



Bài 3. Văn phạm sản sinh

ONE LOVE. ONE FUTURE.

Lý thuyết ngôn ngữ

- **Phân tích cú pháp** là quá trình kiểm tra xem **một câu** chứa các **ký hiệu**, trong ngôn ngữ tự nhiên hoặc ngôn ngữ lập trình hay bất cứ cấu trúc nào có dạng tương tự có được xây dựng từ một **văn phạm hình thức** không?
- **Lý thuyết ngôn ngữ hình thức** xem xét ngôn ngữ dưới dạng mô hình toán học. Với cách thức mô tả hình thức, các khái niệm liên quan đến ngôn ngữ được thể hiện dưới dạng máy đọc được, qua đó có thể xử lý trên máy tính
- Một cách trực quan, có thể coi ngôn ngữ là một tập các **câu** (**xâu**).

Bảng chữ

Ký hiệu

Là đối tượng cơ bản nhất của lý thuyết ngôn ngữ, không được định nghĩa một cách hình thức.

Bảng chữ

Tập hữu hạn, không rỗng các ký hiệu

Thường ký hiệu bảng chữ bằng chữ cái Hy Lạp Σ (sigma)

Ví dụ về bảng chữ

- Bảng chữ nhị phân : $\Sigma = \{0,1\}$
- Bảng chữ cái viết thường tiếng Anh: $\Sigma = \{a,b,c,..z\}$
- Tập chữ cái và chữ số : $\Sigma = \{a-z, A-Z, 0-9\}$
- Tập các nucleotit tạo thành phân tử DNA $\Sigma = \{a,c,g,t\}$ (guanine, adenine, thymine, cytosine)
- Tập các ký tự hợp lệ trong ngôn ngữ C.
- Tập các từ tố trong ngôn ngữ C (hằng số, định danh, từ khóa, hằng ký tự, toán tử)

Ví dụ về bảng chữ: tập ký hiệu hợp lệ trong ngôn ngữ C

Loại	Character Set
Chữ cái viết thường	a –z
Chữ cái viết hoa	A - Z
Chữ số	0-9
Các ký tự đặc biệt	~! # \$% ^ & *()_ + \ ' - = { } [] : " ; < > ? , . /
Khoảng trống	Tab Or New line Or Space

Xâu (string, sentence)

- *Xâu là dãy hữu hạn ký hiệu trong một bảng chữ nào đó*
- Xâu rỗng: xâu không gồm ký hiệu nào, ký hiệu ε
- Ví dụ :
 - 1000010101111
 - Chương trình C là một dãy các từ tố
 - Mẫu DNA của con người

```
GGTGTGGGGACAGGGGTGTGGGGACAGGGGTCTGGGGACAGGGGTGTGGG
GACAGGGGTCCTGGGGACAGGGGTGTGGGGATAGGGGTGTGGGGACAGGG
GTGTGGGGACAGGGGTGTGGGGACAGGGGTCTGGGGACAGCAGCGCAAAG
AGCCCCGCCCTGCAGCCTCCAGCTCTCCTGGTCTAATGTGGAAAGTGGCC
CAGGTGAGGGCTTTGCTCTCCTGGAGACATTTGCCCCAGCTGTGAGCAG
GGACAGGTCTGGCCACCGGGCCCCCTGGTTAAGACTCTAATGACCCGCTGG
TCCTGAGGAAGAGGTGCTGACGACCAAGGAGATCTTCCCACAGACCCAGC
ACCAGGGAAATGGTCCGGAATTCAGCCTCAGCCCCAGCCATCTGCCG
ACCCCCCACCCAGGCCCTAATGGGCCAGGCGGCAGGGGTTGAGAGGTA
GGGGAGATGGGCTCTGAGACTATAAAGCCAGCGGGGGCCAGCAGCCCTC
```



