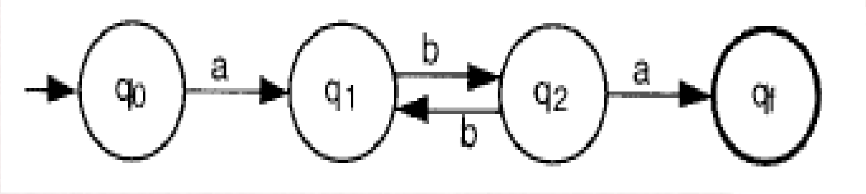
Linguagens Regulares e Não-regulares

- Na aula passada...
  - LR são representadas por formalismos pouco complexos, muito eficientes e de fácil implementação.
  - São restritas e limitadas.
  - Para demonstrar que uma linguagem é regular, represente-a usando os formalismos conhecidos: AF, ER ou GR.
  - A não-regularidade deve ser estudada caso à caso.
    - Suponha a linearidade e contradiga-a.

Lema do bombeamento

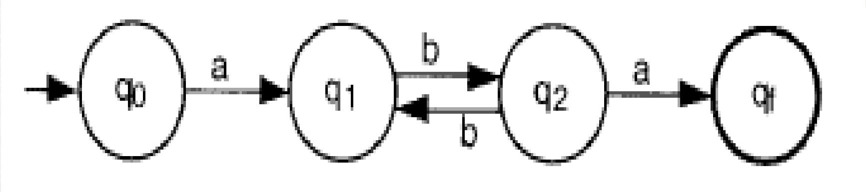
- Se L é LR, é aceita por um AFD com n estados predefinidos.
- Se o autômato reconhece a entrada w com |w| >= n, obrigatoriamente o autômato assume algum estado q mais de uma vez, então existe um ciclo na função programa que passa por q.
- Assim, w pode ser dividida em 3 sub-palavras (w = uvz), tal que |uv| <= n, |v| >=1 e v é a parte de w reconhecida pelo ciclo. uv^iz, para i>=0, é aceita pelo autômato e o ciclo é executado i vezes.

Exemplificando o lema...



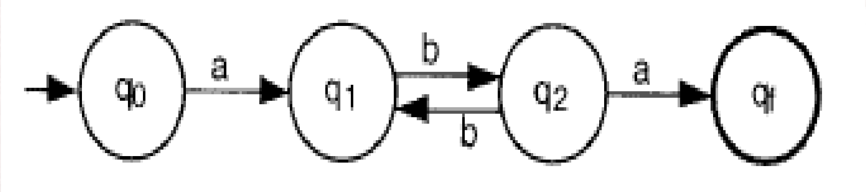
- Considere o autômato acima.
- Qual a linguagem aceita?
- Defina formalmente.
- n = ??
- Suponha uma palavra onde |w| >=n.

Exemplificando o lema...



- Considere o autômato acima.
- Qual a linguagem aceita?
- Defina formalmente.
- n = 4
- |w| >=n, que tal abbba?!

Exemplificando o lema...

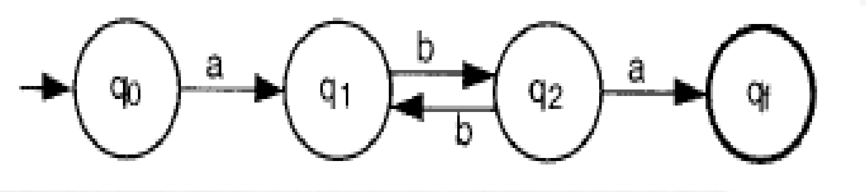


- Considere o autômato acima.
- n = 4, w = abbba, |w| = 5 (a1,a2,...am).
- Existem r e s, tal que 0 <= r < s <=n e qr = qs, $\delta(q0, a1...ar) = qr$

$$\delta(qr, a(r+1)...as) = qs$$

$$\delta(qs, a(s+1)...am) = qm$$

Exemplificando o lema...



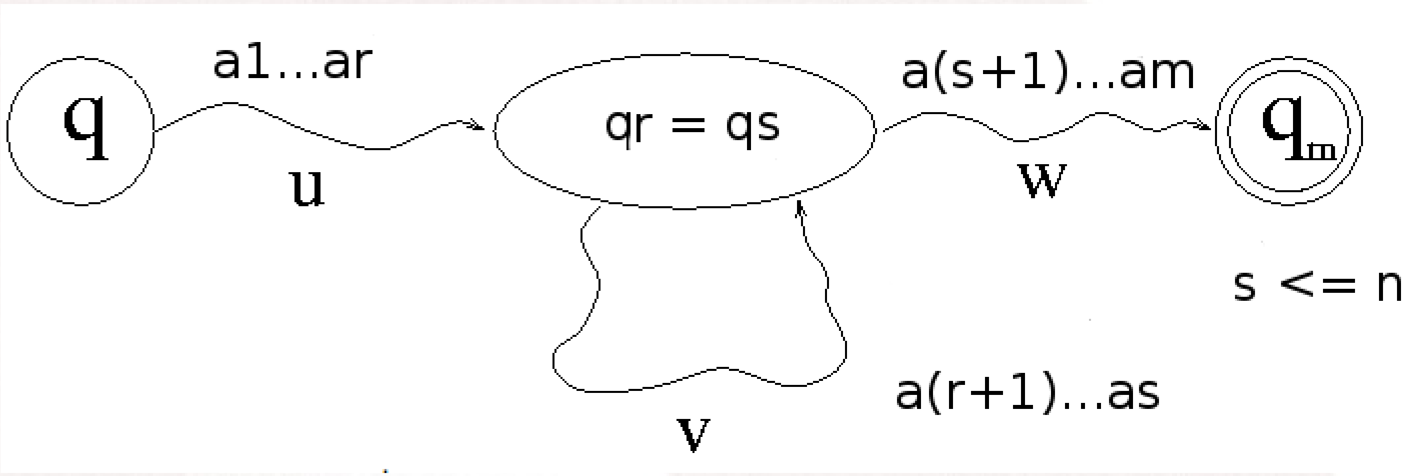
- Considere o autômato acima.
- Existem r e s, tal que 0 <= r < s <=n e qr = qs, $\delta(q0, a1...ar) = qr$

