



# Construção de Compiladores

## Linguagem

*Professor: Luciano Ferreira Silva, Dr.*



# Linguagem

## ■ Definição:

- ✓ Conjunto de símbolos e um conjunto de regras para combinar estes símbolos, que é usado e entendido por uma determinada comunidade.
- ✓ Exemplos:
  - as linguagens naturais ou idiomas;
  - linguagens de programação
  - protocolos de comunicação



# Alfabeto

## ■ Definição:

- ✓ É o conjunto de símbolos básicos da linguagem;
- ✓ Exemplos:
  - Alfabeto da Língua Portuguesa  $\{a, b, c, \dots, z, A, B, C, \dots, Z\}$ ;
  - Alfabeto da Linguagem de Máquina (Linguagem Binária)  $B = \{0, 1\}$ ;

## ■ Strings de um Alfabeto:

- ✓ Combinação (ou concatenação) de símbolos de um alfabeto;
- ✓ Exemplos:
  - São *strings* de  $B$ : 0, 001, 1110101, etc.;
  - Não são *strings* de  $B$ : 012, 114, a21, etc.;
- ✓ Uma *string* vazia é representada usualmente por  $\varepsilon$ .



# Alfabeto

## ■ Seja o conjunto $A$ um alfabeto:

- ✓ A clausura de  $A$ , denotada por  $A^*$ , é o conjunto de todas as *strings*, incluindo a *string* vazia, compostas a partir do alfabeto  $A$ ;
- ✓ Exemplo:
  - A clausura  $B$  é  $B^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$
- ✓  $A^+$  é o conjunto de todas *strings* compostas a partir do alfabeto  $A$ , exceto a *string* vazia;
  - Ou seja,  $A^+ = A^* - \epsilon$ ;



# Linguagens em notação de conjuntos

- Seja  $L$  definida sobre o alfabeto:

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

- ✓ Onde  $b^n$  representa uma seqüência de  $n$  ocorrências do símbolo  $b$ ;
- ✓ Exemplos de *strings* válidas em  $L$ : 01, 0011, 000111, 00001111,...
- ✓ Exemplos de *strings* não válidas: 10, 00111,...

- Obs.:  $L$  é uma linguagem simples de ser representada.



# Gramáticas

## ■ Definição:

- ✓ Mecanismos de representações de linguagens;

## ■ Produção:

- ✓ Regra para transformar uma seqüência de símbolos em outra;
- ✓ Representada pelo símbolo  $\rightarrow$  para indicar que o lado esquerdo da relação pode ser substituído pelo direito;
- ✓ Símbolos não-terminais:
  - São símbolos intermediários na substituição;
  - Não pertencem ao conjunto de *strings*;





# Gramáticas

- As produções usam um símbolo não-terminal (chamado símbolo sentencial) como ponto de partida;
- ✓ Símbolos terminais:
  - Compõem o conjunto *strings* válidas da linguagem;
- ✓ Exemplo de produções para a linguagem  $L$  no alfabeto  $B$ :
  - Seja  $Z$  um símbolo sentencial (perceba  $Z$  não pertence a  $B$ )
  - Primeira produção  $Z \rightarrow \varepsilon$
  - Segunda produção  $Z \rightarrow 0Z1$



# Gramáticas

- Representada formalmente por uma quádrupla:  $G = (V_T, V_N, P, S)$ ;
  - ✓  $V_T$  é um conjunto de símbolos terminais;
  - ✓  $V_N$  é um conjunto de símbolos não-terminais;
  - ✓  $P$  é um conjunto de produções expressas na forma  $\alpha \rightarrow \beta$ , com  $\alpha \in (V_N \cup V_T)^+$ ;  $\beta \in (V_N \cup V_T)^*$ , onde  $\alpha$  é, no caso geral, uma cadeia contendo no mínimo um não-terminal;
  - ✓  $S \in V_N$ , é o símbolo sentencial, ponto de partida para criação de qualquer sentença na linguagem;
  - ✓ Exemplo, gramática referente a linguagem  $L$ :
    - $G_L = (\{0, 1\}, \{Z\}, \{Z \rightarrow \varepsilon, Z \rightarrow 0Z1\}, Z)$





