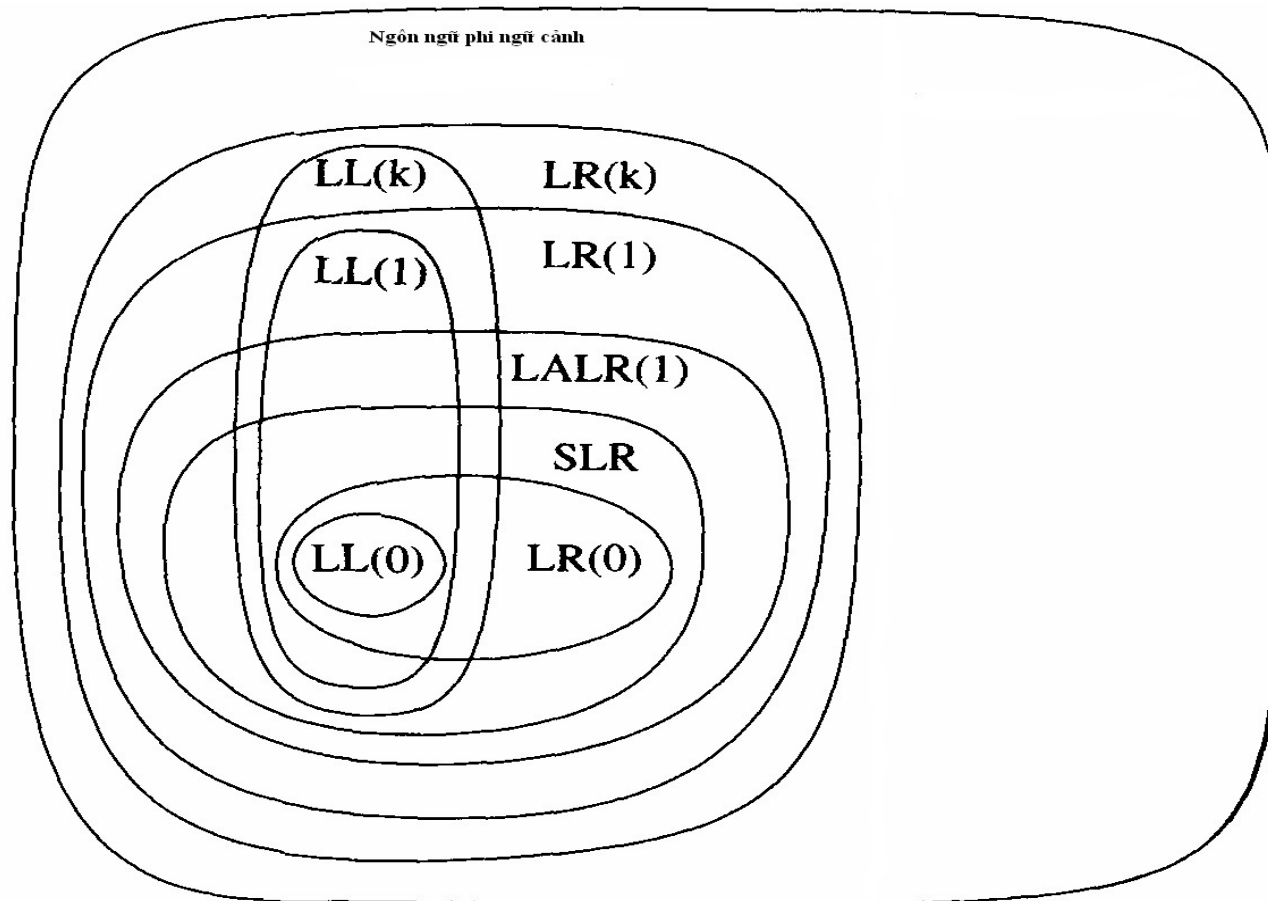




# Bài 8.

## Văn phạm LL(k)

# Phân cấp các ngôn ngữ phi ngữ cảnh





# Ngôn ngữ LL(k)

- Xem trước k ký hiệu trên xâu vào để quyết định sản xuất được sử dụng
- Được sinh ra nhờ văn phạm LL(k)



