a posição corrente das cabeças de cada fita (representadas na* figura 6.10 *pelo símbolo* 🡅*).*



©	a_1	a_{-1}	a_2	a_{-2}	a_3	a_{-3}	...
				🡅			...
©	b_1	b_{-1}	b_2	b_{-2}	b_3	b_{-3}	...
	🡅						...
©	c_1	c_{-1}	c_2	c_{-2}	c_3	c_{-3}	...
		🡅					...

figura 6.10 Simulação de 3 fitas infinitas dos dois lados

6.2.4 Outras modificações sobre a Máquina de Turing

Outras modificações sobre o modelo básico da Máquina de Turing também não aumentam o poder computacional.

a) *Máquina de Turing Multidimensional*. A fita tradicional é substituída por uma estrutura do tipo arranjo k-dimensional, infinita em todas as 2k direções;

b) *Máquina de Turing com Múltiplas Cabeças*. A Máquina de Turing com esta modificação possui k cabeças de leitura e gravação sobre uma única fita. Cada cabeça possui movimento independente. Assim, o processamento depende do estado corrente e do símbolo lido em cada uma das cabeças.

c) *Combinações*. A combinação de algumas ou de todas as modificações apresentadas não aumenta o poder computacional da Máquina de Turing. Por exemplo, uma *Máquina de Turing não-determinística com múltiplas fitas e múltiplas cabeças* pode ser simulada por uma *Máquina de Turing tradicional.*

6.3 Hipótese de Church

- Turing propôs um modelo abstrato de computação com o objetivo de explorar os limites da capacidade de expressar soluções de problemas.
- Trata-se, portanto, de uma proposta de definição formal da noção intuitiva de algoritmo.
- Diversos outros trabalhos, como *Máquina de Post* (1936) e *Funções Recursivas de Kleene*(1936), bem como a *Máquina de Registradores Norma* e o Autômato com Pilhas, resultaram em conceitos equivalentes ao de Turing.

O fato de todos esses trabalhos independentes gerarem o mesmo resultado em termos de capacidade de expressar computabilidade é um forte reforço para a *tese de Church* ou *tese de Turing-Church*:

"A capacidade de computação representada pela Máquina de Turing é o limite máximo que pode ser atingido por qualquer dispositivo de computação"

- A Tese de Church afirma que qualquer outra forma de expressar algoritmos terá, no máximo, a mesma capacidade computacional da Máquina de Turing

Como a noção de algoritmo ou função computável é intuitiva, a Tese de Church não é demonstrável.

- Supondo verdadeira a Hipótese de Church, pode-se afirmar que para:
  a) *Função Computável*: É possível construir uma Máquina de Turing (ou formalismo equivalente) que compute a função;
  b) *Função Não-Computável*: Não existe Máquina de Turing (ou formalismo equivalente) que compute a função.

6.4 Conclusões

A Tese de Church supõe que o limite máximo do que é computável pode ser expresso como um algoritmo em uma Máquina de Turing.

Como noção de algoritmo é intuitiva, a tese não é demonstrável, sendo aceita (suposta) como verdadeira.

Corroboraram para isto, evidências externas e internas, como a verificação de que as máquinas de Turing e Norma são equivalentes e a constatação de que as modificações sobre a Máquina de Turing apresentadas nesse capítulo não ampliam a Classe das Funções Computadas.

6.5 Exercícios

Exercício 6.1

Complete a prova do ***Teorema 6.1 Máquina de Turing ≤ Máquina Norma*** de tal forma que Máquina Norma simule as condições ACEITA e REJEITA de parada da Máquina de Turing.

Exercício 6.2

No contexto do ***Teorema 6.2 Máquina Norma ≤ Máquina de Turig***, suponha que a Máquina Norma tenha três registradores (X, Y e Z). Então:

a) Como seriam representados esses registradores na fita da Máquina de Turing

b) Dê a função programa, na forma de grafo, que implemente as seguintes operações: adX, subX, zeroX;

c) Dê a função programa, na forma de grafo que implemente as seguintes operações: adY, subY, zeroY;

d) Dê a função programa, na forma de grafo que implemente as seguintes operações: adZ, subZ, zeroZ

Exercício 6.3

No contexto do ***Teorema 6.2 Máquina Norma ≤ Máquina de Turig***, suponha que a Máquina Norma tenha cinco registradores. Discuta como seriam representados os registradores na fita e como seriam implementadas as operações da Máquina Norma na Máquina de Turing.

