[Aula 08] Propriedade das LR: Bombeamento para linguagens regulares

Prof. João F. Mari
joaof.mari@ufv.br

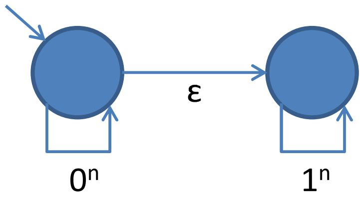
ROTEIRO

- Linguagens não regulares
- **[EX]** B = $\{0^n1^n \mid n >= 0\}$
- Lema do bombeamento para LRs
- **[TEOREMA]** Lema do Bombeamento
- IDEIA da Prova
- **[EX]** B = $\{0^n1^n \mid n >= 0\}$
- Bombeamento para baixo
- **[EX]** E = $\{0^i1^j \mid i>j\}$

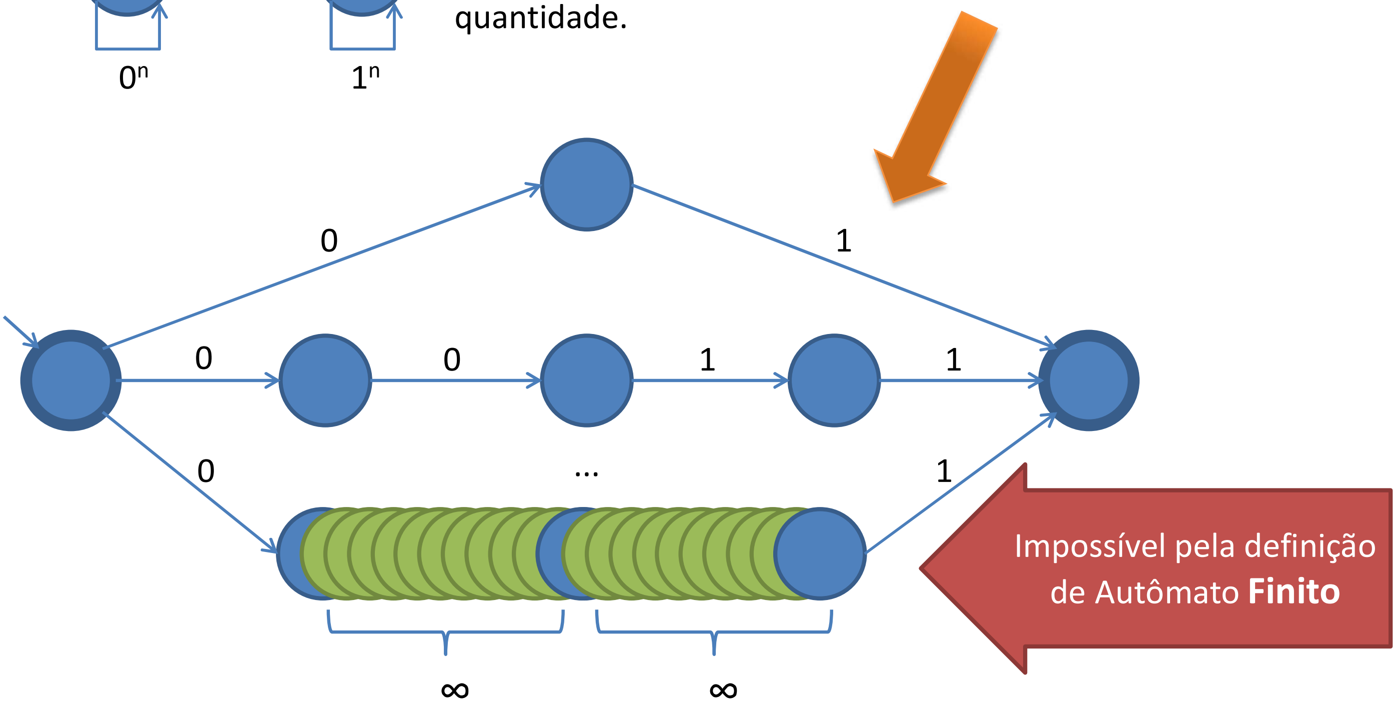
Linguagens Não Regulares

- Como provar que certas linguagens NÃO podem ser reconhecidas por NENHUM autômato finito.
- <u>**[EX]**</u> A linguagem B = $\{0^n1^n \mid n >= 0\}$
  - Ao tentar encontrar um AFD que reconheça B,
  - descobrimos que a máquina precisa lembrar quantos 0s foram vistos até então à medida que lê a entrada.
  - Como o número de 0s não é limitado,
    - A máquina precisa registrar um número infinito de possibilidades.
    - Impossível para uma máquina com quantidade finita de estados.

