

CHƯƠNG TRÌNH DỊCH

Chương 4. Phân tích cú pháp Thuật toán LL

TS. Phạm Văn Cảnh
Khoa Công nghệ thông tin

Email: canh.phamvan@phenikaa-uni.edu.vn

-
- 1. Giới thiệu về thuật toán phân tích tất định**
 - 2. Tiếp cận top-down**
 - 3. Phân tích LL(1)**
 - 4. Bài tập**

1. Giới thiệu về thuật toán tất định

□ Hạn chế của các thuật toán trước

- Các thuật toán phân tích vụn vặt (CYK, Earley).
 - Phân tích mọi văn phạm phi ngữ cảnh.
 - Tốc độ chấp nhận được: $O(n^3)$ với n là độ dài chuỗi vào.

□ Đối với những mã nguồn các ngôn ngữ lập trình, giá trị của n có thể lên tới vài triệu, bài toán phân tích văn phạm trở nên rất đặc biệt.

- Tốc độ chấp nhận được nếu là gần tuyến tính $O(n)$.
- Văn phạm đơn giản, chặt chẽ, đơn nghĩa.

□ Hệ quả là nảy sinh nhu cầu xây dựng các bộ phân tích văn phạm **tất định (deterministic)**.

1. Giới thiệu về thuật toán tất định

❑ Thế nào là “tất định” – do ràng buộc độ phức tạp tính toán là $O(n)$, hệ quả là:

- Khi nhận một kí hiệu đầu vào, bộ phân tích văn phạm cần ngay lập tức quyết định sẽ sử dụng luật sinh nào cho trường hợp này.
- Quyết định chọn luật sinh nào cần phải đủ tốt để không phải thử lại phương án khác.
- Tính chất “tất định” ~ không có quay lui.

❑ Cái giá phải trả cho sự “tất định”:

- Văn phạm sẽ không còn vạn năng nữa.
- Nhưng văn phạm đủ tốt để dùng trong thực tế.