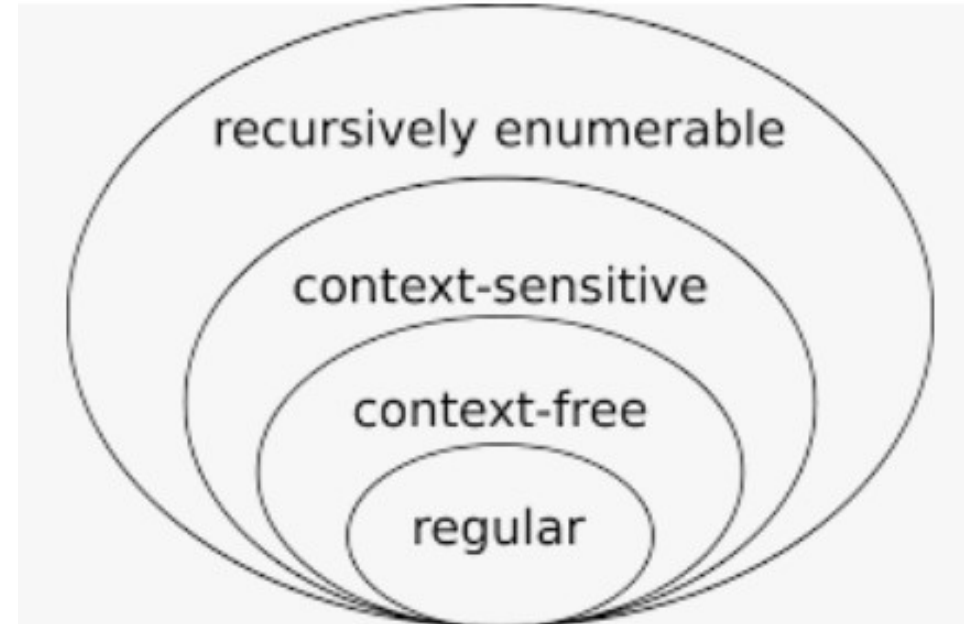
Ngôn ngữ là tập các xâu trên bảng chữ Σ

Ví dụ về ngôn ngữ:

- Tập tất cả các xâu trên bảng chữ $\{0, 1\}$,
- $\{ a^n \mid n \text{ is a prime number} \}$, tập các xâu chỉ chứa ký hiệu a có độ dài là một số nguyên tố
- Ngôn ngữ lập trình C: Tập tất cả các chương trình đúng cú pháp trong ngôn ngữ C

Phân cấp Chomsky

- **Ngôn ngữ loại 0 (đệ quy kể được)**
chứa các thể hiện của một bài toán
- **Ngôn ngữ loại 1 (cảm ngữ cảnh)**
một số câu trong ngôn ngữ tự nhiên,
ngôn ngữ của các phân tử DNA
- **Ngôn ngữ loại 2 (phi ngữ cảnh)**
Ngôn ngữ lập trình, ngôn ngữ tự nhiên
- **Ngôn ngữ loại 3 (chính quy)**
từ tố trong ngôn ngữ lập trình, từ trong ngôn ngữ tự nhiên



- Thực chất là biểu diễn cú pháp của ngôn ngữ
- Biểu diễn phải hữu hạn
- Công cụ sản sinh: văn phạm
- Công cụ đoán nhận: ô tômat

Công cụ sản sinh và đoán nhận của các lớp ngôn ngữ

Lớp ngôn ngữ	Công cụ sản sinh	Công cụ đoán nhận	Ghi chú
Đệ quy kế được	Văn phạm loại 0 (ngữ cấu)	Máy Turing	Các bài toán tổng quát
Cảm ngữ cảnh	Văn phạm cảm ngữ cảnh	Ôtômat tuyến tính giới nội	Ngôn ngữ tự nhiên
Phi ngữ cảnh	Văn phạm phi ngữ cảnh	Ôtômat đẩy xuống	Ngôn ngữ lập trình, phần chính của ngôn ngữ tự nhiên
Chính quy	Văn phạm chính quy Công cụ biểu diễn: Biểu thức chính quy	Ôtômat hữu hạn	Từ vựng của ngôn ngữ tự nhiên, ngôn ngữ lập trình

