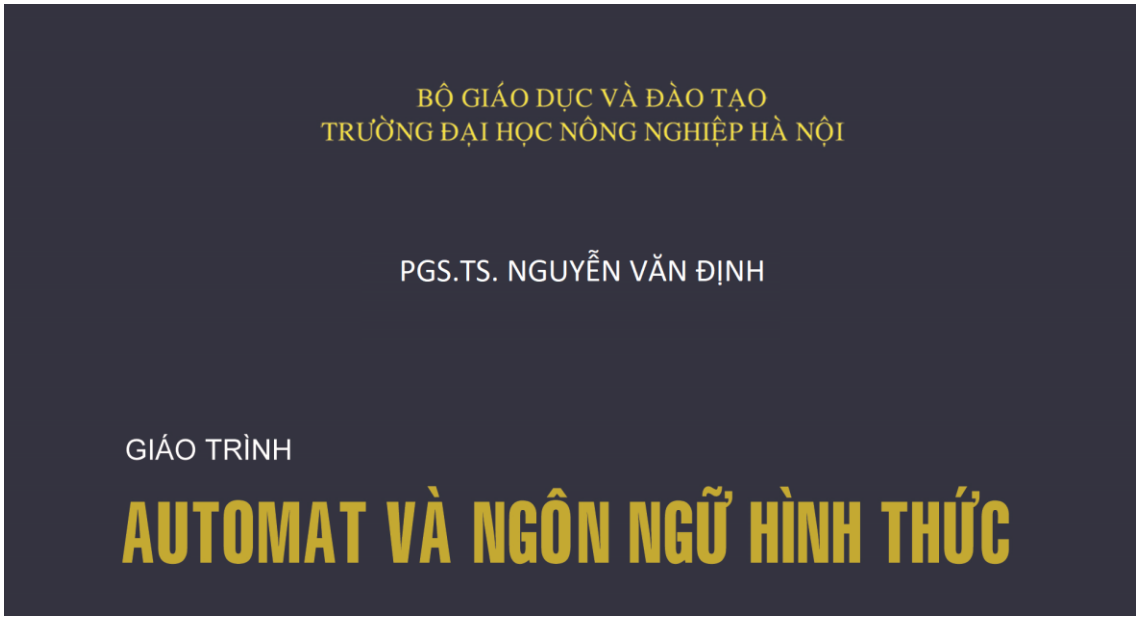


Chương 2

VĂN PHẠM VÀ NGÔN NGỮ HÌNH THỨC

Trong chương này, chúng ta đề cập đến một số khái niệm và kết quả cơ bản liên quan đến văn phạm và ngôn ngữ hình thức.

2.1. Các khái niệm cơ bản về ngôn ngữ hình thức

2.2. Các phép toán trên các từ

2.3. Các phép toán trên ngôn ngữ

2.4. Văn phạm và ngôn ngữ sinh bởi văn phạm

2.1. CÁC KHÁI NIỆM CƠ BẢN VỀ NGÔN NGỮ HÌNH THỨC

Mở đầu

Ngôn ngữ là công cụ để giao tiếp, có thể giữa con người với con người, giữa con người với máy tính hoặc máy tính với máy tính.

Ngôn ngữ trong giao tiếp thông thường giữa con người với con người gọi là ngôn ngữ tự nhiên, chẳng hạn tiếng Anh, tiếng Việt, tiếng Nga… là các ngôn ngữ tự nhiên. Ngôn ngữ tự nhiên thường dùng cả tiếng nói và chữ viết trong các giao tiếp. Ngôn ngữ tự nhiên rất phong phú và thuận tiện trong giao tiếp, nhưng có nhược điểm là cùng một thể hiện (câu viết hay nói) có thể có nhiều nghĩa khác nhau, và nói chung một thông điệp hay một câu trong ngôn ngữ tự nhiên phụ thuộc vào ngữ cảnh mà thông điệp được truyền đạt. Và có rất nhiều ngôn ngữ tự nhiên đồng thời tồn tại làm cho việc giao tiếp rất bất tiện, và hầu như không thể dùng ngôn ngữ tự nhiên cho việc giao tiếp giữa người với máy hoặc giữa máy với máy.

Một loại ngôn ngữ đặc biệt được xây dựng để giao tiếp giữa người với máy, hoặc giữa máy với máy, đó là ngôn ngữ hình thức. Một ngôn ngữ hình thức là một hệ thống các chuỗi ký hiệu, được thiết lập theo những quy tắc nhất định. Ngôn ngữ hình thức có đặc điểm là đơn nghĩa, và ý nghĩa của các thông điệp không phụ thuộc vào ngữ cảnh mà thông điệp đó được truyền tải. Các ngôn ngữ lập trình là những ngôn ngữ hình thức. Mọi ngôn ngữ hình thức đều có các thành phần cơ bản là tập các ký hiệu gọi là bảng chữ cái, và tập các chuỗi các ký hiệu, đó là các từ. Chúng ta sẽ bắt đầu nghiên cứu ngôn ngữ hình thức bằng việc nghiên cứu các khái niệm này.

2.1.1. Bảng chữ cái

Định nghĩa 2.1

Tập Σ khác rỗng gồm hữu hạn hay vô hạn các ký hiệu được gọi là *bảng chữ cái (alphabet)* Mỗi phần tử $a \in \Sigma$ được gọi là một *chữ cái* hay một *ký hiệu (symbol)*.

Thí dụ 2.1: Các tập hợp dưới đây là các bảng chữ cái:

1. Bảng chữ cái Latinh $\sum = \{a, b, c,\ldots, x, y, z\}$
2. Bảng chữ cái Hylạp $\Delta = \{\alpha, \beta, \gamma, \delta, \varepsilon, \eta, \varphi, \kappa, \mu, \chi, \nu, \pi, \theta, \rho, \sigma, \tau, \omega, \xi, \psi\}$
3. Bảng chữ số thập phân $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
4. Bảng chữ số nhị phân $\Gamma = \{0, 1\}$,
5. Bảng các ký hiệu tuỳ chọn $W = \{\text{if, then, else, a, b, c, d, e, f, +, –, *, /, =, } \neq\}$.

2.1.2. Từ

Định nghĩa 2.2

Giả sử có bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_m\}$, một chuỗi các chữ cái $\alpha = a_{i1}\, a_{i2} \ldots a_{it}$, với $a_{ij} \in \Sigma$ $(1 \le j \le t)$ được gọi là một xâu *(string)* hay một từ *(word)* trên bảng chữ cái Σ.

▪ Tổng số vị trí của các ký hiệu xuất hiện trong xâu α được gọi là độ dài của xâu α và ký hiệu là $|\alpha|$. Như vậy, một *từ* trên bảng chữ cái Σ là một xâu hữu hạn gồm một số lớn hơn hay bằng không các chữ cái của Σ, trong đó một chữ cái có thể xuất hiện nhiều lần.

▪ Xâu không có chữ cái nào được gọi là xâu *rỗng* và được ký hiệu là "ε". Rõ ràng xâu rỗng là từ thuộc mọi bảng chữ cái.

▪ Hai từ $\alpha = a_1 a_2 \ldots a_n$ và $\beta = b_1 b_2 \ldots b_m$ được gọi là bằng nhau, và được ký hiệu là $\alpha = \beta$, nếu $n = m$ và $a_i = b_i$ với mọi $i = 1, 2, \ldots, n$.

▪ Nếu α là một từ trên bảng chữ cái Σ, và $\Sigma \subseteq \Delta$ thì α cũng là từ trên bảng chữ cái Δ.

▪ Tập mọi xâu trên bảng chữ cái Σ được ký hiệu là Σ^*, tập mọi xâu khác rỗng được ký hiệu là Σ^+. Như vậy $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ và $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$. Dễ thấy rằng các tập Σ^* và Σ^+ là vô hạn.

Nói chung, *'từ'* hay *'xâu'* đều có ý nghĩa như nhau, nhưng ta thường dùng *'từ'* để chỉ rằng đó là một từ của một ngôn ngữ xác định.

Về cấu trúc đại số thì Σ^* là một vị nhóm tự do đối với phép nhân ghép sinh bởi Σ với đơn vị là từ rỗng ε, còn Σ^+ là một nửa nhóm tự do sinh bởi Σ. Có thể chứng minh được rằng các tập Σ^* và Σ^+ là vô hạn đếm được.

Thí dụ 2.2:

1. Ta có ε, 0, 01, 101, 1010, 110011 là các từ trên bảng chữ cái $\Gamma = \{0,1\}$
2. Các xâu ε, beautiful, happy, holiday là các từ trên bảng chữ cái $\Sigma = \{a, b, c, \ldots, z\}$.
3. Mỗi đại phân tử DNA được biểu diễn bằng một xâu gồm bốn ký hiệu A, C, G, T. Như vậy mỗi trình tự sinh học DNA là một từ trên bảng chữ cái {A, C, G, T}.

2.1.3. Ngôn ngữ

Định nghĩa 2.3

Cho bảng chữ cái Σ, mỗt tập con $L \subseteq \Sigma^*$ được gọi là một *ngôn ngữ hình thức (formal languages) (hay đơn giản là ngôn ngữ)* trên bảng chữ cái Σ.

▪ Tập rỗng, ký hiệu ∅, là một ngôn ngữ không gồm một từ nào và được gọi là *ngôn ngữ rỗng.* Vậy ngôn ngữ rỗng là ngôn ngữ trên mọi bảng chữ cái.

▪ Chú ý rằng ngôn ngữ rỗng: L = ∅ là khác với ngôn ngữ chỉ gồm một từ rỗng: L = {ε}.

Thí dụ 2.3:

1. Σ^* là ngôn ngữ gồm tất cả các từ trên Σ còn Σ^+ là ngôn ngữ gồm tất cả các từ khác từ rỗng trên Σ.
2. L = { ε, 0, 1, 01, 10, 00, 11, 011, 100} là một ngôn ngữ trên bảng chữ cái Γ = {0, 1}.
3. L = {a, b, c, aa, ab, ac, abc} là ngôn ngữ trên bảng chữ cái Σ = {a, b, c}.
4. L_1 = {ε, a, b, abb, aab, aaa, bbb, abab}, $L_2 = \{a^n b^n \mid n \in \mathbf{N}\}$ là hai ngôn ngữ trên bảng chữ Σ = {a, b}, L_1 là ngôn ngữ hữu hạn trong khi L_2 là ngôn ngữ vô hạn. Mỗi từ thuộc ngôn ngữ L_2 có số chữ cái a bằng số chữ cái b với a và b không xen kẽ, a nằm ở phía trái và b ở phía phải của từ.

2.2. CÁC PHÉP TOÁN TRÊN CÁC TỪ

Các phép toán dưới đây thực hiện trên các từ trên cùng một bảng chữ cái Σ, kết quả của các phép toán sẽ tạo nên các từ mới cũng thuộc cùng một bảng chữ cái.

2.2.1. Phép nhân ghép

Định nghĩa 2.4.

Tích ghép (hay nhân ghép) của hai từ $\alpha = a_1a_2\ldots a_m$ và từ $\beta = b_1b_2\ldots b_n$ trên bảng chữ cái Σ, là từ $\gamma = a_1a_2\ldots a_mb_1b_2\ldots b_n$ trên bảng chữ cái Σ.

Kí hiệu phép nhân ghép là γ = α.β (hay γ = αβ).

Nhận xét: Từ định nghĩa 2.1, ta thấy:

▪ Từ rỗng là phần tử đơn vị đối với phép nhân ghép, tức là với mọi từ ω, ta có: ωε = εω = ω.

▪ Phép nhân ghép có tính kết hợp, nghĩa là với mọi từ α, β, γ, ta có (αβ)γ = α(βγ).

▪ Ký hiệu ω^n, với n là số tự nhiên, được dùng theo nghĩa quen thuộc:

$$\omega^n = \begin{cases} \varepsilon \ khi \ n = 0, \\ \omega \ khi \ n = 1, \\ \omega^{n-1}\omega \ khi \ n > 1. \end{cases}$$

▪ Đối với phép nhân ghép thì hàm độ dài có một số tính chất hình thức của lôgarit: với mọi từ α, β và mọi số tự nhiên n, thì:

$|\alpha\beta| = |\alpha| + |\beta|$, và

$|\alpha^n| = n|\alpha|$.

Và rõ ràng là với phần tử đơn vị, tức là từ rỗng ε, thì $|\varepsilon| = 0$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Một vài khái niệm liên quan

▪ Đối với các từ ω, t_1, φ, t_2 trên bảng chữ cái Σ mà $\omega = t_1\varphi t_2$ thì $*\varphi*$ (* không phải là một ký hiệu của Σ) gọi là một *vị trí* của φ trên Σ.

▪ Xâu φ được gọi là xâu con (hay *từ con)* trong ω nếu tồn tại ít nhất một vị trí của φ trong ω.

▪ Nếu $t_1 = \varepsilon$, tức là $\omega = \varphi t_2$ thì φ được gọi là tiền tố (phần đầu) của từ ω, nếu $t_2 = \varepsilon$, tức là $\omega = t_1\varphi$ thì φ được gọi là hậu tố (phần cuối) của từ ω. Dễ thấy rằng từ rỗng ε là phần đầu, phần cuối và là từ con của một từ ω bất kỳ trên bảng chữ cái Σ.

▪ Trường hợp $|\varphi| = 1$, tức là φ chỉ gồm 1 ký hiệu, chẳng hạn $\varphi = b \in \Sigma$, thì $*b*$ được gọi là một vị trí của b trong từ ω, cũng gọi là một điểm trong ω.

▪ Số vị trí của kí hiệu a trong từ ω được ký hiệu là $I_a(\omega)$, hay $|\omega|_a$ hoặc đơn giản hơn là $\omega|_a$.

Thí dụ 2.4:

1. Trên bảng chữ cái W = {if, then, else, a, b, c, d, e, f, +, –, ∗, /, =, ≠}, ta có các từ α là: if a+b=c then c∗d=e và từ β là: else c/d=f , còn $\alpha\beta$ là từ: if a+b=c then c∗d=e else c/d=f.
2. Cho Σ = {a, b, c}, khi đó: Từ ω = abcbcb chứa 2 vị trí của bcb, đó là a*bcb*cb và abc*bcb*, φ = bcb là một từ con của ω. Từ ω chứa một vị trí của ký hiệu a, đó là *a*bcbcb.
3. Từ ω = 010111001 trên bảng chữ cái {0, 1} có độ dài 9, trong đó 0101 là tiền tố và 11001 là hậu tố của ω.

