Bài Giảng Môn học: OTOMAT VÀ NGÔN NGỮ HÌNH THỨC

TS. Nguyễn Văn Định, Khoa CNTT

Lời nói đầu

Ngôn ngữ là phương tiện để giao tiếp, sự giao tiếp có thể hiểu là giao tiếp giữa con người với nhau, giao tiếp giữa người với máy, hay giao tiếp giữa máy với máy. Ngôn ngữ để con người có thể giao tiếp với nhau được gọi là ngôn ngữ tự nhiên, chẳng hạn như tiếng Anh, tiếng Nga, tiếng Việt… là các ngôn ngữ tự nhiên. Các quy tắc cú pháp của ngôn ngữ tự nhiên nói chung rất phức tạp nhưng các yêu cầu nghiêm ngặt về ngữ nghĩa thì lại thiếu chặt chẽ, chẳng hạn cùng một từ hay cùng một câu ta có thể hiểu chúng theo những nghĩa khác nhau tùy theo từng ngữ cảnh cụ thể. Con người muốn giao tiếp với máy tính tất nhiên cũng thông qua ngôn ngữ. Để có sự giao tiếp giữa người với máy hay giữa máy với nhau, cần phải có một ngôn ngữ với các quy tắc cú pháp chặt chẽ hơn so với các ngôn ngữ tự nhiên, nói cách khác, với một từ hay một câu thì ngữ nghĩa của chúng phải là duy nhất mà không phụ thuộc vào ngữ cảnh. Những ngôn ngữ như thế được gọi là ngôn ngữ hình thức. Con người muốn máy tính thực hiện công việc, phải viết các yêu cầu đưa cho máy bằng ngôn ngữ máy hiểu được. Việc viết các yêu cầu như thế gọi là lập trình. Ngôn ngữ dùng để lập trình được gọi là ngôn ngữ lập trình. Các ngôn ngữ lập trình đều là các ngôn ngữ hình thức.

Cả ngôn ngữ hình thức lẫn ngôn ngữ tự nhiên đều có thể xem như những tập các từ, tức là các xâu hữu hạn các phần tử của một bộ chữ cái cơ sở nào đó. Về mặt truyền thống, lý thuyết ngôn ngữ hình thức liên quan đến các đặc tả cú pháp của ngôn ngữ nhiều hơn là đến những vấn đề ngữ nghĩa. Một đặc tả về cú pháp của một ngôn ngữ có hữu hạn từ, ít nhất về nguyên tắc, có thể được cho bằng cách liệt kê các từ. Điều đó không thể áp dụng đối với các ngôn ngữ có vô hạn từ. Nhiệm vụ chính của lý thuyết ngôn ngữ hình thức là nghiên cứu các cách đặc tả hữu hạn của các ngôn ngữ vô hạn.

Lý thuyết tính toán cũng như của nhiều ngành khác nhau của nó, chẳng hạn mật mã học, có liên quan mật thiết với lý thuyết ngôn ngữ. Các tập vào và ra của một thiết bị tính toán có thể được xem như các ngôn ngữ và nói một cách sâu sắc hơn thì các mô hình tính toán có thể được đồng nhất với các lớp các đặc tả ngôn ngữ theo nghĩa mà trong bài giảng này chúng ta sẽ nêu chính xác hơn. Chẳng hạn, các máy Turing có thể được đồng nhất với các văn phạm cấu trúc câu, các otomat hữu hạn có thể đồng nhất với các văn phạm chính quy.

Môn học otomat và ngôn ngữ hình thức nhằm trang bị cho sinh viên các năm cuối của ngành Tin học các khái niệm về ngôn ngữ hình thức, các otomat, máy Turing…Trên cơ sơ đó, sinh viên có thể hiểu sâu hơn cấu trúc các ngôn ngữ lập trình, các chương trình dịch cũng như bản chất của thuật toán và độ phức tạp tính toán của chúng.

Trong khi chưa có điều kiện biên soạn một giáo trình cho môn học này, chúng tôi tạm thời cung cấp cho sinh viên ngành Tin học tập bài giảng này, để làm tài liệu tham khảo và học tập. Do thời gian biên soạn có hạn nên chắc rằng tập bài giảng này còn nhiều thiếu sót, rất mong nhận được những ý kiến đóng góp của các em sinh viên và đồng nghiệp.

Chương 1

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.

§ 1. Các khái niệm cơ bản về ngôn ngữ hình thức

- 1.1 Bảng chữ cái
- 1.2 Từ
- 1.3 Ngôn ngữ

§ 2. Các phép toán trên các từ

- 2.1 Phép nhân ghép
- 2.2 Phép lấy từ ngược
- 2.3 Phép chia từ

§ 3. Các phép toán trên ngôn ngữ

- 3.1 Phép hợp
- 3.2 Phép giao
- 3.3 Phép lấy phần bù
- 3.4 Phép nhân ghép
- 3.5 Phép lặp
- 3.6 Phép lấy ngôn ngữ ngược
- 3.7 Phép chia ngôn ngữ

§ 4. Văn phạm và ngôn ngữ sinh bởi văn phạm

- 4.1 Định nghĩa văn phạm
- 4.2 Ngôn ngữ sinh bởi văn pham
- 4.3 Phân loại văn phạm theo Chomsky

§ 5. Các tính chất của văn phạm và ngôn ngữ

- 5.1 Tính chất của văn phạm và dẫn xuất
- 5.2 Tính đóng của lớp ngôn ngữ sinh bởi văn phạm

§1. Các khái niệm cơ bản về ngôn ngữ hình thức

1.1 Bảng chữ cái

Định nghĩa 1.1 Tập Σ khác rỗng gồm hũu hạn hay vô hạn các ký hiệu được gọi là *bảng chữ cái*. Mỗi phần tử $a \in \Sigma$ được gọi là một *chữ cái* hay một *ký hiệu*.

Thí dụ 1.1 Dưới đây là các bảng chữ cái:

1. $\sum = \{a, b, c, \ldots, x, y, z\}$
2. $\Delta = \{\alpha, \beta, \gamma, \delta, \varepsilon, \eta, \varphi, \kappa, \mu, \chi, \nu, \pi, \theta, \rho, \sigma, \tau, \omega, \xi, \psi\}$,
3. $\Gamma = \{0, 1\}$,
4. W = {if, then, else, a, b, c, d, e, f, +, −, ∗, /, =, ≠}.

1.2 Từ

Định nghĩa 1.2 Giả sử có bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_m\}$, một dãy 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 từ hay một xâu trên bảng chữ cái Σ.

Tổng số vị trí của các ký hiệu xuất hiện trong xâu α được gọi là độ dài của từ 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 *từ rỗng* và được ký hiệu là ε. Rõ ràng từ rỗng là từ thuộc mọi bảng chữ cái.

Hai từ $\alpha = a_1a_2\ldots a_n$ và $\beta = b_1b_2\ldots b_m$ 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 từ trên bảng chữ cái Σ được ký hiệu là Σ^*, còn tập mọi từ khác rỗng trên bảng chữ cái Σ 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.

Về cấu trúc đại số thì Σ^* là một vị nhóm tự do 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 1.2

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 $\Sigma = \{a, b, c, \ldots, z\}$.

1.3 Ngôn ngữ

Định nghĩa 1.3 Cho bảng chữ cái Σ, mỗi tập con $L \subseteq \Sigma^*$ được gọi là một ngôn ngữ hình thức (hay 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: $L = \emptyset$ là khác với ngôn ngữ chỉ gồm một từ rỗng: $L = \{\varepsilon\}$.

Thí dụ 1.3

1. Σ^* là ngôn ngữ gồm tất cả các từ trên Σ còn Σ^+ là ngôn ngữ gồm tất cả các từ khác từ trống 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. Các phép toán trên các từ

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

2.1 Phép nhân ghép

Định nghĩa 2.1 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 $\gamma = \alpha.\beta$ (hay $\gamma = \alpha\beta$).

Nhận xét: Từ định nghĩa 2.1, ta thấy:

- Từ rỗng là phần tử đơn vị đối với phép nhân ghép, tức là: $\omega\varepsilon = \varepsilon\omega = \omega$ đúng với mọi từ ω.
- Phép nhân ghép có tính kết hợp, nghĩa là với mọi từ α, β, γ, ta có $(\alpha\beta)\gamma = \alpha(\beta\gamma)$.
- 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 một *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 2.1

1. Trên bảng chữ cái W = {if, then, else, a, b, c, d, e, f, +, −, $*$, /, =, $\neq$}, ta có các từ α = if a+b=c then c$*$d=e và β = 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 Phép lấy từ ngược

Định nghĩa 2.2 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 2.2

1. Cho các từ α = 100110 và β = 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 $\sum = \{a, b, c, \ldots x, y, z\}$, khi đó ta có:
   $(happy)^R = yppah$ và $(oto)^R = oto$.
   Ngoài ra ta có: $|\ (happy)^R\ | = |\ yppah| = |\ happy\ | = 3$.

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 2.3 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 2.4 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 ${}^{\alpha}/_{\gamma} = \beta$
- $({}_{\beta}\backslash^{\alpha})^R = \alpha^R / \beta^R$.
- $({}^{\alpha}/_{\gamma})^R = \gamma^R \backslash \alpha^R$.