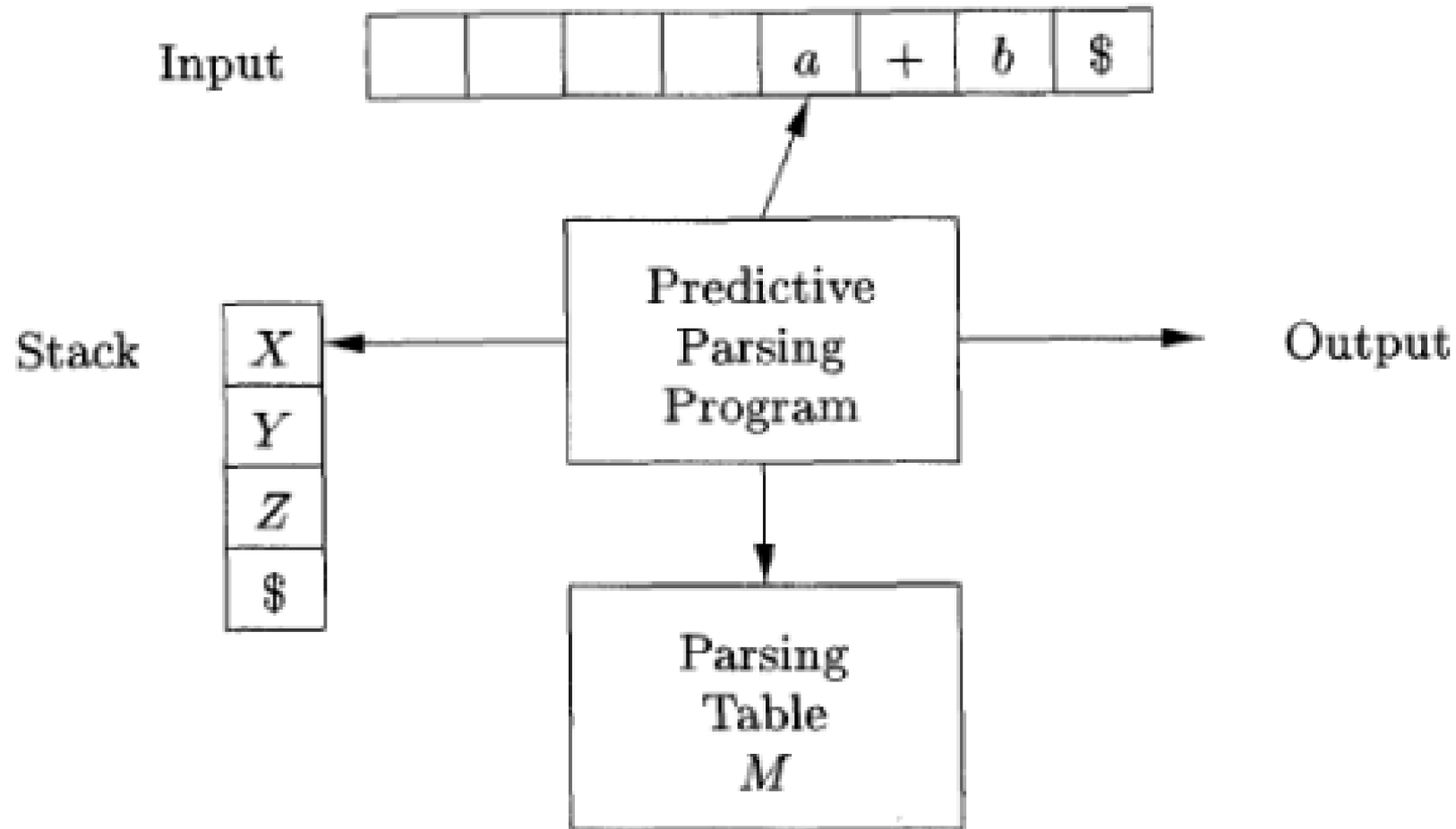
1. Giới thiệu về thuật toán tất định

Kiến trúc chung

- ☐ Việc lựa chọn ngay lập tức phương án suy dẫn dẫn tới yêu cầu cần nghiên cứu trước bộ luật văn phạm và có các phương án phù hợp trong các tình huống có thể xảy ra.
- ☐ Các thuật toán phân tích tất định đều sử dụng kĩ thuật xây dựng trước **bảng phương án**.
- ☐ Có nhiều kĩ thuật xây dựng bảng phương án khác nhau ứng với các phương pháp tiếp cận khác nhau.
- ☐ Với các loại bảng phương án, thuật toán phân tích cũng có sự khác biệt khi thực hiện đoán nhận.

1. Giới thiệu về thuật toán tất định

Kiến trúc chung



2. Tiếp cận Top-down

- ❑ Hãy quan sát quá trình thực hiện phân tích top-down chuỗi $w = () ()$ của văn phạm: $S \rightarrow (S) S \mid \epsilon$
 - Cần tìm quá trình suy dẫn $S \Rightarrow^* w = () ()$
 - Ở đây chúng ta chỉ có 1 non-terminal duy nhất S
 - Có 2 terminal "(" và ")"
- ❑ Bước suy dẫn đầu tiên, $S \Rightarrow (S) S \Rightarrow^* () ()$
- ❑ Vậy ở bước 2, cần tìm quá trình $S) S \Rightarrow^*) ()$
- ❑ Rõ ràng trong tình huống này, ta không thể áp dụng luật sinh $S \rightarrow (S) S$ mà phải sử dụng $S \rightarrow \epsilon$

2. Tiếp cận Top-down

□ Quan sát quá trình suy dẫn từ $\alpha \Rightarrow^* w$, dễ thấy:

- **Nếu α bắt đầu bởi terminal**, thì terminal đó nhất thiết phải trùng với kí hiệu bắt đầu của w , trong tình huống này ta gạt bỏ kí hiệu này ở cả 2 chuỗi
- **Nếu α bắt đầu bởi non-terminal A** , thì A nhất thiết phải suy dẫn (trực tiếp hoặc gián tiếp) ra kí hiệu bắt đầu của w (w_1) hoặc ra ϵ .
- **Ta có thể dựa trên văn phạm G để tính được A có suy ra w_1 được hay không?**
- Lập một bảng phương án 2 chiều, 1 chiều gồm các nonterminal, 1 chiều gồm các terminal, ta đưa ra các tình huống áp dụng luật sinh cho mỗi cặp (A, w_1) .