# Gramáticas

## ■ Derivações:

- ✓ É a substituição de símbolos que combinam com o lado esquerdo de uma produção pelo lado direito correspondente, representada por  $\Rightarrow$ .
- ✓ Em  $G_1$  há duas derivações válidas a partir do símbolo sentencial  $Z$ : (1)  $Z \Rightarrow \varepsilon$ , (2)  $Z \Rightarrow 0Z1$
- ✓ Exemplo:  $Z \xRightarrow{(2)} 0Z1 \xRightarrow{(2)} 00Z11 \xRightarrow{(1)} 0011$
- ✓ Para omissões escreve-se:  $Z \xRightarrow{+} 0011$



# Classificação de gramáticas

## ■ Hierarquia de Chomsky:

- ✓ Tipo 0 ou recursivamente enumeráveis: sem restrição quanto ao tipo de produção;
  - Exemplo:  $RRZ \rightarrow \alpha Z$ ;
- ✓ Tipo 1 ou sensíveis ao contexto: têm uma ou mais produções com lado esquerdo com mais de um símbolo e com lado direito com pelo menos a mesma quantidade de símbolos;
  - Exemplo:  $RZ \rightarrow \alpha Z$  ou  $RZ \rightarrow \alpha Z\beta$



# Classificação de gramáticas

- ✓ Tipo 2 ou livres de contexto: têm todas as suas produções com apenas **um** símbolo do lado esquerdo. O lado direito das produções pode:
  - ter qualquer quantidade de símbolos
  - ser recursivo (O mesmo símbolo no dois lados)
    - podendo ser auto-incorporativo ( $A \rightarrow \alpha_1 A \alpha_2$ );
  - Exemplos:
    - $Z \rightarrow 0Z1$ ;
    - $E \rightarrow ( E )$ ;



# Classificação de gramáticas

- ✓ Tipo 3 ou regulares: têm todas as suas produções com apenas **um** símbolo do lado esquerdo. O lado direito das produções pode:
  - ter qualquer quantidade de símbolos;
  - ser recursivo;
    - **não** deve ser auto-incorporativo ( $A \rightarrow \alpha A$  ou  $A \rightarrow A\alpha$ );
  - Exemplo: Seja  $l$  e  $d$  representantes quaisquer de letras e dígitos respectivamente. Portanto, um conjunto de produções para construir identificadores em linguagens de programação seria:  $N \rightarrow lR$ ,  $R \rightarrow lR$ ,  $R \rightarrow dR$ ,  $R \rightarrow \varepsilon$ , sendo  $N$  o símbolo sentencial;



# Notações alternativas

## ■ Expressões regulares

- ✓ Forma alternativa de expressar uma linguagem regular;
- ✓ São recursivamente definidas pelo conjunto de regras:
  - A *string* vazia é uma expressão regular;
  - Dado um alfabeto  $A$ , então um elemento de  $A$  é uma expressão regular em  $A$ ;
  - Se  $P$  é uma expressão regular em  $A$ , então a repetição de 0 ou mais ocorrências de  $P$ , denota  $P^*$ , também é uma expressão regular em  $A$ ;





# Notações alternativas

- ✓ Se  $P$  e  $Q$  são expressões regulares em  $A$ , então a seqüência de  $P$  e  $Q$ , denotada  $PQ$ , também é uma expressão regular em  $A$ ;
- ✓ Se  $P$  e  $Q$  são expressões regulares em  $A$ , então a alternativa entre  $P$  e  $Q$ , denotada  $P \mid Q$ , também é uma expressão regular em  $A$ ;
- ✓ Portanto, existem três operações aplicáveis a um alfabeto:
  - Concatenação: é associativa (  $xyz = (xy)z$  ) e **não** comutativa ( $xy \neq yx$ );
  - Repetição (  $*$  ): possui maior precedência;
  - Alternativa (  $\mid$  ): é comutativa ( $x \mid y = y \mid x$  ) e associativa (  $x \mid y \mid z = (x \mid y) \mid z$  ) e possui menor precedência;
  - Parênteses podem ser usados para alterar ou explicitar as precedências.