Văn phạm xuất phát từ ngôn ngữ tự nhiên

<câu>::=<chủ ngữ> <vị ngữ>

<chủ ngữ>::=<danh ngữ>

<danh ngữ>::=<danh từ> <tính từ>

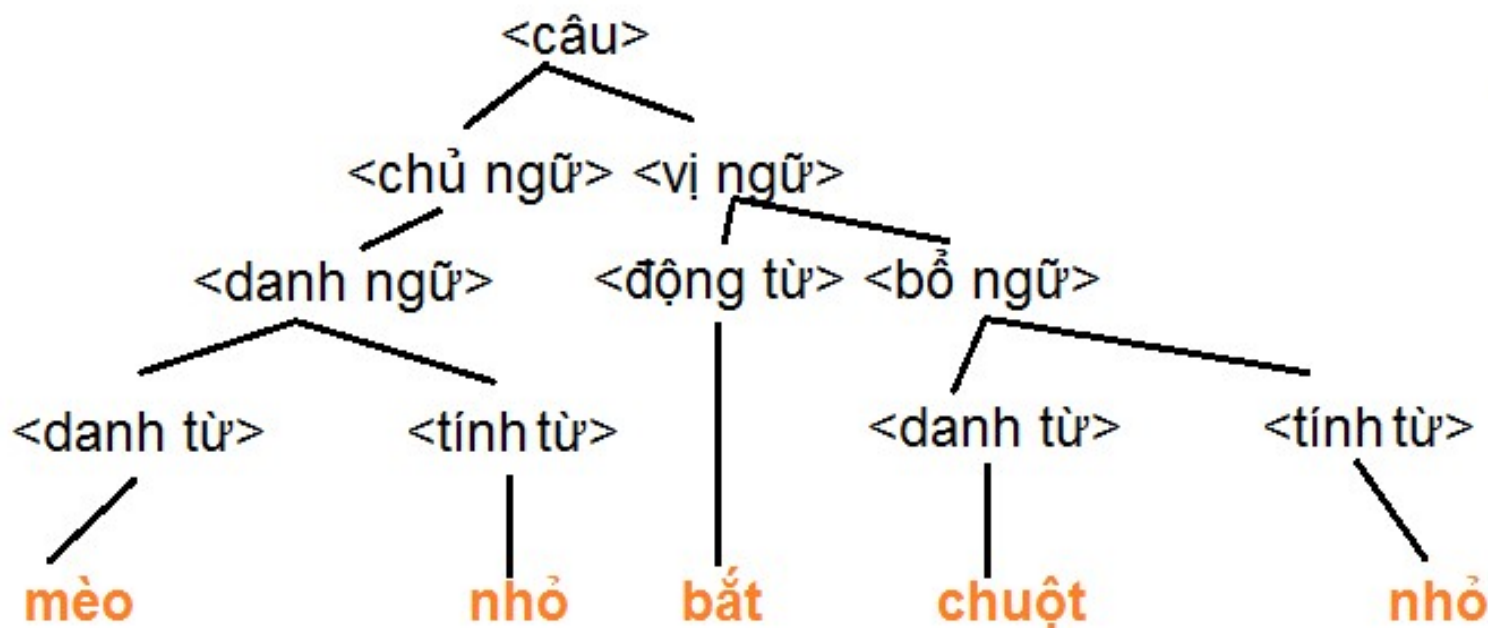
<vị ngữ>::=<động từ> <bổ ngữ>

<bổ ngữ>::=<danh từ> <tính từ>

<động từ> ::= **bắt**

<danh từ>::= **mèo** | **chuột**

<tính từ>::= **nhỏ**



$\langle \text{Số} \rangle ::= -\langle \text{Số thập phân} \rangle \mid \langle \text{số thập phân} \rangle$

$\langle \text{Số thập phân} \rangle ::= \langle \text{Dãy chữ số} \rangle \mid \langle \text{Dãy chữ số} \rangle . \langle \text{Dãy chữ số} \rangle$

$\langle \text{Dãy chữ số} \rangle ::= \langle \text{Chữ số} \rangle \mid \langle \text{Chữ số} \rangle \langle \text{Dãy chữ số} \rangle$

$\langle \text{Chữ số} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Định nghĩa hình thức của văn phạm và văn phạm phi ngữ cảnh

Văn phạm là bộ 4 $G = (\Sigma, \Delta, P, S)$, trong đó

Σ : Tập hữu hạn các ký hiệu kết thúc

Δ : Tập hữu hạn các ký hiệu không kết thúc

$S \in \Delta$: Ký hiệu đầu

P : Tập hữu hạn các *sản xuất (luật sinh)*, ký hiệu $\alpha \rightarrow \beta$, α chứa ít nhất 1 ký hiệu không kết thúc, β gồm ký hiệu kết thúc, ký hiệu không kết thúc và có thể không gồm ký hiệu nào.

Văn phạm G là **phi ngữ cảnh** nếu mọi sản xuất của văn phạm có dạng

$A \rightarrow \beta$, trong đó A là ký hiệu không kết thúc.