3. Thuật toán LL(1)

Bước	Chuỗi nguồn	Chuỗi đích	Hành động
1	S\$	()()\$	$S \rightarrow (S)S$
2	(S)S\$	()()\$	gạt bỏ
3	S)S\$)(()\$	$S \rightarrow \epsilon$
4)S\$)(()\$	gạt bỏ
5	S\$	()\$	$S \rightarrow (S)S$
6	(S)S\$	()\$	gạt bỏ
7	S)S\$)\$	$S \rightarrow \epsilon$
8)S\$)\$	gạt bỏ
9	S\$	\$	$S \rightarrow \epsilon$

$M[N, T]$	()	\$
S	$S \rightarrow (S)S$	$S \rightarrow \epsilon$	$S \rightarrow \epsilon$

3. Thuật toán LL(1)

- ❑ Như vậy bộ phân tích LL(1) hoạt động tương tự như phân tích top-down, nhưng không có bước quay lui (vì không có sự lựa chọn thử-sai).
- ❑ Vấn đề lớn nhất: làm sao xây dựng được bảng phương án?
- ❑ LL(1) nghĩa là gì? Viết tắt của “Left-to-right parse, Leftmost-derivation, 1-symbol lookahead”.
- ❑ Kí hiệu k trong LL(k) nghĩa là bộ phân tích sẽ nhìn trước k kí hiệu khi ra quyết định.

3. Thuật toán LL(1): FIRST(X)

- Nếu X là kí hiệu kết thúc thì $FIRST(X)$ là $\{X\}$
- Nếu $X \rightarrow \varepsilon$ là một luật sinh thì thêm ε vào $FIRST(X)$
- Nếu $X \rightarrow Y_1 Y_2 Y_3 \dots Y_k$ là một luật sinh thì:
 - Thêm tất cả các **ký hiệu kết thúc** khác ε của $FIRST(Y_1)$ vào $FIRST(X)$
 - Nếu $\varepsilon \in FIRST(Y_1)$ thì tiếp tục thêm vào $FIRST(X)$ tất cả các ký hiệu kết thúc khác ε của $FIRST(Y_2)$
 - Nếu $\varepsilon \in FIRST(Y_1) \cap FIRST(Y_2)$ thì thêm tất cả các ký hiệu kết thúc khác $\varepsilon \in FIRST(Y_3)$
 - Tiếp tục như vậy cho tới Y_k
 - Thêm ε vào $FIRST(X)$ nếu $\varepsilon \in \bigcap_{i=1 \rightarrow k} FIRST(Y_i)$

3. Thuật toán LL(1): $FIRST(\alpha)$

- Định nghĩa $FIRST(\alpha)$: giả sử α là một chuỗi các ký hiệu văn phạm, $FIRST(\alpha)$ là tập hợp các ký hiệu kết thúc mà nó bắt đầu một chuỗi dẫn xuất từ α :
- Giả sử $\alpha = X_1X_2 \dots X_n$
 - Thêm vào $FIRST(\alpha)$: $FIRST(X_1) - \{\varepsilon\}$
 - Với mọi $i = 2, 3, \dots, n$; nếu $FIRST(X_k)$ chứa ε với mọi $k = 1, 2, \dots, i - 1$ thì thêm vào $FIRST(\alpha)$: $FIRST(X_i) - \{\varepsilon\}$
 - Nếu với mọi $i = 1, 2, \dots, n$; nếu $FIRST(X_i)$ chứa ε thì thêm ε vào $FIRST(\alpha)$

3. Thuật toán LL(1): Ví dụ

□ Xét văn phạm G:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid \text{id}$$

Theo định nghĩa tập FIRST, ta có:

$$\text{Vì } F \Rightarrow (E) \mid \text{id} \Rightarrow \text{FIRST}(F) = \{ (, \text{id} \}$$

$$\text{Từ } T \rightarrow F T' \text{ và } \varepsilon \notin \text{FIRST}(F) \Rightarrow \text{FIRST}(T) = \text{FIRST}(F)$$

$$\text{Từ } E \rightarrow T E' \text{ và } \varepsilon \notin \text{FIRST}(T) \Rightarrow \text{FIRST}(E) = \text{FIRST}(T)$$

$$\text{Vì } E' \rightarrow \varepsilon \Rightarrow \varepsilon \in \text{FIRST}(E')$$

