

CHƯƠNG TRÌNH DỊCH

Chương 4. Phân tích cú pháp Thuật toán LR

TS. Phạm Văn Cảnh
Khoa Công nghệ thông tin

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

-
- 1. Bộ phân tích kiểu gạt-thu (shift-reduce)**
 - 2. Máy phân tích cú pháp LR**
 - 3. Xây dựng bảng PTCP**
 - 4. Đánh giá về LR**
 - 5. Bài tập**

1. Bộ phân tích kiểu gặt-thu

- ❑ Cách làm việc xuất phát từ việc quan sát hoạt động của phân tích bottom-up
- ❑ Bắt đầu từ nút lá phải nhất
- ❑ Thu gọn dần về nút gốc
- ❑ Chỉ 2 kiểu hoạt động chính:
 - Gặt (shift): lấy kí hiệu tiếp theo.
 - Thu (reduce): thu gọn nhánh thành một kí hiệu trung gian.

1. Bộ phân tích kiểu gọt-thu

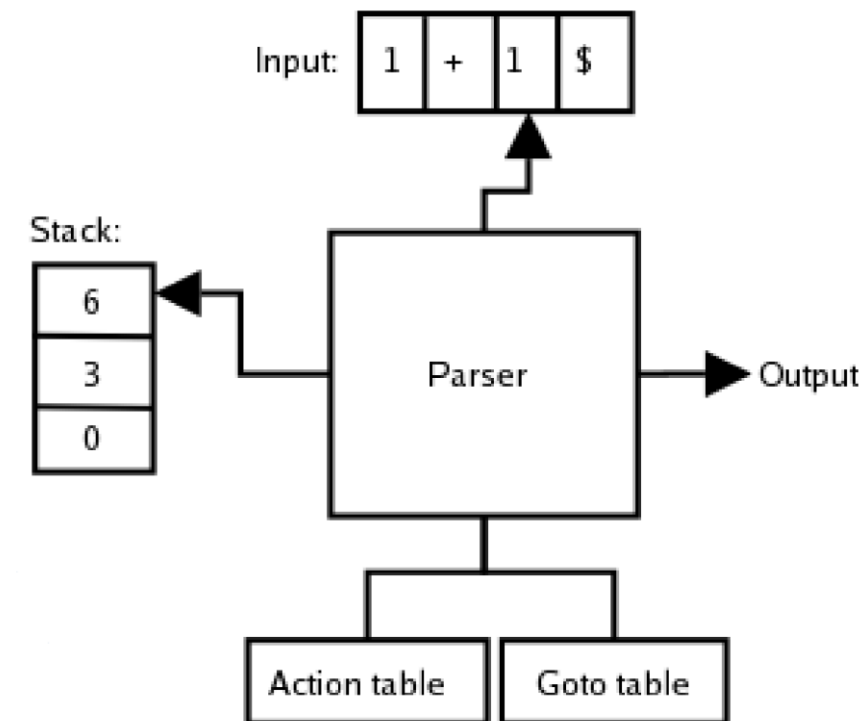
❑ Là một dạng automat làm việc theo bảng phương án (đã được đề cập tới trong bài trước)

❑ Vấn đề: xây dựng bảng phương án như thế nào

- Khi nào thì shift
- Khi nào thì reduce
- Còn hoạt động nào khác?
- Có trạng thái bị tranh chấp?

❑ Hoạt động của stack ra sao?