2.2.2. Phép lấy từ ngược

Định nghĩa 2.5. Giả sử có từ khác rỗng $\omega = a_1a_2\ldots a_m$ trên bảng chữ cái Σ, khi đó từ $a_m a_{m-1}\ldots a_2 a_1$ được gọi là từ ngược (hay từ soi gương) của từ ω, và được ký hiệu là ω^R, hay $\omega^\wedge$.

Khi $\omega = \varepsilon$ ta quy ước $\varepsilon^R = \varepsilon$.

Nhận xét: Dễ thấy rằng phép lấy từ ngược có các tính chất sau:

- $(\omega^R)^R = \omega$.
- $(\alpha\beta)^R = \beta^R \alpha^R$
- $| \alpha^R | = | \alpha |$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Thí dụ 2.5:

1. Cho các từ $\alpha = 100110$ và $\beta = aabb$ trên bảng chữ cái {0,1,a,b}, theo định nghĩa ta có:

   $\alpha^R = 011001$ và $(\alpha^R)^R = (011001)^R = 100110 = \alpha$.

   $\beta^R = bbaa$ và $(\beta^R)^R = (bbaa)^R = aabb = \beta$.

2. Cho các từ happy và oto trên bảng chữ cái $\sum$ = {a, b, c,… x, y, z}, khi đó ta có:

   $(happy)^R$ = yppah và $(oto)^R$ = oto.

   Ngoài ra ta có: $(happy)^R$ | = | yppah| = | happy | = 5.

2.2.3. Phép chia từ

Là phép toán ngắt bỏ phần đầu hay phần cuối của một từ. Ta có các định nghĩa sau:

Định nghĩa 2.6

Phép chia trái của từ α cho từ β (hay thương bên trái của α và β) cho kết quả là phần còn lại của từ α sau khi ngắt bỏ phần đầu β trong từ α, và được ký hiệu là ${}_\beta\backslash^\alpha$.

Định nghĩa 2.7

Phép chia phải của từ α cho từ γ (hay thương bên phải của α và γ) cho kết quả là phần còn lại của từ α sau khi ngắt bỏ phần cuối γ trong từ α, và được ký hiệu là ${}^\alpha/_\gamma$

Nhận xét: Dễ thấy rằng các phép chia từ có tính chất sau:

- Trong phép chia trái của từ α cho từ β thì β phải là tiền tố của từ α, tương tự, trong phép chia phải từ α cho từ γ thì γ phải là hậu tố của từ α.
- ${}_\varepsilon\backslash^\alpha = {}^\alpha/_\varepsilon = \alpha$.
- ${}_\alpha\backslash^\alpha = {}^\alpha/_\alpha = \varepsilon$.
- Nếu $\alpha = \beta.\gamma$ thì ${}_\beta\backslash^\alpha = \gamma$, còn ${}^\alpha/_\gamma = \beta$

- $(\beta \backslash^{\alpha})^R = \alpha^R / \beta^R$.
- $({}^{\alpha}/_{\gamma})^R = \gamma^R \setminus \alpha^R$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Thí dụ 2.6:

Cho các từ α = abcaabbcc, β = abc, γ = bcc trên bảng chữ cái ∑ = {a, b, c}, khi đó ta có

1. $\beta \backslash^{\alpha}$ = aabbcc và ${}^{\alpha}/\gamma$ = abcaab.
2. $(\beta \backslash^{\alpha})^R = (aabbcc)^R = ccbbaa = {}^{ccbbaacba}/cba = \alpha^R / \beta^R$

2.3. CÁC PHÉP TOÁN TRÊN NGÔN NGỮ.

Các họ ngôn ngữ cụ thể, thường được đặc trưng một cách tiện lợi qua các phép toán xác định trên ngôn ngữ. Họ đó gồm các ngôn ngữ nhận được bằng việc tổ hợp từ một số ngôn ngữ cho trước bởi một số phép toán nào đó.

Vì mỗi ngôn ngữ là một tập hợp nên các phép toán trên ngôn ngữ bao gồm các phép toán đại số tập hợp như là phép hợp, phép giao, phép hiệu, phép lấy bù trên các ngôn ngữ. Chẳng hạn, với L_1 và L_2 là hai ngôn ngữ trên bảng chữ cái Σ thì ta cũng có các ngôn ngữ mới trên bảng chữ cái Σ dựa trên các phép toán tập hợp, đó là các ngôn ngữ: $L_1 \cup L_2$, $L_1 \cap L_2$, $\Sigma^* \setminus L_1$, $\Sigma^* \setminus L_2$.

Ngoài ra, đối với lớp các ngôn ngữ còn có các phép toán đặc biệt trên các các tập từ, như phép tích ghép, phép lặp, phép chia ngôn ngữ.

Dưới đây chúng ta sẽ trình bày các phép toán trên ngôn ngữ

2.3.1. Phép hợp

Định nghĩa 2.8

Hợp của hai ngôn ngữ L_1 và L_2 trên bảng chữ cái ∑, ký hiệu $L_1 \cup L_2$, là một ngôn ngữ trên bảng chũ cái ∑, đó là tập từ:

$$L = \{\omega \in \Sigma^* \mid \omega \in L_1 \text{ hoặc } \omega \in L_2\}$$

Định nghĩa phép hợp có thể mở rộng cho một số hữu hạn các ngôn ngữ, tức là hợp của các ngôn ngữ $L_1, L_2, \ldots, L_n$ trên bảng chữ cái Σ, là tập từ:

$$\bigcup_{i=1}^{n} L_i = \{\omega \in \Sigma^* \mid \omega \in L_i, \text{ với i nào đó, } 1 \le i \le n\}$$

Nhận xét: Dễ dàng thấy rằng phép hợp các ngôn ngữ có các tính chất sau:

▪ Phép hợp hai ngôn ngữ có tính giao hoán: $L_1 \cup L_2 = L_2 \cup L_1$.

▪ Phép hợp các ngôn ngữ có tính kết hợp: $(L_1 \cup L_2) \cup L_3 = L_1 \cup (L_2 \cup L_3)$.

▪ Với mọi ngôn ngữ L trên Σ thì: $L \cup \varnothing = \varnothing \cup L = L$ và $L \cup \Sigma^* = \Sigma^*$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

2.3.2. Phép giao

Định nghĩa 2.9

Giao của hai ngôn ngữ L_1 và L_2 trên bảng chữ cái ∑, ký hiệu $L_1 \cap L_2$, là một ngôn ngữ trên bảng chữ cái ∑, đó là tập từ:

$$L = \{\omega \in \Sigma^* \mid \omega \in L_1 \text{ và } \omega \in L_2\}$$

Định nghĩa phép giao có thể mở rộng cho một số hữu hạn các ngôn ngữ, tức là giao của các ngôn ngữ $L_1, L_2, \ldots, L_n$ trên bảng chữ cái Σ, là tập từ:

$$\bigcap_{i=1}^{n} L_i = \{\omega \in \Sigma^* \mid \omega \in L_i, \text{ với mọi } i, 1 \le i \le n\}$$

Nhận xét: Dễ dàng thấy rằng, phép giao các ngôn ngữ có tính chất sau:

▪ Phép giao hai ngôn ngữ có tính giao hoán: $L_1 \cap L_2 = L_2 \cap L_1$.

▪ Phép giao các ngôn ngữ có tính kết hợp: $(L_1 \cap L_2) \cap L_3 = L_1 \cap (L_2 \cap L_3)$.

▪ Phép giao các ngôn ngữ có tính phân phối đối với phép hợp:

▪ $(L_1 \cap L_2) \cup L_3 = (L_1 \cup L_3) \cap (L_2 \cup L_3)$.

▪ $(L_1 \cup L_2) \cap L_3 = (L_1 \cap L_3) \cup (L_2 \cap L_3)$.

▪ Với mọi ngôn ngữ L trên Σ thì: $L \cap \varnothing = \varnothing \cap L = \varnothing$ và $L \cap \Sigma^* = L$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

2.3.3. Phép lấy phần bù

Định nghĩa 2.10

Ngôn ngữ phần bù của ngôn ngữ L trên bảng chữ cái Σ, ký hiệu $C_\Sigma L$ (hay đơn giản là CL, hoặc $\overline{L}$ nếu không gây nhầm lẫn), là một ngôn ngữ trên bảng chữ cái ∑, đó là tập từ:

$$C_\Sigma L = \{\omega \in \Sigma^* \mid \omega \notin L\}$$

Nhận xét: Dễ dàng thấy rằng phép lấy phần bù các ngôn ngữ có các tính chất sau:

▪ $C_\Sigma\{\varepsilon\} = \Sigma^+$, $C_\Sigma\,\Sigma^+ = \{\varepsilon\}$.

▪ $C_\Sigma\,\varnothing = \Sigma^*$, $C_\Sigma\,\Sigma^* = \varnothing$.

▪ $C(CL_1 \cup CL_2) = L_1 \cap L_2$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Thí dụ 2.7:

1. Cho ngôn ngữ $L_1 = \{\varepsilon, 0, 01\}$, $L_2 = \{\varepsilon, 01, 10\}$ trên bảng chữ cái $\Sigma = \{0, 1\}$, khi đó ta có:

$L_1 \cup L_2 = \{\varepsilon, 0, 01, 10\}$,

$L_1 \cap L_2 = \{\varepsilon, 01\}$.

2. Cho ngôn ngữ $L = \{\omega \in \sum^*, \text{với } |\,\omega\,| \text{ là một số chẵn}\}$, khi đó ta có:

$C_\Sigma L = \{\omega \in \sum^+, \text{với } |\,\omega\,| \text{ là một số lẻ}\}$.

2.3.4. Phép nhân ghép

Định nghĩa 2.11

Cho hai ngôn ngữ L_1 trên bảng chữ Σ_1 và L_2 trên bảng chữ Σ_2. Nhân ghép hay tích của hai ngôn ngữ L_1 và L_2 là một ngôn ngữ trên bảng chữ $\Sigma_1 \cup \Sigma_2$, ký hiệu L_1L_2, được xác định bởi tập từ sau:

$$L_1L_2 = \{\alpha\beta \mid \alpha \in L_1 \text{ và } \beta \in L_2\}.$$

Nhận xét: Dễ dàng nhận thấy phép nhân ghép (tích) các ngôn ngữ có các tính chất sau:

▪ Phép nhân ghép có tính kết hợp: với mọi ngôn ngữ L_1, L_2 và L_3, ta có:

▪ $(L_1L_2)L_3 = L_1(L_2L_3)$.

▪ $\varnothing L = L\varnothing = \varnothing$, $\{\varepsilon\}L = L\{\varepsilon\} = L$,

▪ Phép nhân ghép có tính phân phối đối với phép hợp, nghĩa là

▪ $L_1(L_2 \cup L_3) = L_1L_2 \cup L_1L_3$, $(L_2 \cup L_3)L_1 = L_2L_1 \cup L_3L_1$.

▪ Đặc biệt: Phép nhân ghép không có tính phân phối đối với phép giao. Phép hợp, phép giao không có tính phân phối đối với phép nhân ghép (xem thí dụ 2.2). Tức là với mọi ngôn ngữ L_1, L_2 và L_3, thì:

▪ $L_1(L_2 \cap L_3) \neq (L_1L_2) \cap (L_1L_3)$ và

$L_1 \cup (L_2L_3) \neq (L_1 \cup L_2)(L_1 \cup L_3)$,

$L_1 \cap (L_2L_3) \neq (L_1 \cap L_2)(L_1 \cap L_3)$.

Thí dụ 2.8: Đây là một phản ví dụ để chỉ ra rằng phép nhân ghép không có tính phân phối đối với phép giao. Phép hợp, phép giao không có tính phân phối đối với phép nhân ghép.

Xét các ngôn ngữ $L_1 = \{0, 01\}$, $L_2 = \{01, 10\}$, $L_3 = \{0\}$ trên bảng chữ cái $\Sigma = \{0, 1\}$.

1. Có thể kiểm tra được rằng phép nhân ghép không có tính phân phối đối với phép giao:

Ta có: $L_2 \cap L_3 = \varnothing$, do đó:

$L_1(L_2 \cap L_3) = \varnothing$,

Mặt khác, ta có $L_1L_2 = \{001, 010, 0101, 0110\}$ và $L_1L_3 = \{00, 010\}$, do đó:

$(L_1L_2) \cap (L_1L_3) = \{010\}$.

Vậy $L_1(L_2 \cap L_3) \neq (L_1L_2) \cap (L_1L_3)$, tức là phép nhân ghép không có tính phân phối đối với phép giao.

2. Kiểm tra tính phân phối của phép hợp, phép giao đối với phép nhân ghép:

Ta có: $L_2L_3 = \{010, 100\}$, do đó:

$L_1 \cup (L_2L_3) = \{0, 01, 010, 100\}$,

Mặt khác ta cũng có $L_1 \cup L_2 = \{0, 01, 10\}$ và $L_1 \cup L_3 = \{0, 01\}$, do đó:

$(L_1 \cup L_2)(L_1 \cup L_3) = \{00, 001, 010, 0101, 100, 1001\}$.

Vậy $L_1 \cup (L_2L_3) \neq (L_1 \cup L_2)(L_1 \cup L_3)$, tức là phép hợp không có tính phân phối đối với phép nhân ghép.

Tương tự, đối với phép giao, ta có:

$L_2L_3 = \{010, 100\}$, do đó:

$L_1 \cap (L_2L_3) = \varnothing$.

Mặt khác $L_1 \cap L_2 = \{01\}$, $L_1 \cap L_3 = \{0\}$, do đó:

$(L_1 \cap L_2)(L_1 \cap L_3) = \{010\}$.

Vậy $L_1 \cap (L_2L_3) \neq (L_1 \cap L_2)(L_1 \cap L_3)$. Phép giao không phân phối với phép nhân ghép.