# FIRST<sub>k</sub>( $\alpha$ )

Định nghĩa : Cho văn phạm  $G$  phi ngữ cảnh, số nguyên dương  $k$  ,  $\alpha$  là một xâu bao gồm ký hiệu kết thúc và không kết thúc

*FIRST<sub>k</sub>( $\alpha$ ) là tập các xâu  $x$  gồm  $k$  ký hiệu kết thúc trái nhất của các xâu suy dẫn từ  $\alpha$  (Kể cả trường hợp  $x$  không có đủ  $k$  ký hiệu nhưng  $\alpha$  suy dẫn ra  $x$  , không còn ký hiệu nào sau  $x$ )*

# FIRST<sub>k</sub>( $\alpha$ )

Định nghĩa : Cho văn phạm  $G = (\Sigma, \Delta, P, S)$ , số nguyên dương  $k$ ,  $\alpha \in V^*$

$$\text{FIRST}_k(\alpha) = \{ x \in \Sigma^* \mid \alpha \Rightarrow^* x\beta \text{ và } |x| = k \text{ hoặc } \alpha \Rightarrow^* x \text{ và } |x| < k \}$$

( Tập các chuỗi  $x \in \Sigma^*$  có  $k$  ký hiệu trái nhất suy dẫn từ  $\alpha$  ( Kể cả trường hợp  $x$  không có đủ  $k$  ký hiệu nhưng  $\alpha x$ , không còn ký hiệu nào sau  $x$ ))



# $\text{FOLLOW}_k(\alpha)$

*k ký hiệu kết thúc đầu tiên tiếp sau xâu được suy dẫn từ  $\alpha$ .*

Đặc biệt , khi A là ký hiệu không kết thúc, S suy dẫn ra  $\beta A$  thì  $\text{FOLLOW}_1(A) = \{\$ \}$



# $\text{FOLLOW}_k(\alpha)$

$$\text{FOLLOW}_k(\alpha) = \{x \in \Sigma^* \mid S \Rightarrow^* \beta\alpha\delta \text{ và } x \in \text{FIRST}_k(\delta)\}$$

Đặc biệt , khi  $\alpha = A \in \Delta^*$  ,  $S \Rightarrow^* \beta A$  thì  
 $\text{FOLLOW}_1(A) = \{\$ \}$

# Văn phạm LL(k)

Định nghĩa văn phạm phi ngữ cảnh  $G = (\Sigma, \Delta, P, S)$  là LL(k) với k cho trước nếu với mọi cặp suy dẫn trái

$$S \Rightarrow^* xA\alpha \Rightarrow x\beta_1\alpha \Rightarrow^* xZ_1$$

$$S \Rightarrow^* xA\alpha \Rightarrow x\beta_2\alpha \Rightarrow^* xZ_2$$

Nếu  $\text{FIRST}_k(Z_1) = \text{FIRST}_k(Z_2)$  thì  $\beta_1 = \beta_2$





# Ví dụ

Văn phạm G với các sản xuất :

$$S \rightarrow aAS \mid b$$

$$A \rightarrow bSA \mid a$$

là LL(1)



# Văn phạm LL(1) đơn giản

Văn phạm  $G = (\Sigma, \Delta, P, S)$  là LL(1) đơn giản nếu mọi sản xuất của văn phạm có dạng

$$A \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid \dots \mid a_n\alpha_n, a_i \in \Sigma \ 1 \leq i \leq n$$