❑ Ý nghĩa các trạng thái của máy



$$int + (int + int + int)$$

```

graph TD
    S[S] --> E1[E]
    S --> T1[T]
    E1 --> E2[E]
    E1 --> T2[T]
    E1 --> T3[T]
    E2 --> E3[E]
    E2 --> T4[T]
    E2 --> T5[T]
    E3 --> E4[E]
    E3 --> T6[T]
    E3 --> T7[T]
    E4 --> E5[E]
    E4 --> T8[T]
    E4 --> T9[T]
    E5 --> E6[E]
    E5 --> T10[T]
    E5 --> T11[T]
    E6 --> E7[E]
    E6 --> T12[T]
    E6 --> T13[T]
    E7 --> E8[E]
    E7 --> T14[T]
    E7 --> T15[T]
    E8 --> E9[E]
    E8 --> T16[T]
    E8 --> T17[T]
    E9 --> E10[E]
    E9 --> T18[T]
    E9 --> T19[T]
    E10 --> E11[E]
    E10 --> T20[T]
    E10 --> T21[T]
    E11 --> E12[E]
    E11 --> T22[T]
    E11 --> T23[T]
    E12 --> E13[E]
    E12 --> T24[T]
    E12 --> T25[T]
    E13 --> E14[E]
    E13 --> T26[T]
    E13 --> T27[T]
    E14 --> E15[E]
    E14 --> T28[T]
    E14 --> T29[T]
    E15 --> E16[E]
    E15 --> T30[T]
    E15 --> T31[T]
    E16 --> E17[E]
    E16 --> T32[T]
    E16 --> T33[T]
    E17 --> E18[E]
    E17 --> T34[T]
    E17 --> T35[T]
    E18 --> E19[E]
    E18 --> T36[T]
    E18 --> T37[T]
    E19 --> E20[E]
    E19 --> T38[T]
    E19 --> T39[T]
    E20 --> E21[E]
    E20 --> T40[T]
    E20 --> T41[T]
    E21 --> E22[E]
    E21 --> T42[T]
    E21 --> T43[T]
    E22 --> E23[E]
    E22 --> T44[T]
    E22 --> T45[T]
    E23 --> E24[E]
    E23 --> T46[T]
    E23 --> T47[T]
    E24 --> E25[E]
    E24 --> T48[T]
    E24 --> T49[T]
    E25 --> E26[E]
    E25 --> T50[T]
    E25 --> T51[T]
    E26 --> E27[E]
    E26 --> T52[T]
    E26 --> T53[T]
    E27 --> E28[E]
    E27 --> T54[T]
    E27 --> T55[T]
    E28 --> E29[E]
    E28 --> T56[T]
    E28 --> T57[T]
    E29 --> E30[E]
    E29 --> T58[T]
    E29 --> T59[T]
    E30 --> E31[E]
    E30 --> T60[T]
    E30 --> T61[T]
    E31 --> E32[E]
    E31 --> T62[T]
    E31 --> T63[T]
    E32 --> E33[E]
    E32 --> T64[T]
    E32 --> T65[T]
    E33 --> E34[E]
    E33 --> T66[T]
    E33 --> T67[T]
    E34 --> E35[E]
    E34 --> T68[T]
    E34 --> T69[T]
    E35 --> E36[E]
    E35 --> T70[T]
    E35 --> T71[T]
    E36 --> E37[E]
    E36 --> T72[T]
    E36 --> T73[T]
    E37 --> E38[E]
    E37 --> T74[T]
    E37 --> T75[T]
    E38 --> E39[E]
    E38 --> T76[T]
    E38 --> T77[T]
    E39 --> E40[E]
    E39 --> T78[T]
    E39 --> T79[T]
    E40 --> E41[E]
    E40 --> T80[T]
    E40 --> T81[T]
    E41 --> E42[E]
    E41 --> T82[T]
    E41 --> T83[T]
    E42 --> E43[E]
    E42 --> T84[T]
    E42 --> T85[T]
    E43 --> E44[E]
    E43 --> T86[T]
    E43 --> T87[T]
    E44 --> E45[E]
    E44 --> T88[T]
    E44 --> T89[T]
    E45 --> E46[E]
    E45 --> T90[T]
    E45 --> T91[T]
    E46 --> E47[E]
    E46 --> T92[T]
    E46 --> T93[T]
