



Linguagens Regulares

Lema do Bombeamento
Linguagens regulares e não-regulares

Lema do Bombeamento para LR

- Como decidir que uma linguagem não é regular?
 - Toda linguagem regular satisfaz o Lema do bombeamento (LB). Lemas são artefatos práticos, bons para provas.
 - Se alguém apresenta a você uma LR falsa, use o LB para mostrar a contradição, pois ela não vai satisfazer o LB.
- Veja que toda cadeia maior que o número de estados do AF força um estado do AF a se repetir... (brincos ou voltas)

- O lema do bombeamento (Pumping Lemma) nos diz que:
 - qualquer cadeia suficientemente longa z de uma LR pode ser decomposta em 3 partes:
 - $z = uvw$, de maneira que podemos construir outras cadeias da linguagem pela repetição da parte central v .
- Todas as cadeias da forma $u v^i w$ são também da linguagem.
 - Ou seja, podemos acionar a *bomba* quantas vezes quisermos, para criar quantas sentenças novas da linguagem desejarmos: $uw, uvvw, uvvww, \dots$

Mostrando que uma Linguagem não é Regular

- Para mostrar que uma linguagem **não é** regular, mostramos que
 - não há como decompor uma cadeia (qualquer, arbitrariamente longa) da linguagem de forma que seja possível *bombear* e continuar na linguagem.
 - Usamos ele para mostrar que **CERTAS** linguagens não são regulares. E são **MUITAS** para as quais aplicamos.

E para LR?

- Entretanto, o lema não dá uma condição suficiente para a linguagem ser regular,
 - pois para algumas **não regulares** o lema é verdadeiro.
 - Assim, o lema não pode ser usado para PROVAR que uma linguagem é regular, embora ele seja aplicável/verdadeiro para todas as LR's.
 - Todavia, se o lema se aplica, nós podemos facilmente construir o AF que reconhece a linguagem, então dizemos que ela é regular.
 - É necessário MAS não suficiente. Não é uma prova do tipo se e somente se (com ida e volta)

Lema do Bombeamento

Se L é uma linguagem regular, então:

existe uma constante natural n tal que,

para toda cadeia z de L com comprimento **maior ou igual a n** (n é chamada "tamanho do bombeamento" - nro de estados), **pode ser** decomposta em três cadeias u, v, w

($z = uvw$) de forma que

- $|uv| \leq n$
- $v \neq \lambda$ (isto é, $|v| \geq 1$) e
- para qualquer $i \geq 0$, $u v^i w \in L$ (veja que $i=0$ significa remover v).

Observem a afirmação acima: **existe uma**

Lema em termos de algoritmo

Lemma.

For every **regular** language L ,

there exists a number $p \geq 1$ such that

for every word $w \in L$ with at least p letters

there exist x, y, z with $w = xyz$ and $|y| \neq 0$ and $|xy| \leq p$ such that

for every number $i \geq 0$

$xy^i z \in L$.

- **Demonstração (simplificada):** Baseia-se no fato de que para as **cadeias longas z** é necessário usar pelo menos um loop de estados num AFD que aceite a linguagem.
- Assim, os símbolos de **u** são usados para chegarmos a um estado **q** do loop;
- os símbolos de **v** são usados para dar a volta no loop, de volta ao estado **q** ;
- os símbolos de **w** são usados para ir de **q** até um estado final.
- Portanto, podemos dar quantas voltas no loop quisermos, e repetir **v** um número qualquer **i** de vezes: **$u v^i w$** .

As **cadeias curtas (comprimento $< n$)** não são consideradas porque podem ser aceitas sem passar por nenhum loop.

Exemplo de demonstração do lema

JFLAP : <untitled1>

File Input Test Convert Help

Editor Simulate: abbbba

```
graph LR; q0((q0)) -- a --> q1((q1)); q1 -- b --> q2((q2)); q2 -- b --> q1; q2 -- a --> q3(((q3)))
```

q0

abbbba

L1 = ??