3. Thuật toán LL(1): Ví dụ

□ Xét văn phạm G:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid id$$

Mặt khác do $E' \rightarrow + T E'$ mà $FIRST(+)=\{+\} \Rightarrow$

$$FIRST(E') = \{+, \varepsilon\}$$

Tương tự $FIRST(T') = \{*, \varepsilon\}$

Vậy ta có :

$$FIRST(E) = FIRST(T) = FIRST(F) = \{ (, id \}$$

$$FIRST(E') = \{+, \varepsilon\}$$

$$FIRST(T') = \{*, \varepsilon\}$$

Theo định nghĩa tập FIRST, ta có:

$$\text{Vì } F \Rightarrow (E) \mid id \Rightarrow FIRST(F) = \{ (, id \}$$

$$\text{Từ } T \rightarrow F T' \text{ và } \varepsilon \notin FIRST(F) \Rightarrow FIRST(T) = FIRST(F)$$

$$\text{Từ } E \rightarrow T E' \text{ và } \varepsilon \notin FIRST(T) \Rightarrow FIRST(E) = FIRST(T)$$

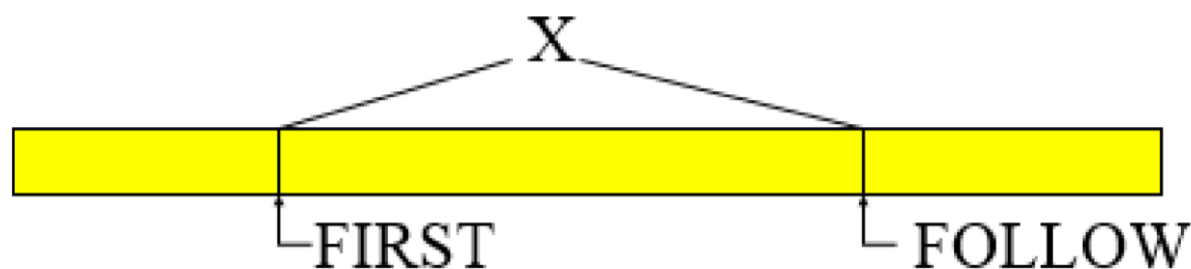
$$\text{Vì } E' \rightarrow \varepsilon \Rightarrow \varepsilon \in FIRST(E')$$

3. Thuật toán LL(1): FOLLOW

- Định nghĩa FOLLOW(A): tập hợp các ký hiệu kết thúc nó xuất hiện ngay sau A (bên phải của A) trong một dạng câu nào đó.
 - Tức là tập hợp các ký hiệu kết thúc a , sao cho tồn tại một dẫn xuất dạng $S \Rightarrow^* \alpha A a \beta$
 - Chú ý rằng nếu A là ký hiệu phải nhất trong một dạng câu nào đó thì $\$ \in FOLLOW(A)$ ($\$$ là ký hiệu kết thúc chuỗi nhập)

3. Thuật toán LL(1): FOLLOW

- Định nghĩa FOLLOW(A): tập hợp các ký hiệu kết thúc a mà nó xuất hiện ngay sau A (bên phải của A) trong một dạng câu nào đó
 - Tức là tập hợp các ký hiệu kết thúc a, sao cho tồn tại một dẫn xuất dạng $S \Rightarrow^* \alpha A a \beta$
 - Chú ý rằng nếu A là ký hiệu phải nhất trong một dạng câu nào đó thì $\$ \in FOLLOW(A)$ (\$) là ký hiệu kết thúc chuỗi nhập)



3. Thuật toán LL(1): FOLLOW

- ❑ Tính FOLLOW (A): áp dụng các quy tắc sau cho đến khi không thể thêm gì vào mọi tập FOLLOW được nữa:
 1. Đặt \$ vào follow(S), trong đó S là ký hiệu bắt đầu của văn phạm và \$ là ký hiệu kết thúc chuỗi nhập.
 2. Nếu có một luật sinh $A \rightarrow \alpha B \beta$ thì thêm mọi phần tử khác ϵ của $FIRST(\beta)$ vào trong $FOLLOW(B)$.
 3. Nếu có luật sinh $A \rightarrow \alpha B$ hoặc $A \rightarrow \alpha B \beta$ mà $\epsilon \in FIRST(\beta)$ thì thêm tất cả các phần tử trong $FOLLOW(A)$ vào $FOLLOW(B)$.

