

Máquinas de Turing

Matheus Henrique Schaly - 18200436

1) a) $L = \{a^i b^j c^k \mid i, j, k \in \mathbb{N} \text{ e } i \times j = k^2\}$

Algoritmo:

1. Verifica, não-deterministicamente, se o input é $\geq abc$ e aceita se for. (
2. Chega no primeiro c e marca-o com um 3, percorre todos os c 's e vai para o passo 3. Caso não houver mais c 's para marcar vai para o passo 4.
3. Percorre todos os d 's, ao chegar no final da fita, marca um d e retorna ao passo 2.
4. Transforma todos os 3's em c 's novamente.
5. Para cada c marca-o como 3 e percorre todos os d 's; para cada d marca-o como 4 e vai para o passo 6. Caso não houver mais c 's para marcar vai para o passo 7.
6. Percorre todos os C 's, ao chegar no final da fita, marca um C e retorna ao passo 5.
7. Transforma todos os 4's e 3's em c 's.
8. Para cada c encontrado, marca-o como C , e elimina um C do final da fita.
9. Transforma todos os C 's em c 's. Os passos 2 ao 9, transformaram c em c^2 .
10. Volta ao começo da fita.
11. Para cada a encontrado marca-o como 1 e percorre todos os b 's, marcando-os como 2. Para cada b marque um c como 3.
12. Caso não haja um c correspondente para cada b marcado, rejeite a entrada.
13. Caso acabe a quantidade de b 's e ainda exista pelo menos um c , rejeite a entrada.
14. Aceite a entrada.

Entradas mostradas usando o Step:

- Aceitando: $\geq aaabbbccc$
- Rejeitando: $\geq aaabbccc$

Entradas mostradas usando o Multiple Run:

- Aceita:
 - $\geq abc$
 - $\geq abbbbcc$
 - $\geq aaaabcc$
 - $\geq aabbcc$
 - $\geq aaaaaaabbccccc$
- Rejeita:
 - \geq
 - $\geq a$

- >b
- >c
- >ab
- >ac
- >abcc
- >aabcc
- >abbcc
- >aabbccc
- >aaabbcc
- >aabbbcc

2) a) $L = \{a^i b^j c^k \mid i, j, k \in \mathbb{N} \text{ e } i^2 \times j^2 = k\}$

Obs: Os estados q14, q15, q16, q17 são desnecessários. Porém, ao tirá-los, a máquina bugava, portanto eles serão deixados na máquina.

Algoritmo:

- 1) Verifique, não-deterministicamente, se o input é >abc e aceite se for. (desnecessário, a máquina aceitaria com ou sem esse não-determinismo, como comentado acima).
- 2) Na fita um, para cada a marque-o como A e percorra todos os b's marcando-os como B. Simultaneamente, para cada b marcado adicione um C na fita dois. Resete todos os B's para b e repita o processo até que todos os a's estejam marcados.
- 3) Na fita dois, para cada C marque-o como D e coloque um # ao final da mesma fita.
- 4) Na fita dois, para cada D marque-o como 1 e percorra todos os #'s marcando-os como 2, para cada # marcado adicione um C ao final da mesma fita. Resete todos os 2 para # e repita o processo até que todos os D estejam marcados.
- 5) Coloque o cabeçalho da fita 1 no primeiro c. Coloque o cabeçalho da fita 2 no primeiro C. Percorra as duas fitas simultaneamente, se houver a mesma quantidade de c's na fita 1 quanto C's na fita dois, aceite o input. Caso contrário rejeite.

Entradas mostradas usando o Step:

- Aceitando: >aabbcccccccccccccc
- Rejeitando: >aabbccc

Entradas mostradas usando o Multiple Run:

- Aceita:
 - >abc
 - >abbcccc
 - >aabcccc
 - >aaabcccccccc
 - >aabbbcccccccc

- >aabbcccccccccccccccc
- >aabbbcc
- >aaabbcc

- Rejeita:

- >
- >a
- >b
- >c
- >ab
- >ac
- >abcc
- >aabcc
- >abbcc
- >aabbcc
- >aabbccc
- >aaabbcccccc
- >aabbbcc

- 2) b) A série de Fibonacci. A máquina recebe como entrada uma sequência de símbolos que representa n . Ao término, deve constar na fita uma sequência de símbolos que indica o valor do n -ésimo termo, ou seja $F_{ibonacci}(n)$.
Exemplo: A máquina recebe como entrada a sequência "aaaaa"(5) e deve retornar "bbbbbbbb"(8).

Obs. 1: O algoritmo não funcionou no Fast Run nem no Multiple Run. Portanto os testes demonstrados no vídeo foram realizados apenas com o modo Step.

Obs. 2: O output desejado encontra-se na fita três.