Trong đó  $a_i \neq a_j$  với  $i \neq j$

# Điều kiện nhận biết văn phạm LL(1)

- Định lý Văn phạm  $G = (\Sigma, \Delta, P, S)$  là LL(1) khi và chỉ khi mọi tập A- sản xuất trong P có dạng
- $A \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n, n \geq 2$  thoả mãn
$$\text{FIRST}_1(\alpha_i) \cap \text{FIRST}_1(\alpha_j) = \emptyset, i \neq j$$
- Nếu  $\alpha_i \Rightarrow^* \varepsilon$  thì
$$\text{FIRST}_1(\alpha_j) \cap \text{FOLLOW}_1(A) = \emptyset, j \neq i$$



## Điều kiện LL(1) trên sơ đồ cú pháp

- Ở mỗi lối rẽ, các nhánh phải bắt đầu bằng các ký hiệu khác nhau
- Nếu biểu đồ có chứa một đường rỗng thì mọi ký hiệu đứng sau ký hiệu được biểu diễn bởi biểu đồ phải khác các ký hiệu đứng đầu các nhánh của sơ đồ

# Kiểm tra điều kiện LL(1) trên văn phạm KPL

02)  $\langle \text{Block} \rangle ::= \text{KW\_CONST} \langle \text{ConstDecl} \rangle \langle \text{ConstDecls} \rangle \langle \text{Block2} \rangle$   
03)  $\langle \text{Block} \rangle ::= \langle \text{Block2} \rangle$

- $\text{FIRST}(\text{VP1}) = \{\text{KW\_CONST}\}$
- $\text{FIRST}(\text{VP2}) = \{\text{KW\_TYPE}, \text{KW\_VAR}, \text{KW\_PROCEDURE}, \text{KW\_FUNCTION}, \text{KW\_BEGIN}\}$
- $\text{FIRST}(\text{VP1}) \cap \text{FIRST}(\text{VP2}) = \emptyset$
- Các sản xuất của  $\langle \text{Block} \rangle$  thỏa điều kiện LL(1)

# Lệnh

- 49) Statement ::= AssignSt
- 50) Statement ::= CallSt
- 51) Statement ::= GroupSt
- 52) Statement ::= IfSt
- 53) Statement ::= WhileSt
- 54) Statement ::= ForSt
- 55) Statement ::=  $\varepsilon$

- FIRST (AssignSt)={TK\_IDENT}
- FIRST (CallSt)={KW\_CALL}
- FIRST (GroupSt)={KW\_BEGIN}
- FIRST (IfSt)={KW\_IF}
- FIRST (WhileSt)={KW\_WHILE}
- FIRST (ForSt)={KW\_FOR}
- FOLLOW (Statement) = {SB\_SEMICOLON, KW\_END, KW\_ELSE}

# Một số sản xuất vi phạm đk LL(1)

- 56) AssignSt ::= Variable SB\_ASSIGN Expression **VP1**  
57) AssignSt ::= FunctionIdent SB\_ASSIGN Expression **VP2**