    E47 --> E48[E]
    E47 --> T94[T]
    E47 --> T95[T]
    E48 --> E49[E]
    E48 --> T96[T]
    E48 --> T97[T]
    E49 --> E50[E]
    E49 --> T98[T]
    E49 --> T99[T]
    E50 --> E51[E]
    E50 --> T100[T]
    E50 --> T101[T]
    E51 --> E52[E]
    E51 --> T102[T]
    E51 --> T103[T]
    E52 --> E53[E]
    E52 --> T104[T]
    E52 --> T105[T]
    E53 --> E54[E]
    E53 --> T106[T]
    E53 --> T107[T]
    E54 --> E55[E]
    E54 --> T108[T]
    E54 --> T109[T]
    E55 --> E56[E]
    E55 --> T110[T]
    E55 --> T111[T]
    E56 --> E57[E]
    E56 --> T112[T]
    E56 --> T113[T]
    E57 --> E58[E]
    E57 --> T114[T]
    E57 --> T115[T]
    E58 --> E59[E]
    E58 --> T116[T]
    E58 --> T117[T]
    E59 --> E60[E]
    E59 --> T118[T]
    E59 --> T119[T]
    E60 --> E61[E]
    E60 --> T120[T]
    E60 --> T121[T]
    E61 --> E62[E]
    E61 --> T122[T]
    E61 --> T123[T]
    E62 --> E63[E]
    E62 --> T124[T]
    E62 --> T125[T]
    E63 --> E64[E]
    E63 --> T126[T]
    E63 --> T127[T]
    E64 --> E65[E]
    E64 --> T128[T]
    E64 --> T129[T]
    E65 --> E66[E]
    E65 --> T130[T]
    E65 --> T131[T]
    E66 --> E67[E]
    E66 --> T132[T]
    E66 --> T133[T]
    E67 --> E68[E]
    E67 --> T134[T]
    E67 --> T135[T]
    E68 --> E69[E]
    E68 --> T136[T]
    E68 --> T137[T]
    E69 --> E70[E]
    E69 --> T138[T]
    E69 --> T139[T]
    E70 --> E71[E]
    E70 --> T140[T]
    E70 --> T141[T]
    E71 --> E72[E]
    E71 --> T142[T]
    E71 --> T143[T]
    E72 --> E73[E]
    E72 --> T144[T]
    E72 --> T145[T]
    E73 --> E74[E]
    E73 --> T146[T]
    E73 --> T147[T]
    E74 --> E75[E]
    E74 --> T148[T]
    E74 --> T149[T]
    E75 --> E76[E]
    E75 --> T150[T]
    E75 --> T151[T]
    E76 --> E77[E]
    E76 --> T152[T]
    E76 --> T153[T]
    E77 --> E78[E]
    E77 --> T154[T]
    E77 --> T155[T]
    E78 --> E79[E]
    E78 --> T156[T]
    E78 --> T157[T]
    E79 --> E80[E]
    E79 --> T158[T]
    E79 --> T159[T]
    E80 --> E81[E]
    E80 --> T160[T]
    E80 --> T161[T]
    E81 --> E82[E]
    E81 --> T162[T]
    E81 --> T163[T]
    E82 --> E83[E]
    E82 --> T164[T]
    E82 --> T165[T]
    E83 --> E84[E]
    E83 --> T166[T]
    E83 --> T167[T]
    E84 --> E85[E]
    E84 --> T168[T]
    E84 --> T169[T]
    E85 --> E86[E]
    E85 --> T170[T]
    E85 --> T171[T]
    E86 --> E87[E]
    E86 --> T172[T]
    E86 --> T173[T]
    E87 --> E88[E]
    E87 --> T174[T]
    E87 --> T175[T]
    E88 --> E89[E]
    E88 --> T176[T]
    E88 --> T177[T]
    E89 --> E90[E]
    E89 --> T178[T]
    E89 --> T179[T]
    E90 --> E91[E]
    E90 --> T180[T]
    E90 --> T181[T]
    E91 --> E92[E]
    E91 --> T182[T]
    E91 --> T183[T]
    E92 --> E93[E]
    E92
```