  $\delta(qr, a(r+1)...as) = qs$

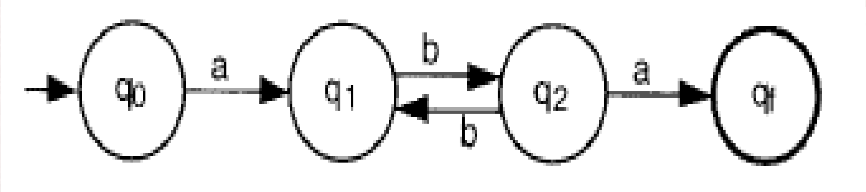
  $\delta(qs, a(s+1)...am) = qm$
- u = a1...ar, v = a(r+1)...as e z = a(s+1)...am.
- r < s <= n, |v| >= 1 e |uv| <= n

Exemplificando o lema...

- O que estou querendo dizer:

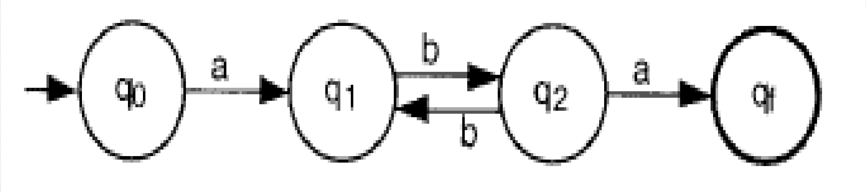


Exemplificando o lema...



- Considere o autômato acima.
- u = a1...ar, v = a(r+1)...as e z = a(s+1)...am.
- r < s <= n, |v| >= 1 e |uv| <= n
- qr = qs = qf (então v é reconhecida no ciclo).

Exemplificando o lema...



- Considere o autômato acima.
- u = a1...ar, v = a(r+1)...as e z = a(s+1)...am.
- r < s <= n, |v| >= 1 e |uv| <= n
- qr = qs = qf
- u = a, v = bb e z = ba

O lema sobre uma LNR

- A linguagem L sobre {a,b} não é Regular:

  $L = \{ a^n b^n | n>=0\}$

- PROVA:
- Lema do bombeamento.
- Se L é regular, então:
  - Existe um AFD com n estados que aceita L.

O lema sobre uma LNR

- Tome $w = a^n b^n$
- Pelo lema, w = uvz,
  - |uv|<= n, |v| >=1

    para todo i>=0, $uv^i z$ é palavra de L.
  - Ops, observe:
    - Se |uv|<=n e uv é composta por a's, uv^2z não possui o mesmo número de símbolos a e b.
    - Se |uv|<=n e uv é composta por b's, uv^2z não possui o mesmo número de símbolos a e b.
    - Se |uv|<=n e uv é composta por a's e b's pode-se ter o mesmo número de a e b, mas podem não acontecer na ordem ab (aaabbb, p.e.).

Outra forma de exemplificar...

- D = {w|w tem o mesmo número de a's e b's}.
  - Se D é regular, então:
    - Tome p como o comprimento do bombeamento, a string é dada por $s = a^n b^n$ $s > n$
    - $s = uvz$, e para $i >= 0 \rightarrow s = uv^i z$ é aceito em D.
    - Um momento! E se u e z fossem o símbolo vazio?
      - $v = a^n b^n$ (o lema foi confirmado!!! Mas olhemos mais de perto...)

Outra forma de exemplificar...

- Lembre que |uv| <= n.
- Se s = $a^n b^n$ e |uv| <= n, então v deve conter somente a's o que não permite o pump! (exemplo anterior)
- E se s = $(ab)^n$???
  - Mesmo número de a's e b's (hummm...)
  - u = ε
  - v = ab
  - z = $(ab)^{n-1}$
  - s = uv^iz (yeah! É regular!!)

Outra forma de exemplificar...

- Ao buscar uma string para testar uma LR, não desista no primeiro não!!!
- Uma alternativa para testar a não regularidade da linguagem $L(C)=(ab)^n$.
  - Parta do conhecimento de uma LNR $L(B) = a^n b^n$.
  - Se C é LR, $C \cap a^*b^*$ também é.