Insper

# Lógica da Computação - 2018/2

Aula 06 - 22/Ago/2018

Raul Ikeda - rauligs@insper.edu.br

**Objetivos**

1. Consolidar o conhecimento sobre Gramáticas Regulares.
2. Apresentar o *Pumping Lemma*.

**Cenário**

Um engenheiro desenvolveu um protocolo de comunicação entre um Aplicativo e um IoT. O IoT envia os dados de dois sensores via Bluetooth, contudo não há como saber quantos bytes serão enviados por cada sensor. É sabido que existem alguns valores reservados que os sensore não transmitem. Portanto o engenheiro estabeleceu que:

- Há um byte de início de transmissão representado por 0xFA.
- Há um byte que separa os dados do sensor 1 do sensor 2: 0xFB.
- Finalmente há um byte que representa o fim da transmissão: 0xFC.

*Exemplo*: '**0xFA** 0x24 0x12 0x1D 0x00 0x23 **0xFB** 0x23 0x0A 0x04 **0xFC**'

1. Escreva uma gramática linear à direita que representa o protocolo de comunicação acima.

2. Demonstre que essa gramática é regular.

3. Converta a gramática para um autômato finito.

4. Justifique se o autômato finito é determinístico ou não determinístico.

5. Escreva a expressão regular equivalente à gramática acima.

6. Finalmente escreva a linguagem correspondente à gramática acima.

7. Se ainda não o fez, escreva o AFD correspondente.

**Pumping Lemma**

**Teorema**: Seja $L$ uma linguagem regular infinita. Então, existe uma constante $n$ (que depende de $L$) tal que, para todo string $w$ em $L$ tal que $|w| \geq n$, podemos dividir $w$ em três strings, $w = xyz$, tais que:

- $y \neq \lambda$
- $|xy| \leq n$
- Para todo $k \geq 0$, o string $xy^kz$ também está em $L$.

**Ideia da Demonstração:**

```
>> Ver Cap. 3.9 Ramos et al
```

***Exemplo:*** $L = \{a^m b^m | m \geq 0\}$

**Etapa de Manutenção**

Devido à problemas na comunicação do dispositivo, havia uma grande perda de dados. Então delegaram ao estagiário implementar um *checksum* no final da transmissão. Em um ímpeto de otimização e produtividade o estagiário desenhou o *checksum* da seguinte maneira:

- após o byte final 0xFC iria colocar um bit 1 para cada byte enviado, exceto os bytes de controle.
- Exemplo: bin(0xFA 0x01 0x0D 0x04 0xFB 0x20 0x10 0xFC) + 11111

1. Explique porque isso foi uma péssima decisão.

2. Prove formalmente que a gramática **não é** regular.

3. Como se prova que uma gramática **é** regular?

TGF: Ramos et al. Pag. 239-254

**Propriedades de Fechamento**

As linguagens regulares são fechadas em relação às seguintes operações:

- União, concatenação e fecho
- Complemento
- Intersecção

Teorema: "A classe das linguagens regulares é fechada em relação à operação de intersecção."

**Verificação**: Considere $L_1$ sobre $\Sigma_1$ e $L_2$ sobre $\Sigma_2$, sendo $\Sigma_1, \Sigma_2 \subseteq \Sigma$. Através da Lei de Morgan, a seguinte relação é verdadeira:

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Como as linguagens regulares são fechadas em relação à união e complemento, a intersecção duas linguagens regulares também é uma linguagem regular.

- Substituição
- Homomorfismo e homomorfismo inverso
- Quociente
- Reversão

**Questões Decidíveis**

1. As linguagens são idênticas?

Teorema: "Sejam $L(G_1)$ e $L(G_2)$ duas linguagens regulares quaisquer. Então, a questão $L(G_1) = L(G_2)$ é **decidível**"

**Verificação**: Seja $L_1 = L(M_1)$ e $L_2 = L(M_2)$, onde $M_1$ é um AFD que representa $L_1$ e $M_2$ é um AFD que representa $L_2$. Portanto, se $L_1 = L_2$, então:

$$(L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2) = \emptyset$$

2. A linguagem é vazia?
3. A linguagem é infinita?
4. A linguagem é finita?
5. Uma cadeia pertence à uma linguagem?
6. A lingugem é $\Sigma^*$
7. A linguagem é um subconjunto de outra linguagem?

**Lista de Exercícios**

Ramos et al Cap. 3.13: Exercícios 4, 6, 7, 14, 25, 34, 37, 41 e 82.

**Próxima aula: J. J. Neto - Cap. 5**

1. Operadores Unários
2. Parênteses