Teoria da Computação
Linguagens Regulares (Parte 2)
Linguagens não Regulares - *Pumping Lemma*

Prof. Jefferson Magalhães de Morais

Linguagens que não são regulares

- Um resultado teórico importante advém da existência de linguagens que não são regulares, resultando em

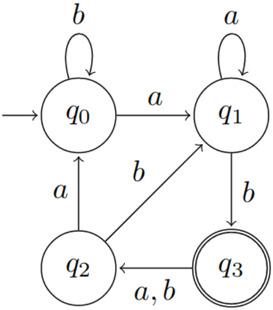
  Linguagens Regulares $\subset$ Linguagens Livre de Contexto

- O *Pumping Lemma* (Lema do Bombeamento) é uma das formas mais usuais para se provar que determinadas linguagens não são regulares
  - Estabelece uma **propriedade** para linguagens regulares
  - A propriedade é **sempre** verdadeira para linguagens regulares
  - Caso contrário, a linguagem não é regular

- **Teorema ("Pumping Lemma" para linguagens regulares)**: "Seja L um conjunto regular infinito. Existe uma constante n, dependente apenas de L, tal que, para quaisquer sentenças $w \in L$, com $|w| \geq n$, w pode ser subdividida em três subcadeias x, y, z, de tal forma que $w = xyz$ satisfazendo as seguintes condições
  - $|y| \geq 1$
  - $|xy| \leq n$, ou seja, $1 \leq |y| \leq n$
  - $xy^i z \in L, \forall i \geq 0$"

- **O reconhecimento de qualquer cadeia** $w \in L$, com $|w| \geq n$, sendo L aceita por um autômato finito M com n estados, **ocorre percorrendo-se pelo menos dois estados idênticos entre** $n + 1$ **configurações** assumidas por M durante o reconhecimento dos primeiros n símbolos de w

- **Exemplo**: considere o autômato M que reconhece a $L(M) = b^*a^+b((a|b)(ab^*a^+b|ba^*b))^*$
  - Considere $n = 4$ e a cadeia $w = abbbab, |w| \geq n, \therefore 6 \geq 4$
  - $\underbrace{q_0 \xrightarrow{a} {\color{red}q_1} \xrightarrow{b} q_3 \xrightarrow{b} q_2 \xrightarrow{b} {\color{red}q_1}}_{n+1 \text{ estados}} \xrightarrow{a} q_1 \xrightarrow{b} q_3$



Pumping Lemma

- Considerando-se os $n+1$ estados inicialmente percorridos por $M(q_0, q_1, \ldots, q_n)$, é fato que pelo menos dois desses estados devem ser idênticos. Existem então duas possibilidades extremas a considerar
  1. A distância entre eles é a menor possível: $(q_i, a_k \ldots a_m) \vdash (q_j, a_{k+1} \ldots a_m), q_i = q_j, j \leq n$
  2. A distância entre eles é a maior possível: $(q_0, a_1 \ldots a_m) \vdash^* (q_n, a_{n+1} \ldots a_m), q_0 = q_n$

- **Exemplo anterior**: $w = abbbab$ e
  $q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_3 \xrightarrow{b} q_2 \xrightarrow{b} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_3$
  - Reescrevendo w como xyz
    - x é a parte da cadeia que leva M à primeira ocorrência de um estado repetido
    - y é a parte da cadeia que leva M à sua segunda ocorrência, obedecendo:
      $|y| \geq 1$
      $|xy| \leq n$
      $xy^iz \in L, \forall i \geq 0$
    - Temos que $x = a, y = bbb$, e $z = ab$

Pumping Lemma

$$q_0 \xrightarrow{a} \color{red}{q_1} \color{black} \xrightarrow{b} q_3 \xrightarrow{b} q_2 \xrightarrow{b} \color{red}{q_1} \color{black} \xrightarrow{a} q_1 \xrightarrow{b} q_3$$

- y leva o autômato de um estado q_i (anterior ao seu reconhecimento) para o mesmo estado $q_j = q_i$ (posterior ao seu reconhecimento)
- Esse fato caracteriza como um **ciclo** o caminho percorrido pelos estados de M, com os símbolos de y
- Por ser um ciclo, repetições arbitrárias do mesmo conduzem ao reconhecimento de sentenças também pertencentes à linguagem definida pelo autômato

A constante n

- Admitindo que exista um autômato finito mínimo que reconhece uma linguagem regular L, é natural que se considere o **número de estados** do correspondente autômato finito como o valor n inerente à linguagem L
- Note que, embora o teorema prove a existência da constante n, a sua aplicação em casos práticos **não** exige que se determine o valor dessa constante

Principal aplicação do Pumping Lemma

- Geralmente prova-se que uma linguagem não é regular por **contradição**
  1. Admite-se inicialmente, por hipótese, que a linguagem seja regular
  2. Através de manipulações, demonstra-se que a linguagem não exibe as propriedades descritas pelo *Pumping Lemma*
  3. Conclui-se que a hipótese não é verdadeira e a linguagem não é regular

- Em termos práticos
  1. Admite-se n como um parâmetro do problema, caso não seja conhecido
  2. Escolhe-se uma $w \in L$ com o comprimento mínimo exigido
  3. Manipula-se w buscando identificar outras cadeias conforme o *Pumping Lemma*, mas que **violem a definição da linguagem**
  4. Havendo sucesso, conclui-se que a linguagem considerada não é regular

Exemplo 1 de linguagem que não é regular

- Seja $L = \{a^k b^k \mid k \geq 0\}$. Supondo que L seja uma linguagem regular, tome-se a sentença $a^n b^n$, onde n é a constante definida no *Pumping Lemma* e $|a^n b^n| = 2n, \therefore 2n \geq n$

  Segundo o *Pumping Lemma*, $a^n b^n = xyz$, tais que $|xy| \leq n, |y| \geq 1$

  Logo, $y = a^i, 1 \leq i \leq n$ e xyz pode ser reescrita como $a^{n-i} a^i b^n$

  No entanto, nenhuma das seguintes cadeias pertencem a L

  1. $xy^0 z = a^{n-i} b^n$
  2. $xyyz = a^{n-i} a^i a^i b^n = a^{n+i} b^n$

  uma vez que as ocorrências do símbolo a estão desbalanceadas em relação às ocorrências dos símbolos b. Logo, L não é regular

Exemplo 2 de linguagem que não é regular

- Seja $L = \{0^k 1 0^k \mid k \geq 1\}$ e uma sentença w de comprimento suficientemente longo pertencente a esta linguagem, $w = 0 \dots 010 \dots 0$

  Admite-se $w = xyz$, tem-se que $1 \leq y \leq n$, onde n é a constante de L, e y pode assumir uma das cinco formas seguintes

  1. $y = 1$
  2. $y \in 0^+$
  3. $y \in 0^+1$
  4. $y \in 10^+$
  5. $y \in 0^+10^+$

  Percebe-se que, se $y = 1$, então $xy^0z \notin L$, pois faltará o "1"

  Se $y \in 0^+$, então $xyyz \notin L$, pois haverá quantidades diferentes de "0"

  Se $y \in 0^+1, y \in 10^+$, ou $y \in 0^+10^+$, então $xyyz \notin L$, uma vez que $xyyz$ terá mais que um único "1"