Exercício 6.4

Considere a seguinte Máquina de Turing (introduzida anteriormente), ilustrada na ***Figura 6.11***:

Conc = ({ a, b, # }, { q0, q1, q2, q3, q4 }, Π, q0, { q4 }, ∅, β, ©)

No contexto do ***Teorema 6.1 Máquina de Turing ≤ Máquina Norma***, qual o correspondente programa (na forma de instruções rotuladas) na Máquina Norma?

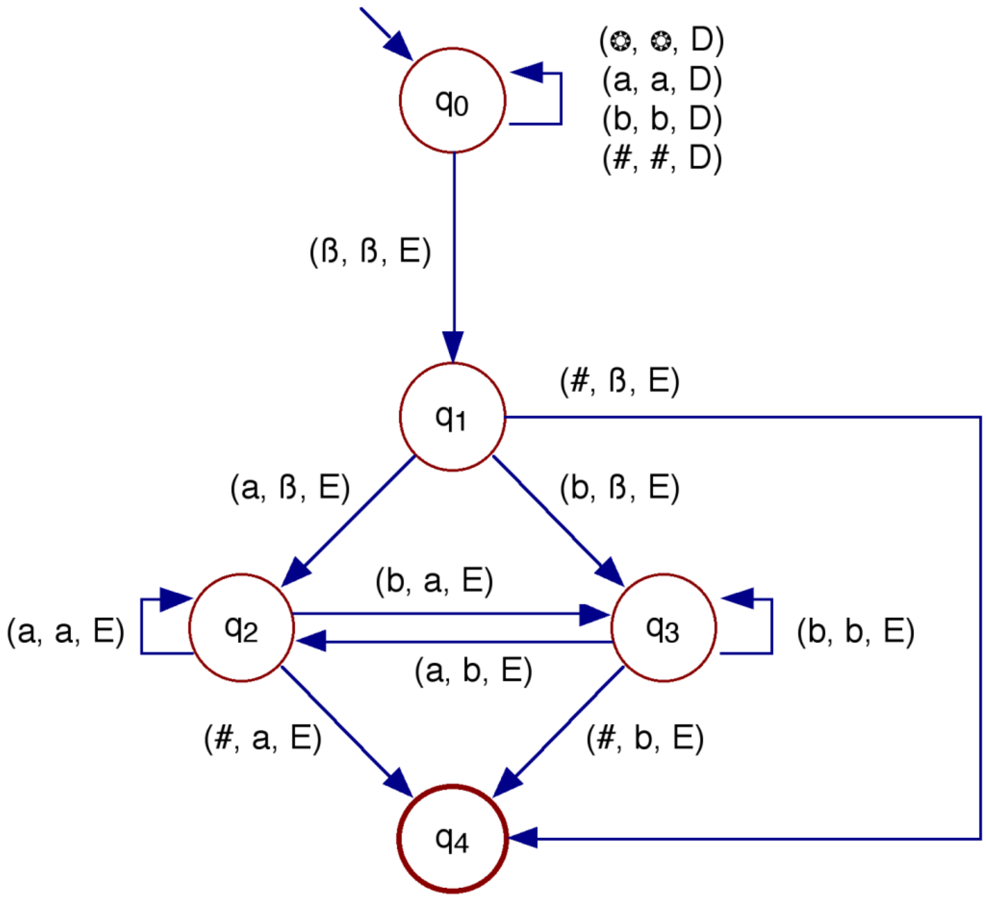


Figura 6.11 Grafo da Máquina de Turing - Concatenação

Exercício 6.5

No contexto do ***Teorema 6.1 Máquina de Turing ≤ Máquina Norma***, marque a alternativa *errada*:

a) A fita é simulada por um arranjo unidimensional em X, sendo que cada célula da fita corresponde a uma posição do arranjo;

