Bài 6
Phân tích cú pháp trên xuống có quay lui

ONE LOVE. ONE FUTURE.

Bài toán phân tích cú pháp

Bài toán đặt ra

Cho văn phạm phi ngữ cảnh G và xâu w

$w \in L(G)$ (L(G): ngôn ngữ chứa các xâu được sản sinh nhờ văn phạm G) đúng hay sai?

Phân tích trên xuống (top down)

$S \Rightarrow^* w$?

w đúng cú pháp ⇒cây PT cú pháp

E -> E + T

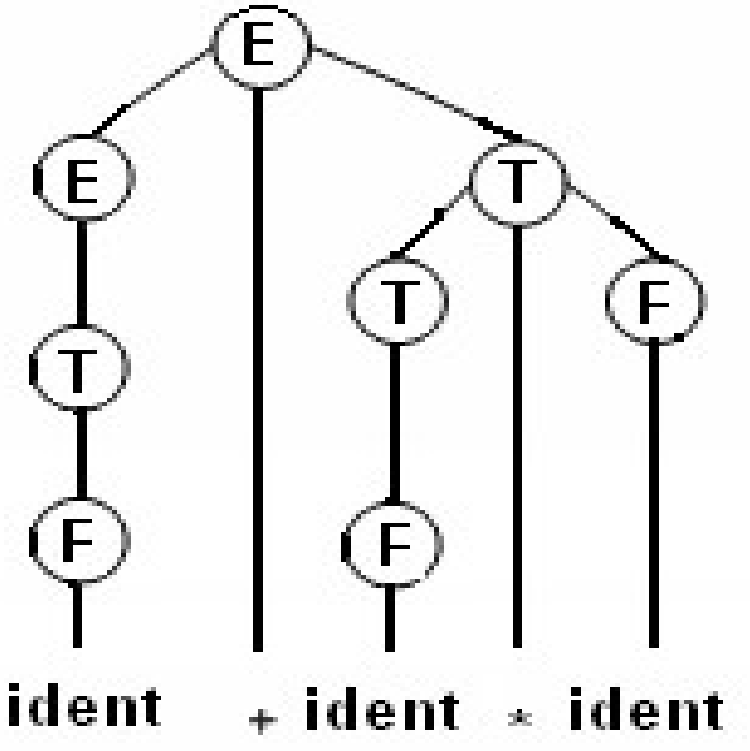
E -> T

T -> T * F

T -> F

F -> (E)

F -> ident



Biểu diễn cây PT cú pháp bằng cách nào?

Phân tích trái

- *Phân tích trái* của α là dãy các sản xuất được sử dụng trong suy dẫn trái ra α từ S
- *Phân tích phải* của α là *nghịch đảo* của dãy các sản xuất được sử dụng trong suy dẫn phải ra α từ S
- *Phân tích là danh sách các số từ 1 đến p*

Ví dụ

Xét văn phạm G, các sản xuất được đánh số như sau

1. $E \rightarrow T+E$
2. $E \rightarrow T$
3. $T \rightarrow F^* T$
4. $T \rightarrow F$
5. $F \rightarrow (E)$
6. $F \rightarrow a$

Suy dẫn trái: $E \Rightarrow T \Rightarrow F^*T \Rightarrow a^*T \Rightarrow a^* F \Rightarrow a^*(E) \Rightarrow$
$a^* (T+ E) \Rightarrow a^* (F + E) \Rightarrow a^* (a + E) \Rightarrow a^* (a+T) \Rightarrow$
$a^* (a + F) \Rightarrow a^* (a+a)$

- Phân tích trái của xâu a*(a+a) là 23645146246
- Phân tích phải của xâu a*(a+a) là 66464215432

Giải thuật phân tích top down quay lui

- Tư tưởng chủ yếu của giải thuật là xây dựng cây phân tích cú pháp (cây suy dẫn) cho xâu w

- Đánh số thứ tự các sản xuất có cùng vế trái, như vậy, các A - sản xuất của văn phạm sẽ được xếp thứ tự