Vì phép nhân ghép ngôn ngữ có tính kết hợp nên ký hiệu L^n được dùng với mọi ngôn ngữ L và số tự nhiên n theo nghĩa quen thuộc sau:

$$L^n = \begin{cases} \{\varepsilon\} \text{ khi } n = 0, \\ L \quad \text{khi } n = 1, \\ L^{n-1}L \text{ khi } n > 1. \end{cases}$$

2.3.5. Phép lặp

Định nghĩa 2.12

Cho ngôn ngữ L trên bảng chữ cái Σ, khi đó:

- Tập từ $\{\varepsilon\} \cup L \cup L^2 \cup \ldots \cup L^n \cup \ldots = \bigcup_{n=0}^{\infty} L^n$ được gọi là *ngôn ngữ lặp* của ngôn ngữ L (hay *bao đóng ghép* của ngôn ngữ *L)*, ký hiệu L^*.

  Vậy *ngôn ngữ lặp* của L là hợp của mọi luỹ thừa của *L*: $L^* = \bigcup_{n=0}^{\infty} L^n$.

- Tập từ $L \cup L^2 \cup \ldots \cup L^n \cup \ldots = \bigcup_{n=1}^{\infty} L^n$ được gọi là *ngôn ngữ lặp cắt* của ngôn ngữ L, (hay bao đóng dương của ngôn ngữ L), ký hiệu L^+,

  Vậy *ngôn ngữ lặp cắt* của L là hợp của mọi luỹ thừa dương của *L*: $L^+ = \bigcup_{n=1}^{\infty} L^n$.

Thí dụ 2.9: Thực hiện một số phép toán trên các ngôn ngữ.

1. Xét ngôn ngữ L = {0, 1} trên bảng chữ Σ = {0, 1}. Ta có:

   L^2 = {00, 01, 10, 11}, tập hợp các xâu nhị phân độ dài 2;

   L^3 = {000, 001, 010, 011, 100, 101, 110, 111}, tập hợp các xâu nhị phân độ dài 3.

   Tương tự, L^n là tập hợp các xâu nhị phân độ dài n.

   Vì vậy, L^* là tập hợp tất cả các xâu nhị phân.

2. Xét hai ngôn ngữ trên bảng chữ Σ = {a}:

   $L_1 = \{a^{2n} \mid n \geq 1\}$,

   $L_2 = \{a^{5n+3} \mid n \geq 0\}$.

   Khi đó, ta có $L_1 = \{a^2\}^+$, $L_2 = \{a^5\}^*\{a^3\}$.

2.3.6. Phép lấy ngôn ngữ ngược

Định nghĩa 2.13

Cho ngôn ngữ L trên bảng chữ cái Σ, khi đó ngôn ngữ ngược của L là một ngôn ngữ trên bảng chữ cái ∑, được ký hiệu là L^R hay $L^{\wedge}$, là tập từ:

$$L^R = \{\omega \in \Sigma^* / \omega^R \in L\}$$

Nhận xét: Dễ dàng thấy rằng phép lấy ngôn ngữ ngược có các tính chất sau:

- $(L^R)^R = L$.
- $\{\varepsilon\}^R = \{\varepsilon\}$.
- $(\varnothing)^R = \varnothing$.

Chứng minh các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Thí dụ 2.10: Cho L = {ε, ab, abc, cbaa} là một ngôn ngữ trên bảng chữ cái Σ = {a, b, c}, khi đó L^R = {ε, ba, cba, aabc} là ngôn ngữ ngược của L.

2.3.7. Phép chia ngôn ngữ

Định nghĩa 2.14

Cho ngôn ngữ X và Y trên bảng chữ cái Σ, khi đó thương bên trái của ngôn ngữ X cho ngôn ngữ Y (Y khác ngôn ngữ ∅) là một ngôn ngữ trên bảng chữ cái ∑, được ký hiệu là ${}_Y \backslash {}^X$, là tập từ:

$${}_Y \backslash {}^X = \{z \in \Sigma^* / x \in X, y \in Y \text{ mà } x = yz\}$$

Định nghĩa 2.15

Cho ngôn ngữ X và Y trên bảng chữ cái Σ, khi đó thương bên phải của ngôn ngữ X cho ngôn ngữ Y là một ngôn ngữ trên ∑, được ký hiệu là ${}^X /_Y$, là tập từ:

$${}^X / {}_Y = \{z \in \Sigma^* / x \in X, y \in Y \text{ mà } x = zy\}$$

Nhận xét: Dễ dàng thấy rằng phép chia ngôn ngữ có các tính chất sau:

- ${}_{\{\varepsilon\}}\backslash {}^L = {}^L/_{\{\varepsilon\}} = L$
- ${}_L\backslash {}^{\Sigma^*} = {}^{\Sigma^*}/_L = \Sigma^*$
- ${}_L\backslash {}^{\Sigma^+} = {}^{\Sigma^+}/_L = \Sigma^+$
- $({}_Y \backslash {}^X)^R = X^R / Y^R$, $({}^X / {}_Y)^R = Y^R \backslash X^R$.

Việc kiểm tra các kết quả trên là khá dễ dàng, xin dành cho sinh viên như là bài tập.

Thí dụ 2.11: Cho X = {a, b, abc, cab, bcaa} và Y = {ε, c, ab} là các ngôn ngữ trên bảng chữ cái Σ = {a, b, c}, khi đó:

1. $_{Y}\backslash^{X}$ = {a, b, abc, cab, bcaa, ab, c}
2. $^{X}/_{Y}$ = {a, b, abc, cab, bcaa, ab, c}
3. $_{X}\backslash^{Y}$ = {b}
4. $^{Y}/_{X}$ = {a}
5. $_{X}\backslash^{X}$ = {ε , bc, caa}
6. $_{Y}\backslash^{Y}$ = {ε, c, ab}
7. Kiểm tra tính chất $(_{Y}\backslash^{X})^{R} = X^{R} / Y^{R}$:

Từ phần 1 của thí dụ này, ta có $(_{Y}\backslash^{X})^{R}$ = {a, b, abc, cab, bcaa, ab, c}^{R} = {a, b, cba, bac, aacb, ba, c}

Mặt khác, ta có: X^{R} = {a, b, cba, bac, aacb}và Y^{R} = {ε, c, ba},

do đó X^{R}/ Y^{R} = {a, b, cba, bac, aacb, ba, c}= $(_{Y}\backslash^{X})^{R}$.

2.4. VĂN PHẠM VÀ NGÔN NGỮ SINH BỞI VĂN PHẠM

2.4.1. Biểu diễn ngôn ngữ

Theo định nghĩa 2.3, ngôn ngữ là một tập từ trên bảng chữ cái Σ cho trước, hay cụ thể hơn, ngôn ngữ L trên bảng chữ cái Σ là tập con của tập các xâu trên bảng chữ cái Σ, hay L ⊂ Σ*. Như vậy, vấn đề đặt ra là đối với một ngôn ngữ L, có thể xác định được các xâu ω∈ Σ* có thuộc về ngôn ngữ L hay không, hay tổng quát hơn, làm thế nào để biểu diễn một ngôn ngữ?

Đối với các ngôn ngữ hữu hạn, để biểu diễn chúng, ta chỉ cần liệt kê tất cả các từ của chúng, như là liệt kê các phần tử của một tập hợp.

Chẳng hạn: L_1 = {ε}

L_2 = {a, aa, b, bb},

L_3 = {ε, ab, aabb, aaabbb},

là các ngôn ngữ trên bảng chữ cái Σ = {a, b}

Tuy nhiên, khi các ngôn ngữ là vô hạn, ta không thể liệt kê tất cả các từ của nó được, mà phải tìm các phương pháp hiệu quả và ngắn gọn hơn, tức là tìm cách biểu diễn

hữu hạn cho các ngôn ngữ vô hạn. Trong những trường hợp không quá phức tạp, ta có thể xác định ngôn ngữ vô hạn bằng việc chỉ ra đặc điểm chung của các từ của nó, được mô tả bằng một điều kiện nào đó.

Chẳng hạn: Nếu có ngôn ngữ L, mà các từ có dạng giống như của L_3 trên đây, nhưng số từ là vô hạn, tức là L = {ε, ab, aabb, aaabbb,… a^nb^n} thì ta có thể biểu diễn L như sau:

$L_4 = \{a^nb^n \mid n \geq 0\}$, hay là:

$L_5 = \{ \omega \in \{a, b\}^* \mid$ các ký hiệu a đứng trước và bằng số ký hiệu b$\}$

Cả L_4 và L_5 trên đây đều là các cách biểu diễn hữu hạn cho ngôn ngữ vô hạn L.

Mặt khác, ngôn ngữ L trên đây còn có thể được định nghĩa đệ quy như sau:

(i). $\varepsilon \in L$,

(ii). Nếu $\omega \in L$ thì $a\omega b \in L$,

(iii). Không còn từ nào khác thuộc L.

Định nghĩa đệ quy nói trên cho ta một cách sản sinh ra các từ của L như sau: Do (i) nên ta có từ đầu tiên trong L là ε, tiếp theo, áp dụng (ii) ta có từ thứ hai là aεb hay ab, và tương tự sẽ có các từ aabb, aaabbb… như vậy bằng các áp dụng hữu hạn lần quy tắc (ii) ta sẽ sản sinh được mọi từ của L, điều kiện (iii) chỉ ra rằng ngôn ngữ xác định như định nghĩa trên, sẽ không chứa từ nào khác, tức là ngôn ngữ đó chính là L.

Nói chung, người ta cố gắng tìm cách biểu diễn hữu hạn cho một ngôn ngữ. Tuy nhiên, trước khi có thể biểu diễn một ngôn ngữ, ta cần xác định ngôn ngữ đó, tức là nhận diện được các từ của ngôn ngữ. Việc xác định một ngôn ngữ trên bảng chữ cái cho trước có thể được thực hiện bằng một trong các cách thức sau:

Cách 1. Đối với mỗi từ thuộc ngôn ngữ đã cho, ta có thể chọn một quy cách hoạt động của "thiết bị tự động" để sau một số hữu hạn bước làm việc nó dừng và sinh ra chính từ đó. "Thiết bị tự động" này có khả năng lần lượt sinh ra tất cả các từ trong ngôn ngữ đã cho.

Cách 2. Với mỗi từ ω cho trước, "thiết bị tự động" có thể cho biết từ đó có thuộc ngôn ngữ đã cho hay không.

Trong lý thuyết ngôn ngữ hình thức, người ta đã chứng minh được rằng các cách thức trên là tương đương nhau. Trong chương này, ta quan tâm đến cách thứ nhất, tức là ta xét văn phạm như là một "thiết bị tự động" sinh ra các từ. Vì lẽ đó mà người ta còn gọi các "thiết bị tự động" loại này là các văn phạm sinh. Chẳng hạn, dựa vào định nghĩa

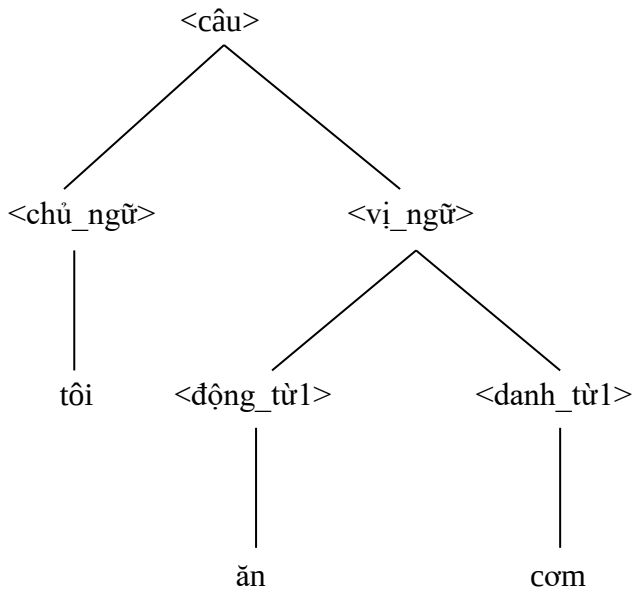
đệ quy cho ngôn ngữ L trên đây, ta có thể xây dựng một "thiết bị tự động" thực hiện theo các bước của định nghĩa trên, để sinh ra mọi từ của ngôn ngữ L nói trên, thiết bị tự động theo kiểu này chính là các văn phạm mà ta sẽ nghiên cứu kỹ lưỡng hơn trong các mục dưới đây.

Với cách thứ hai, tức là việc xác định một ngôn ngữ bằng cách đoán nhận một xâu có thuộc ngôn ngữ đã cho hay không, thì "thiết bị tự động" ở đây là các otomat hữu hạn, hay các máy hình thức như các máy Turing, mà ta sẽ xét đến trong các chương sau.

2.4.2. Văn phạm

Trước khi xem xét định nghĩa của văn phạm hình thức, ta hiểu văn phạm theo nghĩa thông thường là các cú pháp đối với từng ngôn ngữ tự nhiên, đó là tập các quy tắc cấu tạo từ, và quy tắc liên kết các từ lại thành một câu.

Chẳng hạn, ta xét một câu đơn trong ngôn ngữ tiếng Việt: 'tôi ăn cơm', câu này có cây phân tích cú pháp như trong hình 2.1 dưới đây:



Hình 2.1. Cây phân tích cú pháp trong văn phạm tiếng Việt.

▪ Xuất phát từ nút gốc: <câu>, sẽ sinh ra hai nút con là các thành phần <chủ_ngữ> và <vị_ngữ>,

▪ Mỗi nút con <chủ_ngữ> và <vị_ngữ>, lại sinh ra các thành phần 'tôi' và 'ăn cơm'.

Hai thành phần cuối cùng này là các lá, nó không phải sinh ra thêm thành phần nào khác, cho nên nó được coi như các *ký hiệu kết thúc* của hệ thống văn phạm, còn các thành phần <câu> và <chủ ngữ>, <vị ngữ> đóng vai trò là các *ký hiệu không kết thúc* vì

nó còn phải sinh ra các thành phần khác. Chuỗi kết quả của cây cú pháp dưới đây là gồm một chuỗi toàn các *ký hiệu kết thúc*, được phát sinh từ một nút khởi đầu là một ký hiệu không kết thúc đặc biệt (*start-S*) được gọi là gốc của cây phân tích cú pháp, qua các nút trung gian có chứa các ký hiệu không kết thúc. Đối với văn phạm hình thức, thì chuỗi kết quả (*gồm toàn các ký hiệu kết thúc)* được coi là một từ sinh bởi văn phạm đã cho. (hình 2.1).