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

Thí dụ 2.3 Cho các từ α = abcaabbcc, β = abc, γ = bcc trên bảng chữ cái $\sum = \{a, b, c\}$, khi đó ta có

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

§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 ta có các phép toán đại số tập hợp như là phép giao, phép hợp, 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 sau đây trên bảng chữ cái Σ: $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1.L_2$, $\Sigma^* \setminus L_1$.

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

3.1 Phép hợp

Định nghĩa 3.1 Hợp của hai ngôn ngữ L_1 và L_2 trên bảng chữ cái $\sum$, ký hiệu $L_1 \cup L_2$, là một ngôn ngữ trên bảng chũ cái $\sum$, 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.

3.2 Phép giao

Định nghĩa 3.2 Giao của hai ngôn ngữ L_1 và L_2 trên bảng chữ cái $\sum$, ký hiệu $L_1 \cap L_2$, là một ngôn ngữ trên bảng chữ cái $\sum$, 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 ràng, 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.

3.3 Phép lấy phần bù

Định nghĩa 3.3 Ngôn ngữ phần bù của ngôn ngữ L trên bảng chữ cái Σ, ký hiệu $C_\Sigma L$ (hay đơn giản là CL, nếu không gây nhầm lẫn), là một ngôn ngữ trên bảng chữ cái $\sum$, 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 3.1

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 $L = \{\omega \in \sum^*$, với $|\omega|$ là một số chẵn $\}$, khi đó ta có:

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

3.4 Phép nhân ghép

Định nghĩa 3.4 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:

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

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:

  $(L_1L_2)L_3 = L_1(L_2L_3)$.

$\emptyset L = L\emptyset = \emptyset$, $\{\varepsilon\}L = L\{\varepsilon\} = L$,

- 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 3.2). 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 3.2 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: $L_2 \cap L_3 = \emptyset$, do đó:

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

   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:

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

   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 $L_1 \cap (L_2L_3) \neq (L_1 \cap L_2)(L_1 \cap L_3)$. Tức là phép giao không có tính phân phối đối với phép nhân ghép.

Vì phép 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:

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

3.5 Phép lặp

Định nghĩa 3.5 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, 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 3.3

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

3.6 Phép lấy ngôn ngữ ngược

Định nghĩa 3.6 Cho ngôn ngữ L trên bảng chữ cái Σ, khi đó ngôn ngữ ngược của L là một ngôn ngữ trên bảng chữ cái Σ, 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 3.4 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.

3.7 Phép chia ngôn ngữ

Định nghĩa 3.7 Cho ngôn ngữ X và Y trên bảng chữ cái Σ, khi đó thương bên trái của ngôn ngữ X cho ngôn ngữ Y là một ngôn ngữ trên ∑, 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 3.8 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:

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

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

Thí dụ 3.5 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}

§4. Văn phạm và ngôn ngữ sinh bởi văn phạm

Mở đầu

Ta có thể hình dung một văn phạm như một "thiết bị tự động" mà nó có khả năng sinh ra một tập hợp các từ trên một bảng chữ cái cho trước. Mỗi từ được sinh ra sau một số hữu hạn bước thực hiện các quy tắc của văn phạm.

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.

Cách 2. "Thiết bị tự động" có khả năng lần lượt sinh ra tất cả các từ trong ngôn ngữ đã cho.

Cách 3. 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 văn phạm, người ta đã chứng minh được rằng ba cách thức trên là tương đương nhau hay văn phạm làm việc theo các cách trên là tương đương nhau. Vì vậy, ở đây 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" đó là văn phạm sinh.

Việc sinh ra các từ có thể được thực hiện bằng nhiều cách khác nhau. Các từ có thể được sinh ra bởi các văn phạm, bởi các Otomat, bởi các máy hình thức như máy Turing, …Ở đây ta đề cập đến cách của CHOMSKY đưa ra vào những năm 1956-1957.

4.1. Định nghĩa văn phạm

Định nghĩa 4.1 *Văn phạm* G là một bộ sắp thứ tự gồm 4 thành phần:

$$G = < \Sigma, \Delta, S, P >,$$

trong đó:

+ Σ là một bảng chữ cái, gọi là *bảng chữ cái cơ bản (*hay *bảng chữ cái kết thúc)*, mỗi phần tử của nó được gọi là một *ký hiệu kết thúc* hay *ký hiệu cơ bản*;

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

+ $S \in \Delta$ được gọi là *ký hiệu xuất phát hay tiên đề*;

+ P là tập hợp các *quy tắc sinh* có dạng $\alpha \to \beta$, α được gọi là *vế trái* và β được gọi là *vế phải* của quy tắc này, với $\alpha, \beta \in (\Sigma \cup \Delta)^*$ và *trong α chứa ít nhất một ký hiệu không kết thúc*.

$$P = \{\alpha \to \beta \mid \alpha = \alpha' A \alpha'', \text{ với } A \in \Delta, \alpha', \alpha'', \beta \in (\Sigma \cup \Delta)^* \}$$

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

Thí dụ 4.1 Các bộ bốn sau là các văn phạm:

1. $G_1 = <\{0, 1\}, \{S\}, S, \{S\to 0S1, S\to\varepsilon\}>$,
2. $G_2 = <\{a, b\}, \{S, A\}, S, \{S\to Ab, A\to aAb, A\to\varepsilon\}>$,
3. $G_3 = <\{a, b, c\}, \{S, A, B, C\}, S, \{S\to ABC, A\to aA, B\to bB, C\to cC, A\to a, B\to b, C\to c\}>$
4. $G_4 = <\Sigma, \Delta, S, P>$, trong đó:

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

   Δ = {<câu>, <chủngữ>, <vịngữ>, <độngtừ1>, <độngtừ2>, <danhtừ1>, <danhtừ2>},

   S = <câu>,

   P = {<câu>→<chủngữ><vịngữ>, <chủngữ>→tôi, <chủngữ>→anh, <chủngữ>→chị, <vịngữ>→<độngtừ1><danhtừ1>, <vịngữ>→<độngtừ2><danhtừ2>, <độngtừ1>→ăn, <độngtừ2>→uống, <danhtừ1>→cơm, <danhtừ1>→phở, <danhtừ2>→sữa, <danhtừ2>→café}.

Chú ý: Nếu các quy tắc có vế trái giống nhau có thể viết gọn lại: hai quy tắc $\alpha\to\beta$, $\alpha\to\gamma$ có thể được viết là $\alpha\to\beta \mid \gamma$. Chẳng hạn, như trong văn phạm G_1 ở thí dụ 4.1, ta có thể viết hai quy tắc của nó dưới dạng $S\to 0S1 \mid \varepsilon$.

4.2 Ngôn ngữ sinh bởi văn phạm

Định nghĩa 4.2 Cho văn phạm $G = < \Sigma, \Delta, S, P >$ và $\eta, \omega\in(\Sigma \cup \Delta)^*$. Ta nói ω được *suy dẫn trực tiếp* từ η trong G, ký hiệu $\eta \vdash^G \omega$ hay ngắn gọn là $\eta \vdash \omega$ (nếu không sợ nhầm lẫn), nếu tồn tại quy tắc $\alpha\to\beta\in P$ và $\gamma, \delta\in(\Sigma \cup \Delta)^*$ sao cho $\eta = \gamma\alpha\delta$, $\omega = \gamma\beta\delta$.

Điều này có nghĩa là nếu η nhận vế trái α của quy tắc $\alpha\to\beta$ như là từ con thì ta thay α bằng β để được từ mới ω.

Định nghĩa 4.3 Cho văn phạm $G = < \Sigma, \Delta, S, P >$ và $\eta, \omega\in(\Sigma \cup \Delta)^*$. Ta nói ω được *suy dẫn* từ η trong G, ký hiệu $\eta \models^G \omega$ hay ngắn gọn là $\eta \models \omega$ (nếu không sợ nhầm lẫn), nếu $\eta = \omega$ hoặc tồn tại một dãy $D = \omega_0, \omega_1,\dots, \omega_k\in(\Sigma \cup \Delta)^*$ sao cho $\omega_0 = \eta$, $\omega_k = \omega$ và $\omega_{i-1} \vdash \omega_i$, với i = 1, 2,..., k.

Dãy $D = \omega_0, \omega_1, \dots, \omega_k$ được gọi là một *dẫn xuất* của ω từ η trong G và số k được gọi là *độ dài* của dẫn xuất này. Nếu $\omega_0 = S$ và $\omega_k \in \Sigma^*$ thì dãy D gọi là dẫn xuất đầy đủ.

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 4.4 Cho văn phạm $G = < \Sigma, \Delta, S, P >$. Từ $\omega\in\Sigma^*$ được gọi là *sinh bởi* văn phạm G nếu tồn tại suy dẫn $S \models \omega$. Ngôn ngữ *sinh bởi* văn phạm G, ký hiệu L(G), là tập hợp tất cả các từ sinh bởi văn phạm G:

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

Định nghĩa 4.5 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 4.2

1. Xét văn phạm G_1 trong thí dụ 4.1. Từ $\omega = 00001111$ được suy dẫn từ S bằng dãy dẫn xuất độ dài 5: $S \vdash 0S1 \vdash 00S11 \vdash 000S111 \vdash 0000S1111 \vdash 00001111$ (có thể viết ngắn gọn là $\omega = 0^4 1^4$).

   Bằng việc sử dụng n lần ($n \geq 0$) quy tắc 1 rồi quy tắc 2, ta có: $S \models 0^n 1^n$.