1. Bộ phân tích kiểu gọt-thu

Grammar:

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow T \\ E &\rightarrow E + T \\ T &\rightarrow \text{int} \\ T &\rightarrow (E) \end{aligned}$$

$$\begin{aligned} &\text{int} + (\text{int} + \text{int} + \text{int})\$ \\ &T + (\text{int} + \text{int} + \text{int})\$ \\ &E + (\text{int} + \text{int} + \text{int})\$ \\ &E + (T + \text{int} + \text{int})\$ \\ &E + (E + \text{int} + \text{int})\$ \\ &E + (E + T + \text{int})\$ \\ &E + (E + \text{int})\$ \\ &E + (E + T)\$ \\ &E + (E)\$ \\ &E + T\$ \\ &E\$ \\ &S \end{aligned}$$

1. Bộ phân tích kiểu gọt-thu

Grammar:

$$E \rightarrow E + T \mid T$$

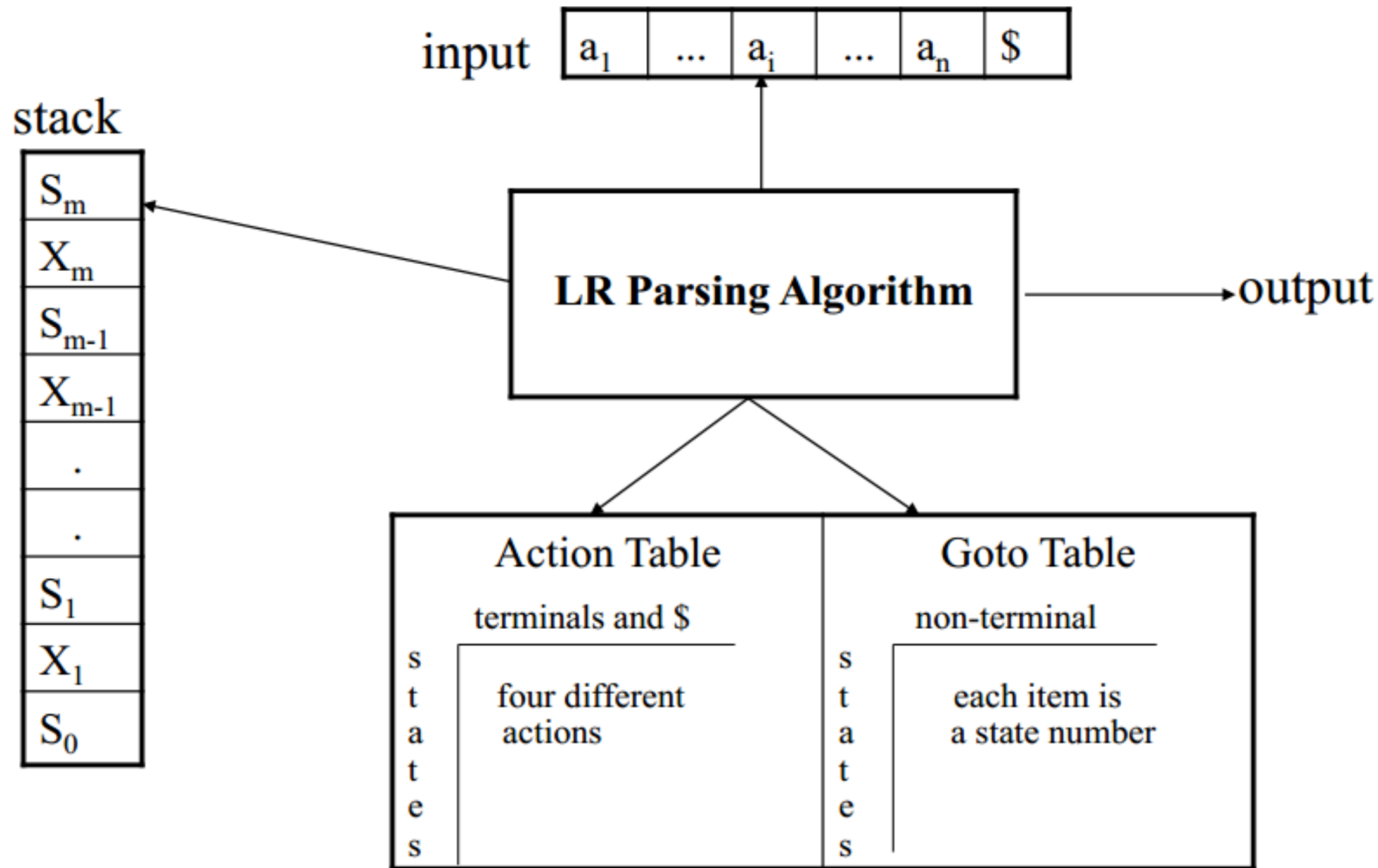
$$T \rightarrow T * F \mid F$$

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

Bottom-up parsing of
 $id + id * id$

Left substring	Right substring	Action
\$	$id + id * id\$$	Shift
$\$id$	$+id * id\$$	Reduce by $F \rightarrow \text{id}$
$\$F$	$+id * id\$$	Reduce by $T \rightarrow F$
$\$T$	$+id * id\$$	Reduce by $E \rightarrow T$
$\$E$	$+id * id\$$	Shift
$\$E+$	$id * id\$$	Shift
$\$E + id$	$*id\$$	Reduce by $F \rightarrow \text{id}$
$\$E + F$	$*id\$$	Reduce by $T \rightarrow F$
$\$E + T$	$*id\$$	Shift
$\$E + T*$	$id\$$	Shift
$\$E + T * id$	\$	Reduce by $F \rightarrow \text{id}$
$\$E + T * F$	\$	Reduce by $T \rightarrow T * F$
$\$E + T$	\$	Reduce by $E \rightarrow E + T$
$\$E$	\$	Accept

2. Máy phân tích cú pháp LR



2. Máy phân tích cú pháp LR

□ Stack lưu một chuỗi $s_0X_1s_1 \dots X_ms_m$ trong đó

- s_m nằm trên đỉnh Stack.
- X_i là một ký hiệu văn phạm, s_i là một trạng thái tóm tắt thông tin chứa trong Stack bên dưới nó.