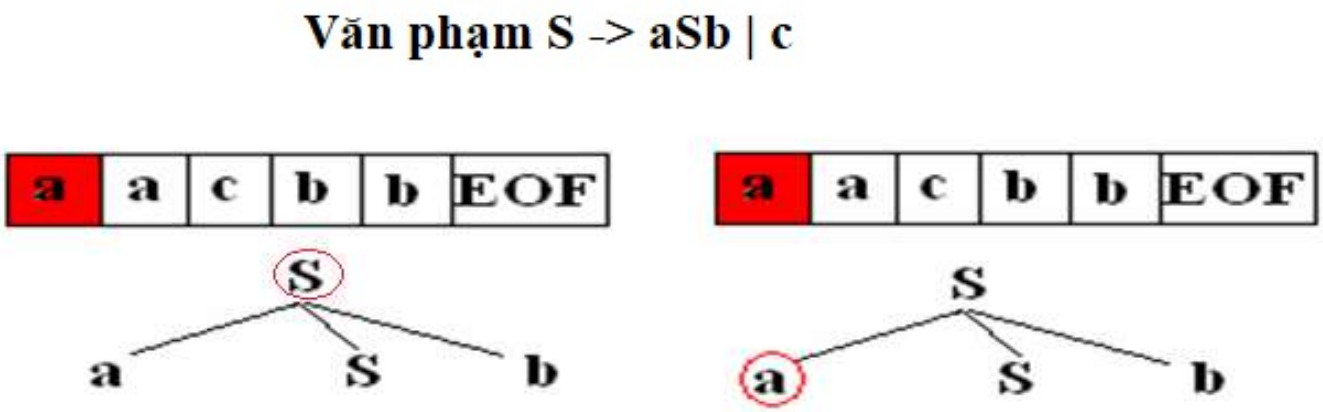
  $A \rightarrow \alpha 1 \mid \alpha 2 \mid . . . . \mid \alpha n$

Mô tả giải thuật

- Bắt đầu từ nút gốc S
- Nút S được coi là nút hoạt động (K/h không kết thúc)
- Tạo ra các nút con một cách đệ quy

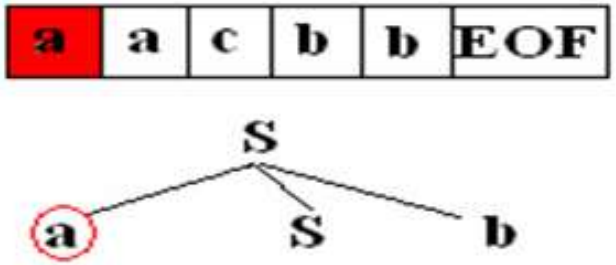
Nút hoạt động là ký hiệu không kết thúc A

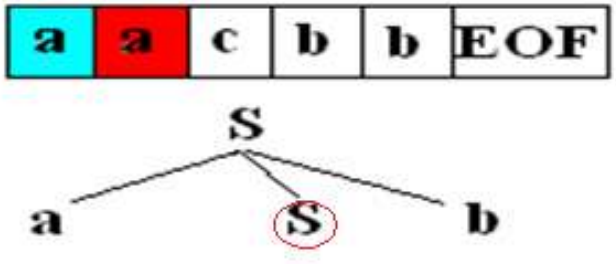
- Chọn vế phải đầu tiên của A- sản xuất : $X_1X_2. . . .X_k$.
- Tạo k nút con trực tiếp của A với nhãn $X_1, X_2, ...X_k$.
- Nút hoạt động là nút nhãn X_1.
- Nếu k = 0, (sản xuất $A \rightarrow \varepsilon$) thì nút hoạt động sẽ là nút ngay sau A khi duyệt cây theo thứ tự trái



Nút hoạt động là ký hiệu kết thúc a

- **So sánh với ký hiệu đang xét trên xâu vào.**
  - Nếu trùng với ký hiệu đang xét thì chuyển đầu đọc sang phải 1 ô, chuyển sang xét nút tiếp theo.
  - Nếu a không trùng với ký hiệu đang xét thì **quay lui** tới nút mà tại đó đã sử dụng sản xuất trước (Thay thế một ký hiệu không kết thúc - chẳng hạn A - bằng vế phải một sản xuất).
  - Chuyển đầu đọc sang trái (nếu cần) và thử với lựa chọn tiếp theo của A. Nếu không còn lựa chọn nào khác thì quay lui tới bước trước đó





  - **Nếu đã quay lui tới S và không còn lựa chọn khác: câu sai cú pháp**

Điều kiện để thực hiện giải thuật

- *Văn phạm G cần thoả điều kiện không đệ quy trái để tránh rơi vào chu trình vô hạn*

Ví dụ

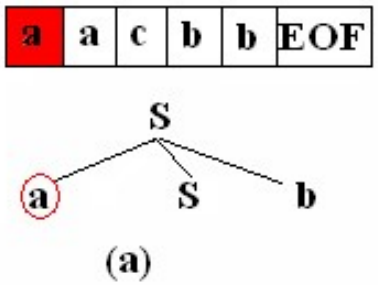
- Quay lại văn phạm

1. $S \rightarrow aSb$
2. $S \rightarrow c$

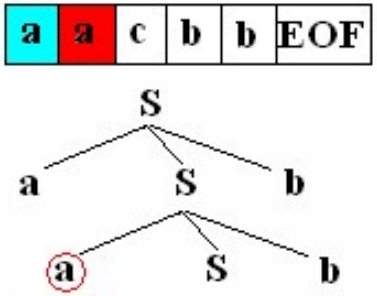
Các sản xuất được đánh số từ 1 đến 2.