$S \rightarrow -A \mid A$ (Sản xuất - Luật sinh)

$A \rightarrow B.B \mid B$

$B \rightarrow CB \mid C$

$C \rightarrow 0 \mid 1 \mid 2 \mid \dots \mid 9$

Quy ước viết trong mô hình văn phạm sản sinh

Chữ in hoa : K/h không kết thúc

Chữ thường (số, dấu): K/h kết thúc

$\alpha \rightarrow \beta, \alpha \rightarrow v$ Viết $\alpha \rightarrow \beta \mid v$

Làm thế nào để sản sinh ra các xâu ?

Văn phạm phi ngữ cảnh có thể dùng để sản sinh ra các xâu thuộc ngôn ngữ như sau:

X = Ký hiệu đầu

While còn ký hiệu không kết thúc Y trong X ***do***

Áp dụng một trong các sản xuất của, văn phạm chẳng hạn Y $\rightarrow w$

Khi X chỉ chứa ký hiệu kết thúc, nó là xâu được sản sinh bởi văn phạm.

Ví dụ : quá trình sản sinh sâu -3.14

- Quá trình thay thế

S

-A

-B.B

-B.CB

-C.CB

-C.CC

-3.CC

-3.1C

-3.14

- Sản xuất được sử dụng

$S \rightarrow -A$

$A \rightarrow B.B$

$B \rightarrow CB$

$B \rightarrow C$

$B \rightarrow C$

$C \rightarrow 3$

$C \rightarrow 1$

$C \rightarrow 4$



Quá trình suy dẫn (derivation)

- Mỗi lần thực hiện việc thay thế là một bước suy dẫn.
- Nếu mỗi dạng câu có nhiều ký hiệu không kết thúc để thay thế có thể thay thế bất cứ ký hiệu không kết thúc nào

- Nếu giải thuật phân tích cú pháp chọn ký hiệu không kết thúc cực trái hay cực phải để thay thế, kết quả của nó là suy dẫn trái hoặc suy dẫn phải

Ví dụ suy dẫn trái:

$$S \Rightarrow -A \Rightarrow -B.B \Rightarrow -C.B \Rightarrow -3.B \Rightarrow -3.BC \Rightarrow -3.CC$$

$$\Rightarrow -3.1C \Rightarrow -3.14$$

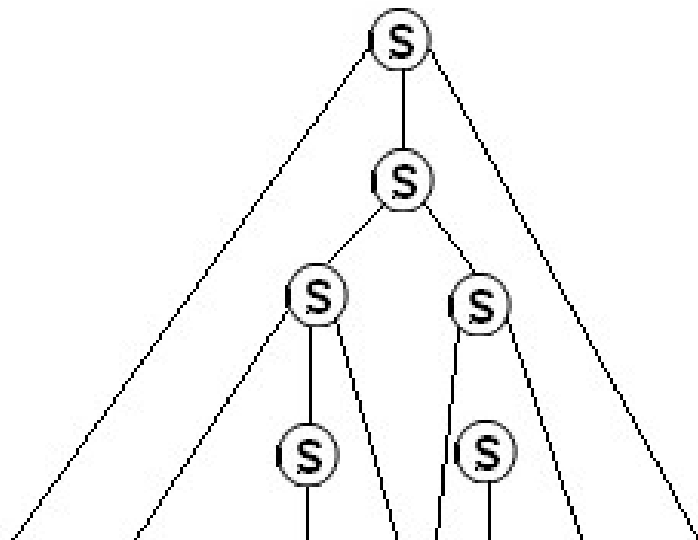
Ví dụ suy dẫn phải:

$$S \Rightarrow -A \Rightarrow -B.B \Rightarrow -B.BC \Rightarrow -B.B4 \Rightarrow -B.C4 \Rightarrow -B.14$$