Tương tự như vây, văn phạm hình thức là một hệ thống sinh các từ của một ngôn ngữ, được định nghĩa như sau:

Định nghĩa 2.16. *Văn phạm* (*Grammar*) G là một bộ sắp thứ tự gồm 4 thành phần:

$$\mathbf{G = < \Sigma, \Delta, S, P >}$$

trong đó:

▪ Σ là một bảng chữ cái, gọi là *bảng chữ cái chính (*hay *bảng chữ cái kết thúc)*, mỗi phần tử của nó được gọi là một *ký hiệu chính* hay *ký hiệu kết thúc (terminal),*

▪ Δ là một bảng chữ cái, $\Delta \cap \Sigma = \varnothing$, gọi là *bảng ký hiệu phụ (hay báng chữ cái không kết thúc)*, mỗi phần tử của nó được gọi là một *ký hiệu phụ* hay *ký hiệu không kết thúc* (*non terminal),*

▪ $S \in \Delta$ được gọi là *tiên đề* hay *ký hiệu xuất phát (start),*

▪ P là tập hợp các *quy tắc sinh* (*production*) có dạng $\alpha \rightarrow \beta$, với $\alpha, \beta \in (\Sigma \cup \Delta)^*$; α được gọi là *vế trái* và β được gọi là *vế phải* của quy tắc này, và *trong α phải chứa ít nhất một ký hiệu phụ.* Như vậy, các quy tắc hợp lệ của của P có dạng:

$\alpha \rightarrow \beta$ với $\alpha = \alpha' A \alpha''$, trong đó $A \in \Delta$, α', α'', $\beta \in (\Sigma \cup \Delta)^*$

Chẳng hạn, với $\Sigma = \{0,1\}$, $\Delta = \{S, A, B\}$ thì các quy tắc $S \rightarrow 0S1A$, $0AB \rightarrow 1A1B$, $A \rightarrow \varepsilon$,… là các quy tắc hợp lệ vì vế trái luôn chứa ít nhất 1 ký hiệu phụ thuộc Δ, nhưng các quy tắc dạng: $0 \rightarrow A$, $01 \rightarrow 0B$,… là các quy tắc không hợp lệ.

Thí dụ 2.12: Các bộ bốn sau là các văn phạm:

1. $G_1 = <\{0, 1\}, \{S\}, S, \{S \rightarrow 0S1, S \rightarrow \varepsilon\}>$, ở đây, tập quy tắc $P_1 = \{S \rightarrow 0S1, S \rightarrow \varepsilon\}$, có thể được

viết tường minh, bao gồm cả số hiệu các quy tắc, dưới dạng:

P_1: | $S \rightarrow 0S1$, (1)

| $S \rightarrow \varepsilon$. (2)

2. G_2 = <{a, b}, {S, A}, S, P_2 >, với tập quy tắc:

P_2:
- S→ Ab, (1)
- A→ aAb, (2)
- A→ ε. (3)

3. G_3 = <{a, b, c}, {S, A, B, C}, S, P_3>, với tập quy tắc P_3

P_3:
- S→ ABC, (1)
- A→ aA, (2)
- B→ bB, (3)
- C→ cC, (4)
- A→ a, (5)
- B→ b, (6)
- C→ c. (7)

4. G_4 = <Σ, Δ, S, P_4>, trong đó:

Σ = {tôi, anh, chị, ăn, uống, cơm, phở, sữa, café},

Δ={<câu>,<chủ_ngữ>,<vị_ngữ>,<động_từ1>,<động_từ2>,<danh_từ1>,<danh_từ2>}

S = <câu>; với tập quy tắc P_4:

P_4 :
- <câu>→ <chủ_ngữ> <vị_ngữ>, (1)
- <chủ_ngữ>→ tôi, (2)
- <chủ_ngữ>→ anh, (3)
- <chủ_ngữ>→ chị, (4)
- <vị_ngữ>→ <động_từ1> <danh_từ1>, (5)
- <vị_ngữ>→ <động_từ2> <danh_từ2>, (6)
- <động_từ1>→ ăn, (7)
- <động_từ2>→ uống, (8)
- <danh_từ1>→ cơm, (9)
- <danh_từ1>→ phở, (10)
- <danh_từ2>→ sữa, (11)
- <danh_từ2>→ café. (12)

Chú ý: 1. Nếu các quy tắc có vế trái giống nhau có thể viết gọn lại, chẳng hạn hai quy tắc: $\alpha \rightarrow \beta$, $\alpha \rightarrow \gamma$ có thể được viết là $\alpha \rightarrow \beta \mid \gamma$. Chẳng hạn, như trong văn phạm G_1 ở thí dụ trên, ta có thể viết hai quy tắc của nó dưới dạng $S \rightarrow 0S1 \mid \varepsilon$.

2. Người ta thường dùng các chữ cái in hoa: S, A, B, C để chỉ các ký hiệu phụ (kí hiệu không kết thúc), đặc biệt S dùng để chỉ kí hiệu xuất phát, và các chữ cái in thường: a, b, c… để chỉ các ký hiệu chính (kí hiệu kết thúc). Vì vậy, trong nhiều trường hợp, khi cho một văn phạm, ta chỉ cần cho tập quy tắc là đủ để xác định được tất cả các thành phần của văn phạm. Chẳng hạn, trong thí dụ trên, có thể cho văn phạm G_2 bởi tập quy tắc:

$$P_2: \begin{cases} S \rightarrow Ab, & (1) \\ A \rightarrow aAb, & (2) \\ A \rightarrow \varepsilon. & (3) \end{cases}$$

2.4.3. Ngôn ngữ sinh bởi văn phạm

Định nghĩa 2.17

Cho văn phạm $G = <\Sigma, \Delta, S, P>$, nếu có quy tắc $\alpha \rightarrow \beta \in P$ thì ta nói xâu $\eta = \gamma\alpha\delta$ *suy dẫn trực tiếp ra xâu* $\omega = \gamma\beta\delta$ (với $\gamma, \delta \in (\Sigma \cup \Delta)^*$), và ký hiệu là $\eta \vdash^G \omega$ hay ngắn gọn là $\eta \vdash \omega$ (nếu suy dẫn được hiểu là trong văn phạm G)

Điều này có nghĩa là áp dụng quy tắc sinh $\alpha \rightarrow \beta$ vào xâu $\eta = \gamma\alpha\delta$ thì được xâu $\omega = \gamma\beta\delta$

Định nghĩa 2.18

Cho văn phạm $G = <\Sigma, \Delta, S, P>$, nếu có các xâu $\omega_0, \omega_1, \ldots \omega_n \in (\Sigma \cup \Delta)^*$, sao cho $\omega_0 \vdash \omega_1$, $\omega_1 \vdash \omega_2, \ldots, \omega_{n-1} \vdash \omega_n$ thì ta nói ω_n được *suy dẫn gián tiếp* (hay *suy dẫn*) từ ω_0 trong G, ký hiệu $\omega_0 \models^G \omega_n$ hay ngắn gọn là $\omega_0 \models \omega_n$ (nếu các suy dẫn được hiểu là trong văn phạm G).

▪ Dãy $D = \omega_0, \omega_1, \ldots, \omega_n$ được gọi là một *dẫn xuất* của ω_n từ ω_0 trong G và số n được gọi là *độ dài* của dẫn xuất này. Nếu $\omega_0 = S$ và $\omega_n \in \Sigma^*$ thì dãy D gọi là *dẫn xuất đầy đủ*.

▪ Như vậy, $\omega_0 \models \omega_n$ khi và chỉ khi tồn tại dãy các suy dẫn trực tiếp: $\omega_0 \vdash \omega_1 \vdash \ldots \vdash \omega_n = \omega$.

▪ Nếu ω_i được suy dẫn trực tiếp từ ω_{i-1} bằng việc áp dụng một quy tắc p nào đó trong G thì ta nói quy tắc p được áp dụng *ở bước thứ i*.

Định nghĩa 2.19

Cho văn phạm G = <Σ, Δ, S, P>. Từ ω∈ Σ* được gọi là *sinh bởi* văn phạm G nếu tồn tại suy dẫn S ⊨ ω. Ngôn ngữ *sinh bởi* văn phạm G, ký hiệu L(G), là tập hợp tất cả các từ sinh bởi văn phạm G:

$$L(G) = \{\omega \in \Sigma^* \mid S \models^G \omega\}.$$

▪ Như vậy, ω là một từ sinh bởi văn phạm G, hay ω∈ L(G), khi và chỉ khi tồn tại dãy các suy dẫn trực tiếp được bắt đầu bằng một quy tắc có vế trái là S: $S = \omega_0 \vdash \omega_1 \vdash \ldots \vdash \omega_n = \omega \in \Sigma^*$.

Định nghĩa 2.20

Hai văn phạm $G_1 = <\Sigma_1, \Delta_1, S_1, P_1>$ và $G_2 = <\Sigma_2, \Delta_2, S_2, P_2>$ được gọi là tương đương nếu $L(G_1) = L(G_2)$.

Thí dụ 2.13:

1. Xét văn phạm G_1 trong thí dụ 2.12. Từ ω = 00001111 được suy dẫn từ S bằng dãy dẫn xuất độ dài 5: S ⊢ 0S1 ⊢ 00S11 ⊢ 000S111 ⊢ 0000S1111 ⊢ 00001111 (có thể viết ngắn gọn là $\omega = 0^4 1^4$). Theo dãy suy dẫn trên, xâu ω = 00001111 sinh bởi văn phạm G_1 nhờ việc áp dụng quy tắc 1 bốn lần, sau đó áp dụng quy tắc 2.

   Bằng việc sử dụng n lần (n ≥ 0) quy tắc 1 rồi quy tắc 2, ta có: $S \models 0^n 1^n$, như vậy, mọi từ sinh bởi G_1 đều có dạng $0^n 1^n$. Do đó $L(G_1) = \{0^n 1^n \mid n \geq 0\}$.

2. Xét văn phạm G_2 trong thí dụ 2.12. Sử dụng quy tắc 1, rồi n lần (n ≥ 0) quy tắc 2, sau đó quy tắc 3 để kết thúc, ta có: $S \vdash Ab \models a^n Ab^n b \vdash a^n b^{n+1}$.

   Do đó $L(G_2) = \{a^n b^{n+1} \mid n \geq 0\}$.

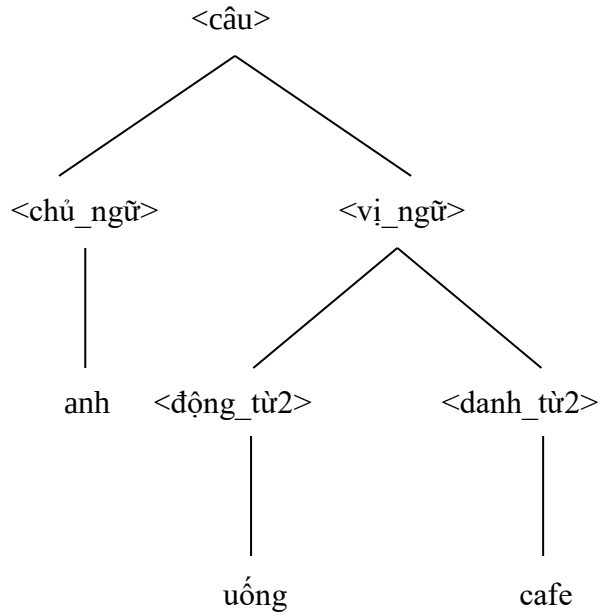
3. Xét văn phạm G_3 trong thí dụ 2.12. Sử dụng quy tắc 1, rồi m -1 lần (m ≥ 1) quy tắc 2, n-1 lần (n ≥ 1) quy tắc 3, k-1 lần (k ≥ 1) quy tắc 4 (các quy tắc có thể xen kẽ), sau đó kết thúc bởi các quy tắc 5, 6, 7, ta có: $S \vdash ABC \models a^m Ab^n Bc^k C \models a^m b^n c^k$.

   Do đó $L(G_3) = \{a^m b^n c^k \mid m \geq 1, n \geq 1, k \geq 1\}$.

4. Dễ dàng thấy rằng: $L(G_4)$ = {tôi ăn cơm, anh ăn cơm, chị ăn cơm, tôi ăn phở, anh ăn phở, chị ăn phở, tôi uống sữa, anh uống sữa, chị uống sữa, tôi uống café, anh uống café, chị uống café}.

Ta có thể biểu diễn việc dẫn xuất từ ký hiệu '<câu>' đến một từ trong $L(G_4)$, chẳng hạn từ 'tôi ăn cơm' bằng một cây gọi là *cây dẫn xuất* hay *cây phân tích cú pháp*

như trong hình 2.1 ở phần trên, hay từ ký hiệu <câu> có thể dẫn xuất đến từ ‘anh uống café’như hình 2.2 dưới đây.



Hình 2.2. Cây dẫn xuất cho thí dụ 2.13

Thí dụ 2.14: Cho hai văn phạm $G_3 = <\Sigma, \{S\}, S, P_3>$, $G_4 = <\Sigma, \{S\}, S, P_4>$, trong đó:

$$\Sigma = \{0, 1, 2, 3, 4, 5 ,6, 7, 8, 9\},$$

P_3 :

$S \rightarrow 1\mid 2\mid 3\mid 4\mid 5\mid 6\mid 7\mid 8\mid 9,$	(1-9)
$S0\mid S1\mid S2\mid S3\mid S4\mid S5\mid S6\mid S7\mid S8\mid S9.$	(10-19)

P_4 :

$S \rightarrow 0\mid 1\mid 2\mid 3\mid 4\mid 5\mid 6\mid 7\mid 8\mid 9,$	(1-10)
$1S\mid 2S\mid 3S\mid 4S\mid 5S\mid 6S\mid 7S\mid 8S\mid 9S.$	(11-19)

Dễ thấy rằng $L(G_3) = \{n \mid n \geq 1\}$. Thật vậy, sử dụng k-1 lần ($k \geq 1$) các quy tắc trong nhóm 10 quy tắc cuối của G_3, rồi một quy tắc trong nhóm 9 quy tắc đầu tiên của nó, ta có:

$S \vdash Si_1 \vdash Si_2i_1 \vdash \ldots \vdash Si_{k-1}\ldots i_2i_1 \vdash Si_ki_{k-1}\ldots i_2i_1$, (với $i_1, i_2,\ldots, i_k \in \Sigma$).

trong đó, $i_1, i_2,\ldots, i_{k-1} \geq 0$ và $i_k \geq 1$. Do đó, $L(G_3) = \{n \mid n \geq 1\}$.