- Xét xâu vào aacbb

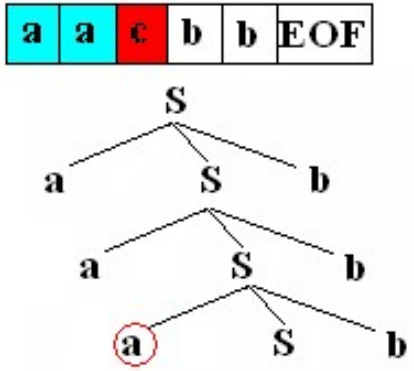
Dựng cây phân tích cú pháp



(a)

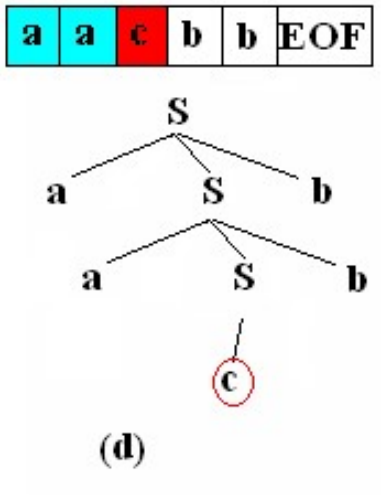


(b)



(c)

Thử lựa chọn khác



Giải thuật phân tích cú pháp quay lui

- **Vào**

  Văn phạm G phi ngữ cảnh không đệ quy trái,
  xâu w = a1. . . .an, n ≥ 0
  Các sản xuất của G được đánh số 1,. . . . q

- **Ra**

  Một phân tích trái cho w (nếu có)
  Thông báo lỗi nếu ngược lại

Phương pháp

- *Bộ phân tích cú pháp sử dụng 2 stack D_1 và D_2.*

  D_2 biểu diễn dạng câu trái hiện tại có được bằng cách thay thế các ký hiệu không kết thúc bởi vế phải tương ứng

  D_1 ghi lại lịch sử những lựa chọn đã sử dụng và những ký hiệu vào trên đó đầu đọc đã đổi vị trí

Đánh số các sản xuất có cùng vế trái

- $\forall A \in N$, giả sử có các A-sản xuất

$$A \rightarrow \alpha 1 \mid \alpha 2 \mid \ldots . \mid \alpha n$$

Coi các sản xuất trên là

$$A_1 \rightarrow \alpha_1$$

$$\ldots .$$

$$A_n \rightarrow \alpha_n$$

Hình trạng của giải thuật

Bộ bốn (s, i, α, β)

- s $\in$ Q: Trạng thái hiện thời
  - q: Trạng thái bình thường
  - b: Quay lui
  - t: Kết thúc
- i : Vị trí đầu đọc (Băng vào có dấu hiệu kết thúc $)

α: Nội dung stack thứ nhất

β: Nội dung stack thứ hai

Thực hiện giải thuật

- Bắt đầu từ hình trạng đầu, tính liên tiếp các hình trạng tiếp theo cho đến khi không tính được nữa.

- Nếu hình trạng cuối là (t,n+1,γ,ε), đưa ra h(γ) và dừng. Ngược lại đưa ra thông báo sai

Ví dụ

- Xét xâu vào aacbb và văn phạm G với các sản xuất

$$S \rightarrow aSb$$
$$S \rightarrow c$$

Đánh số lại các sản xuất

1. $S1 \rightarrow aSb$
2. $S2 \rightarrow c$

Quá trình thay đổi hình trạng

(q,1, ε, S#)

⊢ (q, 1, S_1, aSb#)

⊢ (q, 2, S_1a, Sb#)

⊢ (q, 2, S_1aS_1,aSbb#)

⊢ (q, 3, S_1aS_1a, Sbb#)

⊢ (q, 3, S_1aS_1aS_1,aSbbb#)

⊢ (b, 3, S_1aS_1aS_1,aSbbb#)

⊢ (q, 3, S_1aS_1aS_2, cbb#)