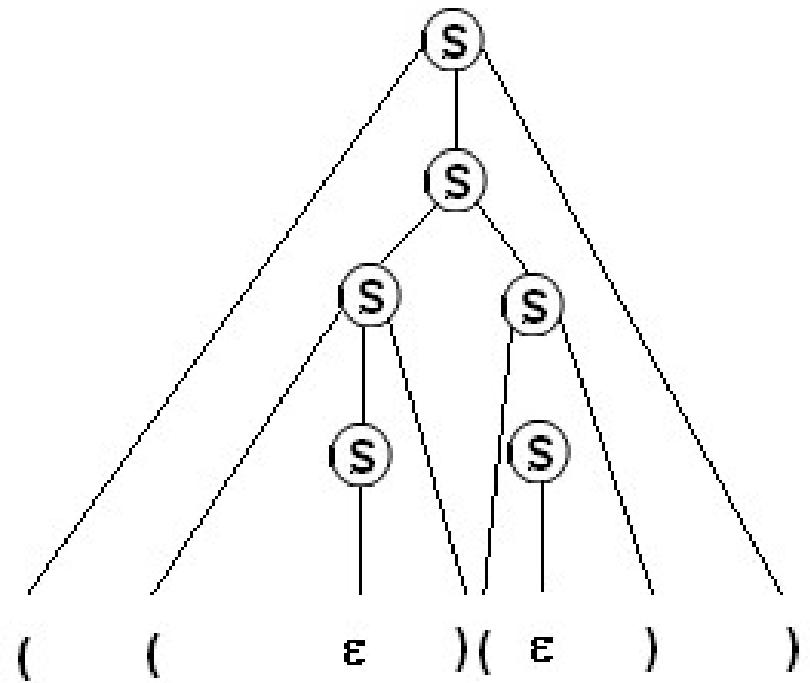
$$\Rightarrow -C.14 \Rightarrow -3.14$$

Cây suy dẫn (Cây phân tích cú pháp)

Cây suy dẫn có những đặc điểm sau

- 1) Mỗi nút của cây có nhãn là ký hiệu kết thúc, ký hiệu không kết thúc hoặc ϵ (xâu rỗng)
 - 2) Nhãn của nút gốc là S (ký hiệu đầu)
 - 3) Nút trong có nhãn là ký hiệu không kết thúc hoặc ϵ
 - 4) Nút A có các nút con từ trái qua phải là dạng $A \rightarrow X_1 X_2 \dots X_k$
 - 5) Nút lá có thể có nhãn ϵ chỉ khi tồn tại sản phẩm chỉ có một nút con duy nhất
- 

Ví dụ: Cây phân tích cú pháp của văn phạm

$$G: S \rightarrow SS \mid (S) \mid \varepsilon \quad w = (())()$$


Văn phạm nhập nhằng

Văn phạm

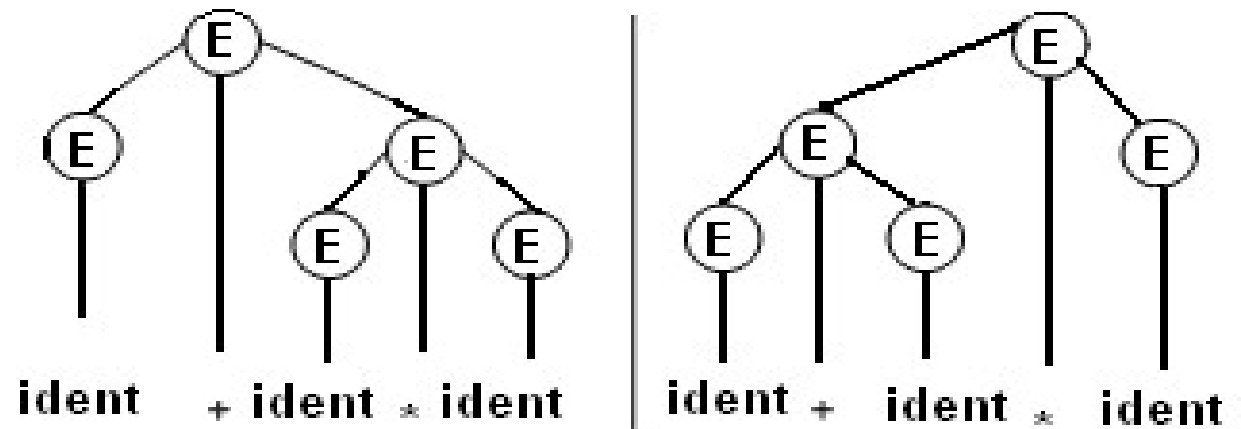
$E \rightarrow E + E$

$E \rightarrow E * E$

$E \rightarrow (E)$

$E \rightarrow \text{ident}$

E: Expression



Cho phép đưa ra hai suy dẫn khác nhau cho xâu $\text{ident} + \text{ident} * \text{ident}$ (chẳng hạn $x + y * z$)



