O PROBLEMA DO CASTOR

BUSY BEAVER'S PROBLEM

João Gross, Rodrigo Leite - Grupo 16

Histórico e Caracterização do Problema

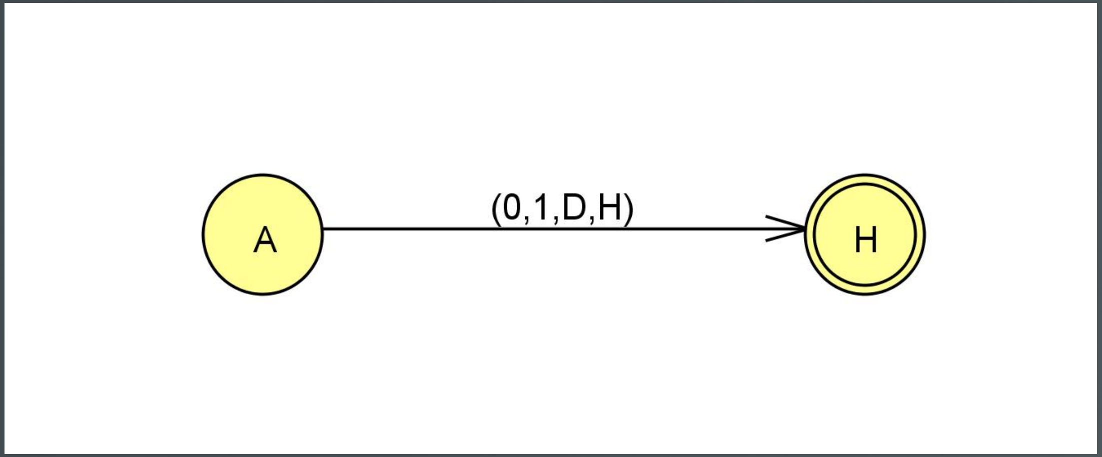
- Em 1962, Tibor Radó introduziu em seu artigo o jogo do castor, que consistia em encontrar uma máquina de Turing de n estados que deixa o maior número de “1s” na fita antes de parar.
- Definição da máquina de Turing:
  - Alfabeto binário {0,1};
  - Fita bidirecional infinita;
  - Estado adicional de parada;
- Função de transição:
  - Entradas:
    - Estado atual ;
    - Símbolo da fita na posição atual;
  - Saídas: símbolo a ser escrito na posição atual da fita;
    - Direção do movimento;
    - Novo estado para o qual transitar;
- Formato da 5-upla:
  - Estado atual, o símbolo atual, símbolo de escrever, o sentido da mudança, o próximo estado.

Exemplos

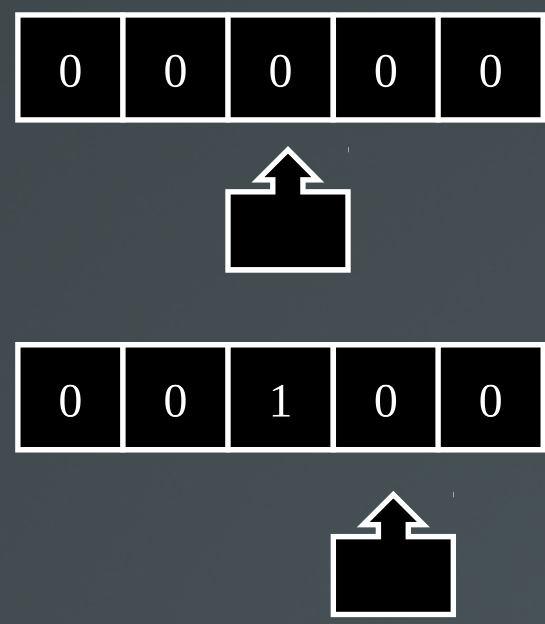
- Máquina com 1 estado
- Tabela

Símbolo/Estado	A
0	1,D,H
1	Não Usado

- Diagrama



- Computação:

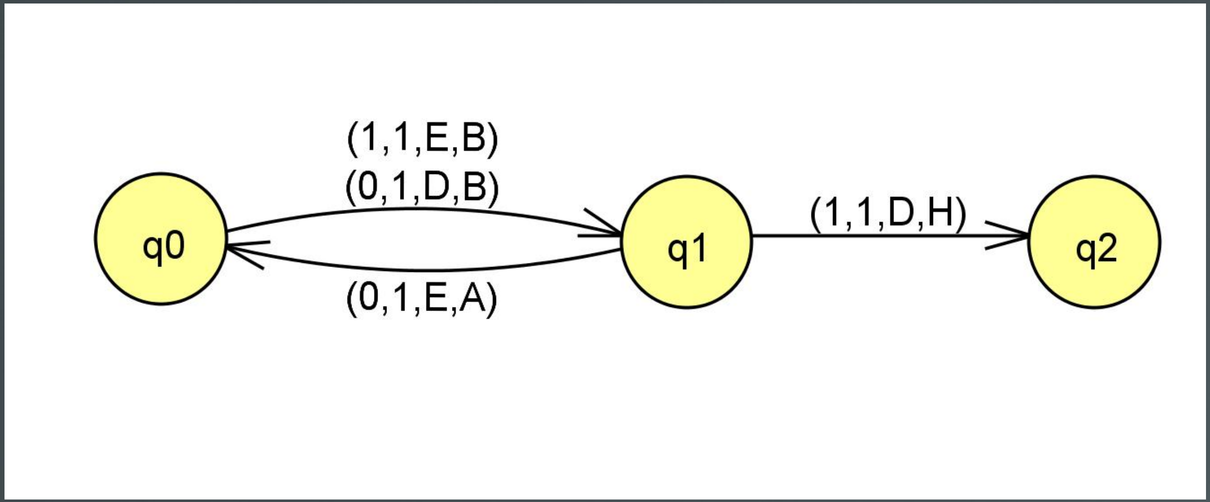


- Resultado:
  - 0 0 1 **0** 0 (1 passo, um "1" no total);