[EX] B = $\{0^n1^n \mid n >= 0\}$



Como que eu sei quantos 0s eu li para repetir com 1s?
Lembrando que posso ler ∞ 0s e 1s, desde que na mesma quantidade.



Linguagens Não Regulares

- Só porque uma linguagem **parece** precisar de memória ilimitada não significa que ela **realmente necessita**.

- Considere as linguagens C e D sobre {0, 1}:
  - C = { w | w tem um número igual de 0s e 1s}
    - **É não regular**. Precisa de memória auxiliar para contar o número de 0s e replicar o número de 1s.
  - D = { w | w tem um número igual de 01 e 10 como subcadeias **e somente elas**}
    - **É REGULAR**. Apesar de parecer necessitar de memória auxiliar, NÃO precisa!!!!
    - Palavras que pertencem a linguagem são do tipo:
      - 101, 10101, 1010101, ...,
      - 010, 01010,0101010, ...

[AULA 08] Propriedades das LR – Bombeamento para LR SIN 131 – Introdução à Teoria da Computação (PER-3)

Lema do bombeamento para LRs

- Lema do bombeamento:
  - Teorema sobre as Linguagens Regulares.
  - Todas as Linguagens Regulares têm uma propriedades especial:
    - Se mostrarmos que uma linguagem não possui essa propriedade, temos a garantia de que ela não é regular.

- O lema do bombeamento diz:
  - Todas as palavras (cadeias) de uma linguagem podem ser "bombeadas" se:
    - Elas são no mínimo tão longas do que um determinado valor especial.
      - (Comprimento de bombeamento).
  - **Cada** palavra (cadeia) contém uma parte que pode ser repetida um número qualquer de vezes:
    - Com a cadeia resultante permanecendo na linguagem.

[TEOREMA] Lema do Bombeamento

- Lema do bombeamento:
  - Se ***A*** é uma linguagem regular;
  - Então, existe um número ***p*** (comprimento de bombeamento) tal que;
    - Se ***s*** é qualquer palavra de ***A*** de comprimento no mínimo ***p***, então ***s*** pode ser dividida em três partes:

s = xyz

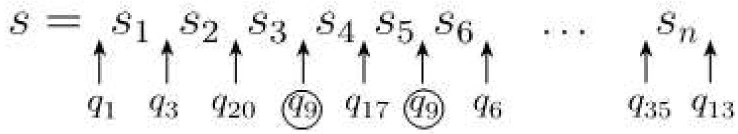
  - E satisfaça as seguintes condições:
    - 1) para cada i >= 0, $xy^iz \in A$,
    - 2) |y| > 0, e
    - 3) | xy | <= p.

IDEIA da Prova

- Seja um autômato ***M*** que reconhece ***A***
  - Ao comprimento de bombeamento ***p*** atribuímos o número de estados de ***M***.
  - Mostramos que:
    - Qualquer cadeia ***s*** em ***A*** de comprimento pelo menos ***p*** pode ser quebrada nas três partes ***xyz***, satisfazendo as nossas 3 condições.
  - E se nenhuma cadeia em ***A*** tem comprimento no mínimo p?
    - O teorema se torna verdadeiro por ***vacuidade***,
      - As 3 condições são verdadeiras para as cadeias >= a pois não existem tais cadeias.

IDEIA da Prova

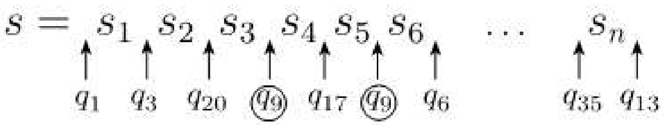
- A sequencia de estados pelos quais M passa quando computa uma entrada s.
  - Se s em A tem comprimento n.
    - E n tem comprimento pelo menos p (n >= p).
- q_0, q_3, q_{20}, q_9, ..., q_{13} tem comprimento n+1
  - Como n>=p, então n+1>p.
    - A sequencia têm que conter um estado repetido.