# Notações alternativas

## ■ Exemplos:

✓  $aa^* \rightarrow a, aa, aaa, aaaa, \dots$

- $aa^* \neq (aa)^* \rightarrow \varepsilon, aa, aaaa, aaaaaa, \dots$

- $(\{a\}, \{S\}, \{S \rightarrow aS, S \rightarrow a\}, S)$

✓  $a(a|b)^*b \rightarrow ab, aab, abb, aabb, abab, \dots$

- $(\{a, b\}, \{S, Q\}, \{S \rightarrow aQb, Q \rightarrow aQ, Q \rightarrow Qb, Q \rightarrow \varepsilon\}, S)$

✓  $ba|a^*b \rightarrow ba, b, ab, aab, \dots$

✓ Para  $(\{l, d\}, \{N, R\}, \{N \rightarrow lR, R \rightarrow lR, R \rightarrow dR, R \rightarrow \varepsilon\}, N)$   
 $\rightarrow l(l|d)^*$



# Notações alternativas

## ■ Backus-Naur Form (BNF):

- ✓ Operador binário ( $::=$ ): descreve as produções da gramática;
  - O operando do lado esquerdo é um símbolo não-terminal;
  - O operando do lado direito é uma expansão, que pode conter símbolos terminais e não-terminais;
- ✓ Símbolos não-terminais são delimitados por colchetes angulares,  $<$  e  $>$ . Qualquer outro símbolo são considerados terminais.
  - $\langle \text{expr} \rangle ::= ( \langle \text{expr} \rangle )$



# Notações alternativas

## ✓ Operador de alternativas de expansão ( | )

- $\langle S \rangle ::= \langle A \rangle | \langle B \rangle$ 
  - $\langle S \rangle ::= \langle A \rangle$  ou  $\langle S \rangle ::= \langle B \rangle$

## ✓ Operador opcional [ ]

- $\langle S \rangle ::= [\langle A \rangle] \langle B \rangle$ 
  - $\langle S \rangle ::= \langle A \rangle \langle B \rangle$  ou  $\langle S \rangle ::= \langle B \rangle$

## ✓ Operador de repetição \*

- $\langle S \rangle ::= \langle A \rangle^* \langle B \rangle$ 
  - $\langle S \rangle ::= \langle A \rangle \langle S \rangle$  ou  $\langle S \rangle ::= \langle B \rangle$