Lập luận như trên, ta nhận được $L(G_4) = \{n \mid n \geq 0\}$. Vì vậy, G_3 và G_4 không tương đương nhau.

2.4.4. Phân loại văn phạm theo Chomsky

Dựa vào đặc điểm của tập quy tắc mà người ta chia các văn phạm thành các nhóm khác nhau. Noam Chomsky (*) đã phân loại văn phạm thành bốn nhóm dưới đây:

▪ Nhóm 0: Văn phạm không hạn chế *(Unrestricted Grammar-UG)* (hay văn phạm *ngữ cấu*, văn phạm *tổng quát*),

▪ Nhóm 1: Văn phạm cảm ngữ cảnh (*Context Sensitive Grammar-CSG*),

▪ Nhóm 2: Văn phạm phi ngữ cảnh (*Context Free Grammar-CFG*),

▪ Nhóm 3: Văn phạm chính quy. (*Regular Grammar-RG*)

Dưới đây là các định nghĩa cho các nhóm văn phạm nói trên.

Định nghĩa 2.21

Văn phạm $G = <\Sigma, \Delta, S, P>$ mà không có một ràng buộc nào đối với các quy tắc của nó được gọi là *văn phạm tổng quát hay văn phạm không hạn chế.*

Như vậy, các quy tắc trong văn phạm nhóm 0 chỉ cần là các quy tắc hợp lệ (vế trái chứa ít nhất một ký hiệu phụ: $\alpha \rightarrow \beta$, với $\alpha = \alpha' A \alpha''$, $A \in \Delta$, α', α'', $\beta \in (\Sigma \cup \Delta)^*$). Các quy tắc của văn phạm nhóm 0 được gọi là quy tắc không hạn chế.

▪ Ngôn ngữ do văn phạm nhóm 0 sinh ra được gọi là ngôn ngữ tổng quát.

Định nghĩa 2.22

Văn phạm $G = <\Sigma, \Delta, S, P>$ với các quy tắc hợp lệ $\alpha \rightarrow \beta$, và $| \alpha | \leq | \beta |$, được gọi là văn phạm *nhóm 1 hay văn phạm cảm ngữ cảnh*.

Các quy tắc trong văn phạm *nhóm 1* được gọi là quy tắc *cảm ngữ cảnh*.

▪ Ngôn ngữ do văn phạm cảm ngữ cảnh sinh ra được gọi là ngôn ngữ cảm ngữ cảnh. (*Context Sensitive Language-CSL*)

Các văn phạm mà các quy tắc của chúng có dạng trên, đồng thời chứa thêm quy tắc rỗng $S \rightarrow \varepsilon$, cũng được xếp vào lớp văn phạm nhóm 1.

Định nghĩa 2.23

Văn phạm $G = <\Sigma, \Delta, S, P>$ mà các quy tắc của nó có dạng $A \rightarrow \omega$, trong đó $A \in \Delta$, $\omega \in (\Sigma \cup \Delta)^*$, được gọi là văn phạm *nhóm 2*.hay *văn phạm phi ngữ cảnh.*

(*)Noam Chomsky: Institute Professor, Massachusetts Institute of Technology (MIT). Born December 7, 1928, Philadelphia, Pennsylvania, USA.

Như vậy, các quy tắc trong văn phạm *phi ngữ cảnh* có vế trái chỉ chứa một ký hiệu phụ còn vế phải là tùy ý, và được gọi là quy tắc *phi ngữ cảnh*.

Ngôn ngữ do văn phạm phi ngữ cảnh sinh ra được gọi là ngôn ngữ phi ngữ cảnh. (*Context Free Language-CFL*),

Định nghĩa 2.24

Văn phạm G = <Σ, Δ, S, P> mà các quy tắc của nó chỉ có dạng A→ aB, A→ a (còn gọi là dạng tuyến tính phải-right linear) (hoặc chỉ có dạng A→ Ba, A→ a, còn gọi là dạng tuyến tính trái-left linear), trong đó A, B∈ Δ, a∈ Σ, được gọi là văn phạm nhóm 3 hay văn phạm chính quy.

Các văn phạm chính quy, đồng thời chứa thêm quy tắc rỗng S→ ε cũng được gọi là văn phạm chính quy suy rộng. Các quy tắc *tuyến tính phải* hoặc *tuyến tính trái* trong văn phạm chính quy được gọi là *quy tắc chính quy.*

▪ Ngôn ngữ do văn phạm chính quy sinh ra được gọi là *ngôn ngữ chính quy*. (*Regular Language-RL*)

Thí dụ 2.15: Cho văn phạm G = <{a, b, c}, {S, A, B, C}, S, P>, với tập quy tắc sau:

P:
- S→ aSAC, (1)
- S→ abC, (2)
- CA→ BA, (3)
- BA→ BC, (4)
- BC→ AC, (5)
- bA→ bb, (6)
- C→ c. (7)

Các quy tắc của văn phạm G đều có dạng α→ β, với |α|≤|β|, vậy G là văn phạm cảm ngữ cảnh.

Sử dụng n-1 lần (n ≥ 1) quy tắc 1, rồi quy tắc 2, kế đến sử dụng liên tiếp các quy tắc 3, 4, 5 (để đổi chỗ A và C), sau đó sử dụng n-1 lần quy tắc 6 và n lần quy tắc 7, ta có dãy suy dẫn:

$S \models a^{n-1}S(AC)^{n-1} \vdash a^nbC(AC)^{n-1} \models a^nbA^{n-1}C^n \models a^nb^nc^n$.

Từ đó suy ra $L(G) = \{a^nb^nc^n \mid n \geq 1\}$.

Thí dụ 2.16

1. Cho văn phạm $G_1 = <\{a, b\}, \{S, A\}, S, P>$, trong đó:

$$P_1 : \begin{cases} S \rightarrow Sa, & (1) \\ S \rightarrow Aa, & (2) \\ A \rightarrow aAb, & (3) \\ A \rightarrow ab. & (4) \end{cases}$$

Các quy tắc đều có vế trái là một ký hiệu phụ, vậy G_1 là văn phạm phi ngữ cảnh.

Sử dụng m-1 lần ($m \geq 1$) quy tắc 1, rồi quy tắc 2, sau đó sử dụng n-1 lần ($n \geq 1$) quy tắc 3, cuối cùng là quy tắc 4, ta có dãy suy dẫn:

$S \models Sa^{m-1} \vdash Aaa^{m-1} \models a^{n-1}Ab^{n-1}a^m \vdash a^nb^na^m$.

Từ đó suy ra $L(G_1) = \{a^nb^na^m \mid n \geq 1, m \geq 1\}$.

Chú ý rằng chỉ những quy tắc phi ngữ cảnh mà vế phải có chứa ký hiệu phụ ở vế trái, (gọi là các quy tắc đệ quy) thì mới có thể áp dụng nhiều lần quy tắc này. (Ở thí dụ trên là quy tắc 1 và quy tắc 3)

2. Cho văn phạm $G_2 = <\{0, 1\}, \{S\}, S, P_2>$, với tập quy tắc:

$$P_2 : \begin{cases} S \rightarrow SS, & (1) \\ S \rightarrow 0S1, & (2) \\ S \rightarrow 1S0, & (3) \\ S \rightarrow \varepsilon. & (4) \end{cases}$$

Dễ thấy G_2 là văn phạm phi ngữ cảnh. Từ các quy tắc của G_2, ta có $L(G_2) = \{\varepsilon, 01, 10, 0011, 1100, 1001, 111000,\ldots\}$ hay $L(G_2) = \{\omega \in \{0, 1\}^* \mid$ số các chữ số 0 và 1 trong ω là bằng nhau$\}$.

3. Cho văn phạm $G_3 = <\{a, b\}, \{S\}, S, P_3>$, với P_3:

$$P_3: \begin{cases} S \rightarrow \varepsilon, & (1) \\ S \rightarrow aSa, & (2) \\ S \rightarrow bSb, & (3) \\ S \rightarrow aa, & (4) \\ S \rightarrow bb. & (5) \end{cases}$$

Dễ thấy G_3 là văn phạm phi ngữ cảnh vì nó có mọi quy tắc với vế trái chỉ gồm một ký hiệu phụ. G_3 sinh ra ngôn ngữ phi ngữ cảnh $L(G_3) = \{\omega\omega^R \mid \omega \in \{a, b\}^*\}$ có các từ có độ dài chẵn và có các ký hiệu đối xứng nhau từ hai đầu của từ. Chẳng hạn các từ abba, bbaabb, ababbaba… là thuộc $L(G_3)$.

Thí dụ 2.17:

1. Cho văn phạm: $G_1 = <\{1\}, \{S, A, B\}, S, P_1>$, với tập quy tắc

$$P_1: \begin{cases} S \to \varepsilon, & (1) \\ S \to 1A, & (2) \\ A \to 1B, & (3) \\ B \to 1A, & (4) \\ A \to 1. & (5) \end{cases}$$

Các quy tắc trong P_1 đều là quy tắc chính quy, vậy G_1 là văn phạm chính quy và ngôn ngữ sinh bởi G_1 là ngôn ngữ chính quy, $L(G_1) = \{1^{2n} \mid n \geq 0\}$. Thật vậy, sử dụng quy tắc 1, ta có $S \vdash \varepsilon$, nếu sử dụng quy tắc 2, rồi n-1 lần ($n \geq 1$) liên tiếp cặp quy tắc 3 và 4, cuối cùng là quy tắc 5, ta có dãy suy dẫn:

$$S \vdash 1A \vdash 11B \vdash 111A \vdash \ldots \models 1(1^{2n-2})A \vdash 1(1^{2n-2})1 = 1^{2n}.$$

Như vậy, ngôn ngữ $L(G_1)$ gồm mọi xâu có độ dài chẵn, chứa toàn ký hiêu '1' .

2. Cho văn phạm $G_2 = <\{0, 1\}, \{S, A\}, S, P_2>$, với tập quy tắc

$$P_2: \begin{cases} S \to 0A, & (1) \\ A \to 0A, & (2) \\ A \to 1A, & (3) \\ A \to 0. & (4) \end{cases}$$

Khi đó, G_2 là văn phạm chính quy và $L(G_2) = \{0\omega0 \mid \omega \in \{0, 1\}^*\}$. Thật vậy, sử dụng quy tắc 1, rồi một số hữu hạn lần tuỳ ý, có thể xen kẽ các quy tắc 2 và 3, cuối cùng là quy tắc 4, ta có: $S \vdash 0A \models 0\omega A \vdash 0\omega0$. Như vậy, ngôn ngữ sinh bởi văn phạm G_2 gồm mọi xâu nhị phân, bắt đầu và kết thúc bởi ký hiệu '0'.

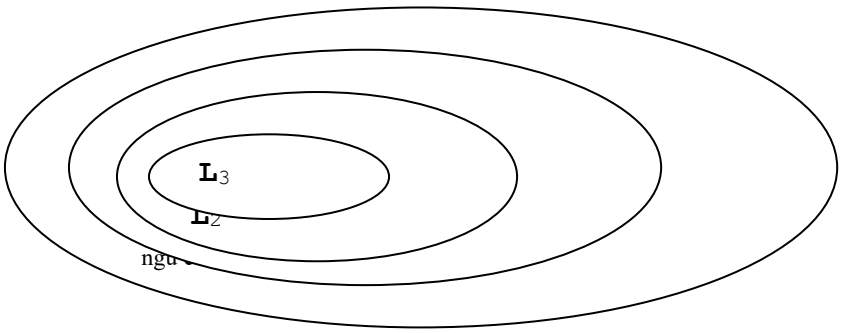
Nhận xét: Từ các định nghĩa trên, ta thấy lớp văn phạm không hạn chế là rộng nhất, nó chứa đựng các văn phạm cảm ngữ cảnh, lớp văn phạm cảm ngữ cảnh chứa các văn phạm phi ngữ cảnh và lớp văn phạm phi ngữ cảnh chứa các văn phạm chính quy.

Một ngôn ngữ hình thức được gọi tên bằng tên văn phạm sinh ra nó, tức là L là được gọi ngôn ngữ tổng quát (hay ngôn ngữ cảm ngữ cảnh, phi ngữ cảnh, chính quy) nếu tồn tại văn phạm loại tổng quát (tương ứng: văn phạm cảm ngữ cảnh, phi ngữ cảnh, chính quy) sinh ra L. Vì vậy, đối với các lớp ngôn ngữ, nếu ký hiệu L_0, L_1, L_2, L_3 lần lượt là các lớp ngôn ngữ *tổng quát,* ngôn ngữ *cảm ngữ cảnh,* ngôn ngữ *phi ngữ cảnh* và ngôn ngữ *chính quy* thì ta có bao hàm thức:

$$L_3 \subset L_2 \subset L_1 \subset L_0.$$

Theo sự phân loại của Chomsky, ta thấy lớp ngôn ngữ chính quy L_3 là nhỏ nhất, nó bị chứa thực sự trong lớp ngôn ngữ phi ngữ cảnh L_2, lớp ngôn ngữ phi ngữ cảnh lại bị chứa thực sự trong lớp ngôn ngữ cảm ngữ cảnh L_1 và cuối cùng lớp ngôn ngữ tổng quát L_0 (ngôn ngữ ngữ cấu) là rộng nhất. Ta cũng thấy về mặt cấu trúc ngữ pháp thì các quy tắc của các văn phạm phi ngữ cảnh và văn phạm chính quy là đơn giản hơn cả và chúng có nhiều ứng dụng trong việc thiết kế các ngôn ngữ lập trình và trong nghiên cứu về chương trình dịch… Vì vậy, trong các phần tiếp theo chúng ta dành thêm sự quan tâm tới hai lớp văn phạm đó.

Hình vẽ dưới đây cho một sự so sánh về mức độ rộng lớn của các lớp ngôn ngữ theo phân loại của Chomsky.



Hình 2.3. So sánh các lớp ngôn ngữ

Thí dụ 2.18:. Cho bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_n\}$.