   Do đó $L(G_1) = \{0^n 1^n \mid n \geq 0\}$.

2. Xét văn phạm G_2 trong thí dụ 4.1. Sử dụng quy tắc 1, rồi n lần ($n \geq 0$) quy tắc 2, sau đó quy tắc 3 để kết thúc, ta có: $S \vdash Ab \models a^n A b^n b \vdash a^n b^{n+1}$.

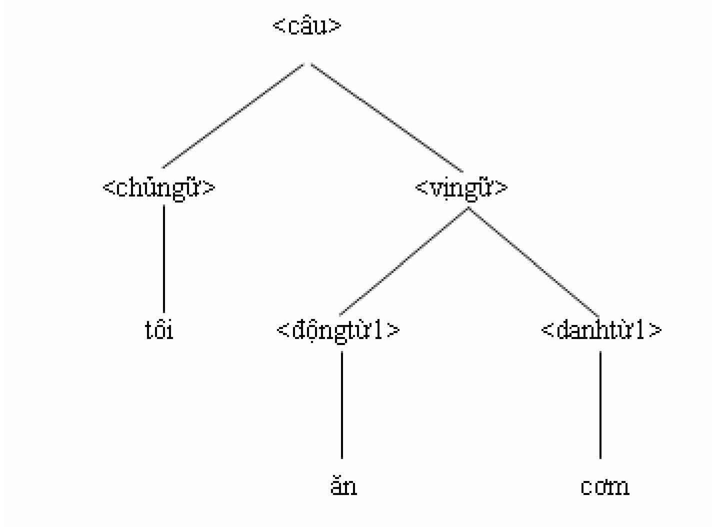
   Do đó $L(G_2) = \{a^n b^{n+1} \mid n \geq 0\}$.

3. Xét văn phạm G_3 trong thí dụ 4.1. Sử dụng quy tắc 1, rồi m -1 lần ($m \geq 1$) quy tắc 2, n-1 lần ($n \geq 1$) quy tắc 3, k-1 lần ($k \geq 1$) quy tắc 4 (các quy tắc có thể xen kẻ), sau đó kết thúc bởi các quy tắc 5, 6, 7, ta có: $S \vdash ABC \models a^m A b^n B c^k C \models a^m b^n c^k$.

   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 <câu> đến một từ trong $L(G_4)$, chẳng hạn "tôi ăn cơm" bằng một cây gọi là *cây dẫn xuất* hay *cây phân tích cú pháp* như dưới đây. Tất nhiên, theo quan điểm phân tích cú pháp thực tế, việc xem xét các quy tắc theo hướng ngược lại là từ phải qua trái. Điều đó có nghĩa là cây dưới đây được xử lý từ dưới lên trên chứ không phải là từ trên xuống dưới. (H.1).



H. 2.1 Cây dẫn xuất cho ví dụ 4.2

Thí dụ 4.3 Cho hai văn phạm $G_3 = <\Sigma, \{S\}, S, P_3>$, $G_4 = <\Sigma, \{S\}, S, P_4>$, trong đó:

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

$P_3 = \{S\rightarrow 1| 2| 3| 4| 5| 6| 7| 8| 9| S0| S1| S2| S3| S4| S5| S6| S7| S8| S9\}$,

$P_4 = \{S\rightarrow 0| 1| 2| 3| 4| 5| 6| 7| 8| 9| 1S| 2S| 3S| 4S| 5S| 6S| 7S| 8S| 9S\}$.

Dễ thấy rằng $L(G_3) = \{n \mid n \geq 1\}$. Thất 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.

4.3 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 (Institute Professor, Massachusetts Institute of Technology. Born December 7, 1928 Philadelphia, Pennsylvania, USA) đã phân loại văn phạm thành bốn nhóm:

- Nhóm 0: Văn phạm không hạn chế (hay văn phạm ngữ cấu, văn phạm tổng quát),
- Nhóm 1: Văn phạm cảm ngữ cảnh,
- Nhóm 2: Văn phạm phi ngữ cảnh,
- Nhóm 3: Văn phạm chính quy.

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

Định nghĩa 4.6 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 có dạng: $\alpha\rightarrow\beta$, 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 4.7 Văn phạm $G = < \Sigma, \Delta, S, P >$ mà các quy tắc của nó đều có dạng: $\alpha\rightarrow\beta$, với $\alpha = \alpha' A\alpha''$, $A \in \Delta$, α', α'', $\beta \in (\Sigma \cup \Delta)^*$, và $| \alpha | \leq | \beta |$, được gọi là văn phạm *nhóm 1hay văn phạm cảm ngữ cảnh.*

Các quy tắc trong văn phạm *nh óm 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.

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.

Thí dụ 4.4 Cho văn phạm $G = <\{a, b, c\}, \{S, A, B, C\}, S, P>$, trong đó:

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

Khi đó G là văn phạm cảm ngữ cảnh.

Sử dụng n-1 lần ($n \geq 1$) 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:

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

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

Định nghĩa 4.8 Văn phạm $G = < \Sigma, \Delta, S, P >$ mà các quy tắc của nó có dạng $A \to \omega$, trong đó $A \in \Delta$, $\omega \in (\Sigma \cup \Delta)^*$, được gọi là văn phạm *nhóm 2*.hay *văn phạm phi ngữ cảnh.*

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.

Thí dụ 4.5

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

   $$P = \{S \to Sa, S \to Aa, A \to aAb, A \to ab\}.$$

   Khi đó 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:

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

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

2. Cho văn phạm $G_2 = <\{0, 1\}, \{S\}, S, \{S \to SS, S \to 0S1, S \to 1S0, S \to \varepsilon\}>$.

   G_2 là văn phạm phi ngữ cảnh. Từ các quy tắc của G_2, ta có $L(G_2) =\{\varepsilon, 01, 10, 0011, 1100, 1001, 111000, \dots\}$ 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 $P_3 = \{S \to \varepsilon, S \to aSa, S \to bSb, S \to aa, S \to bb\}$.

   G_3 là văn phạm phi ngữ cảnh và nó 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 $L(G_3)$.

Định nghĩa 4.9 Văn phạm $G = < \Sigma, \Delta, S, P >$ mà các quy tắc của nó chỉ có dạng $A \to aB$, $A \to a$ (hoặc chỉ có dạng $A \to Ba$, $A \to a$), trong đó $A, B \in \Delta$, $a \in \Sigma$, được gọi là văn phạm *nhóm 3* hay *văn phạm chính quy*.

Các văn phạm mà các quy tắc của chúng có dạng trên, đồng thời chứa thêm quy tắc rỗng $S \to \varepsilon$ cũng được gọi là văn phạm chính quy (hay còn gọi là văn phạm chính quy suy rộng).

Các quy tắc 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*.

Thí dụ 4.6

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

   Khi đó, G_1 là văn phạm chính quy và $L(G_1) = \{1^{2n} \mid n \geq 0\}$. Thật vậy, sử dụng quy tắc 1, ta có $S \vdash 1^{2n}$, ($\varepsilon = 1^{2n}$, với n = 0), sử dụng quy tắc 2, rồi n-1 lần (n ≥ 1) liên tiếp cặp quy tắc 3 và 4, cuối cùng là quy tắc 5, ta có:

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

2. Cho văn phạm $G_2 = <\{0, 1\}, \{S, A\}, S, P_2 >, P_2 = \{S\to 0A, A\to 0A, A\to 1A, A\to 0\}>$.

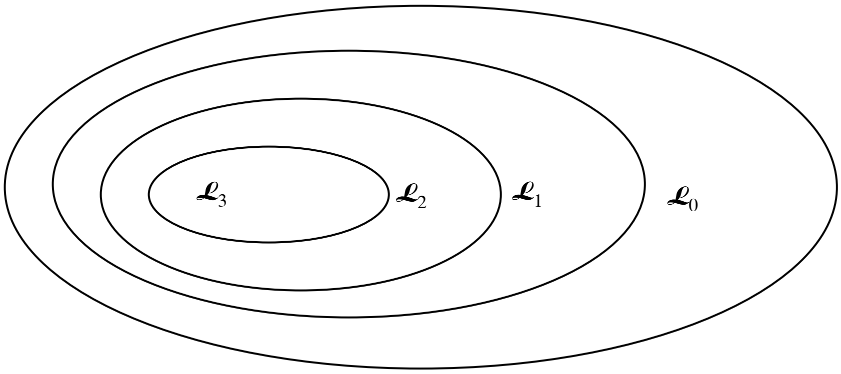
   Khi đó, G_1 là văn phạm chính quy và $L(G_2) = \{0\omega 0 \mid \omega\in\{0, 1\}^*\}$. Thật vậy, sử dụng quy tắc 1, rồi một số hữu hạn lần tuỳ ý, có thể xen kẽ các quy tắc 2 và 3, cuối cùng là quy tắc 4, ta có: $S \vdash 0A \models 0\omega A \vdash 0\omega 0$.

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.