3. Thuật toán LL(1)

Ví dụ tính FOLLOW

□ Xét văn phạm G:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid id$$

3. Thuật toán LL(1)

Ví dụ tính FOLLOW

□ Xét văn phạm G:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid id$$

- Áp dụng luật 2 cho luật sinh $F \rightarrow (E) \Rightarrow) \in FOLLOW(E) \Rightarrow FOLLOW(E) = \{ \$,) \}$
- Áp dụng luật 3 cho $E \rightarrow T E' \Rightarrow), \$ \in FOLLOW(E') \Rightarrow FOLLOW(E') = \{ \$,) \}$
- Áp dụng luật 2 cho $E \rightarrow T E' \Rightarrow + \in FOLLOW(T)$.
- Áp dụng luật 3 cho $E' \rightarrow + T E', E' \rightarrow \varepsilon$
- $\Rightarrow FOLLOW(E') \subset FOLLOW(T) \Rightarrow FOLLOW(T) = \{ +,), \$ \}$.
- Áp dụng luật 3 cho $T \rightarrow F T'$ thì $FOLLOW(T') = FOLLOW(T) = \{ +,), \$ \}$
- Áp dụng luật 2 cho $T \rightarrow F T' \Rightarrow * \in FOLLOW(F)$
- Áp dụng luật 3 cho $T' \rightarrow * F T', T' \rightarrow \varepsilon$
- $\Rightarrow FOLLOW(T') \subset FOLLOW(F) \Rightarrow FOLLOW(F) = \{ *, +,), \$ \}$.
- Vậy ta có: $FOLLOW(E) = FOLLOW(E') = \{ \$,) \}$
- $FOLLOW(T) = FOLLOW(T') = \{ +,), \$ \}$
- $FOLLOW(F) = \{ *, +,), \$ \}$

3. Thuật toán LL(1): Xây dựng bảng PTCP

Bảng phân tích cú pháp M là một mảng hai chiều dạng $M[A,a]$, trong đó A là ký hiệu chưa kết thúc, a là ký hiệu kết thúc hoặc \$.

Thuật toán xây dựng bảng PTCP:

1. Với mỗi luật sinh $A \rightarrow \alpha$ của văn phạm, thực hiện:

- Với mỗi ký hiệu kết thúc $a \in \text{FIRST}(\alpha)$, thêm $A \rightarrow \alpha$ vào $M[A,a]$
- Nếu $\epsilon \in \text{FIRST}(\alpha)$ thì đưa luật sinh $A \rightarrow \alpha$ vào $M[A,b]$ với mỗi ký hiệu kết thúc $b \in \text{FOLLOW}(A)$
- Nếu $\epsilon \in \text{FIRST}(\alpha)$ và $\$ \in \text{FOLLOW}(A)$ thì đưa luật sinh $A \rightarrow \alpha$ vào $M[A,\$]$

2. Các ô trống trong bảng tương ứng với lỗi (error)

Chú ý: một ô trong bảng có thể chứa nhiều suy dẫn, tình huống này gọi là bảng có nhập nhằng.

3. Thuật toán LL(1): Xây dựng bảng PTCP

□ Xét văn phạm G:

$E \rightarrow T E'$

$E' \rightarrow + T E' \mid \varepsilon$

$T \rightarrow F T'$

$T' \rightarrow * F T' \mid \varepsilon$

$F \rightarrow (E) \mid id$

Luật sinh $E \rightarrow TE'$: Tính $FIRST(TE') = FIRST(T) = \{ (, id \}$

$\Rightarrow M[E, id]$ và $M[E, (]$ chứa luật sinh $E \rightarrow TE'$

Luật sinh $E' \rightarrow + TE'$: Tính $FIRST(+TE') = FIRST(+) = \{ + \}$

$\Rightarrow M[E', +]$ chứa $E' \rightarrow +TE'$

Luật sinh $E' \rightarrow \varepsilon$: Vì $\varepsilon \in FIRST(E')$ và $FOLLOW(E') = \{), \$ \}$

$\Rightarrow E \rightarrow \varepsilon$ nằm trong $M[E',)]$ và $M[E', \$]$

Luật sinh $T' \rightarrow * FT'$: $FIRST(* FT') = \{ * \}$

$\Rightarrow T' \rightarrow * FT'$ nằm trong $M[T', *]$

Luật sinh $T' \rightarrow \varepsilon$: Vì $\varepsilon \in FIRST(T')$ và $FOLLOW(T') = \{ +,), \$ \}$

$\Rightarrow T' \rightarrow \varepsilon$ nằm trong $M[T', +]$, $M[T',)]$ và $M[T', \$]$