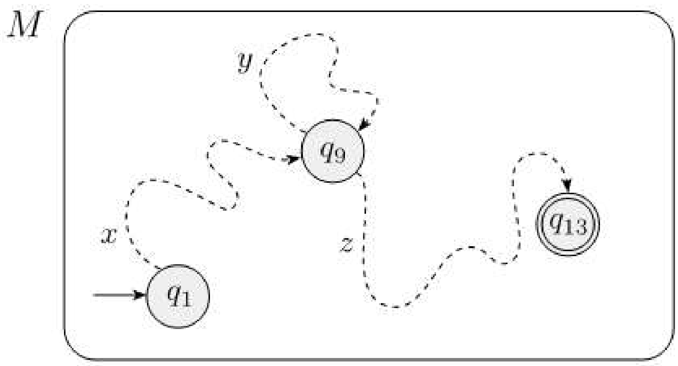


- Principio da **casa de Pombos**:
  - Se p pombos for colocados em menos de p casas, alguma casa recebe mais de um pombo.

IDEIA da Prova

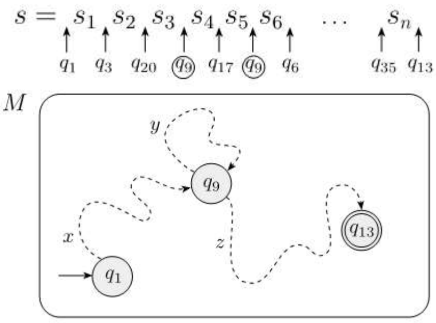
- Dividimos s em x, y e z.
  - X é a parte que aparece antes de q_9.
    - X leva M de q_1 para q_9.
  - Y aparece antes das duas ocorrências de q_9.
    - Y leva M de q_9 de volta para q_9.
  - Z é o resto de s (após a segunda ocorrência de q_9)
    - Z leva M de q_9 para q_{13} (final).





IDEIA da Prova

- Essa divisão de ***s*** satisfaz as 3 condições?
  - Computemos ***xyyz*** pelo autômato M:
    - O ***x*** leva ***M*** de ***q1*** para ***q9***.
    - O primeiro ***y*** leva de ***q9*** de volta para ***q9***, assim com o segundo ***y***.
    - O ***z*** leva de ***q9*** para ***q13***, sendo ***q13*** um estado final... ***M*** aceita ***xyyz***.



  - Condição 1:
    - Analogamente ***M*** aceitará ***xy^i z*** para qualquer ***i>0***
    - Para ***i=0***, ***xy^i z = xz***
      - Portanto ***xz*** é aceito por razões semelhantes.
  - Condição 2:
    - |y| > 0, pois é a parte de s entre duas ocorrências de q9.
  - Condição 3:
    - Pelo princípio da casa de pombos, os primeiros p+1 estados deve ter uma repetição, portanto |xy| <= p.

[EXEMPLO] B = $\{0^n1^n \mid n >= 0\}$

- Usar o lema do bombeamento para provar que B não é regular.
  - Suponha, inicialmente, que B seja Regular.
  - Lembrando: p é o comprimento do bombeamento.
  - Escolha ***s = 0^p1^p***
    - Como ***s ∈ B*** e ***|s| >= p*** (s tem comprimento maior que p),
    - O lema do bombeamento, garante que s pode ser dividida:
      - ***s = xyz***
    - e para ***i >=0*** a palavra ***xy^iz*** continua em ***B***
  - 3 formas de bombear y que provam que B é não regular.
    - 1) y contém apenas 0s;
    - 2) y contém apenas 1s;
    - 3) y contém 0s e 1s.

[EXEMPLO] B = $\{0^n1^n \mid n >= 0\}$

- 1) A palavra y contém apenas 0s:
  - s = 00001111, então:
    - x = 00; y = 00; z = 1111.
  - Bombeando 2 vezes:
    - $s_2 = xy^2z = xyyz$ = 00 0000 1111 $\notin$ B (6 zeros e 4 uns).
      - Portanto B não é regular. Não é possível construir um AFD que reconheça 0^n1^n.