Ngôn ngữ hình thức được gọi là ngôn ngữ tổng quát (hay cảm ngữ cảnh, phi ngữ cảnh, chính quy) nếu tồn tại văn phạm loại tương ứng sinh ra nó. Vì vậy, đối với các lớp ngôn ngữ, nếu ký hiệu $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3$ lần lượt là các lớp ngôn ngữ *tổng quát, cảm ngữ cảnh, phi ngữ cảnh* và *chính quy* thì ta có bao hàm thức:

$$\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0.$$

Hình vẽ dưới đây cho một sự so sánh về độ lớn của các lớp ngôn ngữ theo phân loại của Chomsky, cho thấy lớp ngôn ngữ chính quy $\mathcal{L}_3$ là nhỏ nhất, nó bị chứa thực sụ trong lớp ngôn ngữ phi ngữ cảnh $\mathcal{L}_2$, lớp ngôn ngữ phi ngữ cảnh lại bị chứa thực sự trong lớp ngôn ngữ cảm ngữ cảnh $\mathcal{L}_1$ và cuối cùng lớp ngôn ngữ tổng quát $\mathcal{L}_0$ (ngôn ngữ ngữ cấu) là rộng nhất.



H. 2.2 So sánh các lớp ngôn ngữ

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.

Thí dụ 4.7 Cho bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_n\}$.

Chứng minh rằng các ngôn ngữ: $L_1 = \{\omega = a_1a_2 \ldots a_n\}$, $L_2 = \Sigma^+$, $L_3 = \Sigma^*$, $L = \varnothing$ 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:

$G_1 = <\Sigma, \{S, A_1, \ldots, A_{n-1}\}, S, \{S\to a_1A_1, A_1\to a_2A_2, \ldots, A_{n-2}\to a_{n-1}A_{n-1}, A_{n-1}\to a_n\}>$.

Dễ thấy G_1 là văn phạm chính quy, và $L_1 = L(G_1)$.

$G_2 = <\Sigma, \{S\}, S, \{S\to aS, S\to a \mid a\in\Sigma\}>$, dễ thấy G_2 là văn phạm chính quy, và $L_2 = L(G_2)$.

$G_3 = <\Sigma, \{S, A\}, S, \{S\to\varepsilon, S\to a, S\to aA, A\to aA, A\to a \mid a\in\Sigma\}>$, dễ thấy G_3 là văn phạm chính quy, và $L_3 = L(G_3)$.

$G_4 = <\Sigma, \{S\}, S, \{S\to aS \mid a\in\Sigma\}>$, dễ thấy G_4 là văn phạm chính quy, và nó làm việc không bao giờ dừng, tức là không có $\omega\in\Sigma^*$ sinh bởi G_4, vậy G_4 sinh ra ngôn ngữ $\varnothing$.

§5. Các tính chất của văn phạm và ngôn ngữ sinh bởi văn phạm

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 các văn phạm.

Định lý 5.1 Với mọi văn phạm $G = < \Sigma, \Delta, S, P >$, luôn tồn tại một văn phạm $G' = < \Sigma', \Delta', S', P' >$ tương đương với văn phạm G, tức là $L(G) = L(G')$.

Chứng minh:

Giả sử có văn phạm $G = < \Sigma, \Delta, S, P >$, ta xây dựng văn phạm $G' = < \Sigma', \Delta', S', P' >$, trong đó:

+ $\Sigma' = \Sigma$, và với mỗi $a \in \Sigma$, ta bổ xung một ký hiệu $\overline{a} \notin \Sigma \cup \Delta$ và gọi là đối ngẫu của a, đặt $\Gamma = \{\overline{a} \mid a \in \Sigma\}$

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

+ $S' = S$,

+ $P' = P_1 \cup P_2$, với $P_1 = \{\overline{a}\to a \mid \forall a \in \Sigma\}$, $P_2 = \{\overline{\alpha} \to \overline{\beta} \mid \forall\alpha\to\beta \in P\}$, $\overline{\alpha}$ và $\overline{\beta}$ là các xâu α và β đã được thay các ký hiệu thuộc Σ bằng các ký hiệu đối ngẫu của nó. Dễ thấy rằng $L(G) = L(G')$, thật vậy ta sẽ chứng minh hai bao hàm thức:

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

b./ *Chứng minh L(G') ⊆ L(G)*: Lấy bất kỳ $\omega \in L(G')$, khi đó ta có $S \models^{G'} \omega$, tức là ta có một dãy suy dẫn trong G': $S = \omega'_0 \vdash^{G'} \omega'_1 \vdash^{G'} \ldots \vdash^{G'} \omega'_k = \omega$, trong các suy dẫn $\omega_i \vdash^{G'} \omega_{i+1}$, ($0 \leq i \leq k-1$), ta thay mọi kí hiệu $\overline{a} \in \Gamma$ bởi các ký hiệu tương ứng $a \in \Sigma_1$, khi đó mọi quy tắc đều thuộc P, ta nhận được dãy các suy dẫn trưc 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 5.1 Cho văn phạm $G_1 = <\{a, b\}, \{S\}, S, \{S\to aSb, S\to ab\}>$, ta có thể xây dựng G_2 tương đương với G_1 như sau:

$$G_2 = <\{a, b\}, \{S, A, B\}, S, \{S\to ASB, A\to a, B\to b, S\to AB\}>.$$

Dễ dàng có được $L(G_1) = L(G_2) = \{a^n b^n \mid n \geq 1\}$, hay 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, để nhận được một văn phạm tương đương, nhờ bổ đề sau:

Bổ đề 5.1 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 $P' = P \cup \{S'\to\alpha \mid S\to\alpha \in P\}$. 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 $S\to\alpha\in P$, do đó $S'\to\alpha\in P'$ 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 $S'\to\alpha \in P'$, do đó tồn tại $S\to\alpha \in P$. 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 $L(G') \subseteq L(G)$.

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 cơ bản ở vế trái các quy tắc, nhờ bổ đề sau:

Bổ đề 5.2 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 cơ bản ở vế trái.

Chứng minh:

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

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

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

$P_2 = \{ \overline{\alpha} \to \overline{\beta} \mid \forall \alpha \to \beta \in P \}$, $\overline{\alpha}$ và $\overline{\beta}$ là các xâu α và β đã được thay các ký hiệu $a \in \Sigma$ (mà đã xuất hiện ở vế trái một quy tắc nào đó) , bằng các ký hiệu đối ngẫu $\overline{a}$ của nó.

Xây dựng văn phạm G’ = < Σ’, Δ’, S’, P’ >, với:

Σ’ = Σ,

Δ’ = Δ ∪ Γ,

S’ = S,

$P' = P_1 \cup P_2$

Văn phạm G’ sẽ là văn phạm tương đương với văn phạm G (theo định lý 5.1), hơn nữa, theo cách xây dựng thì trong tất cả các vế trái của G’ sẽ không chứa ký hiệu cơ bản.

Vậy bổ đề được chứng minh.

Ta đưa ra hai khái niệm về dẫn xuất:

Định nghĩa 5.1 Cho văn phạm G = < Σ, Δ, S, P > và hai dãy 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 5.2 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 5.2 Với mọi dẫn xuất trong văn phạm G tùy ý, luôn luôn tồn tại một dẫn xuất không lặp và đồng lưc 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.

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 “∘” 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 ∘. Lớp ngôn ngữ sinh bởi văn phạm là đóng đối với hầu hết các phép toán trên ngôn ngữ mà ta đã học trong §3, dưới đây ta chỉ xét tính đóng đối với một số phép toán quan trọng nhất.

Định lý 5.3 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 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 sinh ra ngôn ngữ $L_1 \cup L_2$ như sau: $G = <\Sigma, \Delta, S, P>$, với:

$\Sigma = \Sigma_1 \cup \Sigma_2$

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

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

Ta sẽ chứng minh $L(G) = L_1 \cup L_2$ 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:

+ 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)

+ 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 $\omega \in L_1 = L(G_1)$, khi đó ta có suy dẫn $S_1 \models^{G1} \omega$ 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 $\omega \in L_2 = L(G_2)$, khi đó ta có suy dẫn $S_2 \models^{G2} \omega$ trong G_2, do đó ta cũng 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:

$\Sigma = \Sigma_1 \cup \Sigma_2$

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

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

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: $\Gamma_1 = \{ \overline{a} \mid a \in \Sigma_1\}$ là tập các ký hiệu đối ngẫu của các ký hiệu trong Σ_1, còn $\Gamma_2 = \{\overline{b} \mid b \in \Sigma_2\}$ là tập các ký hiệu đối ngẫu của Σ_2.

$P = \overline{P_1} \cup \overline{P_2} \cup \{S \to S_1S_2\} \cup P' \cup P''$, 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 $\overline{a} \in \Gamma_1$, $\overline{P_2}$ 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 $\overline{b} \in \Gamma_2$, và:

$P' = \{ \overline{a}\,\overline{b} \to \overline{b}\,\overline{a} \mid a \in \Sigma_1, b \in \Sigma_2\}$,

$P'' = \{ \overline{a}\,\overline{a} \to a \mid a \in \Sigma_1 \cap \Sigma_2\}$.

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 trên ngôn ngữ: 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 5.1 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 5.2 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 $\Sigma = \{a, b, c\}$, có thể thấy rằng L_1 và L_2 lần lượt được sinh bởi các văn phạm sau đây:

$G_1 = <\Sigma, \{S_1, A, B\}, S_1, \{S_1 \to AS_1B, S_1 \to c, A \to a, B \to bb\}>$,

$G_2 = <\Sigma, \{S_2, C, D\}, S_2, \{S_2 \to CS_2D, S_2 \to c, C \to aa, D \to b\}>$.