Luật sinh $F \rightarrow (E)$; $FIRST((E)) = \{ (\}$

$\Rightarrow F \rightarrow (E)$ nằm trong $M[F, (]$

Luật sinh $F \rightarrow id$; $FIRST(id) = \{ id \}$

$\Rightarrow F \rightarrow id$ nằm trong $M[F, id]$

3. Thuật toán LL(1): Xây dựng bảng PTCP

□ Bảng phân tích cú pháp cho văn phạm:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

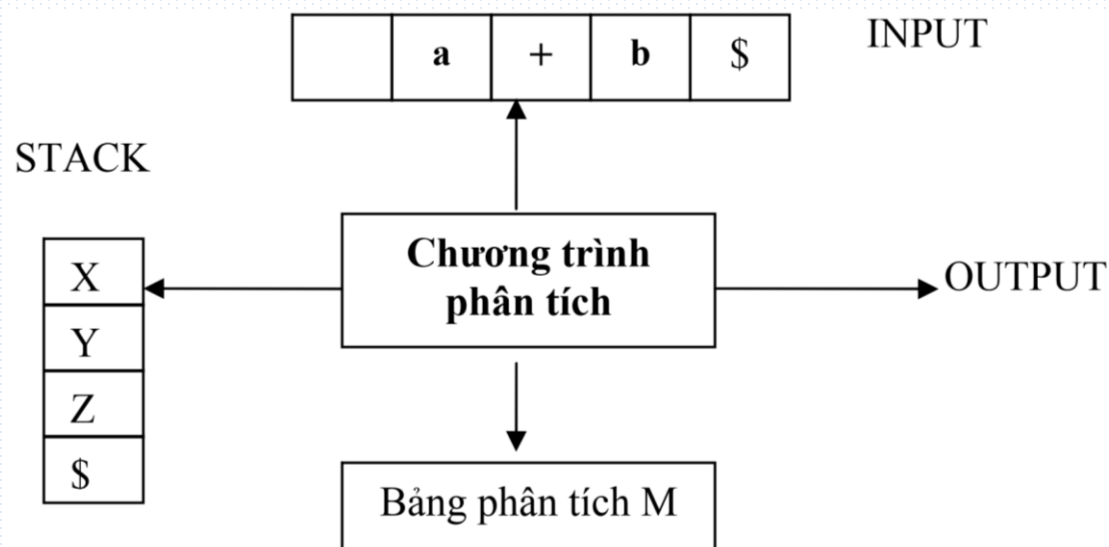
$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid id$$

Ký hiệu chưa kết thúc	Ký hiệu nhập					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$			$E \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

3. Thuật toán LL(1): Xây dựng cây PTCP

- ❑ **INPUT** là bộ đệm chứa chuỗi cần phân tích, kết thúc bởi ký hiệu \$.
- ❑ **STACK** chứa một chuỗi các ký hiệu văn phạm với ký hiệu \$ nằm ở đáy Stack.
- ❑ **Bảng phân tích M**



3. Thuật toán LL(1): Xây dựng cây PTCP

- ❑ Xét ký hiệu X trên đỉnh Stack và ký hiệu nhập hiện hành a .
- ❑ Bộ phân tích cú pháp hoạt động như sau:
 1. Nếu $X = a = \$$ thì chương trình phân tích cú pháp kết thúc thành công.
 2. Nếu $X = a \neq \$$, Pop X ra khỏi Stack và đọc ký hiệu nhập tiếp theo.
 3. Nếu X là ký hiệu chưa kết thúc thì chương trình truy xuất đến phần tử $M[X,a]$ trong bảng phân tích M :
 - Nếu $M[X,a]$ là một luật sinh có dạng $X \rightarrow UVW$ thì Pop X ra khỏi đỉnh Stack và Push W, V, U vào Stack (với U trên đỉnh Stack), đồng thời bộ xuất sinh ra luật sinh $X \rightarrow UVW$.
 - Nếu $M[X,a] = \text{error}$, gọi chương trình phục hồi lỗi.

3. Thuật toán LL(1): Xây dựng cây PTCP

□ Bảng phân tích cú pháp cho văn phạm:

$$E \rightarrow T E'$$

$$E' \rightarrow + T E' \mid \varepsilon$$

$$T \rightarrow F T'$$

$$T' \rightarrow * F T' \mid \varepsilon$$

$$F \rightarrow (E) \mid \text{id}$$

Ký hiệu chưa kết thúc	Ký hiệu nhập					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$			$E \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \varepsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

Ví dụ: Phân tích cú pháp cho chuỗi nhập: id + id * id ?