□ Máy phân tích LR là một cặp (STACK, INPUT)

- Trạng thái ban đầu ($s_0, a_1a_2 \dots a_n\$$)
- Trạng thái trung gian ($s_0X_1s_1 \dots X_ms_m, a_ia_{i+1} \dots a_n\$$)
- Trạng thái kết thúc thành công ($s_0Ss_1, \$$)

2. Máy phân tích cú pháp LR

□ Bảng phương án gồm 2 phần

○ Bảng action: **ACTION**[s, a]:

- Với s là một trạng thái và a là một kí hiệu kết thúc (terminal).
- Giá trị trong bảng chỉ có thể là 1 trong 4 hành động gạt (shift), thu (reduce), nhận (accept), lỗi (error).
 1. **shift s**: đẩy s, trong đó s là một trạng thái.
 2. **reduce $A \rightarrow \beta$** : thu gọn bằng luật sinh $A \rightarrow \beta$.
 3. **accept**: Chấp nhận
 4. **error**: Báo lỗi

○ Bảng goto: **GOTO**[s, A] với s là một trạng thái và A là một non-terminal, chỉ ra cách dịch chuyển trạng thái.

2. Máy phân tích cú pháp LR

- Cấu hình (configuration) của một bộ phân tích cú pháp LR là một cặp thành phần, trong đó:
- Thành phần đầu là nội dung của Stack, phần sau là chuỗi nhập chưa phân tích: $(s_0 X_1 s_1 X_2 s_2 \dots X_m s_m, a_i a_{i+1} \dots a_n \$)$
 - Với s_m là ký hiệu trên đỉnh Stack, a_i là ký hiệu nhập hiện tại, cấu hình có được sau mỗi dạng bước đẩy sẽ như sau:

2. Máy phân tích cú pháp LR

☐ Thuật toán LR

1. Nếu **action** $[s_m, a_i]$ = **Shift s**: Thực hiện phép đẩy để được cấu hình mới:

$$(s_0 X_1 s_1 X_2 s_2 \dots X_m s_m a_i s, a_{i+1} \dots a_n \$)$$

Phép đẩy làm cho s nằm trên đỉnh Stack, a_{i+1} trở thành ký hiệu hiện hành.

2. Nếu **action** $[s_m, a_i]$ = **Reduce**(**A** \rightarrow **β**) thì thực hiện phép thu gọn để được cấu hình: $(s_0 X_1 s_1 X_2 s_2 \dots X_{m-i} s_{m-i} A s, a_i a_{i+1} \dots a_n \$)$

Trong đó, $s = \text{goto}[s_{m-i}, A]$ và r là chiều dài số lượng các ký hiệu của β . Ở đây, trước hết 2r phần tử của Stack sẽ bị lấy ra, sau đó đẩy vào A và s.

3. Nếu **action** $[s_m, a_i]$ = **accept**: quá trình phân tích kết thúc.

4. Nếu **action** $[s_m, a_i]$ = **error**: gọi thủ tục phục hồi lỗi.

2. Máy phân tích cú pháp LR

Thuật toán LR

Khởi tạo s_0 là trạng thái khởi tạo nằm trong Stack và $w\$$ nằm trong bộ đệm nhập.

Đặt ip vào ký hiệu đầu tiên của $w\$$;

Repeat forever begin

Gọi s là trạng thái trên đỉnh Stack và a là ký hiệu được trở bởi ip;

If $\text{action}[s, a] = \text{Shift } s'$ **then begin**

Đẩy a và sau đó là s' vào Stack;

Chuyển ip tới ký hiệu kế tiếp;

end

else if $\text{action}[s, a] = \text{Reduce } (A \rightarrow \beta)$ **then begin**

Lấy $2 * |\beta|$ ký hiệu ra khỏi Stack;

Gọi s' là trạng thái trên đỉnh Stack;

Đẩy A , sau đó đẩy $\text{goto}[s', A]$ vào Stack;

Xuất ra luật sinh $A \rightarrow \beta$;

end

else if $\text{action}[s, a] = \text{accept}$ **then**

return tập luật

else error ()

end

2. Máy phân tích cú pháp LR: Ví dụ

□ Ví dụ:

$$1. E \rightarrow E + T$$

$$2. E \rightarrow T$$

$$3. T \rightarrow T * F$$

$$4. T \rightarrow F$$

$$5. F \rightarrow (E)$$

$$6. F \rightarrow \text{id}$$

Action Table							Goto Table			
state	id	+	*	()	\$		E	T	F
0	s5			s4				1	2	3
1		s6				acc				
2		r2	s7		r2	r2				
3		r4	r4		r4	r4				
4	s5			s4				8	2	3
5		r6	r6		r6	r6				
6	s5			s4					9	3
7	s5			s4						10
8		s6			s11					
9		r1	s7		r1	r1				
10		r3	r3		r3	r3				
11		r5	r5		r5	r5				

2. Máy phân tích cú pháp LR: Ví dụ

- Với chuỗi nhập id * id + id, các bước chuyển trạng thái trên Stack và nội dung bộ đệm nhập được trình bày như sau:

STACK		INPUT	ACTION
(1)	0	id * id + id \$	Shift
(2)	0 id 5	* id + id \$	Reduce by $F \rightarrow id$
(3)	0 F 3	* id + id \$	Reduce by $T \rightarrow F$
(4)	0 T 2	* id + id \$	Shift
(5)	0 T 2 * 7	id + id \$	Shift
(6)	0 T 2 * 7 id 5	+ id \$	Reduce by $F \rightarrow id$
(7)	0 T 2 * 7 F 10	+ id \$	Reduce by $T \rightarrow T * F$
(8)	0 T 2	+ id \$	Reduce by $E \rightarrow T$
(9)	0 E 1	+ id \$	Shift
(10)	0 E 1 + 6	id \$	Shift
(11)	0 E 1 + 6 id 5	\$	Reduce by $F \rightarrow id$
(12)	0 E 1 + 6 F 3	\$	Reduce by $T \rightarrow F$
(13)	0 E 1 + 6 T 9	\$	Reduce by $E \rightarrow E + T$
(14)	0 E 1	\$	Thành công

3. Xây dựng bảng phân tích cú pháp

- ❑ Có 03 phương pháp xây dựng một bảng PTCP:
 - LR đơn giản (SLR)
 - LR chính tắc
 - LALR (Lookahead-LR)

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

□ **Mục (Item):** Cho một văn phạm G , mục LR(0) văn phạm là một luật sinh của G với một dấu chấm mục tại vị trí nào đó trong vế phải.

Ví dụ 4.19: Luật sinh $A \rightarrow XYZ$ có 4 mục như sau :

$A \rightarrow \bullet XYZ$

$A \rightarrow X \bullet YZ$

$A \rightarrow XY \bullet Z$

$A \rightarrow XYZ \bullet$

Luật sinh $A \rightarrow \epsilon$ chỉ tạo ra một mục $A \rightarrow \bullet$

□ **Văn phạm tăng cường (Augmented Grammar).** Giả sử G là một văn phạm với ký hiệu bắt đầu S , ta thêm một ký hiệu bắt đầu mới S' và luật sinh $S' \rightarrow S$ để được văn phạm mới G' gọi là văn phạm tăng cường.

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

- **Phép toán bao đóng (Closure).** Giả sử I là một tập các mục của văn phạm G thì bao đóng $\text{closure}(I)$ là tập các mục được xây dựng từ I theo qui tắc sau:
- Tất cả các mục của I được thêm cho $\text{closure}(I)$.
 - Nếu $A \rightarrow \alpha \bullet B\beta \in \text{closure}(I)$ và $B \rightarrow \gamma$ là một luật sinh thì thêm $B \rightarrow \bullet \gamma$ vào $\text{closure}(I)$ nếu nó chưa có trong đó.
 - Lặp lại bước này cho đến khi không thể thêm vào $\text{closure}(I)$ được nữa.

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

□ Ví dụ Xét văn phạm tăng cường của biểu thức:

$$E' \rightarrow E$$

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

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

Nếu I là tập hợp chỉ gồm văn phạm $\{ E' \rightarrow \bullet E \}$ thì $\text{closure}(I)$ bao gồm:

$$E' \rightarrow \bullet E \text{ (Luật 1)}$$

$$E \rightarrow \bullet E + T \text{ (Luật 2)}$$

$$E \rightarrow \bullet T \text{ (Luật 2)}$$

$$T \rightarrow \bullet T * F \text{ (Luật 2)}$$

$$T \rightarrow \bullet F \text{ (Luật 2)}$$

$$F \rightarrow \bullet (E) \text{ (Luật 2)}$$

$$F \rightarrow \bullet id \text{ (Luật 2)}$$

□ **Chú ý:** Nếu một B - luật sinh được đưa vào $\text{closure}(I)$ với dấu chấm mục nằm ở đầu vế phải thì tất cả các B - luật sinh đều được đưa vào.

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

❑ Phép toán Goto

- $\text{Goto}(I, X)$, trong đó I là một tập các mục và X là một ký hiệu văn phạm là bao đóng của tập hợp các mục $A \rightarrow \alpha X \beta$ sao cho $A \rightarrow \alpha \bullet X \beta \in I$.
- Cách tính $\text{goto}(I, X)$:
 1. Tạo một tập $I' = \emptyset$.
 2. Nếu $A \rightarrow \alpha \bullet X \beta \in I$ thì đưa $A \rightarrow \alpha X \bullet \beta$ vào I' , tiếp tục quá trình này cho đến khi xét hết tập I .
 3. $\text{Goto}(I, X) = \text{closure}(I')$

❑ Ví dụ 4.21: Giả sử $I = \{ E' \rightarrow E \bullet, E \rightarrow E \bullet + T \}$.