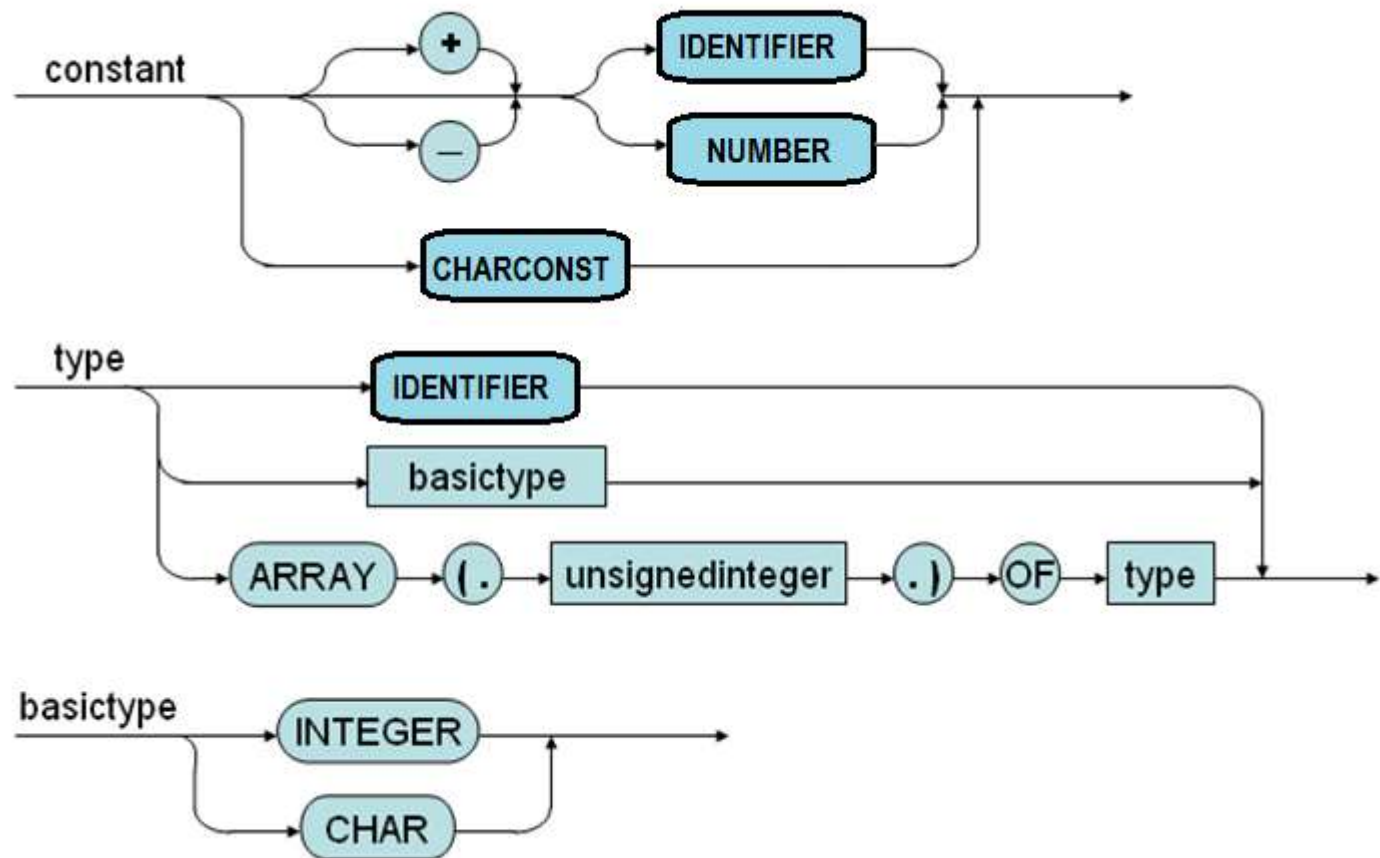
Và

- 86) Factor ::= UnsignedConstant **VP3**  
87) Factor ::= Variable **VP4**  
88) Factor ::= FunctionApptication **VP5**  
89) Factor ::= SB\_LPAR Expression SB\_RPAR **VP6**

FIRST(VP1) = {TK\_IDENT} = FIRST(VP2)  
FIRST(VP3) = {TK\_NUMBER, TK\_CHAR, TK\_IDENT}  
FIRST(VP4) = {TK\_IDENT}  
FIRST(VP5) = {TK\_IDENT}  
FIRST(VP6) = {SB\_LPAR}