Step Reset Freeze Thaw Trace Remove

Iniciar aut_er_1.ppt Bombeamento.ppt SCE_521_185 JFLAP : <untitled1> PT 20:02

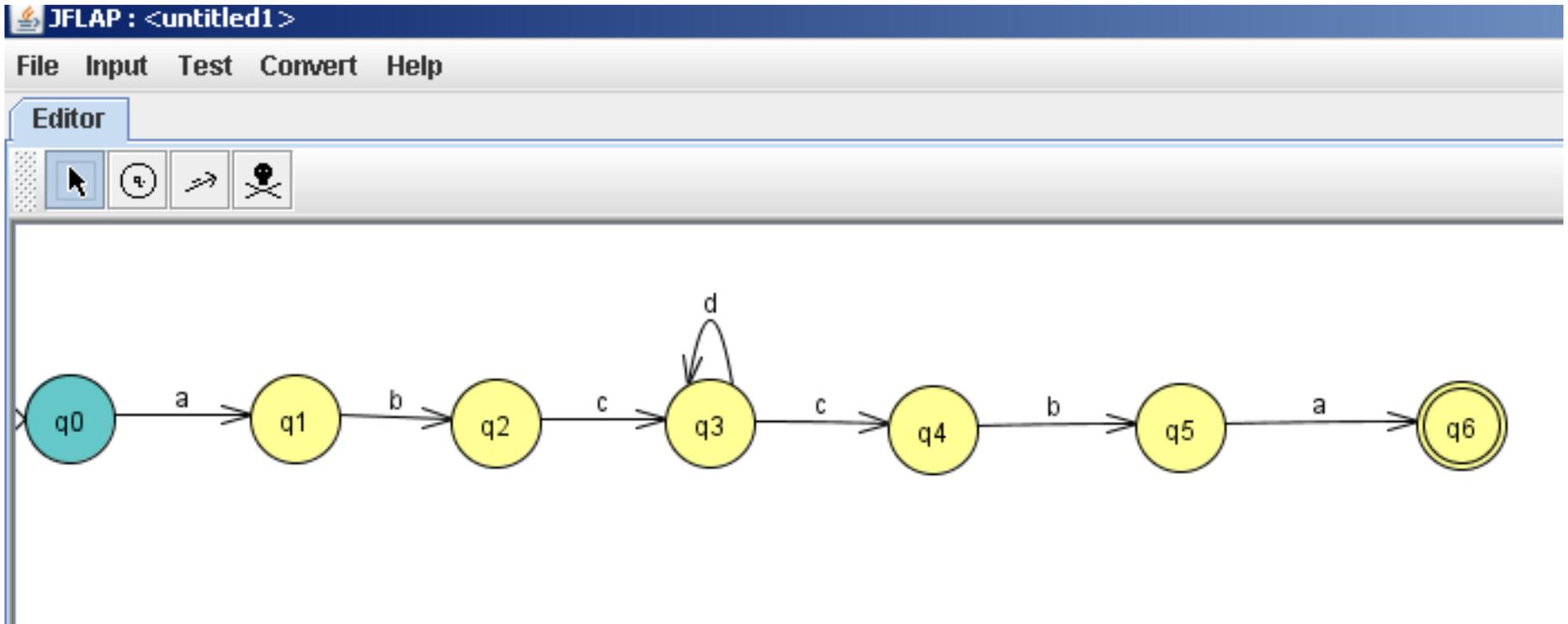
- Pelo Lema do Bombeamento, temos que:
- $n = 4$ (depende da linguagem, pois depende do **número de estados**)
- Para $z = abbba$ $|z| = 5 \geq 4$
 - $q = q_1$ (estado do loop)
 - $u = a$
 - $v = bb$ $|uv| \leq 4; |v| \geq 1$
 - $w = ba$

Para todo $i \geq 0$ $a (bb)^i ba \in L_1$ pois $L_1 = \{ab^n a \mid n \geq 1 \text{ e impar}\} \cup \{ab^{2n+1} a \mid n \geq 0\}$

Exercício

- Mostrem o lema se aplica para $L2 = \{abc d^n cba \mid n \geq 0\}$, que é regular

AF para L2



- Pelo Lema do Bombeamento, temos que:
- $n = 7$ (depende da linguagem, pois depende do número de estados)
- Para $z = abcdcba$ $|z| = 7 \geq 7$
 - $q = q_3$ (estado do loop)
 - $u = abc$
 - $v = d$ $|uv| \leq 7$; $|v| \geq 1$
 - $w = cba$
- Para todo $i \geq 0$ $abc (d)^i cba \in L_2$ pois $L_2 = \{abc d^n cba \mid n \geq 0\}$

Ou $z = abcddcba$; $|z| = 8 \geq 7$; $abc (dd)^i cba \in L_2$,
 $i \geq 0$

Como escolhemos o n do lema?

- Pode ser usado o número de estados do AF que estamos trabalhando, para a demonstração de que o lema se aplica para uma LR.
- Observem que podemos ter vários AF para reconhecer uma dada LR...

*Exemplo de linguagem não regular $\{uu^Rv \mid u,v \in \{0,1\}^+\}$
para a qual o lema é aplicável*

- Esta linguagem pode ser bombeada com $n = 4$.
- Seja $w = xyz$
- Suponha que $w = uu^Rv$ tem comprimento de pelo menos 4, por exemplo, 0011
- Se u tem tamanho 1, então $|v| \geq 2$ e y pode ser o primeiro caractere em v .
- $x = 00$; $y = 1$; $z = 1$
- $|xy| \leq 4$ OK, pois $3 \leq 4$
- $|y| \leq 1$ OK, pois $|y| = 1$
- $001^i 1 \in L$ para qualquer $i \geq 0$ OK, pois $001 \in L$; $0011 \in L$; $00111 \in L$ e assim por diante.

Provando que L não é regular

$L = \{a^i b^i \mid i \geq 0\}$ (Já vimos que L é um LLC.)

Vamos mostrar, por **contradição**, que L não é regular.