Tính $\text{goto}(I, +)$?

- Ta có $I' = \{ E \rightarrow E + \bullet T \}$
- $(\text{goto}(I, +) = \text{closure}(I'))$ bao gồm các mục :
- $E \rightarrow E + \bullet T$ (Luật 1)
- $T \rightarrow \bullet T * F$ (Luật 2)
- $T \rightarrow \bullet F$ (Luật 2)
- $F \rightarrow \bullet (E)$ (Luật 2)
- $F \rightarrow \bullet \text{id}$ (Luật 2)

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

□ Giải thuật xây dựng họ tập hợp các mục LR(0) của văn phạm G'

□ Gọi C là họ tập hợp các mục LR(0) của văn phạm tăng cường G' . Ta có thủ tục xây dựng C như sau:

□ Procedure Item (G')

begin

$C := \text{closure}(\{ S' \rightarrow \bullet S \});$

Repeat

For Với mỗi tập các mục I trong C và mỗi ký hiệu văn phạm X sao cho $\text{goto}(I, X) \neq \emptyset$ và $\text{goto}(I, X) \notin C$ do Thêm $\text{goto}(I, X)$ vào C ;

Until Không còn tập hợp mục nào có thể thêm vào C ;

end;

□ Ví dụ: Xét văn phạm tăng cường của biểu thức:

$E' \rightarrow E$

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

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

Ví dụ: Xét văn phạm tăng cường của biểu thức:

$E' \rightarrow E$

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

	I₀:	$E' \rightarrow \bullet E$ $E \rightarrow \bullet E + T$ $E \rightarrow \bullet T$ $T \rightarrow \bullet T * F$ $T \rightarrow \bullet F$ $F \rightarrow \bullet (E)$ $F \rightarrow \bullet id$			
			Goto (I ₀ , id)	I₅:	$F \rightarrow id \bullet$
			Goto (I ₁ , +)	I₆:	$E \rightarrow E + \bullet T$ $T \rightarrow \bullet T * F$ $T \rightarrow \bullet F$ $F \rightarrow \bullet (E)$ $F \rightarrow \bullet id$
Goto (I ₀ , E)	I₁:	$E' \rightarrow E \mid \bullet$ $E \rightarrow E \bullet + T$	Goto (I ₂ , *)	I₇:	$T \rightarrow T * \bullet F$ $F \rightarrow \bullet (E)$ $F \rightarrow \bullet id$
Goto (I ₀ , T)	I₂:	$E \rightarrow T \bullet$ $T \rightarrow T \bullet * F$			
Goto (I ₀ , F)	I₃:	$T \rightarrow F \bullet$	Goto (I ₄ , E)	I₈:	$T \rightarrow (E \bullet)$ $E \rightarrow E \bullet + T$
Goto (I ₀ , ()	I₄:	$F \rightarrow (\bullet E)$ $E \rightarrow \bullet E + T$ $E \rightarrow \bullet T$ $T \rightarrow \bullet T * F$ $T \rightarrow \bullet F$ $F \rightarrow \bullet (E)$ $F \rightarrow \bullet id$	Goto (I ₆ , T)	I₉:	$E \rightarrow E + T \bullet$ $T \rightarrow T \bullet * F$
			Goto (I ₇ , F)	I₁₀:	$T \rightarrow T * F \bullet$
			Goto (I ₈ ,))	I₁₁:	$F \rightarrow (E) \bullet$

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

□ Thuật toán xây dựng bảng phân tích SLR

Input: Một văn phạm tăng cường G'

Output: Bảng phân tích SLR với hàm action và goto

Phương pháp:

1. Xây dựng $C = \{ I_0, I_1, \dots, I_n \}$, họ tập hợp các mục LR(0) của G' .
2. Trạng thái i được xây dựng từ I_i . Các action tương ứng trạng thái i được xác định như sau:
 - 2.1. Nếu $A \rightarrow \alpha \cdot a\beta \in I_i$ và $\text{goto}(I_i, a) = I_j$ thì $\text{action}[i, a] = \text{"shift } j\text{"}$. Ở đây a là ký hiệu kết thúc.
 - 2.2. Nếu $A \rightarrow \alpha \cdot \in I_i$ thì $\text{action}[i, a] = \text{"reduce } (A \rightarrow \alpha)\text{"}$, $\forall a \in \text{FOLLOW}(A)$. Ở đây A không phải là S'
 - 2.3. Nếu $S' \rightarrow S \cdot \in I_i$ thì $\text{action}[i, \$] = \text{"accept"}$.