Algoritmo:

- 1) Verifique se o input é >a e retorne >; verifique se o input é >aa e retorne >b; verifique se o input é >aaa e retorne >b.
- 2) Percorra o primeiro a da fita um e adicione um A na fita dois.
- 3) Percorra o segundo a da fita um e adicione um B na fita dois e um b na fita três.
- 4) Ao chegar no quarto a da fita um retorne o cabeçote da fita dois ao primeiro A e limpe a fita três e retorne o cabeçote da fita três para o começo.
- 5) Na fita dois, para cada A marque-o como a e adicione um C ao final da mesma fita e um b na fita três.
- 6) Na fita dois, para cada B marque-o como b e adicione um C ao final da mesma fita e um b na fita três.
- 7) Na fita dois, troque os a's por >, troque os b's por A's e troque os C's por B's.
- 8) Caso não haja mais a's na fita um, vá para o estado de aceitação e pare a máquina. Caso contrário, avance para o próximo a, resete a fita três e retorne ao passo 5.

Entradas mostradas usando o Step:

- Fita 1: >a Retorna na Fita 3: >
- Fita 1: >aa Retorna na Fita 3: >b
- Fita 1: >aaa Retorna na fita 3: >b
- Fita 1: >aaaa Retorna na fita 3: >bb
- Fita 1: >aaaaa Retorna na fita 3: >bbb
- Fita 1: >aaaaaa Retorna na fita 3: >bbbbb
- Fita 1: >aaaaaaa Retorna na fita 3: >bbbbbbbb

3) Implemente uma Máquina de Turing (de sua escolha) para computar o algoritmo de Euclides para o Máximo Divisor Comum.

Obs. 1: O algoritmo não funcionou no Multiple Run. Portanto os testes demonstrados no vídeo foram realizados apenas com o modo Step e Fast Run.

Obs 2: O input da fita um deve ser o maior número. Ou seja, o input da fita um deve ser maior ou igual o da fita dois. O resultado é dado por *R*'s na fita três.

Algoritmo:

- 1) Verifique se o input é vazio (>) na fita um ou na fita dois. Caso for, retorne a quantidade de elementos da fita que não é vazia, isto é, para cada elemento da fita não vazia, copie-o para a fita três.
- 2) Para cada *a* na fita dois, adicione um *c* na fita três.
- 3) Para cada *a* na fita dois (marque-o como *Y*) e marque um *b* na fita um (como um *X*). Caso haja mais *a*'s do que *b*'s, vá para o passo 6. Caso todos os *a*'s e *b*'s foram marcados, vá para o passo 9.
- 4) Resete os *Y*'s da fita dois para *a*'s.
- 5) Caso haja *b*'s na fita um, retorne para o passo 3.
- 6) Para cada *a* da fita dois que não possua um correspondente *b* na fita um, marque o *a* como *W*.
- 7) Volte a fita dois até encontrar o primeiro *Y* e substitua-o por >, ao mesmo tempo, troque os *W*'s encontrados na fita dois por *a*'s.
- 8) Volte a quantidade de *c*'s da fita três na fita um e marque o *b* da fita um como >. Retorne para o passo 2.
- 9) Limpe a fita três (substituindo *c*'s por branco).
- 10) Para cada *Y* na fita dois, adicione um *R* na fita três.
- 11) Ao chegar em branco, branco e branco, vá para um estado de aceitação e pare a máquina.

Entradas mostradas usando o Step:

- Fita 1: >
 - Fita 2: >aaaa
 - Retorno fita 3: >RRRR
- Fita 1: >bbbbbb

- Fita 2 >
 - Retorno fita 3: >RRRRR
- Fita 1: >b
 - Fita 2: >a
 - Retorno fita 3: >R
- Fita 1: >bb
 - Fita 2: >aa
 - Retorno fita 3: >RR
- Fita 1: >bbb
 - Fita 2: >aa
 - Retorno fita 3: >R
- Fita 1: >bbbb
 - Fita 2: >aa
 - Retorno fita 3: >RR
- Fita 1: >bbbbbbbbb (9)
 - Fita 2: >aaaaaa (6)
 - Retorno fita 3: >RRR (3)

Entradas mostradas usando o Fast Run:

- Fita 1: >bbbbbbbbbbbbbbbb (16)
 - Fita 2: >aaaaaaaa (8)
 - Retorno fita 3: >RRRRRRRR (8)
- Fita 1: >bbbbbbbbbbbbbbbb (15)
 - Fita 2: >aaaaaaaaaa (10)
 - Retorno fita 3: >RRRRR (5)
- Fita 1: >bbbbbbbbbbbbbbbbbbbb (20)
 - Fita 2: >aaaaaaaa (8)
 - Retorno fita 3: >RRRR (4)