⊢ (q, 4, S_1aS_1aS_2c,bb#)

⊢ (q, 5, S_1aS_1aS_2cb,b#)

⊢ (q, 6, S_1aS_1aS_2cbb,#)

⊢ (t, 6, S_1aS_1aS_2cbb, ε)

Tìm phân tích trái

- *h(a) = ε $\forall$a là ký hiệu kết thúc*

  $h(A_i)$= p ,

  p là số hiệu của sản xuất liên hệ với sản xuất $A \rightarrow \gamma$ với γ là lựa chọn thứ i của A
- *Văn phạm*

1. $S_1 \rightarrow aSb$
2. $S_2 \rightarrow \varepsilon$

- h($S_1aS_1aS_2bb$)=12

Thử phân tích quay lui với KPL

- Phân tích từ vựng và mã hóa từ tố
- Tập sản xuất của văn phạm

Chuyển sơ đồ cú pháp thành luật

<program >::= *program ident ;* <block>.

<block>::= <const-decl><type-decl>
<proc-decl><func-decl><var-decl> *begin* <statement-list> *end*

Mã hóa ký hiệu không kết thúc

```
if(str=="<program>") return 1;
if(str=="<block>") return 2;
if(str=="<const-decl>") return 3;
if (str == "<const-assign-list>") return 4;
if (str == "<constant>") return 5;
if(str=="<type-decl>") return 6;
if (str =="<type-assign-list>") return 7;
if (str == "<type>") return 8;
if (str == "<basictype>") return 9;
if(str=="<var-decl>") return 10;
if (str == "<ident-list>") return 11;
if(str=="<proc-decl>") return 12;
if (str == "<para-list>") return 13;
if (str == "<para-one>") return 14;
if(str=="<func-decl>") return 15;
if(str=="<statement-list>") return 16;
if(str=="<statement>" ) return 17;
if (str == "<condition>") return 18;
if (str == "<relation>") return 19;
if(str=="<expression>") return 20;
if (str == "<adding-op>") return 21;
if(str=="<term>") return 22
if(str=="<multiplying-op>") return 23;
if (str == "<factor>") return 24;
```

Mã hóa từ tố: tên, số, hằng ký tự

```
// ident;
 if(str == "ident") return 25;
//const
 if(str == "number")return 26;
 if (str == "charcon") return 27;
//operator
 if(str == "plus")return 28;
 if (str == "minus") return 29;
 if (str == "times") return 30;
 if (str == "slash") return 31;
 if (str == "oddsym") return 32;
 if (str == "assign") return 33;
 if (str == "leq") return 34;
```

```
//specific symbol
if (str =="lparen") return 35;
if (str == "rparen") return 36;
if (str == "comma") return 37;
if (str == "semicolon") return 38;
if (str == "period") return 39;
if (str == "becomes") return 40;
if (str == "lbrace") return 41;
if (str == "rbrace") return 42;
if (str == "lbrack") return 43;
if (str == "rbrack") return 44;
```

Mã hóa từ tố: từ khóa

```
if (str == "beginsym") return 45;
if (str == "endsym") return 46;
if (str == "ifsym") return 47;
if (str == "thensym") return 48;
if (str == "whilesym") return 49;
if (str == "dosym") return 50;
if (str == "callsym") return 51;
if (str == "constsym") return 52;
```

```
if (str == "varsym") return 53;
if (str == "progsym") return 54;
if (str == "funcsym") return 55;
if (str == "typesym") return 56;
if (str == "arraysym") return 57;
if (str == "ofsym") return 58;
if (str == "intsym") return 59;
if (str == "charsym") return 60;
```

Mã hóa từ tố: phép toán quan hệ

```
//relations
if (str == "eql") return 61;
if (str == "leq") return 62;
if (str == "neq") return 63;
if (str == "lss") return 64;
if (str == "gtr") return 65;
if (str == "geq") return 66;
```

Mã hóa sản xuất

<program >::= *program ident ;* <block>.

setlaw[1,1]="54 25 38 2 39 ";

<block>::= <const-decl><type-decl>
<proc-decl><func-decl><var-decl> *begin* <statement-list> *end*

setlaw[2,1]=" 3 6 12 15 10 45 16 46 ";

Nhận xét

- Cài đặt phức tạp
- Độ phức tạp tính toán hàm mũ theo độ dài xâu vào. Do vậy chi phí thời gian quá lớn nếu chương trình phải phân tích gồm nhiều ký hiệu (từ tố)
- Không thể thông báo lỗi chi tiết