b) O símbolo de cada célula é codificado como um número natural: cada símbolo a_i do alfabeto $\Sigma = \{ a_1, a_2, \ldots, a_n \}$, é codificado como o natural i; e os símbolos especiais de início de fita e branco são codificados como n+1 e zero, respectivamente;

c) Para cada estado de um programa em Turing há uma correspondente instrução rotulada, sendo que ao estado inicial q_0 é associado ao rótulo 2^{0+1} de P e o estado final q_f é associado o rótulo final 2^{f+1} de P; São 3 instruções

d) A cabeça da fita é simulada pelo registrador C, que contém a posição corrente do arranjo X e é inicializado com o valor 1;

e) O estado corrente da máquina de Turing é simulado pelo registrador Q, o qual assume os valores em { 2^1, 2^2,..., 2^{n+1} }, correspondendo aos estados { q_0, q_1,..., q_n }, respectivamente.

Exercício 6.6

No contexto do ***Teorema 6.2 Máquina Norma ≤ Máquina de Turig***, marque a alternativa correta:

a) Cada rótulo é representado por um estado na Máquina de Turing;
b) Se o registrador X contém o valor x, então seu conteúdo é armazenado nas x primeiras células pares da fita e, se o registrador Y contém o valor y, então seu conteúdo é armazenado nas y primeiras células ímpares da fita;
c) Os alfabetos de entrada e auxiliar juntos precisam ter, no mínimo, dois símbolos para poder simular a máquina Norma;
d) A função de transição precisa ser total;
e) A instrução ad_k pode ser simulada procurando-se a primeira célula em branco e substituindo-a pelo símbolo 1.

Exercício 6.7

No contexto do ***Teorema 6.1 Máquina de Turing ≤ Máquina Norma*** e do ***Teorema 6.2 Máquina Norma ≤ Máquina de Turig***, *marque a alternativa correta:*

a) Na Máquina Norma não existe meio de prever todas as condições de ACEITA e de REJEITA da Máquina de Turing;
b) A Máquina de Turing simula os estados da Máquina Norma através de uma fila recursiva;
c) Pode-se representar a fita da Máquina de Turing na Máquina Norma por um arranjo unidimensional no registrador X, onde cada célula da fita corresponde a uma posição do arranjo em X;
d) Não são necessárias funções de codificação e de decodificação para os conjuntos de entrada e de saída nos formalismos em questão.
e) É impossível a simulação da Máquina Norma por uma Máquina de Turing com fita infinita para os dois lados, uma vez que parte da fita ficará vazia e se terá perda de informação.

Exercício 6.8

A Hipótese de Church afirma que (marque a alternativa correta):

a) Qualquer programa pode ser representado na forma de fluxograma;
b) Qualquer máquina abstrata é uma máquina universal;
c) A codificação de conjuntos estruturados é o modo mais eficiente de representar uma máquina universal;

d) Qualquer função computável pode ser processada por uma máquina de Turing;
e) Todo programa iterativo pode ser representado através de um programa monolítico.

Exercício 6.9

A Hipótese de Church não pode ser demonstrada por que (marque a alternativa correta):

a) O conceito de função computável não é matematicamente preciso;
b) O não determinismo não aumenta o poder computacional da Maquina de Turing;
c) Atualmente, não há poder computacional que execute o algoritmo proposto pela Hipótese de Church;
d) Para qualquer sistema computador, o tempo necessário para simular a Hipótese de Church é maior do que o tempo de existência do sistema solar;
e) Uma função computável não tem equivalência na Máquina de Turing.

Exercício 6.10

Sobre a Hipótese de Church, analise as seguintes afirmações:

I. A capacidade de computação representada pela Máquina de Turing é o limite máximo do que pode ser atingido por qualquer dispositivo de computação;
II. Qualquer outra forma de expressar algoritmos terá, no máximo, a mesma capacidade computacional da Máquina de Turing;
III. Como a noção de algoritmo é intuitiva, a Hipótese de Church não é demonstrável.

Marque a alternativa correta:

a) Apenas I está correta;
b) Apenas II está correta;
c) Apenas I e III estão corretas;
d) Apenas II e III estão corretas;
e) I, II e III estão corretas.