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, ta có:

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

Tương tự, trong G_2 ta có $S_2 \models^{G2} a^{2n}cb^n$. ($n \geq 0$)

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, do đó theo hệ quả 5.1 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 5.2 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 (phi ngữ cảnh, cảm ngữ cảnh).

Thí dụ 5.3

1. Cho hai ngôn ngữ $L_1 = \{a^n b^n \mid n \geq 1\}$ 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:

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

   $G_2 = <\{c\}, \{S_2\}, S_2, \{S_2 \to cS_2, S_2 \to c\}>$ 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 $L_1L_2 = \{a^n b^n c^m \mid n \geq 1, m \geq 1\}$ là ngôn ngữ phi ngữ cảnh.

2. Cho hai ngôn ngữ chính quy $L_3 = \{ba^n \mid n \geq 0\}$ và $L_4 = \{b^n a \mid n \geq 0\}$. 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:

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

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

Khi đó theo hệ quả 5.2, ta sẽ có $L_3L_4 = \{ba^n b^m a \mid n \geq 0, m \geq 0\}$ là ngôn ngữ chính quy.

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

Hệ quả 5.3 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 4.7 (ngôn ngữ một từ là chính quy) và từ hệ quả 5.1 (hợp hữu hạn của các ngôn ngữ chính quy là chính quy), ta có hệ quả sau:

Hệ quả 5.4 Mọi ngôn ngữ hữu hạn đều là ngôn ngữ chính quy.

Thí dụ 5.4 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 $P = \{S \to 0, S \to 0A, A \to 1, A \to 1B, B \to 1, B \to 1C, C \to 1\}$.

Dễ dàng thấy rằng L(G) = L.

Bài tập chương 1

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. Tìm cách biểu diễn hữu hạn cho các ngôn ngữ vô hạn sau đây:

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. L_1, L_2 là ngôn ngữ loại nào theo phân loại Chomsky?

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

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

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

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

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

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

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

7. Cho ngôn ngữ $L = \{a^n b^n c^m \mid n, m \geq 1 \}$. 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 $L = \{\omega b\omega^R \mid \omega \in \Sigma^* = \{a_1, a_2, \ldots, a_k, b\}^*, \omega^R$ là từ ngược của $\omega\}$. Xây dựng văn phạm phi ngữ cảnh G sinh ngôn ngữ L.

11. Cho các văn phạm:

a/. G_1 với tập quy tắc $P_1 = \{S \to aS, S \to Sb, S \to aSb, S \to c\}$

b/. G_2 với tập quy tắc $P_2 = \{S \to SS, S \to a, S \to b\}$,

c/. G_3 với tập quy tắc $P_3 = \{S \to aA, S \to bB, A \to Sa, B \to Sb, S \to c\}$,

d/. G_4 với tập quy tắc $P_4 = \{S \to AB, A \to Sc, A \to a, B \to dB, B \to b\}$,

e/. G_5 với tập quy tắc $P_5 = \{S \to SaS, S \to b\}$,

f/. G_6 với tập quy tắc $P_6 = \{S \to aSS, S \to b\}$,

g/. G_7 với tập quy tắc $P_7 = \{S \to AA, A \to aAa, A \to bAb, A \to c\}$.

Hỏi: 1/. Hãy phân loại 7 văn phạm trên theo nhóm 0, 1, 2, 3 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 $\Sigma = \{a, b\}$, viết các văn phạm sinh các ngôn ngữ:

$L_4 = \{\omega$, với $|\omega|$ là một số chẵn $\}$,

$L_5 = \{\omega$, với $|\omega|$ là một số lẻ$\}$.

Phân loại L_4 và L_5 theo Chomsky.

13. Hãy xác định xem các văn phạm dưới đây sinh ra các ngôn ngữ nào?

a/. $G_1 = <\{0, 1\}, \{S, A\}, S, \{S\to 0A, A\to 1S, S\to\varepsilon\}>$.

b/. $G_2 = <\{a, b\}, \{S\}, S, \{S\to SaS, S\to b\}>$.

c/. $G_3 = <\{a, b, c\}, \{S\}, S, \{S\to aca, S\to bcb, S\to aSa, S\to bSb\}>$.

d/. $G_4 = <\{0, 1, 2, \ldots, 9\}, \{S, A\}, S, \{S\to SA \mid A, A\to 0|1|2|3|4|5|6|7|8|9\}>$.

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

a/. $L_6 = \{\omega \in \{a\}^*$, và $|\omega| \bmod 3 = 0\}$. (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/. $L_7 = \{a^{2n+1} \mid n \geq 0\}$.

c/. $L_8 = \{a^m b^n \mid n \geq 0, m \geq n\}$.

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

a/. $L_9 = \{0\omega 1 \mid \omega \in \Sigma^*\}$.

b/. $L_{10} = \{1\}^*\{010\}\{0\}^*$

c/. $L_{11} = \{010\}^* \cup \{1100\}^*$

d/. $L_{12} = \{a^m b^n c^k \mid m \geq 0, n \geq 0, k \geq 0\}$

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

16. 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 $a\omega a$ cũng là một xâu hình tháp.

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

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

1/. 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?

2/. 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?

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

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

Theo phương pháp chứng minh trong định lý 5.5:

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

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

3/. Hãy xây dựng văn phạm $G_4 = <\Sigma_5, \Delta_5, S_5, P_5>$ sao cho $L(G_5) = L(G_1) \cap L(G_2)$.

Chương 2

OTOMAT HỮU HẠN VÀ NGÔN NGỮ CHÍNH QUY

Trong chương này, chúng ta sẽ nghiên cứu một mô hình "máy trừu tượng" để đoán nhận ngôn ngữ, đó là các otomat hữu hạn. Chúng ta sẽ thấy rằng lớp ngôn ngữ được đoán nhận bởi otomat hữu hạn khá đơn giản, đó chính là lớp ngôn ngữ chính quy do văn phạm chính quy sinh ra. Chương này gồm các nội dung chủ yếu sau:

§ 1. Otomat hữu hạn đơn định

- 1.1 Otomat hữu hạn đơn định
- 1.2 Biểu diễn otomat hữu hạn đơn định
- 1.3 Ngôn ngữ được đoán nhận bởi otomat đơn định

§ 2. Otomat hữu hạn không đơn định

- 2.1 Otomat hữu hạn không đơn đinh
- 2.2 Ngôn ngữ đoán nhận bởi otomat không đơn định
- 2.3 Đơn định hóa các otomat
- 2.4 Sự tương đương giữa các otomat đơn định và không đơn định

§ 3. Ngôn ngữ chính quy và biểu thức chính quy

- 3.1 Ngôn ngữ chính quy và biểu thức chính quy
- 3.2 Sự liên hệ giữa otomat hữu hạn và ngôn ngữ chính quy

§ 4. Điều kiện cần của ngôn ngữ chính quy

- 4.1 Otomat tối tiểu
- 4.2 Điều kiện cần của ngôn ngữ chính quy

§1. Otomat hữu hạn đơn định

Mở đầu

Một otomat hữu hạn là một mô hình tính toán thực sự hữu hạn. Mọi cái liên quan đến nó đều có kích thước hữu hạn cố định và không thể mở rộng trong suốt quá trình tính toán. Các loại otomat khác được nghiên cứu sau này có ít nhất một bộ nhớ vô hạn về tiềm năng. Sự phân biệt giữa các loại otomat khác nhau chủ yếu dựa trên việc thông tin có thể được đưa vào bộ nhớ như thế nào.

Một otomat hữu hạn làm việc theo thời gian rời rạc như tất cả các mô hình tính toán khác. Như vậy, ta có thể nói về thời điểm "kế tiếp" khi "đặc tả" hoạt động của một otomat hữu hạn.

Trường hợp đơn giản nhất là thiết bị không có bộ nhớ mà ở mỗi thời điểm, thông tin ra chỉ phụ thuộc vào thông tin vào lúc đó. Các thiết bị như vậy là mô hình của các mạch tổ hợp.

Tuy nhiên, nói chung, thông tin ra sản sinh bởi một otomat hữu hạn phụ thuộc vào cả thông tin vào hiện tại lẫn các thông tin vào trước đó. Như vậy otomat có khả năng (với một phạm vi nào đó) ghi nhớ các thông tin vào trong quá khứ của nó. Một cách chi tiết hơn, điều đó có nghĩa như sau.

Mỗi otomat có một số hữu hạn trạng thái được lưu ở bộ nhớ trong. Tại mỗi thời điểm i, nó ở một trong các trạng thái đó, chẳng hạn q_i. Trạng thái q_{i+1} ở thời điểm sau được xác định bởi q_i và thông tin vào a_i cho ở thời điểm i. Thông tin ra ở thời điểm i được xác định bởi trạng thái q_i (hay bởi cả a_i và q_i).

1.1 Otomat hữu hạn đơn định

Định nghĩa 1.1 Một otomat hữu hạn đơn định (Deterministic Finite Automata-DFA) là một bộ năm:

$$A = <Q, \Sigma, \delta, q_0, F>,$$

trong đó:

+ Q là một tập hữu hạn khác rỗng, được gọi là tập các trạng thái;

+ Σ là một bảng chữ cái, được gọi là bảng chữ vào;

+ δ: D $\to$ Q, là một ánh xạ từ D vào Q, trong đó $D \subseteq Q \times \Sigma$, được gọi là hàm chuyển trạng thái (hay hàm chuyển);

+ $q_0 \in Q$, được gọi là trạng thái khởi đầu;

+ $F \subseteq Q$ được gọi là tập các trạng thái kết thúc.