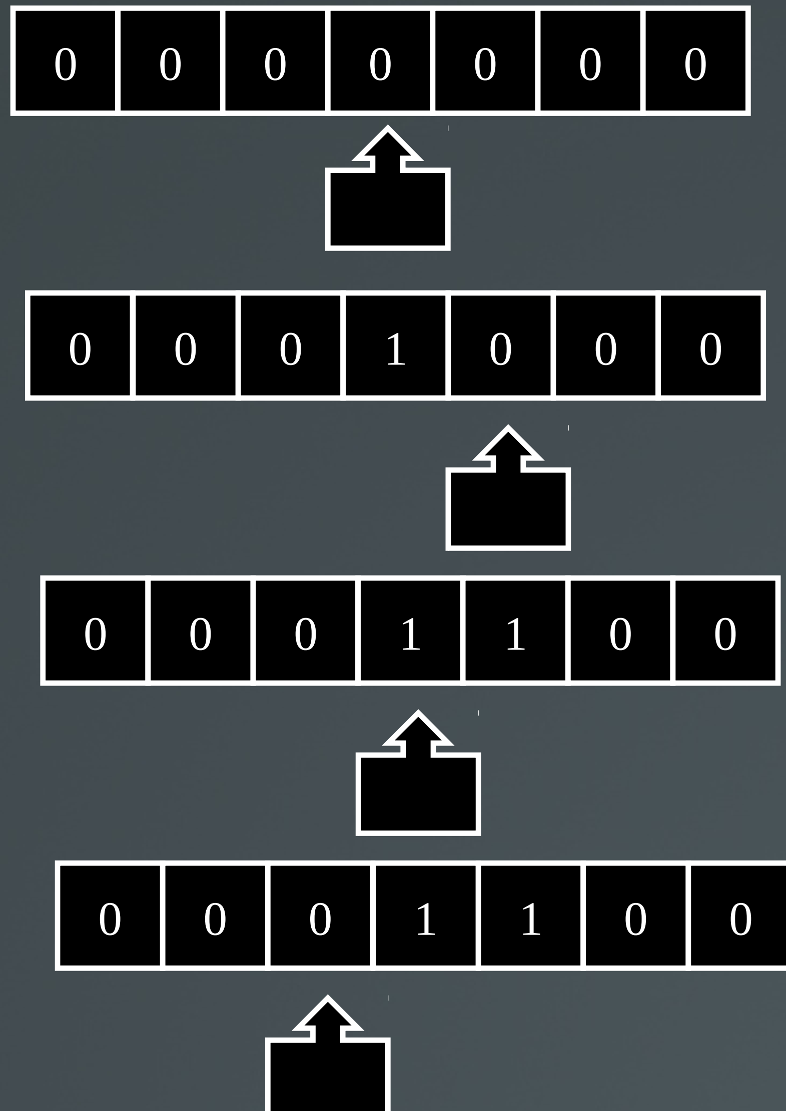
- Máquina com 2 estados
- Tabela

	A	B
0	1,D,B	1,E,A
1	1,E,B	1,D,H

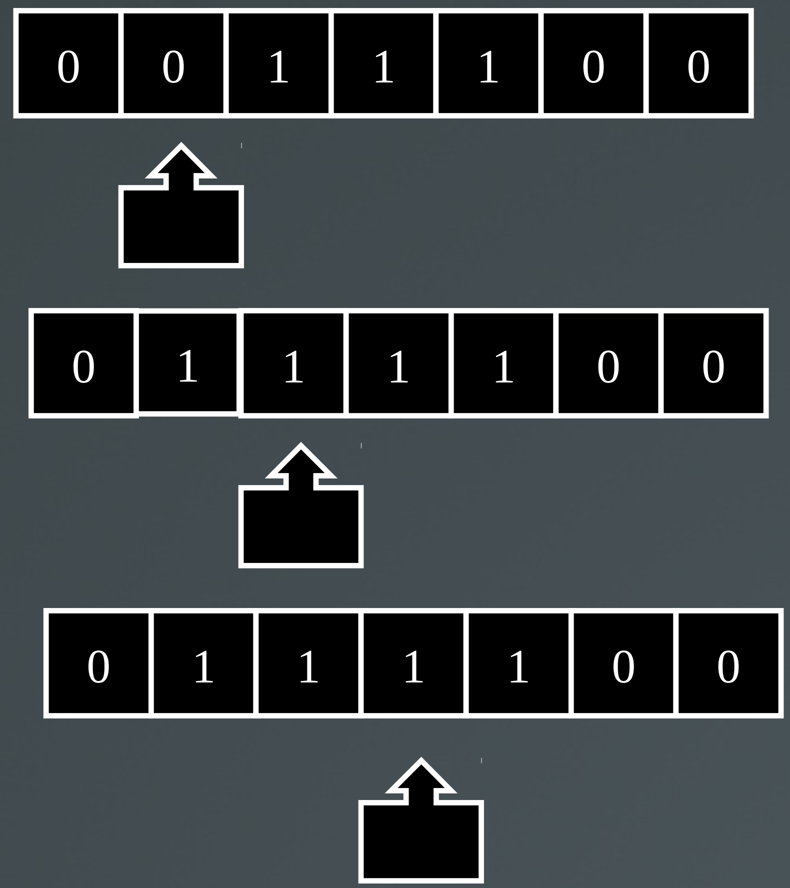
- Diagrama



- Computação:

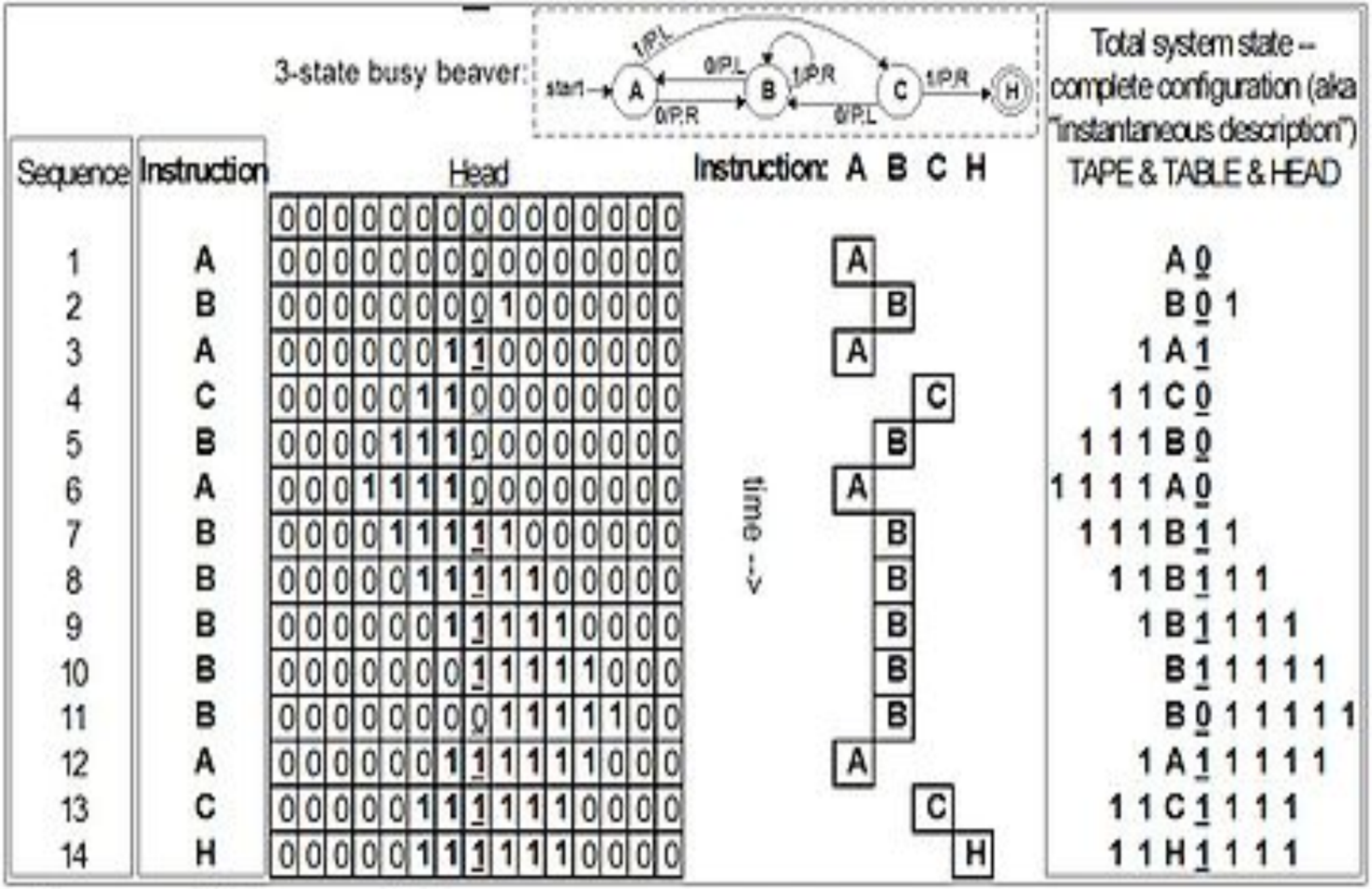


- Computação (continuação...):



- Resultado:
  - 0 0 1 1 **1** 1 0 0 (6 passos, quatro "1"s no total);

- Máquina com 3 estados



Progress of the computation (state-trajectory) of a 3-state busy beaver

Resultados Conhecidos

- Máquina com 2 estados
  - 6 passos e 4 “1s”
- Máquina com 3 estados
  - 21 passos e 6 “1s”
- Máquina com 4 estados
  - 107 passos e 13 “1s”
- Máquina com 5 estados
  - >= 47.176.870 passos e >= 4.098 “1s”

Prova da Não-Computabilidade

- Suponhamos que BB(n), função do castor aplicado a n estados, seja computável, onde n pertence a N*.
- Suponha também que existe uma máquina de Turing M e queremos saber se M roda indefinidamente ou pára (Problema da parada).
- Rodamos M para BB(n) + 1 passo; se ela parar com essa quantidade de passos, teremos uma resposta, senão ela não vai nunca mais parar.

Prova da Não-Computabilidade

- Porém se a máquina rodou a função BB(n) + 1 passo e ainda não parou, geramos uma contradição, pois executando apenas a função BB(n) o programa deveria parar (de acordo com a sua definição).
- Assim BB(n) é não-solucionável.

Outras Formas de Demonstração

- Outra forma de demonstrar que este problema é não-solucionável é aplicando redução. Queremos saber se BB(n) para para qualquer n pertencente a N*, ou seja, reduzimos o problema do castor ao problema da parada. Sabemos por demonstração [TAD] que o problema da parada é não-solucionável, logo o problema do castor ocupado também é não-solucionável.
- Já o complemento da função do castor ocupado também é não-computável, visto que o problema do castor ocupado é não-computável e o complemento de qualquer problema não-computável é não-computável. [TAD]

Referências

- [TAD] Diverio, Tiarajú Asmuz. Teoria da computação: máquinas universais e computabilidade / Tiarajú Asmuz Diverio, Paulo Blauth Menezes. – Porto Alegre: Instituto de Informática da UFRGS : Bookman, c2011. 3ª edição.
- [AKD] Dewdney, A. K. The (new) Turing omnibus: 66 excursions in computer science. Publisher: Hnery Holt, 1993. 480 p.