

Linguagens Formais e Autômatos

Prof. Alex Luciano Roesler Rese, MSc.

Adaptado: Rafael de Santiago, Dr.



Linguagens Sensíveis ao Contexto







- Linguagens Sensíveis ao Contexto englobam as linguagens regulares, livres de contexto
- Além destas, há, no conjunto das LSCs, um grupo de linguagens que possuem sentenças com trechos que são consideradas válidas se ocorrem simultaneamente, ou seja, há uma sensibilidade a um contexto (dependência de contexto)







Vejamos a seguinte gramática:

```
E→ E+E
|E-E
|E*E
|E/E
|(E)
|a
|b
```







Vejamos a seguinte gramática:

Há uma certa dependência de contexto: ao abrir os parênteses, deve-se fechá-los.







Vejamos a seguinte gramática:

Mas esta é uma Linguagem Livre de Contexto.

```
| E /
| ( E
| a
| b
```







 Note que a dependência que vimos, pode ser representada por gramáticas livres de contexto

 Mas existem dependências mais complexas, que não podem ser expressas por LLCs



- É importante vermos que estas dependências existem nas linguagens de programação em alto nível mais comuns
- Tais linguagens oferecem declarações de variáveis, funções, funções... há um propósito duplo nas declarações, que envolvem dependência, o tipo de variável, a quantidade e o tipo de retorno de funções







 Uma vez declarados, os novos nomes devem fazer parte do programa. Por exemplo: somente posso utilizar do lado esquerdo da atribuição, uma variável já declarada.

```
int main() {
   int x;
   cin>>x;
   cout<<x;
   return 0;
}</pre>
```



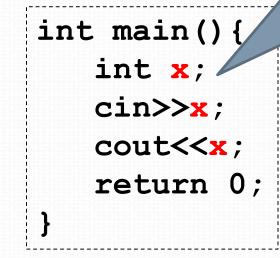




 Uma vez declarados, os novos nomes devem fazer parte do programa. Por q utilizar do lado esquerdo variável já declarada.

```
mesma na tela!
```

posso Depende-se de x ter sido declarado, para na que se realize a leitura e gravação na variável e a impressão da









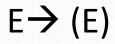
Gramáticas Sensíveis ao Contexto







- As Gramáticas Livres de Contexto possuem uma incapacidade de representar Linguagens Sensíveis ao Contexto
- Por utilizarem apenas um não-terminal do lado esquerdo das produções, as GLCs consideram que apenas estes podem ser substituídos
- Na substituição, não importa o que há na esquerda nem na direita. Exemplo:









 Isto torna impossível representar contextos como no caso das declarações de variáveis (explícito anteriormente)

 Para superar tal limitação, empregam-se as Gramáticas Sensíveis ao Contexto – GSC!







Definição formal de uma GSC:

$$G = (V, \Sigma, P, S)$$

 $V = \Sigma \cup N$, sendo $N = n\tilde{a}o$ - terminais
 $\Sigma = alfabeto$
 $P = \{\alpha \rightarrow \beta \mid \alpha \in V * NV *, \beta \in V *, |\beta| \ge |\alpha| \}$
 $S = n\tilde{a}o$ - terminal inicial







Definição formal de uma GSC:

$$G = (V, \Sigma, P, S)$$

 $V = \Sigma \cup N$, sendo $N = \tilde{nao} - t$

A principal diferença com as GLCs está aqui: o lado esquerdo das produções não está limitado a ser um único símbolo não terminal

$$\Sigma = alfabeto$$

$$P = {\alpha \rightarrow \beta \mid \alpha \in V * NV*, \beta \in V*, |\beta| \ge |\alpha|}$$

$$S = n\tilde{a}o$$
 - terminal inicial







 São Linguagens <u>Estritamente</u> Sensíveis ao Contexto aquelas que são Sensíveis ao Contexto, mas não Livres de Contexto.

 Vejamos, a seguir, um exemplo clássico de uma linguagem que é sensível ao contexto, mas não livre de contexto...







- Exemplo:
 - $G_1 (\{a,b,c,S,B,C\}, \{a,b,c\}, P,S), com$

```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```







Exemplo:

• $G_1 - (\{a,b,c,S,B,C\}, \{a,b,c\}, C\})$

Que linguagem esta GSC reconhece?







S

Partindo do símbolo inicial "S", vamos verificar se abc pertence a linguagem







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

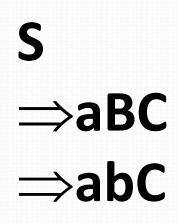
S ⇒aBC Usando a transição S → aBC







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```



Usando a transição aB → ab







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

```
S
⇒aBC
⇒abC
⇒abc
```

Usando a transição bC → bc

Provando assim que abc pertence a linguagem







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

S ⇒aSBC Utilizamos a produção

S→aSBC







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

```
S
⇒aSBC
⇒aaBCBC
```

Utilizamos a produção

S→aBC







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

```
S

⇒aSBC

⇒aaBCBC

⇒aaBBCC
```

Utilizamos a produção

 $CB \rightarrow BC$







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

```
S

⇒aSBC

⇒aaBCBC

⇒aaBBCC

⇒aabBCC
```

Utilizamos a produção

aB→ab







```
P =
{
    S → aSBC,
    S → aBC,
    CB → BC,
    aB → ab,
    bB → bb,
    bC → bc,
    cC → cc
}
```

```
\RightarrowaSBC
\RightarrowaaBCBC
\RightarrowaaBBCC
⇒aabBCC
\RightarrowaabbCC
```

Utilizamos a produção

bB→bb







```
\RightarrowaSBC
\RightarrowaaBCBC
⇒aaBBCC
⇒aabBCC
⇒aabbCC
\RightarrowaabbcC
```

Utilizamos a produção

bC→bc







```
\RightarrowaSBC
\RightarrowaaBCBC
⇒aaBBCC
\RightarrowaabBCC
⇒aabbCC
⇒aabbcC
\Rightarrowaabbcc
```

Utilizamos a produção

 $cC \rightarrow cc$







```
\RightarrowaSBC
\RightarrowaaBCBC
⇒aaBBCC
⇒aabBCC
⇒aabbCC
⇒aabbcC
```

 \Rightarrow aabbcc

Provamos que a sentença "aabbcc" pertence a linguagem desta GSC







Verifiquem se a sentença a seguir também pertence

aaaabbbbcccc







A linguagem que estamos nos referindo é:

```
L(G1) = \{anbncn \mid n \ge 1\}
```