Chứng minh rằng các ngôn ngữ: $L_1 = \{\omega = a_1a_2\ldots a_k\}$, $L_2 = \Sigma^+$, $L_3 = \Sigma^*$, $L = \varnothing$ là các ngôn ngữ chính quy trên bảng chữ Σ.

Thật vậy, ta có thể xây dựng các văn phạm chính quy sinh các ngôn ngữ trên:

- $G_1 = <\Sigma, \{S, A_1, \ldots, A_{n-1}\}, S, P_1>$, với tập quy tắc:

P_1: $S \to a_1A_1$, (1)

$A_1 \to a_2A_2$, (2)

,…,

$A_{k-2} \to a_{k-1}A_{k-1}$, (k-1)

$A_{k-1} \to a_k$. (k)

Dễ thấy G_1 là văn phạm chính quy, và $L_1 = L(G_1)$. Vậy ngôn ngữ L_1 chỉ gồm 1 từ là ngôn ngữ chính quy.

Hiển nhiên là các tập từ chỉ gồm một ký hiệu {a}, với $a \in \Sigma$ cũng là các ngôn ngữ chính quy.

▪ $G_2 = <\Sigma, \{S\}, S, P_2>$, với tập quy tắc:

P_2: $S \to aS$, (1)

$S \to a$. (2), với $a \in \Sigma$.

Dễ thấy G_2 là văn phạm chính quy, và $L_2 = L(G_2)$, vậy tập từ Σ^+ là ngôn ngữ chính quy.

▪ $G_3 = <\Sigma, \{S, A\}, S, P_3>$, với tập quy tắc:

P_3: $S \to \varepsilon$, (1)

$S \to a$, (2)

$S \to aA$, (3)

$A \to aA$, (4)

$A \to a$. (5), với $a \in \Sigma$.

Dễ thấy G_3 là văn phạm chính quy, và $L_3 = L(G_3)$, vậy tập từ Σ^* là ngôn ngữ chính quy.

▪ $G_4 = <\Sigma, \{S\}, S, \{S \to aS \mid a \in \Sigma\}>$, với tập quy tắc chỉ gồm một quy tắc chính quy $S \to aS$, vậy G_4 là văn phạm chính quy, mọi dãy suy dẫn bắt đầu từ S đều không thể kết thúc bằng một xâu chứa toàn ký hiệu chính, tức là không có xâu $\omega \in \Sigma^*$ sinh bởi G_4, vậy G_4 sinh ra ngôn ngữ rỗng ($\varnothing$). Vậy ngôn ngữ Ø là một ngôn ngữ chính quy.

Thí dụ trên đây cho thấy: Ngôn ngữ chỉ gồm một từ, ngôn ngữ rỗng, các tập từ Σ^+ và Σ^* là các ngôn ngữ chính quy.

2.5. CÁC TÍNH CHẤT CỦA VĂN PHẠM VÀ NGÔN NGỮ SINH BỞI VĂN PHẠM

2.5.1. Một số tính chất của văn phạm và dẫn xuất

Trong phần này, chúng ta sẽ trình bày một số tính chất quan trọng của các dẫn xuất và văn phạm.

Trước hết, ta chú ý rằng đối với một văn phạm, người ta quan tâm chủ yếu tới ngôn ngữ mà nó sinh ra. Hai văn phạm là tương đương nếu chúng sinh ra cùng một ngôn ngữ (*Định nghĩa 2.20*).

Các định lý và bổ đề dưới đây chứng minh cho một số tính chất của văn phạm và dẫn xuất.

Định lý 2.1

Với mọi văn phạm G = <Σ, Δ, S, P>, luôn tồn tại văn phạm G' = <Σ', Δ', S', P'> tương đương với văn phạm G, tức là L(G) = L(G').

Chứng minh:

Giả sử có văn phạm G = <Σ, Δ, S, P>, ta xây dựng văn phạm G' = <Σ', Δ', S', P'>, trong đó:

. Σ' = Σ, và với mỗi a∈ Σ, ta bổ sung một ký hiệu $\bar{a} \notin \Sigma \cup \Delta$ và gọi là đối ngẫu của a, đặt Γ = { $\bar{a}$ | a ∈ Σ}

. Δ' = Δ ∪ Γ,

. S' = S,

. P' = $P_1 \cup P_2$, với P_1 = { $\bar{a} \to a$ | ∀a∈ Σ}, P_2 = { $\bar{\alpha} \to \bar{\beta}$ | ∀α→ β∈ P}, $\bar{\alpha}$ và $\bar{\beta}$ là các xâu α và β đã được thay các ký hiệu thuộc Σ bằng các ký hiệu đối ngẫu của nó. Ta sẽ chứng minh rằng L(G) = L(G') bằng việc chứng minh hai bao hàm thức L(G) ⊆ L(G'và L(G') ⊆ L(G).

a). *Chứng minh L(G) ⊆ L(G')*: Lấy bất kỳ ω∈ L(G), khi đó ta có S $\models^G$ω, tức là ta có một dãy suy dẫn trực tiếp trong G: S = $\omega_0 \vdash^G \omega_1 \vdash^G \ldots \vdash^G \omega_k$ = ω, với dãy suy dẫn này, ta thay mọi quy tắc trong các suy dẫn $\omega_i \vdash^G \omega_{i+1}$, (0 ≤ i ≤ k-1), bởi các quy tắc tương ứng trong P_1 và P_2, ta nhận được dãy các suy dẫn trong G': S = $\omega'_0 \vdash^{G'} \omega'_1 \vdash^{G'} \ldots \vdash^{G'} \omega'_m$ = ω, do đó ta có S $\models^{G'}$ω, tức là ω∈ L(G'). Vậy L(G) ⊆ L(G').

b). *Chứng minh L(G') ⊆ L(G)*: Lấy bất kỳ ω∈ L(G'), khi đó ta có S $\models^{G'}$ω, tức là ta có một dãy suy dẫn trong G': S = $\omega'_0 \vdash^{G'} \omega'_1 \vdash^{G'} \ldots \vdash^{G'} \omega'_k$ = ω, trong các suy dẫn $\omega_i \vdash^{G'} \omega_{i+1}$, (0 ≤ i ≤ k-1), ta thay mọi kí hiệu $\bar{a} \in \Gamma$ bởi các ký hiệu tương ứng a∈ Σ_1, khi

đó mọi quy tắc đều thuộc P, ta nhận được dãy các suy dẫn trực tiếp trong G: $S = \omega_0 \vdash_G \omega_1 \vdash_G \ldots \vdash_G \omega_k = \omega$, ta có $S \models_G \omega$, tức là $\omega \in L(G)$. Vậy $L(G') \subseteq L(G)$.

Thí dụ 2.19:. Cho văn phạm $G_1 = <\{a, b\}, \{S\}, S, P_1>$, với tập quy tắc:

P_1: | $S \rightarrow aSb$,

| $S \rightarrow ab$.

ta có thể xây dựng G_2 tương đương với G_1 theo định lý 2.1 như sau:

$G_2 = <\{a, b\}, \{S, A, B\}, S, P_2>$, ở đây A, B là các ký hiệu đối ngẫu của a và b. Ta có:

P_2: | $S \rightarrow ASB$,

| $A \rightarrow a$,

| $B \rightarrow b$,

| $S \rightarrow AB$.

Dễ dàng thấy rằng $L(G_1) = L(G_2) = \{a^n b^n \mid n \geq 1\}$. Vậy G_1 và G_2 là tương đương.

Với mỗi văn phạm G, ta có thể thay thế các quy tắc có chứa ký hiệu xuất phát ở vế phải bằng những quy tắc không có ký hiệu xuất phát ở vế phải để nhận được một văn phạm tương đương, điều này sẽ làm cho việc kiểm soát các vòng lặp được dễ dàng hơn. Xét bổ đề sau:

Bổ đề 2.1

Cho văn phạm $G = <\Sigma, \Delta, S, P>$. Khi đó nếu tồn tại trong P quy tắc chứa ký hiệu xuất phát S ở vế phải thì tồn tại văn phạm G' tương đương với G mà các quy tắc của nó không chứa ký hiệu xuất phát ở vế phải.

Chứng minh:

Lấy $S' \notin \Sigma \cup \Delta$, xét văn phạm $G' = <\Sigma, \Delta \cup \{S'\}, S', P'>$, trong đó $P' = P \cup \{S' \rightarrow \alpha \mid S \rightarrow \alpha \in P\}$. Rõ ràng trong P' không chứa quy tắc nào có S' ở vế phải. Ta chứng minh $L(G) = L(G')$.

a./ Lấy $\omega \in L(G)$: Khi đó ta có $S \models_G \omega$, giả sử dãy dẫn xuất trong G của ω là $S \vdash \alpha \vdash \omega_1 \vdash \ldots \vdash \omega$. Vì $S \vdash_G \alpha$ nên có $S \rightarrow \alpha \in P$, do đó $S' \rightarrow \alpha \in P'$ và vì $P \subset P'$ nên ta có $S' \vdash_{G'} \alpha \models_{G'} \omega$. Vậy $S' \models_{G'} \omega$ hay $\omega \in L(G')$, vậy $L(G) \subseteq L(G')$.

b./ Lấy $\omega \in L(G')$: Khi đó ta có $S' \models_{G'} \omega$, giả sử ta có dãy dẫn xuất trong G' là $S' \vdash_{G'} \alpha \models_{G'} \omega$. Vì $S' \vdash_{G'} \alpha$ nên $S' \rightarrow \alpha \in P'$, do đó tồn tại $S \rightarrow \alpha \in P$. Mặt khác, trong α

không chứa S' nên các suy dẫn trực tiếp trong $\alpha \models^{G'} \omega$ chỉ sử dụng các quy tắc của P. Vậy ta có $S \models^{G} \omega$ hay $\omega \in L(G)$, vậy $L(G') \subseteq L(G)$.

Mặt khác, với mỗi văn phạm G, ta có thể thay thế các quy tắc có chứa ký hiệu cơ bản ở vế trái, để nhận được một văn phạm tương đương không chứa các ký hiệu chính (ký hiệu kết thúc) ở vế trái các quy tắc, nhờ bổ đề sau:

Bổ đề 2.2

Cho văn phạm $G = <\Sigma, \Delta, S, P>$ tùy ý, luôn luôn có thể xây dựng văn phạm G' tương đương với G mà các quy tắc của nó không chứa ký hiệu kết thúc ở vế trái.

Chứng minh:

Giả sử có văn phạm $G = <\Sigma, \Delta, S, P>$ tùy ý, với mỗi ký hiệu chính *a* xuất hiện trong vế trái của một quy tắc nào đó, ta bổ sung một ký hiệu $\bar{a} \notin \Sigma \cup \Delta$ và gọi là đối ngẫu của a,

Đặt $\Gamma = \{ \bar{a} \mid a \in \Sigma$, a xuất hiện ở vế trái quy tắc nào đó $\}$,

$P_1 = \{ \bar{a} \to a \mid \bar{a} \in \Gamma, a \in \Sigma \}$,

$P_2 = \{ \bar{\alpha} \to \bar{\beta} \mid \forall \alpha \to \beta \in P \}$, $\bar{\alpha}$ và $\bar{\beta}$ là các xâu α và β đã được thay các ký hiệu $a \in \Sigma$ bằng các ký hiệu đối ngẫu $\bar{a}$ của nó.

Xây dựng văn phạm $G' = <\Sigma', \Delta', S', P'>$, với:

$\Sigma' = \Sigma$,

$\Delta' = \Delta \cup \Gamma$,

$S' = S$,

$P' = P_1 \cup P_2$

Văn phạm G' sẽ là văn phạm tương đương với văn phạm G (theo định lý 2.1), hơn nữa, theo cách xây dựng thì trong tất cả các vế trái của G' sẽ không chứa ký hiệu chính (ký hiệu kết thúc). Vậy bổ đề được chứng minh.

Ta đưa ra hai khái niệm về dẫn xuất:

Định nghĩa 2.25.

Cho văn phạm $G = < \Sigma, \Delta, S, P >$ và hai dẫn xuất $D = \omega_0, \omega_1, \ldots, \omega_k$ và $D' = \omega'_0, \omega'_1, \ldots, \omega'_m$ trong văn phạm G. Ta nói hai dẫn xuất trên là đồng lực nếu $\omega_0 = \omega'_0$ và $\omega_k = \omega'_m$.

Định nghĩa 2.26.

Cho văn phạm G = <Σ, Δ, S, P> và dẫn xuất D = $\omega_0, \omega_1, \ldots, \omega_k$ trong văn phạm G. Ta nói dẫn xuất D là không lặp nếu không tồn tại cặp (ω_i, ω_j) với $i \neq j$ mà $\omega_i = \omega_j$.

Định lý 2.2

Với mọi dẫn xuất trong văn phạm G tùy ý, luôn luôn tồn tại một dẫn xuất không lặp và đồng lực với nó.

Chứng minh: Giả sử D = $\omega_0, \omega_1, \ldots, \omega_{i-1}, \omega_i, \omega_{i+1}, \ldots, \omega_m$, xét các trường hợp sau:

a). Trong D không có một cặp (ω_i, ω_j) với $i \neq j$ mà $\omega_i = \omega_j$, khi đó D chính là dẫn xuất không lặp và đồng lực với chính nó.

b). Trong D có một cặp (ω_i, ω_j) với $i \neq j$ mà $\omega_i = \omega_j$, khi đó ta xét dẫn xuất D' = $\omega_0, \omega_1, \ldots, \omega_{i-1}, \omega_j, \omega_{j+1}, \ldots, \omega_m$. Rõ ràng đây là dẫn xuất không lặp và đồng lực với D, vì D' nhận được bằng cách bỏ đi một đoạn $\omega_i, \omega_{i+1}, \ldots, \omega_{j-1}$ là đoạn có điểm đầu và điểm cuối trùng nhau, do đó dẫn xuất D' là đồng lực với D. Nếu trong D vẫn còn những cặp $\omega'_i = \omega'_j$ như vậy, ta sẽ lặp lại quá trình trên cho đến khi mọi xâu trong D là khác nhau từng đôi một, ta sẽ nhận được một dẫn xuất mới không lặp và đồng lực với dẫn xuất ban đầu.

Việc thay thế các dẫn xuất bằng các dẫn xuất không lặp và đồng lực, làm cho việc sinh các từ trở nên đơn giản hơn, các dãy suy dẫn để sinh ra một từ là một dẫn xuất có độ dài ngắn nhất.