Văn phạm là nhập nhằng

Khử nhập nhằng (disambiguation): thay thế bằng VP không nhập nhằng

$E \rightarrow E + T$

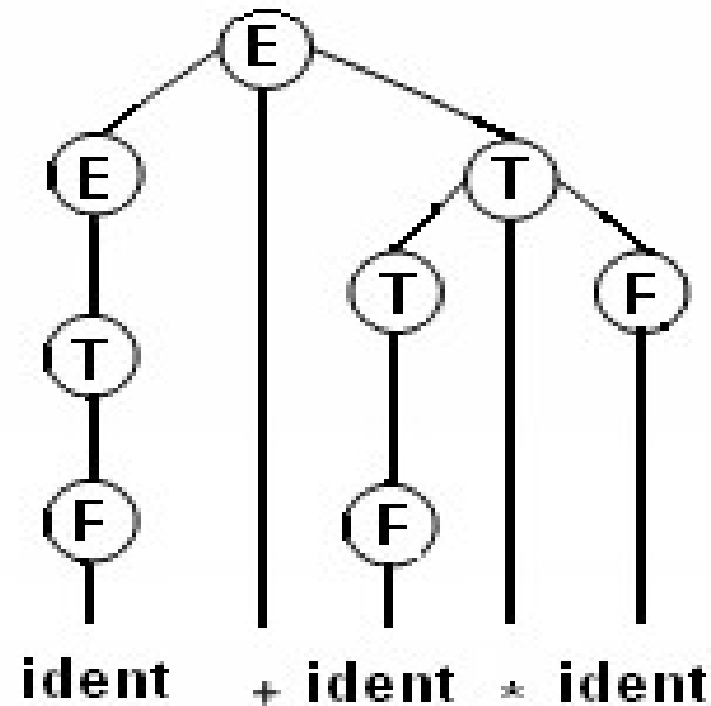
$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{ident}$



E: Expression, T: Term, F: Factor

(Bằng cách thêm các ký hiệu không kết thúc và các sản xuất để đảm bảo thứ tự ưu tiên)

Đệ quy trực tiếp $X \Rightarrow \omega_1 X \omega_2$

- **Một sản xuất là đệ quy nếu** $X \Rightarrow^* \omega_1 X \omega_2$
- Có thể dùng để biểu diễn các quá trình lặp hay cấu trúc lồng nhau

Đệ quy trái $X \Rightarrow b \mid \text{Xa}$.

$X \Rightarrow Xa \Rightarrow Xaa \Rightarrow Xaaa \Rightarrow baaaaa \dots$

Đệ quy phải $X \Rightarrow b \mid a\text{X}$.

$X \Rightarrow aX \Rightarrow aax \Rightarrow aaax \Rightarrow \dots aaaaaab$

Đệ quy giữa $X \Rightarrow b \mid (\text{X})$.

$X \Rightarrow (X) \Rightarrow ((X)) \Rightarrow (((X))) \Rightarrow (((... (b)...)))$

Đệ quy gián tiếp $X \Rightarrow^* \omega_1 X \omega_2$

- Giả sử các A-sản xuất có đệ quy trái của văn phạm G là:

$$A \rightarrow A\alpha_1$$

.....

$$A \rightarrow A\alpha_n$$

- và các sản xuất còn lại không đệ quy trái là:

$$A \rightarrow \beta_1$$

.....

$$A \rightarrow \beta_n$$

Removing Left Recursion

Let K_A denote a symbol which does not already occur in the grammar.

Replace the above productions by:

$$A \rightarrow \beta_1 K_A \mid \dots \mid \beta_s K_A$$

$$K_A \rightarrow \varepsilon \mid \alpha_1 K_A \mid \dots \mid \alpha_r K_A$$

Clearly the grammar G' produced is equivalent to G .

- Giả sử K_A là ký hiệu không kết thúc mới được thêm vào văn phạm
- Thay các sản xuất đệ quy trái bằng

$$A \rightarrow \beta_1 K_A$$

.....

$$A \rightarrow \beta_n K_A$$

và

$$K_A \rightarrow \alpha_1 K_A$$

.....

$$K_A \rightarrow \alpha_n K_A$$

$$K_A \rightarrow \varepsilon$$

Ví dụ: Khử đệ quy trái

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow \text{ident}$



$E \rightarrow E + T$

$E \rightarrow T$

Thêm biến E'

$E \rightarrow TE'$

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

$T \rightarrow T * F$

$T \rightarrow F$

Thêm biến T'

$T \rightarrow FT'$

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



$E \rightarrow TE'$

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

$T \rightarrow FT'$

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

$F \rightarrow (E)$

$F \rightarrow \text{ident}$