3. Thuật toán LL(1): Xây dựng cây PTCP

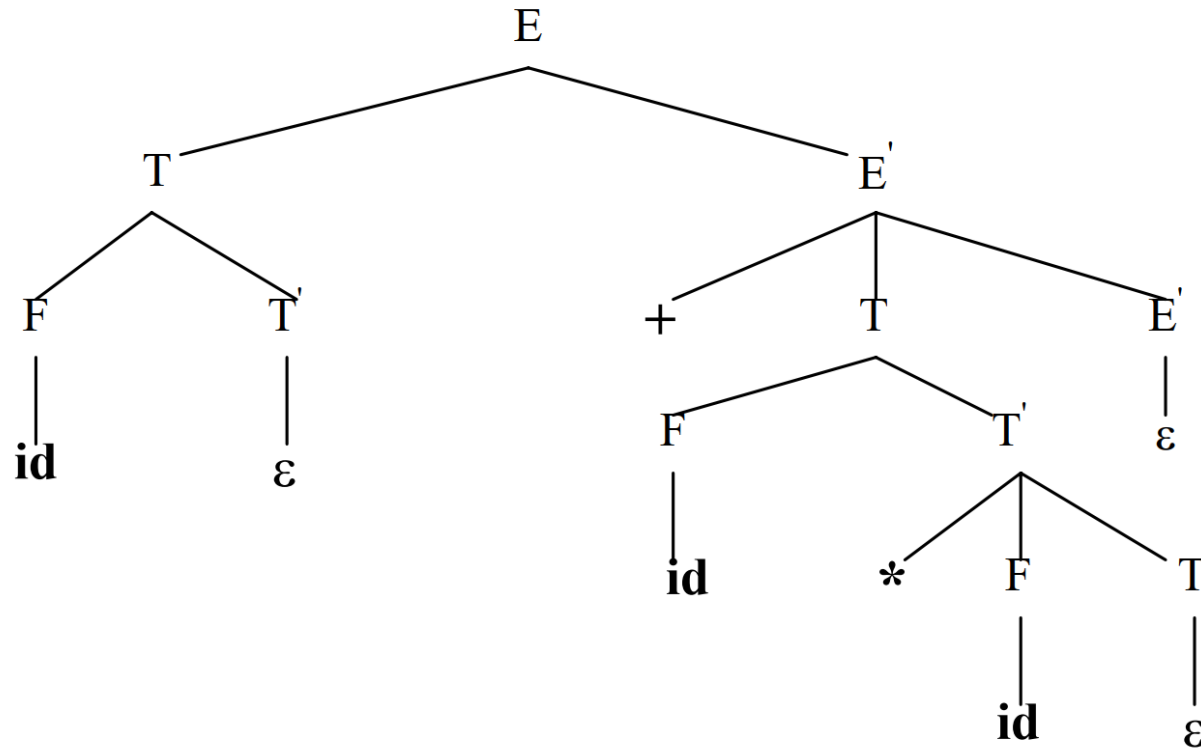
Ví dụ: Phân tích cú pháp cho chuỗi nhập: id + id * id ?

Ký hiệu chưa kết thúc	Ký hiệu nhập					
	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E \rightarrow +TE'$			$E \rightarrow \epsilon$	$E' \rightarrow \epsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \epsilon$	$T' \rightarrow *FT'$		$T' \rightarrow \epsilon$	$T' \rightarrow \epsilon$
F	$F \rightarrow id$			$F \rightarrow (E)$		

STACK	INPUT	OUTPUT
\$ E	id + id * id \$	
\$ E' T	id + id * id \$	$E \rightarrow TE'$
\$ E' T' F	id + id * id \$	$T \rightarrow FT'$
\$ E' T' id	id + id * id \$	$F \rightarrow id$
\$ E' T'	+ id * id \$	
\$ E'	+ id * id \$	$T' \rightarrow \epsilon$
\$ E' T +	+ id * id \$	$E' \rightarrow +TE'$
\$ E' T	id * id \$	
\$ E' T' F	id * id \$	$T \rightarrow FT'$
\$ E' T' id	id * id \$	$F \rightarrow id$
\$ E' T'	* id \$	
\$ E' T' F *	* id \$	$T' \rightarrow *FT'$
\$ E' T' F	id \$	$F \rightarrow id$
\$ E' T' id	id \$	
\$ E' T'	\$	$T' \rightarrow \epsilon$
\$ E'	\$	$E' \rightarrow \epsilon$
\$	\$	

3. Thuật toán LL(1): Xây dựng cây PTCP

Ví dụ: Phân tích cú pháp cho chuỗi nhập: id + id * id ?