2.5.2. Tính đóng của lớp ngôn ngữ sinh bởi văn phạm

Giả sử L_1 và L_2 là hai ngôn ngữ bất kỳ được sinh bởi văn phạm, và "$\circ$" là một phép toán nào đó trên lớp các ngôn ngữ (phép hợp, phép giao, phép nhân ghép, phép lấy ngôn ngữ bù…). Nếu $L_1 \circ L_2$ là ngôn ngữ cũng được sinh bởi một văn phạm thì ta nói lớp ngôn ngữ do văn phạm sinh ra *đóng* đối với phép toán "$\circ$". Lớp ngôn ngữ sinh bởi văn phạm là đóng đối với hầu hết các phép toán trên ngôn ngữ mà ta đã học trong 2.3, dưới đây ta chỉ xét tính đóng đối với một số phép toán quan trọng nhất.

Định lý 2.3

Lớp ngôn ngữ sinh bởi văn phạm là đóng đối với phép hợp ($\cup$), phép giao ($\cap$) và phép nhân ghép ngôn ngữ (.)

Chứng minh:

▪ Trước hết, ta sẽ chứng minh lớp ngôn ngữ sinh bởi văn phạm là đóng đối với phép hợp, việc chứng minh tính đóng của lớp ngôn ngữ sinh bởi văn phạm đối với các phép giao và phép nhân ghép ngôn ngữ là hoàn toàn tương tự.

Giả sử L_1, L_2 là các ngôn ngữ được sinh bởi văn phạm $G_1 = <\Sigma_1, \Delta_1, S_1, P_1>$, $G_2 = <\Sigma_2, \Delta_2, S_2, P_2>$, tức là $L_1 = L(G_1)$, $L_2 = L(G_2)$. Ta chứng minh tồn tại văn phạm G sao cho $L(G) = L_1 \cup L_2$.

Xây dựng văn phạm $G = <\Sigma, \Delta, S, P>$, với:

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\Delta = \Delta_1 \cup \Delta_2 \cup \{S\}$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}$

Ta sẽ chứng minh văn phạm G sinh ra ngôn ngữ $L_1 \cup L_2$, tức là $L(G) = L_1 \cup L_2$, bằng cách chứng minh hai bao hàm thức:

a). *Chứng minh $L(G) \subseteq L_1 \cup L_2$*: Giả sử $\omega \in L(G)$, khi đó tồn tại một suy dẫn trong văn phạm G: $S \models^G \omega$, trong đó $\omega \in \Sigma^* = (\Sigma_1 \cup \Sigma_2)^*$. Do cách xây dựng tập quy tắc P, nên trong suy dẫn $S \models \omega$, có hai khả năng:

(i) hoặc $S \vdash^G S_1 \models^{G1} \omega$, vậy ω là kết quả của suy dẫn $S_1 \models \omega$ trong G_1,
do đó $\omega \in L(G_1)$. (a)

(ii) hoặc $S \vdash^G S_2 \models^{G2} \omega$, vậy ω là kết quả của suy dẫn $S_2 \models \omega$ trong G_2,
do đó $\omega \in L(G_2)$. (b)

Từ (a) và (b), ta thấy $\omega \in L_1 \cup L_2$, hay $L(G) \subseteq L_1 \cup L_2$

b). *Chứng minh $L_1 \cup L_2 \subseteq L(G)$*: Giả sử $\omega \in L_1 \cup L_2$, khi đó ta cũng có hai khả năng: $\omega \in L_1$ hoặc $\omega \in L_2$:

Nếu $\omega \in L_1 = L(G_1)$, do đó $S_1 \models^{G1} \omega$ trong G_1, do đó ta cũng có suy dẫn $S \vdash^G S_1 \models^{G1} \omega$ là một suy dẫn trong G (vì theo cách xây dựng G, mọi quy tắc và mọi ký hiệu trong G_1 cũng đều thuộc G), như vậy $\omega \in L(G)$.

Nếu $\omega \in L_2 = L(G_2)$, ta có suy dẫn $S_2 \models^{G2} \omega$ trong G_2, do đó ta có suy dẫn $S \vdash^G S_2 \models^{G2} \omega$ là một suy dẫn trong G (vì theo cách xây dựng G, mọi quy tắc và mọi ký hiệu trong G_2 cũng đều thuộc G), như vậy $\omega \in L(G)$.

Vậy ta luôn luôn có $\omega \in L(G)$, do đó: $L_1 \cup L_2 \subseteq L(G)$.

Tức là ta đã chứng minh được rằng $L(G) = L_1 \cup L_2$.

▪ Tương tự, để chứng minh tính đóng của lớp ngôn ngữ sinh bởi văn phạm đối với phép nhân ghép ngôn ngữ, ta xây dựng văn phạm $G = <\Sigma, \Delta, S, P>$ sao cho $L(G) = L(G_1).L(G_2)$ như sau:

$\Sigma = \Sigma_1 \cup \Sigma_2$

$\Delta = \Delta_1 \cup \Delta_2 \cup \{S\}$

$P = P_1 \cup P_2 \cup \{S \rightarrow S_1S_2\}$.

Khi đó $L(G) = L(G_1).L(G_2)$

▪ Để chứng minh tính đóng của lớp ngôn ngữ sinh bởi văn phạm đối với phép giao, ta xây dựng văn phạm $G = <\Sigma, \Delta, S, P>$ sao cho $L(G) = L(G_1) \cap L(G_2)$ như sau:

$\Sigma = \Sigma_1 \cap \Sigma_2$

$\Delta = \Delta_1 \cup \Delta_2 \cup \Gamma_1 \cup \Gamma_2 \cup \{S\}$, trong đó: $\Gamma_1 = \{ \bar{a} \mid a \in \Sigma_1\}$ là tập các ký hiệu đối ngẫu của các ký hiệu trong Σ_1, còn $\Gamma_2 = \{\bar{b} \mid b \in \Sigma_2\}$ là tập các ký hiệu đối ngẫu của Σ_2.

$P = \overline{P_1} \cup \overline{P_2} \cup \{S \rightarrow S_1S_2\} \cup P' \cup P''$, trong đó $\overline{P_1}$ là tập các quy tắc nhận được từ P_1, mà mọi ký hiệu $a \in \Sigma_1$ đều được thay bởi ký hiệu đối ngẫu tương ứng của nó $\bar{a} \in \Gamma_1$, $\overline{P_2}$ là tập các quy tắc trong P_2, mà mọi ký hiệu $b \in \Sigma_2$ đều được thay bởi ký hiệu đối ngẫu tương ứng của nó $\bar{b} \in \Gamma_2$, và:

$P' = \{\bar{a}\,\bar{b} \rightarrow \bar{b}\,\bar{a} \mid a \in \Sigma_1, b \in \Sigma_2\}$,

$P'' = \{\bar{a}\,\bar{a} \rightarrow a \mid a \in \Sigma_1 \cap \Sigma_2\}$.

Khi đó ta sẽ có $L(G) = L(G_1) \cap L(G_2)$.

Định lý đã được chứng minh.

Chú ý:

1. Người ta chứng minh được rằng: Lớp ngôn ngữ sinh bởi văn phạm cũng đóng đối với các phép toán: phép lặp, lặp cắt, phép chia trái và chia phải.

2. Nhưng lớp ngôn ngữ sinh bởi văn phạm không đóng đối với phép trừ và phép lấy phần bù ngôn ngữ.

Hệ quả 2.1

Nếu L_1 và L_2 là hai ngôn ngữ chính quy (hay phi ngữ cảnh, cảm ngữ cảnh) thì $L_1 \cup L_2$ cũng là ngôn ngữ chính quy (hay phi ngữ cảnh, cảm ngữ cảnh).

Thí dụ 2.20: Cho hai ngôn ngữ $L_1 = \{a^ncb^{2n} \mid n \geq 0\}$ và $L_2 = \{a^{2n}cb^n \mid n \geq 0\}$ trên bảng chữ cái $\Sigma = \{a, b, c\}$, có thể thấy rằng L_1 và L_2 lần lược được sinh bởi các văn phạm sau đây:

$G_1 = <\Sigma, \{S_1, A, B\}, S_1, P_1>$, với tập quy tắc:

$$P_1: \begin{cases} S_1 \rightarrow AS_1B, & (1) \\ S_1 \rightarrow c, & (2) \\ A \rightarrow a, & (3) \\ B \rightarrow bb. & (4) \end{cases}$$

$G_2 = <\Sigma, \{S_2, C, D\}, S_2, P_2>$, với tập quy tắc:

$$P_2: \begin{cases} S_2 \rightarrow CS_2D, & (1) \\ S_2 \rightarrow c, & (2) \\ C \rightarrow aa, & (3) \\ D \rightarrow b. & (4) \end{cases}$$

Thật vậy, trong G_1, sử dụng n lần ($n \geq 0$) quy tắc 1, sau đó sử dụng n lần quy tắc 3, n lần quy tắc 4 và quy tắc 2, (vì sau khi sử dụng n lần quy tắc 1, sẽ xuất hiện n ký hiệu A và n ký hiệu B), ta có:

$$S_1 \vdash^{G1} A^nS_1B^n \vdash^{G1} a^nc(bb)^n = a^ncb^{2n}.$$

Tương tự, trong G_2 ta có $S_2 \vdash^{G2} a^{2n}cb^n$. ($n \geq 0$). Rõ ràng G_1, G_2 là hai văn phạm phi ngữ cảnh, do đó các ngôn ngữ $L(G_1)$ và $L(G_2)$ cũng là các ngôn ngữ phi ngữ cảnh, vậy theo hệ quả 2.1 thì hợp của chúng $L = L_1 \cup L_2 = \{a^ncb^{2n}, a^{2n}cb^n \mid n \geq 0\}$ cũng là ngôn ngữ phi ngữ cảnh.

Hệ quả 2.2

Nếu L_1 và L_2 là hai ngôn ngữ chính quy (hay phi ngữ cảnh, cảm ngữ cảnh) thì L_1L_2 cũng là ngôn ngữ chính quy (hay phi ngữ cảnh, cảm ngữ cảnh).

Nhờ hệ quả này, ta dễ dàng nhận biết một ngôn ngữ là chính quy (hay phi ngữ cảnh, cảm ngữ cảnh).

Thí dụ 2.21:.

1. Cho hai ngôn ngữ $L_1 = \{a^nb^n \mid n \geq 1\}$ và $L_2 = \{c^n \mid n \geq 1\}$. Dễ dàng thấy rằng $L_1 = L(G_1)$ và $L_2 = L(G_2)$, trong đó:

   $G_1 = <\{a, b\}, \{S_1\}, S_1, \{S_1 \rightarrow aS_1b, S_1 \rightarrow ab\}>$, là văn phạm phi ngữ cảnh.

   $G_2 = <\{c\}, \{S_2\}, S_2, \{S_2 \rightarrow cS_2, S_2 \rightarrow c\}>$ là văn phạm chính quy (và đương nhiên cũng là văn phạm phi ngữ cảnh).

Khi đó theo hệ quả 5.2, ta sẽ có $L_1L_2 = \{a^nb^nc^m \mid n \geq 1, m \geq 1\}$ là ngôn ngữ phi ngữ cảnh.

2. Cho hai ngôn ngữ chính quy $L_3 = \{ba^n \mid n \geq 0\}$ và $L_4 = \{b^na \mid n \geq 0\}$. Ta có ngay $L_3 = L(G_3)$, $L_4 = L(G_4)$, trong đó G_3 và G_4 là hai văn phạm chính quy:

$G_3 = <\{a, b\}, \{S_1, A\}, S_1, \{S_1 \rightarrow b, S_1 \rightarrow bA, A \rightarrow aA, A \rightarrow a\}>$,

$G_4 = <\{a, b\}, \{S_2\}, S_2, \{S_2 \rightarrow bS_2, S_2 \rightarrow a\}>$.

Khi đó theo hệ quả 2.2, ta sẽ có $L_3L_4 = \{ba^nb^ma \mid n \geq 0, m \geq 0\}$ là ngôn ngữ chính quy.

Đối với phép lặp của các ngôn ngữ, ta có thể chứng minh được kết quả sau:

Hệ quả 2.3

Nếu L là ngôn ngữ chính quy thì lặp L^* của L cũng là ngôn ngữ chính quy. Nói một cách khác, lớp các ngôn ngữ chính quy đóng đối với phép toán lặp.

Cuối cùng, do ngôn ngữ hữu hạn là hợp hữu hạn của các ngôn ngữ một từ, nên từ thí dụ 2.18 (ngôn ngữ một từ là chính quy) và từ hệ quả 2.1 (hợp hữu hạn của các ngôn ngữ chính quy là chính quy), ta có hệ quả sau:

Hệ quả 2.4

Mọi ngôn ngữ hữu hạn đều là ngôn ngữ chính quy.

Thí dụ 2.22: Cho ngôn ngữ hữu hạn L = {0, 01, 011, 0111}, khi đó theo hệ quả trên, L là ngôn ngữ chính quy.

Mặt khác, có thể xây dựng văn phạm chính quy $G = <\{0, 1\}, \{S, A, B, C\}, S, P>$, với tập quy tắc:

$$P: \begin{cases} S \rightarrow 0, \\ S \rightarrow 0A, \\ A \rightarrow 1, \\ A \rightarrow 1B, \\ B \rightarrow 1, \\ B \rightarrow 1C, \\ C \rightarrow 1. \end{cases}$$

Dễ dàng thấy rằng L(G) = L.

BÀI TẬP CHƯƠNG 2

1. Cho bảng chữ cái $\Sigma = \{0, 1\}$, hãy viết 10 từ đầu tiên của ngôn ngữ Σ^* dưới dạng liệt kê các từ theo thứ tự độ dài tăng dần, trong các xâu có cùng độ dài thì theo thứ tự từ điển.

2. Mô tả đặc điểm của các từ trong các ngôn ngữ sau và tìm cách biểu diễn hữu hạn cho các ngôn ngữ đó:

a). $L_1 = \{\varepsilon, ab, aabb, aaabbb, \ldots\}$.

b). $L_2 = \{\varepsilon, 0, 1, 00, 01, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots\}$

Viết văn phạm sinh ngôn ngữ L_1, L_2.