Trong trường hợp $D = Q \times \Sigma$, ta nói A là otomat đầy đủ. Sau này ta sẽ thấy rằng mọi otomat hữu hạn đều đưa về được otomat hữu hạn đầy đủ tương đương.

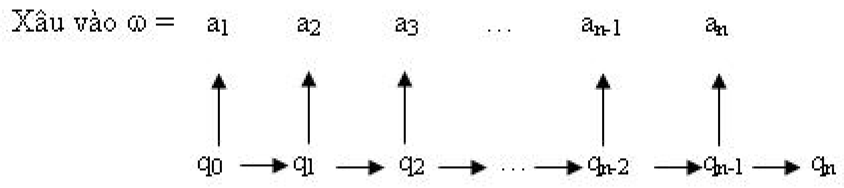
Hoạt động của otomat hữu hạn đơn định $A = <Q, \Sigma, \delta, q_0, F>$ khi cho xâu vào $\omega = a_1a_2\ldots a_n$ có thể được mô tả như sau:

Khi bắt đầu làm việc, otomat ở trạng thái khởi đầu q_0 và đầu đọc đang nhìn vào ô có ký hiệu a_1. Tiếp theo otomat chuyển từ trạng thái q_0 dưới tác động của ký hiệu vào a_1 về trạng thái mới $\delta(q_0, a_1) = q_1 \in Q$ và đầu đọc chuyển sang phải một ô, tức là nhìn vào ô có ký hiệu a_2. Sau đó otomat A có thể lại tiếp tục chuyển từ trạng thái q_1 nhờ hàm chuyển δ về trạng thái mới $q_2 = \delta(q_1, a_2) \in Q$. Quá trình đó sẽ tiếp tục cho tới khi gặp một trong các tình huống sau:

– Otomat A đọc hết xâu vào ω và $\delta(q_{n-1},a_n) = q_n \in F$, ta nói rằng A đoán nhận xâu ω.

– Hoặc otomat A đọc hết xâu vào ω và $\delta(q_{n-1},a_n) = q_n \notin F$, ta nói A không đoán nhận xâu ω.

– Hoặc khi otomat A đọc đến a_j , $(j \le n)$ và hàm $\delta(q_{j-1}, a_j)$ không xác định, ta cũng nói A không đoán nhận xâu ω.



H. 3.1. Mô tả quá trình đoán nhận xâu ω của otomat A

1.2 Biểu diễn otomat hữu hạn đơn định

Hàm chuyển trạng thái là một bộ phận quan trọng của một otomat hữu hạn đơn định. Cho một otomat thực chất là cho hàm chuyển trạng thái của nó, có thể cho dưới dạng bảng chuyển hoặc cho dưới dạng đồ thị chuyển.

Cho otomat bằng bảng chuyển

Cho ôtômát $A = <Q, \Sigma, \delta, q_0, F>$, với $Q = \{q_0, q_1, q_2, \dots , q_m \}$ là tập trạng thái, và bảng chữ cái $\Sigma = \{a_1, a_2, \dots , a_n\}$, khi đó hàm chuyển có thể cho bởi bảng sau; trong đó dòng i cột j của bảng là ô trống nếu $(q_i, a_j) \notin D$, tức là $\delta(q_i,a_j)$ không xác định.

Trạng thái	Ký hiệu vào			
	a_1	a_2	…	a_n
q_0	$\delta(q_0, a_1)$	$\delta(q_0, a_2)$	…	$\delta(q_0, a_2)$
q_1	$\delta(q_1, a_1)$	$\delta(q_1, a_2)$	…	$\delta(q_1, a_2)$
q_2	$\delta(q_2,a_1)$	$\delta(q_2, a_2)$	…	$\delta(q_2, a_2)$
…			…	
q_m	$\delta(q_m, a_1)$	$\delta(q_m, a_2)$	…	$\delta(q_m, a_2)$

H. 3.2. Bảng chuyển trạng thái của otomat A

Cho bảng chuyển trạng thái, và chỉ rõ tập trạng thái kết thúc F, ta sẽ xác định được otomat A

Cho otomat bằng đồ thị chuyển

Cho otomat $A = <Q, \Sigma, \delta, q_0, F>$. Hàm chuyển δ có thể cho bằng một đa đồ thị có hướng, có khuyên G sau đây, được gọi là đồ thị chuyển của otomat A. Tập đỉnh của G được gán nhãn bởi các phần tử thuộc Q, còn các cung được gán nhãn bởi các phần tử thuộc Σ, tức là nếu $a \in \Sigma$ và từ trạng thái q chuyển sang trạng thái p theo công thức $\delta(q, a) = p$ thì sẽ có một cung từ đỉnh q tới đỉnh p được gán nhãn a.

Đỉnh vào của đồ thị chuyển là đỉnh ứng với trạng thái ban đầu q_0. Các đỉnh sẽ được khoanh bởi các vòng tròn, tại đỉnh q_0 có mũi tên đi vào, riêng đỉnh với trạng thái kết thúc được phân biệt bởi vòng tròn đậm, hoặc hình vuông…

Nói chung, với việc cho đồ thị chuyển là hoàn toàn xác định được otomat A.

Thí dụ 1.1 Cho hai otomat hữu hạn đơn định:

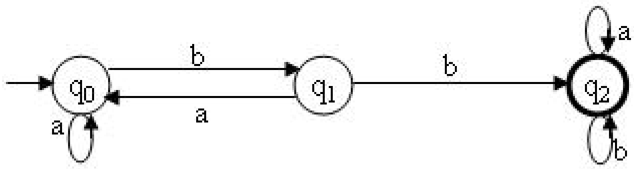
1/. $A_1 = <\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\}>$,

Với $\delta(q_0, a) = q_0$, $\delta(q_0, b) = q_1$, $\delta(q_1, a) = q_0$, $\delta(q_1, b) = q_2$, $\delta(q_2, a) = q_2$, $\delta(q_2, b) = q_2$.

Ta có bảng chuyển trạng thái và đồ thị chuyển trạng thái của otomat A_1 như sau:

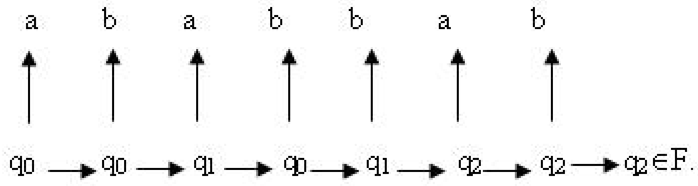
Trạng thái	Ký hiệu vào a	b
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2

H. 3.3 Bảng chuyển trạng thái của A_1



H. 3.4 Đồ thị chuyển trạng thái của A_1

Dãy trạng thái của otomat A_1 trong quá trình đoán nhận xâu vào α = ababbab là:



H. 3.5 Quá trình đoán nhận xâu α = ababbab của A_1

Như vậy, xâu α được đoán nhận bởi otomat A_1.

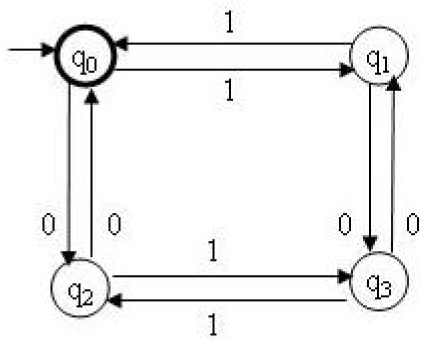
2/. $A_2 = <\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\}>$,

trong đó $\delta(q_0, 0) = q_2$, $\delta(q_0, 1) = q_1$, $\delta(q_1, 0) = q_3$, $\delta(q_1, 1) = q_0$, $\delta(q_2, 0) = q_0$, $\delta(q_2, 1) = q_3$, $\delta(q_3, 0) = q_1$, $\delta(q_3, 1) = q_2$.

Ta có bảng chuyển trạng thái và đồ thị chuyển trạng thái của otomat A_2 được cho trong hình 3.6 và 3.7:

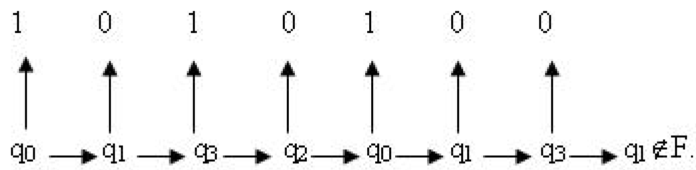
Trạng thái	Ký hiệu vào 0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

H. 3.6 Bảng chuyển trạng thái của A_2



H. 3.7 Đồ thị chuyển trạng thái của A_1

Dãy trạng thái của otomat A_2 trong quá trình đoán nhận xâu vào β = 1010100 là:



H. 3.8 Quá trình đoán nhận xâu vào β = 1010100

Như vậy, otomat A_2 không chấp nhận xâu β.

Ta có thể mô tả quá trình đoán nhận xâu vào của otomat hữu hạn đơn định đầy đủ A bằng thuật toán mô phỏng sau:

Input :

– Một xâu ω, kết thúc bởi ký hiệu kết thúc file là eof.