## ✓ Limitação da BNF: os metacaracteres não podem fazer parte da linguagem;



# Notações alternativas

## ■ Diagramas sintáticos

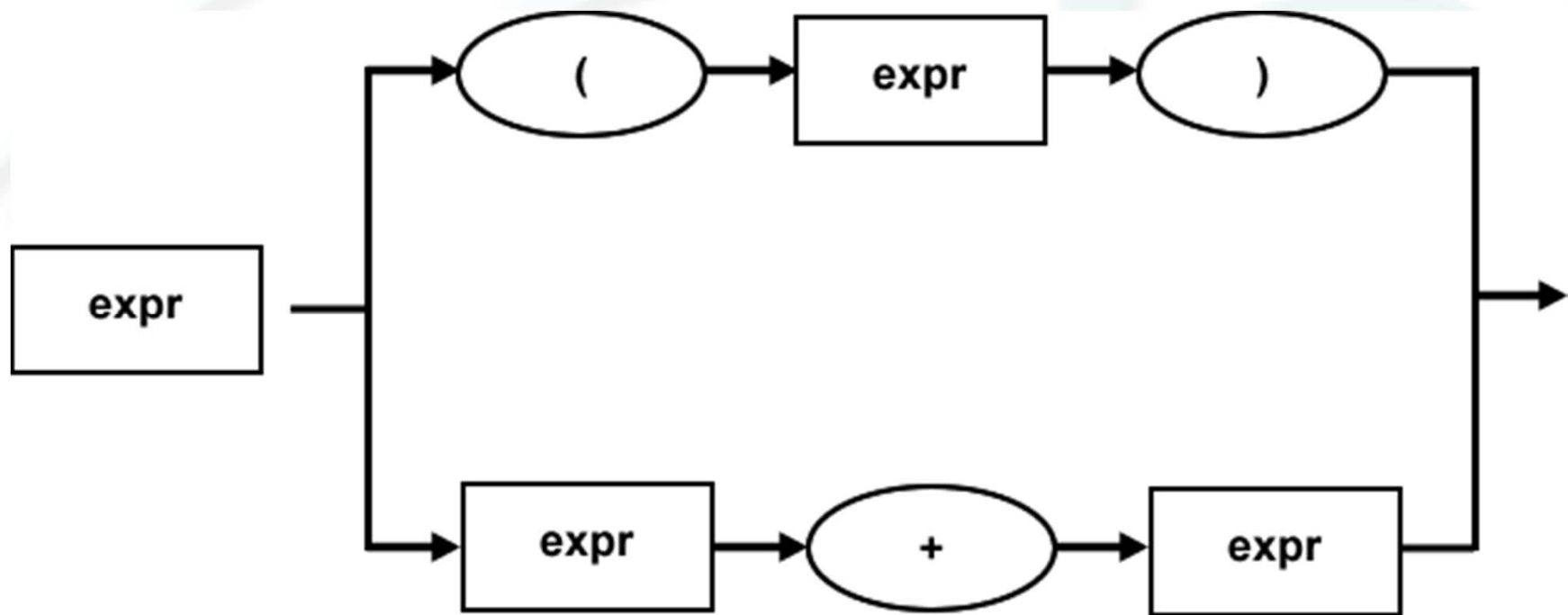
- ✓ Notação gráfica usual para gramáticas livres de contexto;
- ✓ Símbolos não-terminais são representados em retângulos;
- ✓ Símbolos terminais são representados em elipses;
- ✓ As produções são representadas como um grafo dirigido, no qual cada caminho do início ao fim indica seqüência de símbolos que pode expandir o símbolo representado;



# Notações alternativas

## ■ Exemplo:

✓  $\langle \text{expr} \rangle ::= \langle \text{expr} \rangle + \langle \text{expr} \rangle \mid (\langle \text{expr} \rangle)$

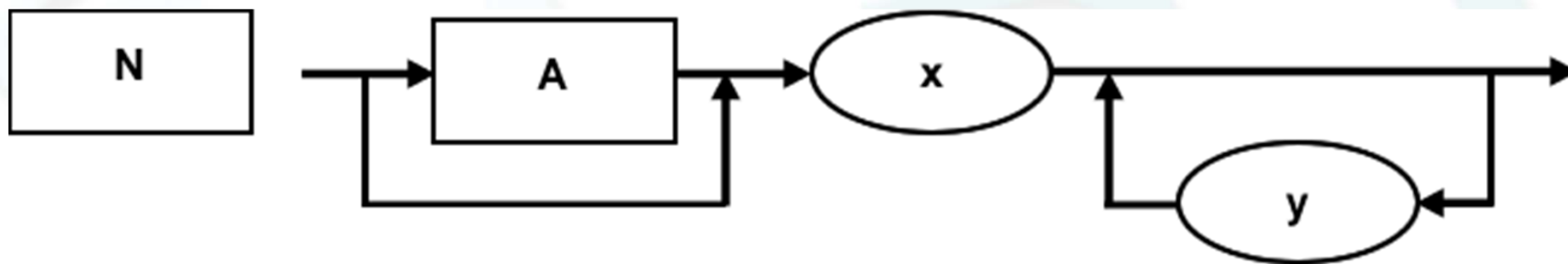




# Notações alternativas

## ■ Exemplo:

✓  $\langle N \rangle ::= [\langle A \rangle]xy^*$







# Notações alternativas

## ■ Vantagens:

- ✓ É uma representação visual para as regras da gramática, com o mesmo poder de BNF;
- ✓ É adequada para fins de documentação e comunicação das produções da gramática para seres humanos;

## ■ Desvantagem:

- ✓ Não é adequada para processamento por computadores;