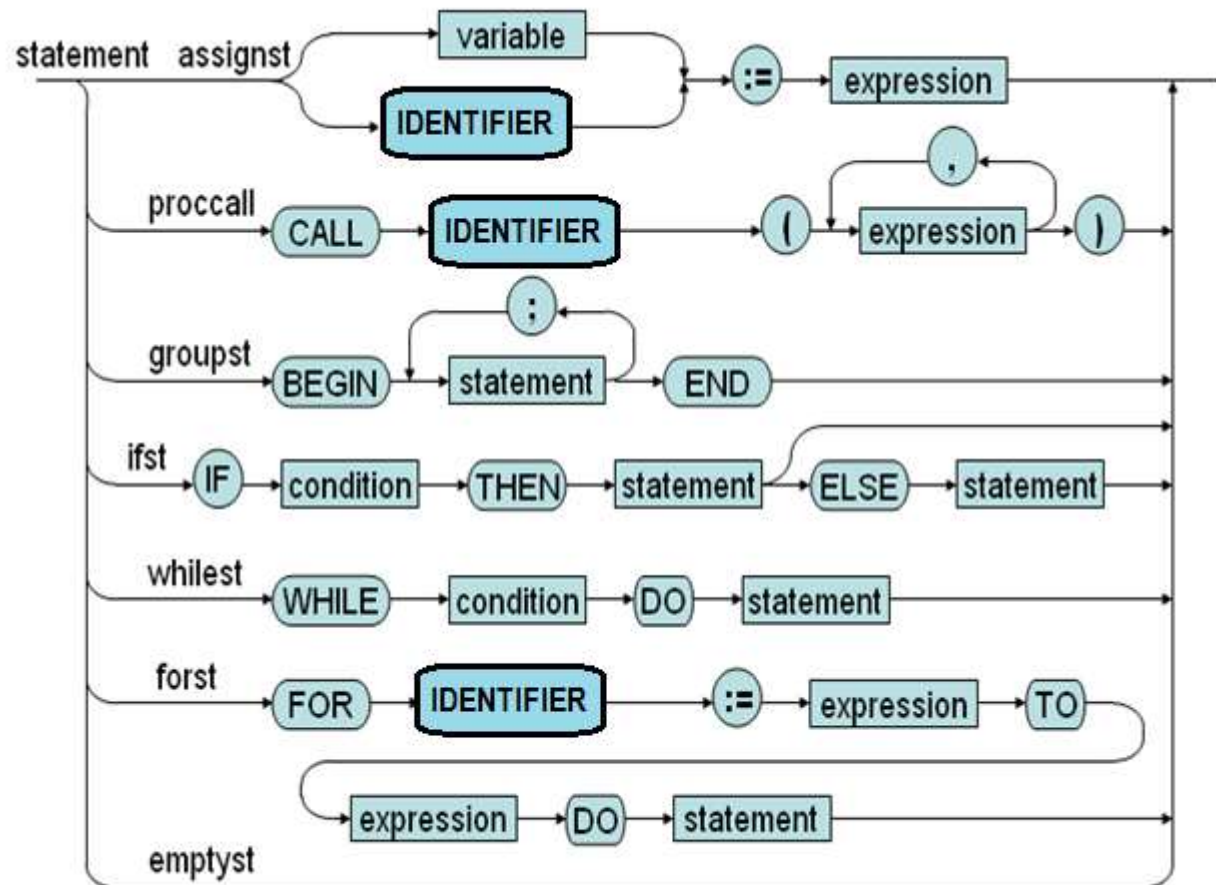
Có thể chứng minh các sản xuất này thỏa đk LL(2)