Suponha que L é regular. Se L é regular, o Lema do Bombeamento se aplica, e existe n tal que a decomposição descrita pode ser realizada **para qualquer cadeia de comprimento igual ou maior que n .**

- Serão apresentadas 2 formas de prova, ou seja, 2 explicações

Prova 1

- $z = a^n b^n$
- $z = uvw$ onde:
- $|uv| \leq n$ $|v| \geq 1$ e
- para todo $i \geq 0$ $uv^i w$ é palavra de L

O que é absurdo pois como $|uv| \leq n$ uv é composta exclusivamente por a 's. Nesse caso, uv^2w não pertence a L pois não possui o mesmo número de a e b .

Prova 2:

- Considere a cadeia $z = a^n b^n$; z pode ser dividida em 3 pedaços = uvw onde para $i \geq 0$ a cadeia $uv^i w$ pertence a L .
- Vamos considerar **3 casos** para mostrar que este resultado é impossível:
 - 1) a cadeia v consiste somente de a 's. Neste caso, a cadeia $uvvw$ tem mais a 's do que b 's **violando o LB**.
 - 2) a cadeia v consiste somente de b 's. Este caso também dá uma **contradição**.
 - 3) a cadeia v consiste de a 's e b 's. Neste caso a cadeia $uvvw$ tem o mesmo número de a 's e b 's mas não estão na ordem desejada pois $v = abab$. E assim ela não é membro de L o que é uma **contradição**.
- A contradição é inevitável se nós aceitamos a suposição de que B é regular. Assim, B não é regular.

Exercícios

Mostre que as linguagens abaixo **não** são regulares:

1) $L = \{0^n 1^{2n} \mid n \geq 1\}$

2) $L = \{xx^r \mid x \in \{0,1\}^*\}$

3) $L = \{0^n 1 0^n \mid n \geq 1\}$

4) $L = \{0^n 1^m 2^n \mid n \text{ e } m \text{ são inteiros quaisquer}\}$

5) $L = \{0^n 1^m \mid n \leq m\}$

Mostre que a Linguagem abaixo satisfaz o Lema

- $L = \{w \mid w \text{ tem um número igual de ocorrências de subcadeias } 01 \text{ e } 10 \text{ e somente elas}\}.$
- Por exemplo, $101 \in L$ pois contém 1 ocorrência de 01 e uma de 10 MAS 1010 não pertence pois contém dois 10 e um 01.
- 010 também pertence e 0101 não.
- Cadeias que pertencem a linguagem são:
 - 101, 10101, 1010101,
 - 010, 01010, 0101010,
- Embora possa parecer que a máquina precise contar as cadeias acima como na linguagem $a^n b^n$, **podemos estar errados...**

Uma das soluções é a união de 2 AF's

The image shows a screenshot of the JFLAP software interface. The main window displays a finite automaton with states $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8$. State q_8 is the start state, indicated by a triangle. States q_3 and q_7 are final states, indicated by double circles. The transitions are as follows:

- $q_0 \xrightarrow{0} q_1$
- $q_1 \xrightarrow{1} q_2$
- $q_2 \xrightarrow{0} q_3$
- $q_1 \xrightarrow{0} q_2$
- $q_2 \xrightarrow{1} q_1$
- $q_8 \xrightarrow{0} q_1$
- $q_8 \xrightarrow{1} q_5$
- $q_4 \xrightarrow{1} q_5$
- $q_5 \xrightarrow{0} q_6$
- $q_6 \xrightarrow{1} q_7$
- $q_5 \xrightarrow{1} q_6$
- $q_6 \xrightarrow{0} q_5$

The simulation window at the bottom left shows the current state q_5 and the input string 10101 . The bottom of the screen shows the Windows taskbar with several open applications: Iniciar, JSTOR: Applied Statistics..., Wiley InterScience: Jour..., Sce185_2005_A_B, Microsoft PowerPoint - [...], and JFLAP: <untitled1>. The system clock shows 12:26.

Pelo Lema do Bombeamento, temos que:

n = 4

- Para $z = 01010$ $|z| = 5 \geq 4$
 - $q = q_1$
 - $u = 0$
 - $v = 10$ $|uv| \leq 4$; $|v| \geq 1$
 - $w = 10$
 - Para todo $i \geq 0$ $(10)^i 10 \in L$
- Para $z = 10101$ $|z| = 5 \geq 4$
 - $q = q_5$
 - $u = 1$
 - $v = 01$ $|uv| \leq 4$; $|v| \geq 1$
 - $w = 01$
 - Para todo $i \geq 1$ $(01)^i 01 \in L$

Jflap

- Verifiquem como o JFLAP trata o lema do bombeamento:
 - Um jogo de adversários (humano, máquina)
 - Utilizem o jogo para exercitar a prova de que as várias linguagens lá disponíveis não são LR.
 - Link útil:
<http://contentbuilder.merlot.org/toolkit/html/stitch.php?s=57516836801094&id=724306996655>