– Một otomat hữu hạn đơn định đầy đủ A với trạng thái đầu q_0 và tập trạng thái kết thúc là F.

Output:

- Trả lời "Đúng" nếu A đoán nhận xâu ω.

- Trả lời "Sai" nếu A không đoán nhận xâu ω.

Thuật toán:

```
Begin
        S:= q0;
        C:= ký hiệu tiếp theo;
        While C < > eof do
                begin
                        S:= δ(S, C);
                        C:= ký hiệu tiếp theo;
                end;
        if S in F return (True)
        else return (False);
End.
```

1.3 Ngôn ngữ được đoán nhận bởi otomat đơn định

Để mô tả hình thức quá trình đoán nhận một từ (xâu vào), người ta đưa vào ánh xạ mở rộng δ' từ $D \subseteq Q\times\Sigma^*$ vào Q như trong định nghĩa sau:

Định nghĩa 1.2 Cho otomat hữu hạn đơn định $A = <Q, \Sigma, \delta, q_0, F>$. Mở rộng δ' của δ là một ánh xạ từ $D \subseteq Q\times\Sigma^*$ vào Q được xác định như sau:

1/. $\delta'(q, \varepsilon) = q, \forall q\in Q$,

2/. $\delta'(q, \omega a) = \delta(\delta'(q, \omega), a), \forall a\in\Sigma, \forall q\in Q, \forall \omega \in \Sigma^*$ sao cho $\delta'(q, \omega)$ được xác định.

Chú ý rằng, ánh xạ δ chỉ khác ánh xạ δ' khi ký hiệu vào là ε, hoặc là một xâu kí hiệu vào ω, do điều kiện 2/. trên $Q\times\Sigma$, ta có thể đồng nhất δ' với δ. Nếu không cần phân biệt, từ đây về sau ta viết δ thay cho δ', và được hiểu là ánh xạ δ trên miền $Q\times\Sigma$, là ánh xạ δ' trên miền $Q\times\Sigma^*$

Định nghĩa 1.3 Cho otomat hữu hạn đơn định $A = <Q, \Sigma, \delta, q_0, F>$, và một xâu $\omega\in\Sigma^*$. Ta nói:

+ ω được đoán nhận bởi A nếu $\delta(q_0, \omega) \in F$;

+ Ngôn ngữ được đoán nhận bởi otomat A và ký hiệu là T(A), là tập từ:

$$T(A) = \{\omega\in\Sigma^* \mid \delta(q_0, \omega)\in F\}$$

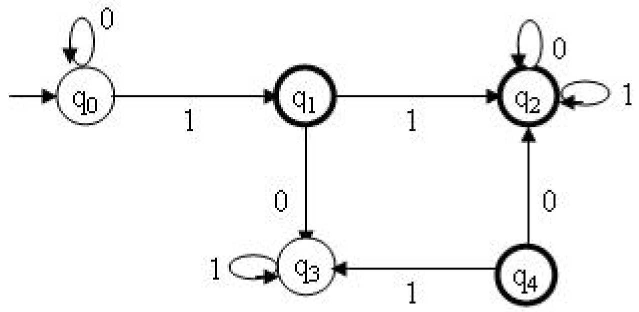
Lưu ý rằng trong đồ thị chuyển của A, $\omega \in \Sigma^*$ được đoán nhận bởi A khi và chỉ khi ω là xâu của các nhãn ứng với một đường đi từ đỉnh q_0 đến một trong các đỉnh kết thúc. Cụ thể, nếu $\omega = a_1a_2\ldots a_n$ thì đường đi là $(q_0, q_1, \ldots, q_k)$ với cung (q_{i-1}, q_i) có nhãn a_i (với $1 \le i \le k$) và $q_k \in F$.

Như vậy, T(A) là tập hợp tất cả xâu ghi trên các đường đi từ q_0 đến các đỉnh kết thúc.

Định nghĩa 1.4 Hai otomat hữu hạn A = <Q, Σ, δ, q_0, F> và A'= <Q', Σ', δ', q'_0, F'> được gọi là tương đương nếu T(A) = T(A').

Thí dụ 1.2 Cho otomat hữu hạn: A_3 = <{q_0, q_1, q_2, q_3, q_4},{0, 1}, δ, q_0, {q_1, q_2, q_4}> với $\delta(q_0,0) = q_0$, $\delta(q_0,1) = q_1$, $\delta(q_1,0) = q_3$, $\delta(q_1,1) = q_2$, $\delta(q_2,0) = q_2$, $\delta(q_2,1) = q_2$, $\delta(q_3,1) = q_3$, $\delta(q_4,0) = q_2$, $\delta(q_4,1) = q_3$.

Đồ thị chuyển của A_3 là:



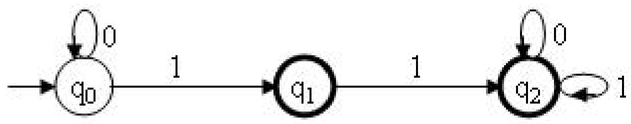
H. 3.9 Đồ thị chuyển của otomat A_3

Trước hết, ta nhận thấy rằng không có đường đi từ q_0 đến đỉnh kết thúc q_4, tức là sẽ không có từ nào được đoán nhận bởi A_3 với đỉnh kết thúc q_4. Ngoài ra, cũng không có một đường đi nào từ q_0 đến đỉnh một đỉnh kết thúc mà đi qua q_3. Như vậy, ta có thể bỏ đi đỉnh q_3 và q_4 mà không ảnh hưởng đến việc đoán nhận các từ của otomat A_3. Do đó otomat A_3 tương đương với otomat A_4 sau:

A_4 = <{q_0, q_1, q_2}, {0, 1}, δ, q_0, {q_1, q_2}>,

trong đó $\delta(q_0,0) = q_0$, $\delta(q_0,1) = q_1$, $\delta(q_1,1) = q_2$, $\delta(q_2,0) = q_2$, $\delta(q_2,1) = q_2$.

Đồ thị chuyển của A_4 được cho trong hình 3.10::



H. 3.10 Đồ thị chuyển của otomat A_4

Các đường đi từ q_0 đến đỉnh kết thúc q_1 ứng với các xâu 0^n1, $n \geq 0$. Các đường đi từ q_0 đến đỉnh kết thúc q_2 ứng với các xâu $0^n11\omega$, $n \geq 0$, $\omega \in \{0, 1\}^*$. Vậy ngôn ngữ được đoán nhận bởi các otomat trên là:

$$T(A_3) = T(A_4) = \{0^n1, 0^n11\omega \,/\, n \geq 0, \omega \in \{0, 1\}^*\}.$$

Bổ đề 1.1 Cho otomat hữu hạn đơn định A = <Q, Σ, δ, q_0, F>. Khi đó $\forall \omega_1, \omega_2 \in \Sigma^*$, $\forall q \in Q$ sao cho $\delta(q, \omega_1\omega_2)$ xác định, ta có:

$$\delta(q, \omega_1\omega_2) = \delta(\delta(q, \omega_1), \omega_2) \qquad (1)$$

Chứng minh: Ta chứng minh đẳng thức trên bằng quy nạp theo độ dài của ω_2.

+ Khi $|\omega_2| = 1$ hay $\omega_2 = a$, $a \in \Sigma$, ta có $\delta(q, \omega_1 a) = \delta(\delta(q, \omega_1),a)$. Đẳng thức (1) đúng.

+ Giả sử đẳng thức (1) đúng với mọi ω_2 có độ dài $|\omega_2| \le n$. Ta cần chứng minh nó cũng đúng với ω_2 có độ dài $|\omega_2| = n + 1$. Đặt $\omega_2 = \omega'_2 a$, với $\omega'_2 \in \Sigma^*$, $|\omega'_2| = n$, $a \in \Sigma$. Ta có $\delta(q, \omega_1\omega_2) = \delta(q, \omega_1\omega'_2 a) = \delta(\delta(q, \omega_1\omega'_2), a) = \delta(\delta(\delta(q, \omega_1), \omega'_2), a) = \delta(\delta(q, \omega_1), \omega'_2 a) = \delta(\delta(q, \omega_1), \omega_2)$.

Do đó đẳng thức (1) đúng với ω_2 có độ dài n + 1.

Bổ đề được chứng minh.

Chú ý: Với otomat hữu hạn đơn định $A = <Q, \Sigma, \delta, q_0, F>$ bất kỳ, ta luôn có thể xây dựng một otomat hữu hạn đơn định đầy đủ A' tương đương với A.

Thật vậy, lấy $S \notin Q$ (do đó $S \notin F$), đặt $Q' = Q \cup \{S\}$ và $\delta': Q' \times \Sigma \to Q'$ xác định bởi: $\forall q \in Q$, $\forall a \in \Sigma$, $\delta'(q, a) = \delta(q, a)$ nếu $\delta(q, a)$ được xác định, $\delta'(q, a) = S$ nếu $\delta(q, a)$ không được xác định và $\delta'(S, a) = S$. Khi đó A' là otomat hữu hạn đơn định đầy đủ mà T(A') = T(A).

Ta thường chọn $S = \varnothing$, và không cần bổ xung S vào Q.