# Một số sơ đồ KPL thỏa điều kiện LL(1)

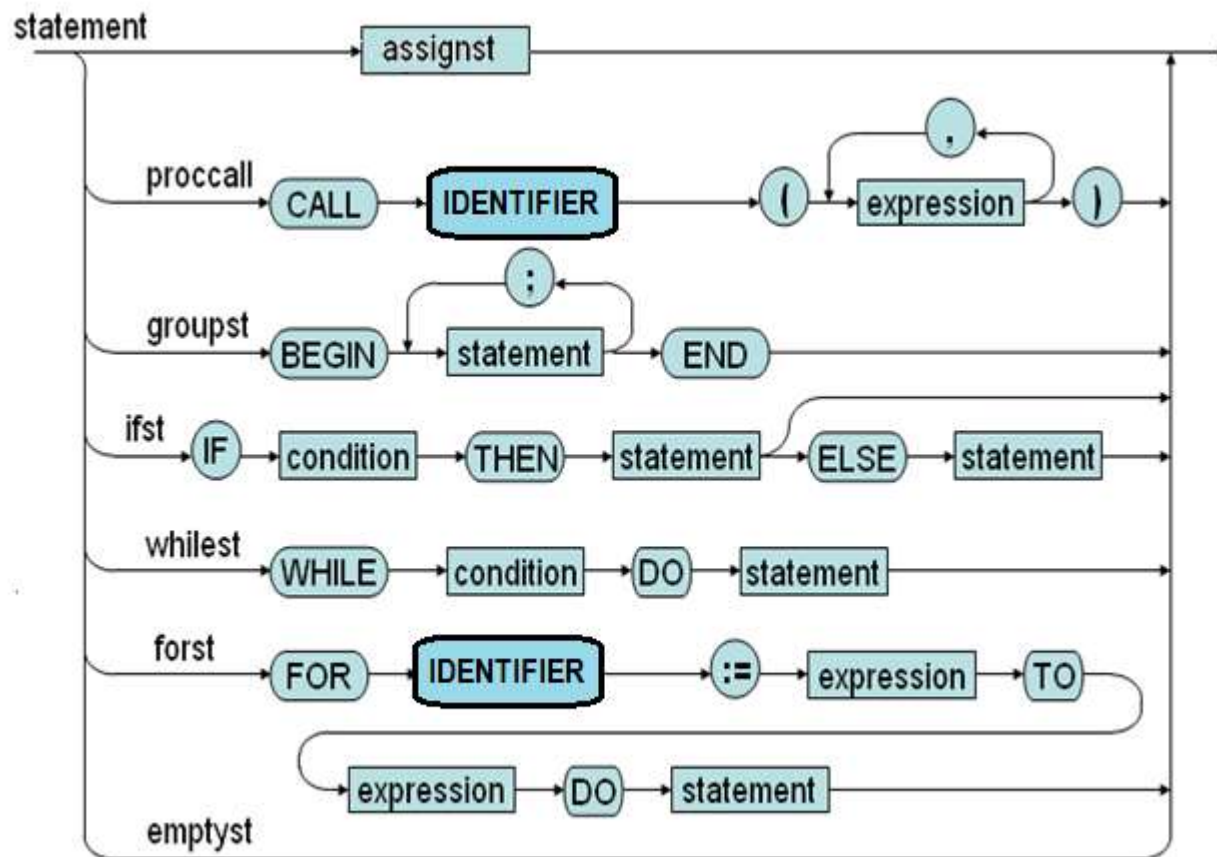




# Statement thỏa điều kiện LL(1)?

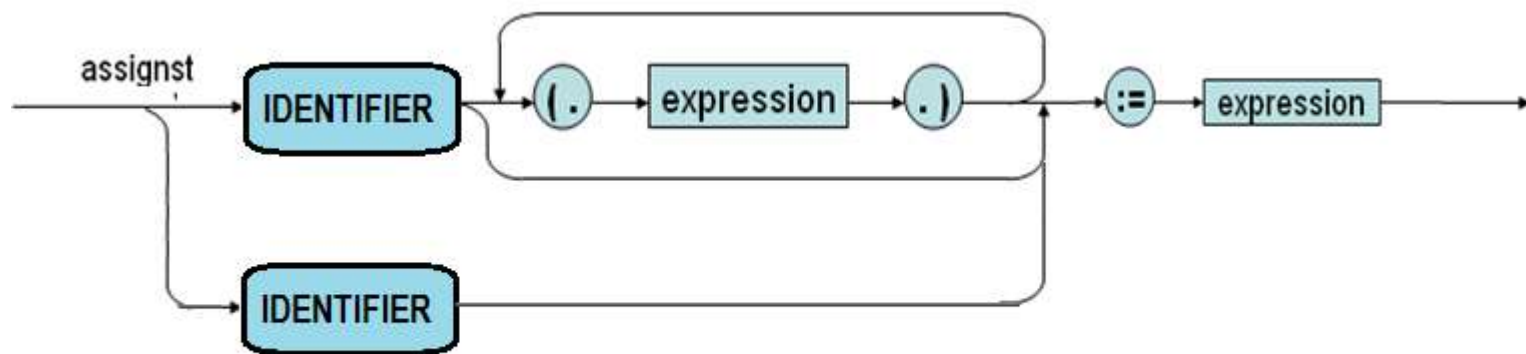


# Biến đổi sơ đồ cú pháp



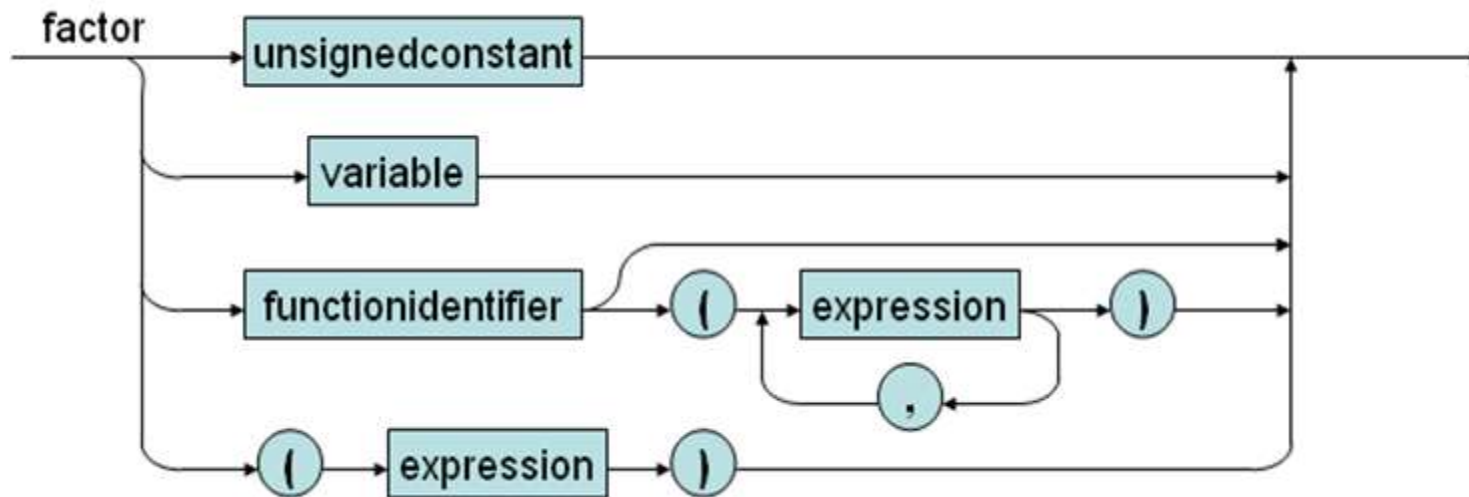
- Nhánh assignst bắt đầu bằng IDENTIFIER
- Nhánh rỗng  $\rightarrow \text{FOLLOW}(\text{statement}) = \{;, \text{ELSE}, \text{END}\} \rightarrow \text{LL}(1)$

# Sơ đồ cú pháp của assignst



- Thỏa điều kiện LL(2)

# Sơ đồ cú pháp của factor



- Khó kiểm tra điều kiện LL(1) khi **unsignedconstant**, **variable** và **functionidentifier** đều có thể là một định danh
- Cần chuyển đổi về dạng tường minh hơn. Việc phân biệt định danh đóng vai trò gì (hằng, biến, hàm) do bộ phân tích ngữ nghĩa đảm nhiệm

# Factor không thỏa điều kiện $LL(1) \rightarrow LL(2)$