3. Hãy mô tả ngôn ngữ $L_2 = \{a\}^+\{b\}^+$ trên bảng chữ cái $\Sigma = \{a, b\}$, viết biểu diễn hữu hạn cho L_2. Xây dựng văn phạm sinh ngôn ngữ L_2, phân loại L_2 theo Chomsky.

4. Cho các ngôn ngữ $X = \{\varepsilon, abc\}$ và $Y = \{abc\}$ trên bảng chữ cái $\Sigma = \{a, b, c\}$, tìm các ngôn ngữ:

a). X^2, Y^2, X.Y, Y.X

b). $_X\backslash^X$, $^X/_X$, $_Y\backslash^X$, $^X/_Y$.

c). $_Y\backslash^Y$, $^Y/_Y$, $_X\backslash^Y$, $^Y/_X$.

5. Cho các văn phạm:

a). $G = < \Sigma, \Delta, S, P >$ với tập quy tắc sinh

$P = \{ S \rightarrow ABC, AB \rightarrow iADj, Dij \rightarrow iDj, DiC \rightarrow BiC, iB \rightarrow Bi, AB \rightarrow \varepsilon, C \rightarrow \varepsilon \}$ với $i,j \in \{a, b\}$.

b). $G = < \Sigma, \Delta, S, P >$ với tập quy tắc sinh:

$$P = \{S \rightarrow SS, S \rightarrow aSb, S \rightarrow bSa, S \rightarrow ab, S \rightarrow ba\}.$$

c). $G = <\Sigma, \Delta, S, R>$ với tập quy tắc sinh:

$$P = \{S \rightarrow aS, S \rightarrow a \mid \text{với } a \in \Sigma = \{a_1, a_2, \ldots a_n\}\}.$$

Hỏi:

1/. Hãy phân loại các văn phạm trên theo dãy phân loại của Chomsky.

2/. Viết lại từng văn phạm theo dạng đầy đủ trong định nghĩa văn phạm.

3/. Tìm các ngôn ngữ do các văn phạm trên sinh ra.

6. Cho văn phạm G = <{a, b}, {S, S_1}, S, {S→ SS_1, S→ aa, S_1→ b}>.

a). G là văn phạm loại gì, theo cách phân loại của Chomsky?

b). Xác định ngôn ngữ L sinh bởi văn phạm G. Viết 5 từ đầu tiên có độ dài tăng dần của L(G)?

c). Xâu nào trong các xâu ω_1 = aabbb và ω_2 = aaabb thuộc L(G), viết dãy suy dẫn của xâu đó trong văn phạm G.

d). Hãy tìm một văn phạm chính quy G' tương đương với G, tức là L(G') = L(G).

7. Cho ngôn ngữ L = {$\omega\omega^R$ | $\omega \in$ {0, 1}*, ω^R là ảnh gương (từ ngược) của ω}. Xây dựng văn phạm phi ngữ cảnh G sinh ngôn ngữ L.

8. Cho ngôn ngữ L = {$a^n b^n c^m$ | n, m ≥ 1}. Xây dựng văn phạm phi ngữ cảnh G sinh ngôn ngữ L.

9. Cho văn phạm phi ngữ cảnh G với tập quy tắc sinh là:

$$P = \{S \to aSa, S \to aa \mid a \in \Sigma = \{a_1, a_2, \ldots, a_n\}\}.$$

Tìm ngôn ngữ do văn phạm G sinh ra, hãy chỉ ra dẫn xuất đầy đủ của xâu $\omega = a_3a_2a_3a_1a_2a_2a_1a_3a_2a_3$ trong văn phạm nói trên.

10. Cho ngôn ngữ L = {$\omega b \omega^R$ | $\omega \in \Sigma^* = \{a_1, a_2, \ldots, a_k, b\}^*$, ω^R là từ ngược của ω}. Xây dựng văn phạm phi ngữ cảnh G sinh ngôn ngữ L.

11. Cho các văn phạm:

a). G_1 với tập quy tắc P_1 = {S→ aS, S→ Sb, S→ aSb, S→ c}

b). G_2 với tập quy tắc P_2 = {S→ SS, S→ a, S→ b},

c). G_3 với tập quy tắc P_3 = {S→ aA, S→ bB, A→ Sa, B→ Sb, S→ c},

d). G_4 với tập quy tắc P_4 = {S→ AB, A → Sc, A→ a, B→ dB, B→ b},

e). G_5 với tập quy tắc P_5 = {S→ SaS, S→ b},

f). G_6 với tập quy tắc P_6 = {S→ aSS, S→ b},

g). G_7 với tập quy tắc P_7 = {S→ AA, A→ aAa, A→ bAb, A→ c}.

Hỏi:

1/. Hãy phân loại 7 văn phạm trên theo cách phân loại của Chomsky.

2/. Tìm các ngôn ngữ ứng với các văn phạm trên, đó là các ngôn ngữ loại gì?

12. Cho bảng chữ cái Σ = {a}, viết các văn phạm sinh các ngôn ngữ:

L_4 = {ω, với | ω | là một số chẵn },

L_5 = {ω, với | ω | là một số lẻ}.

Phân loại L_4 và L_5 theo Chomsky.

13. Cho bảng chữ cái Σ = {a, b}, viết các văn phạm sinh các ngôn ngữ:

L_6 = {ω, với | ω | là một số chẵn },

L_7 = {ω, với | ω | là một số lẻ}.

Phân loại L_6 và L_7 theo Chomsky.

14. Cho văn phạm G_1 = <{0, 1}, {S, A}, S, {S→ 0A, A→ 1S, S→ ε}>.

a). Xác định ngôn ngữ L sinh bởi văn phạm G_1.

b). L là ngôn ngữ loại gì theo cách phân loại của Chomsky.

c). Xây dựng một văn phạm phi ngữ cảnh sinh ra cùng ngôn ngữ với văn phạm G_1.

15. Hãy xác định các ngôn ngữ sinh bởi các văn phạm dưới đây :

a). G_2 = <{a, b}, {S}, S, {S→ SaS, S→ b}>.

b). G_3 = <{a, b, c}, {S}, S, {S→ aca, S→ bcb, S→ aSa, S→ bSb}>.

c). G_4 = <{0, 1, 2, …, 9}, {S, A}, S, {S→ SA | A, A→ 0|1|2|3|4|5|6|7|8|9}>.

16. Hãy xây dựng các văn phạm sinh ra các ngôn ngữ dưới đây:

a). L_6 = {ω∈ {a}*, và |ω| mod 3 = 0}. (x mod y là phần dư của phép chia số nguyên x cho số nguyên y, còn gọi là phép chia lấy phần dư-modulo)

b). L_7 = {a^{2n+1} | n ≥ 0}.

c). L_8 = {$a^m b^n$ | n ≥ 0, m ≥ n}.

17. Hãy xây dựng các văn phạm chính quy sinh ra các ngôn ngữ dưới đây trên bảng chữ Σ = {0, 1}:

a). L_9 = {0ω1 | ω∈ Σ*}.

b). L_{10} = {1}*{010}{0}*

c). L_{11} = {010}* ∪ {1100}*

d). L_{12} = {$a^m b^n c^k$ | m ≥0, n ≥0, k ≥0}

e). $L_{13} = \{(baa)^m(aab)^n \mid m \geq 1, n \geq 1\}$.

18. Một xâu ω trên bảng chữ cái Σ được gọi là xâu hình tháp nếu $\omega^R = \omega$. Hãy chứng minh rằng:

a). ε là một xâu hình tháp,

b). Với mọi $a \in \Sigma$ thì a là một xâu hình tháp,

c). Nếu ω là một xâu hình tháp thì với mọi $a \in \Sigma$ ta có aωa cũng là một xâu hình tháp.

19. Cho văn phạm cảm ngữ cảnh $G = <\{a, b, c\}, \{S, A, B, C\}, S, P>$, trong đó:

$P = \{S \rightarrow aSAC, S \rightarrow abC, CA \rightarrow BA, BA \rightarrow BC, BC \rightarrow AC, bA \rightarrow bb, C \rightarrow c\}$.

a). Hãy xây dựng văn phạm $G_1 = <\Sigma_1, \Delta_1, S_1, P_1>$ tương đương với văn pham G mà mọi vế trái của các quy tắc của G_1 không chứa ký hiệu cơ bản.

G_1 là văn phạm thuộc nhóm nào?

b). Hãy xây dựng văn phạm $G_2 = <\Sigma_2, \Delta_2, S_2, P_2>$ tương đương với văn pham G mà mọi vế phải của các quy tắc của G_2 không chứa ký hiệu xuất phát.

G_2 là văn phạm thuộc nhóm nào?

20. Cho hai văn phạm: $G_1 = < \{a, b\}, \{S_1\}, S_1, \{S_1 \rightarrow aS_1b \mid a\}>$, và:

$G_2 = <\{a\}, \{S_2\}, S_2, \{S_2 \rightarrow aS_2 \mid a\}$.

Theo phương pháp chứng minh trong định lý 2.3:

a). Hãy xây dựng văn phạm $G_3 = <\Sigma_3, \Delta_3, S_3, P_3>$ sao cho $L(G_3) = L(G_1) \cup L(G_2)$.

b). Hãy xây dựng văn phạm $G_4 = <\Sigma_4, \Delta_4, S_4, P_4>$ sao cho $L(G_4) = L(G_1).L(G_2)$.

c). Hãy xây dựng văn phạm $G_5 = <\Sigma_5, \Delta_5, S_5, P_5>$ sao cho $L(G_5) = L(G_1) \cap L(G_2)$.

21. Cho văn phạm: $G = <\{a, b\}, \{S, S_1\}, S, \{S \rightarrow S_1S, S \rightarrow a, S_1 \rightarrow bb\}>$

Theo phương pháp chứng minh trong định lý 2.3:

a). G là văn phạm loại gì, theo cách phân loại của Chomsky.

b). Viết 5 từ đầu tiên có độ dài tăng dần của L(G).

c). Xác định biểu diễn hữu hạn của ngôn ngữ L sinh bởi văn phạm G.

d). Xâu nào trong các xâu ω_1 = bbbba và ω_2 = aaaab thuộc ngôn ngữ L(G), viết dãy suy dẫn đầy đủ của xâu đó trong văn phạm G.

e). Hãy tìm một văn phạm chính quy G'sinh ra cùng ngôn ngữ với G, tức là L(G') = L(G).

22. Bạn hãy chọn *chỉ một phương án trả lời đúng nhất* trong các phương án A, B, C và D trong các câu hỏi trắc nghiệm dưới đây: (chỉ làm trong khoảng thời gian10 phút)

Câu 1. Cho văn phạm G = <{a, b},{S, S_1}, S, P> P = {S→ S_1S, S→a a, S_1→ b}>.

A. Văn phạm G là văn phạm tổng quát
B. Văn phạm G là văn phạm cảm ngữ cảnh.
C. Văn phạm G là văn phạm phi ngữ ngữ cảnh.
D. Văn phạm G là văn phạm chính quy.

Câu 2. Cho văn phạm G = <{a, b},{S, S_1}, S, P> P = {S→ S_1S, S→ aa, S_1→ b}>.

A. Mọi từ của ngôn ngữ L(G) phải kết thúc bởi ký hiệu 'aa'
B. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi các ký hiệu 'b'
C. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi ký hiệu 'aa'
D. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi ký hiệu 'b' và kết thúc bởi các ký hiệu 'aa'

Câu 3. Cho văn phạm G = <{a, b},{S, S_1}, S, P> P = {S→ bS_1, S_1→ aS_1 | bS_1 | a}>.

A. Văn phạm G là văn phạm tổng quát
B. Văn phạm G là văn phạm cảm ngữ cảnh.
C. Văn phạm G là văn phạm phi ngữ ngữ cảnh.
D. Văn phạm G là văn phạm chính quy.

Câu 4. Cho văn phạm G = <{a, b},{S, S_1}, S, P> P = {S→ bS_1, S_1→aS_1 | bS_1 | a}>.

A. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi hiệu 'b'
B. Mọi từ của ngôn ngữ L(G) phải kết thúc bởi ký hiệu 'a'
C. Tất cả các câu trả lời trên đều đúng.
D. Tất cả các câu trả lời trên đều sai

Câu 5. Cho văn phạm G = <{a, b},{S, S_1}, S, P> P = {S→ aS_1, S_1→ aS_1 | bS_1 | a}>.

A. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi ký hiệu 'a'
B. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi các ký hiệu 'aa'
C. Mọi từ của ngôn ngữ L(G) phải bắt đầu bởi ký hiệu 'ab'
D. Mọi từ của ngôn ngữ L(G) phải bắt đầu và kết thúc bởi ký hiệu 'a'

TÀI LIỆU THAM KHẢO

[1]. Đặng Huy Ruận [2002]. *"Lý thuyết ngôn ngữ hình thức và Otomat"* NXB Đại hoc Quốc gia Hà nội 2002.

[2]. Phan Đình Diệu, [1971]. *"Lý thuyết Otomat và Thuật toán",* NXB Đại hoc và trung hoc chuyên nghiệp, Hà nội 1971.

[3]. Đỗ Đức Giáo, [2000], "Toán rời rạc", NXB Đại học Quốc gia Hà nội, Hà nội, 2000.

[4]. Bài giảng "Ngôn ngữ Hình thức và Otomat", Đại học Khoa học, Đại học Huế.

[5]. Nguyễn Xuân Huy [1988], "*Thuật Toán*", NXB Thống Kê, Hà Nội, 1988.

[6]. Nguyễn Văn Định, [2000]. "*Thiết kế và phân tích thuật toán",* Bài giảng chuyên đề tại khoa Toán-Cơ-Tin học, ĐH Khoa học Tự nhiên, ĐHQG Hà nội.

[7]. J.E. Hopcroft, and J.D.Ullman [1979]. *"Introduction to Automata Theory, Languages and Computation"* Addison-Wesley, Reading, MA.1979.

[8]. Nguyen Van Dinh, [1998]. *"Solving Determinazation Problems of Automata on the computer",* VNU. Journal of Science, Nat. Sci., t.XIV, $n^0$1-1998, p: 27-32.

[9]. Nguyen Van Dinh [2001].*"The Automata complexity of the Language transformation schema that contains operations with restricted degree".* Journal of computer science and cybernetics, T.17, S.2 (2001), pages 39-44