- 2) A palavra y contém apenas 1s (*desconsidere a condição 3*):
  - s = 00001111, então:
    - x = 0000; y = 11; z = 11.
  - Bombeando 2 vezes:
    - $s_2 = xy^2z = xyyz$ = 0000 1111 11 $\notin$ B (4 zeros e 6 uns)
      - Portanto B não é regular. Não é possível construir um AFD que reconheça 0^n1^n.

[EXEMPLO] B = $\{0^n1^n \mid n >= 0\}$

- 3) A palavra y contém apenas 0s e 1s (*desconsidere a condição 3*):
  - s = 00001111, então:
    - x = 00; y = 0011; z = 11.
  - Bombeando 1 vez:
    - $s_2 = xy^2z = xyyz$ = 00 0011 0011 11 $\notin$ B
      - **Mesmo número de 0s e 1s**, **porém estão fora de ordem**!!!
      - Portanto B não é regular. Não é possível construir um AFD que reconheça 0^n1^n.

- **!!!** Se **considerarmos** a condição 3 do lema: $|xy| <= p$.
  - Podemos simplificar nossa prova para os casos 2 e 3:
  - Caso 2) y contém 1s
    - Se $s = 0^p1^p$ e p = 4, conseq. s = 00001111
      - Então x = 0000 e y = no mínimo 1 (condição 2) PORTANTO $|xy| = 5 > p$!!!!!
  - Caso 3) y contém 0s e 1s → A prova é semelhante.

[EXEMPLO] B = $\{0^n1^n \mid n >= 0\}$

- **Cuidado!!!** - Nesse exemplo, qualquer cadeia em B não pode ser bombeada.
  - Em outros casos, algumas escolhas para s **podem** ser bombeadas.
    - Devemos ter cuidado em escolher s que não possa ser bombeada
    - Ou esgotar TODAS as possibilidades de existência de s ∈ a linguagem L, e conseguir bombear todas, provando que L é regular.
      - Complete a prova construindo o AFD que reconhece L.

Bombeamento para baixo

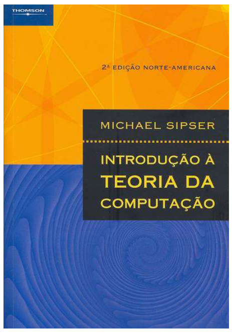
- Algumas vezes o recurso de “bombear para baixo” é útil para provar que certas linguagens não são regulares.
  - Bombear para baixo é testar $s = xy^iz$ para $i = 0$.

[EX] $E = \{0^i1^j \mid i>j\}$

- Suponha ***E*** regular.
  - ***p*** é o comprimento de bombeamento;
  - $s = 0^{p+1}1^p \rightarrow |s| > p$ e $s \in E$.
  - Dividimos ***s*** em ***xyz***:
    - Pela condição 3 y possui somente 0s: $|xy| <= p$.
    - Podemos bombear y quantas vezes quisermos:
      - xyyz, xyyyz, xyyyyz, xyy...yyz → aumentamos apenas o número de 0s, consequentemente s ainda ∈ E.
  - Porém, o lema do bombeamento diz que xy^iz deve ∈ E para $i >= 0$, ou seja, para $i = 0$ também.
    - Concatenando y com ele mesmo 0 vezes, removemos y de s.
      - Bombeamento para baixo.
    - Como $|y|$ é pelo menos 1 (condição 2)...
      - ***xz*** não tem mais 0s do que 1s, portanto $xz \notin E$.

BIBLIOGRAFIA

- SIPSER, M. **Introdução a Teoria da Computação**, 1. Ed., Thomson Pioneira, 2007.
  - <u>Seção 1.4</u>



- MENEZES, P. B. **Linguagens formais e autômatos**, 6. ed., Bookman, 2011.
  - Capítulo 4.
  - + Slides disponibilizados pelo autor do livro.

Material complementar

- Hemerson Pistori. *Lema do Bombeamento para Linguagens Regulares (Pumping Lemma)*. YouTube.
  - https://www.youtube.com/watch?v=MaRdDPivgYg

[FIM]

- FIM:
  - **[AULA 08]** Propriedades das linguagens regulares – Bombeamento para linguagens regulares
- Próxima aula:
  - **[AULA 09]** Propriedades das linguagens regulares – Minimização de ADF