STACK	INPUT	OUTPUT
\$ E	id + id * id \$	
\$ E' T	id + id * id \$	$E \rightarrow T E'$
\$ E' T' F	id + id * id \$	$T \rightarrow F T'$
\$ E' T' id	id + id * id \$	$F \rightarrow id$
\$ E' T'	+ id * id \$	
\$ E'	+ id * id \$	$T' \rightarrow \epsilon$
\$ E' T +	+ id * id \$	$E' \rightarrow + T E'$
\$ E' T	id * id \$	
\$ E' T' F	id * id \$	$T \rightarrow F T'$
\$ E' T' id	id * id \$	$F \rightarrow id$
\$ E' T'	* id \$	
\$ E' T' F *	* id \$	$T' \rightarrow * F T'$
\$ E' T' F	id \$	$F \rightarrow id$
\$ E' T' id	id \$	
\$ E' T'	\$	$T' \rightarrow \epsilon$
\$ E'	\$	$E' \rightarrow \epsilon$
\$	\$	

3. Thuật toán LL(1): Văn phạm LL(1)

- ☐ Giải thuật LL(1) có thể áp dụng cho bất kỳ văn phạm G nào để sinh ra bảng phân tích M .
- ☐ Tuy nhiên, có những văn phạm (đệ quy trái hoặc mơ hồ) thì trong bảng phân tích M sẽ có thể có những ô đa trị (có chứa nhiều hơn 1 luật sinh).

3. Thuật toán LL(1): Văn phạm LL(1)

- Ví dụ: Xét văn phạm sau:
- $$S \rightarrow iEtSS' \mid a$$
- $$S' \rightarrow eS \mid \varepsilon$$
- $$E \rightarrow b$$

Ký hiệu chưa kết thúc	Ký hiệu kết thúc					
	a	b	e	i	T	\$
S	$S \rightarrow a$			$S \rightarrow iEtSS'$		
S'			$S \rightarrow \varepsilon$ $S' \rightarrow eS$			$S' \rightarrow \varepsilon$
E		$E \rightarrow b$				

3. Thuật toán LL(1): Văn phạm LL(1)

- ❑ Một văn phạm mà bảng phân tích M không có các ô đa trị được gọi là văn phạm LL(1) với ý nghĩa như sau :
- ❑ L: Left-to-right parse (mô tả hành động quét chuỗi nhập từ trái sang phải)
- ❑ L: Leftmost-derivation (biểu thị việc sinh ra một dẫn xuất trái cho chuỗi nhập)
- ❑ 1: 1-symbol lookahead (tại mỗi một bước, đầu đọc chỉ đọc trước được một token để thực hiện các quyết định phân tích cú pháp)

4. Bài tập

1. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

$$S \rightarrow Ac \mid BBc$$

$$C \rightarrow b \mid bCd$$

$$D \rightarrow bd \mid bDd$$

$$A \rightarrow BC$$

$$B \rightarrow dBb \mid dDb \mid \varepsilon$$

2. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

$$S \rightarrow AD \mid abc$$

$$B \rightarrow dBc \mid CC$$

$$C \rightarrow DCb \mid CDb \mid \varepsilon$$

$$A \rightarrow Bc$$

$$D \rightarrow Dd \mid \varepsilon$$

4. Bài tập

3. Tạo bảng phân tích LL(1) và chỉ ra quá trình phân tích chuỗi $w = (a+a)$ với văn phạm G sau đây:

$$S \rightarrow F \quad S \rightarrow (S+F) \quad F \rightarrow a$$

4. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

$$S \rightarrow A \quad A \rightarrow T \mid A+T \quad T \rightarrow b \mid (A)$$

5. Tính First, Follow và tạo bảng phân tích LL(1) cho văn phạm sau:

$$E \rightarrow E \text{ or } T \mid T$$

$$T \rightarrow T \text{ and } F \mid F$$

$$F \rightarrow \text{not } F \mid (E) \mid x$$