Nếu một action đưng độ được sinh ra bởi các luật trên, ta nói văn phạm không phải là SLR(1). Giải thuật sinh ra bộ phân tích cú pháp sẽ thất bại trong trường hợp này.

3. Với mọi ký hiệu chưa kết thúc A , nếu $\text{goto}(I_i, A) = I_j$ thì $\text{goto}[i, A] = j$
4. Tất cả các ô không xác định được bởi 2 và 3 đều là "error"
5. Trạng thái khởi đầu của bộ phân tích cú pháp được xây dựng từ tập các mục chứa $S' \rightarrow \cdot S$

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

Ví dụ: Ta xây dựng bảng phân tích cú pháp SLR cho

văn phạm tăng cường G'

$E' \rightarrow E$

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

(0) $E' \rightarrow E$

(1) $E \rightarrow E + T$

(2) $E \rightarrow T$

(3) $T \rightarrow T * F$

(4) $T \rightarrow F$

(5) $F \rightarrow (E)$

(6) $F \rightarrow id$

1. $C = \{ I_0, I_1, \dots, I_{11} \}$

2. $FOLLOW(E) = \{ +,), \$ \}$

$FOLLOW(T) = \{ *, +,), \$ \}$

$FOLLOW(F) = \{ *, +,), \$ \}$

Dựa vào họ tập hợp mục C đã được xây dựng trong ví trước

Trước tiên xét tập mục I_0 : Mục $F \rightarrow \bullet (E)$ cho ra $action[0, (] = \text{"shift 4"}$, và mục

$F \rightarrow \bullet id$ cho $action[0, id] = \text{"shift 5"}$.

Các mục khác trong I_0 không sinh được hành động nào.

Bây giờ xét I_1 : Mục $E' \rightarrow E \bullet$ cho $action[1, \$] = \text{"accept"}$, mục $E \rightarrow E \bullet + T$ cho

$action[1, +] = \text{"shift 6"}$.

Kế đến xét I_2 :

$E \rightarrow T \bullet$

$T \rightarrow T \bullet * F$

Vì $FOLLOW(E) = \{ +,), \$ \}$, mục đầu tiên làm cho $action[2, \$] = action[2, +] = \text{"reduce (E (T))"}$. Mục thứ hai làm cho $action[2, *] = \text{"shift 7"}$

3. Xây dựng bảng phân tích cú pháp

Phương pháp SLR

$$1. E \rightarrow E + T$$

$$2. E \rightarrow T$$

$$3. T \rightarrow T * F$$

$$4. T \rightarrow F$$

$$5. F \rightarrow (E)$$

$$6. F \rightarrow \text{id}$$

Action Table

Goto Table

state	id	+	*	()	\$		E	T	F
0	s5			s4				1	2	3
1		s6				acc				
2		r2	s7		r2	r2				
3		r4	r4		r4	r4				
4	s5			s4				8	2	3
5		r6	r6		r6	r6				
6	s5			s4					9	3
7	s5			s4						10
8		s6			s11					
9		r1	s7		r1	r1				
10		r3	r3		r3	r3				
11		r5	r5		r5	r5				

$\text{FOLLOW}(T) = \{ \epsilon, +,), \Phi \}$

4. Văn phạm họ LR

- ❑ Việc chính là làm thế nào để xây dựng bảng phương án? Có nhiều thuật toán làm việc này
- ❑ LR(0): thuật toán cơ bản, mọi thuật toán LR đều dựa trên nó
- ❑ SLR (Simple LR): cải tiến một chút từ LR(0), mạnh hơn, dễ cài đặt
- ❑ LR(1): còn gọi là LR chính tắc ~ Canonical LR, sử dụng cho nhiều loại văn phạm, kích cỡ bảng rất lớn
- ❑ LALR(1): cân bằng giữa SLR và LR, đủ dùng cho hầu hết các văn phạm nhân tạo

3. Văn phạm họ LR

- ❑ Để dễ dàng cho việc thực thi automat, ta bổ sung thêm luật $S' \rightarrow S$ vào tập luật
- ❑ Khái niệm LR(0) item: một luật đang được phân tích dở, sử dụng dấu chấm (.) để ngăn giữa phần trước và phần sau (tương tự như thuật toán earley)
- ❑ Luật $S \rightarrow ABC$ sẽ có 4 item:
 1. $S \rightarrow .ABC$
 2. $S \rightarrow A.BC$
 3. $S \rightarrow AB.C$
 4. $S \rightarrow ABC.$