§2. Otomat hữu hạn không đơn định

2.1 Otomat hữu hạn không đơn định

Định nghĩa 2.1 Một otomat hữu hạn không đơn định (Nondeterministic Finite Automata-NFA) là một bộ năm:

$$A = <Q, \Sigma, \delta, q_0, F>$$

trong đó Q, Σ, q_0, F như trong định nghĩa 1.1 và $\delta: Q \times \Sigma \to 2^Q$, ở đây 2^Q (hay P(Q), là ký hiệu tập hợp các tập con của Q) gọi là ánh xạ chuyển.

Rõ ràng ở đây ánh xạ δ là một hàm đa trị (hàm không đơn định), vì vậy otomat A trong định nghĩa trên đây được gọi là không đơn định.

Trong trường hợp $\delta(q, a)$ xác định $\forall q \in Q$, $\forall a \in \Sigma$, ta nói ôtômát A là đầy đủ.

Nếu $\delta(q, a) = \{p_1, p_2, \ldots, p_k\}$ thì ta nói rằng otomat A ở trạng thái q gặp ký hiệu a thì có thể chuyển đến một trong các trạng thái $p_1, p_2, \ldots, p_k$. Nếu $\delta(q, a) = \{p\}$ thì ở trạng thái q gặp ký hiệu a, otomat A chỉ chuyển đến một trạng thái duy nhất p. Nếu $\delta(q, a)$ khô ng xác đ ịnh (ta thường viết $\delta(q, a) = \varnothing$) thì ở trạng thái q gặp ký hiệu a, otomat A không thể chuyển đến trạng thái nào, cũng tương tự như với otomat hữu hạn đơn định.

Như vậy, ta thấy rằng một otomat hữu hạn đơn định là một trường hợp đặc biệt của một otomat hữu hạn không đơn định. Hoạt động của otomat hữu hạn không đơn định $A = <Q, \Sigma, \delta, q_0, F>$ khi cho xâu vào $\omega = a_1a_2\ldots a_n$ có thể được mô tả như sau:

Khi bắt đầu làm việc, otomat ở trạng thái đầu q_0 và đầu đọc đang nhìn vào ô có ký hiệu a_1. Từ trạng thái q_0, dưới tác động của ký hiệu vào a_1, $\delta(q_0, a_1) = \{p_1,\ldots, p_k\}$, otomat xác định các trạng thái có thể tiếp theo là $p_1, \ldots, p_k$ và đầu đọc chuyển sang phải một ô, tức là nhìn vào ô có ký hiệu a_2. Tiếp tục với mỗi p_i ($1\leq i \leq k$) và ký hiệu tiếp theo là a_2, các trạng thái tiếp theo có thể đến được là $\delta(p_1, a_2)\cup\ldots\cup\delta(p_k, a_2)$. Quá trình đó sẽ tiếp tục cho tới khi gặp một trong các tình huống sau:

+ Trong trường hợp tập trạng thái tiếp theo sau khi đọc a_j nào đó là rỗng hoặc sau khi đọc ký hiệu a_n là Q' mà $Q'\cap F = \varnothing$, ta nói rằng A không đoán nhận ω.

+ Trường hợp tập trạng thái tiếp theo sau khi đọc ký hiệu a_n là Q' mà $Q'\cap F \neq \varnothing$, ta nói rằng otomat A đoán nhận ω.

Một otomat hữu hạn không đơn định có thể biểu diễn dưới dạng bảng chuyển hoặc đồ thị chuyển như trong trường hợp otomat hữu hạn đơn định. Nếu $\delta(q, a) = \{p_1, p_2, \ldots, p_k\}$ thì trong đồ thị chuyển có k cung từ q sang $p_1, \ldots, p_k$ được ghi cùng một nhãn a.

Thí dụ 2.1 Cho otomat hữu hạn không đơn định:

$$A = <\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, q_0, \{q_2, q_4\}>,$$

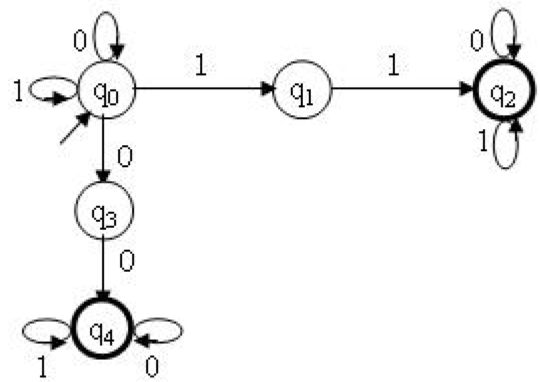
Với $\delta(q_0,0) = \{q_0,q_3\}$, $\delta(q_0, 1) = \{q_0,q_1\}$, $\delta(q_1, 0) = \varnothing$, $\delta(q_1, 1) = \{q_2\}$, $\delta(q_2, 0) = \{q_2\}$,

$\delta(q_2, 1) = \{q_2\}$, $\delta(q_3, 0) = \{q_4\}$, $\delta(q_3,1) = \varnothing$, $\delta(q_4, 0) = \{q_4\}$, $\delta(q_4, 1) = \{q_4\}$.

Bảng chuyển trạng thái và đồ thị chuyển trạng thái của otomat A cho trong hình 3.11 và 3.12:

Trạng thái	Ký hiệu vào 0	1
q_0	$\{q_0, q_3\}$	$\{q_0, q_1\}$
q_1	$\varnothing$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_4\}$	$\varnothing$
q_4	$\{q_4\}$	$\{q_4\}$

H. 3.11 Bảng chuyển của otomat không đơn định A



H. 3.12 Đò thị chuyển của otomat không đơn định A

2.2 Ngôn ngữ được đoán nhận bởi otomat hữu hạn không đơn định

Định nghĩa 2.2 Cho otomat hữu hạn không đơn định A = <Q, Σ, δ, q_0, F>. Mở rộng của δ là ánh xạ δ' từ tập $Q\times\Sigma^*$ vào 2^Q được xác định như sau:

1) δ'(q, ε) = {q}, ∀q ∈ Q,

2) δ'(q, ωa) = $\bigcup_{p\in\delta(q,a)}\delta'(p,\omega)$, ∀q∈Q, ∀a∈Σ, ∀ω∈$\Sigma^*$ sao cho δ'(q, ω) được xác định.

Ta có δ'(q, a) = δ'(q , εa) = $\bigcup_{p\in\delta(q,a)}\delta'(p,\varepsilon) = \bigcup_{p\in\delta(q,a)} p$ = δ(q, a), ∀q ∈ Q, ∀a∈Σ, tức là trên $Q\times\Sigma$ ta có thể đồng nhất δ' với δ. Vì vậy, cũng như trường hợp otomat hữu hạn đơn định, ta có thể sử dụng ký hiệu δ thay cho δ' và được hiểu là ánh xạ δ trên miền $Q\times\Sigma$, là ánh xạ δ' trên $Q\times\Sigma^*$.

Định nghĩa 2.3 Cho otomat hữu hạn không đơn định A = <Q, Σ, δ, q_0, F>, ω ∈ Σ^* và L là một ngôn ngữ trên Σ. Ta nói:

– ω được đoán nhận bởi A nếu $\delta(q_0, \omega) \cap F \neq \emptyset$;

– L được đoán nhận bởi A nếu $L = \{\omega\in\Sigma^* \mid \delta(q_0, \omega) \cap F \neq \emptyset\}$ và ký hiệu L là T(A).

Thí dụ 2.2 Cho otomat hữu hạn không đơn định:

$$A = <\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\}>,$$

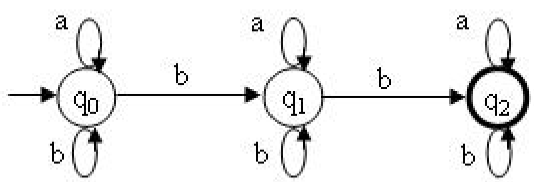
trong đó $\delta(q_0, a) = \{q_0\}$, $\delta(q_0, b) = \{q_0, q_1\}$, $\delta(q_1, a) = \{q_1\}$, $\delta(q_1, b) = \{q_1, q_2\}$,

$\delta(q_2, a) = \{q_2\}$, $\delta(q_2, b) = \{q_2\}$.

Bảng chuyển và đồ thị chuyển của otomat A được cho trong hình 3.13 và 3.14:

δ	a	b
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_1\}$	$\{q_1, q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$

H. 3.13 Bảng chuyển của otomat A trong thí dụ 2.2



H. 3.14 Đồ thị chuyển của otomat A trong thí dụ 2.2

Có thể kiểm tra được rằng từ $\omega = a^nb^n \in T(A)$, tuy nhiên otomat A không đoán nhận ngôn ngữ $L = \{ a^nb^n \mid \forall\ n \geq 1\}$.

Ngôn ngữ được đoán nhận bởi otomat A là:

$$T(A) = \{\omega_1b\omega_2b\omega_3 \mid \omega_1, \omega_2, \omega_3\in\{a, b\}^*\}.$$

2.3 Đơn định hóa các otomat

Trước hết ta cần nhắc lại rằng hai ôtômát hữu hạn A và A'(đơn định hay không đơn định) được gọi là tương đương nếu chúng cùng đoán nhận một ngôn ngữ, tức là T(A) = T(A').

Giả sử A = <Q, Σ, δ, q_0, F> là một otomat không đơn định, khi đó ta có thể xây dựng otomat đơn định và đầy đủ M tương đương với otomat A (theo nghĩa cùng đoán nhận một ngôn ngữ). Việc xây dựng M được thực hiện theo thuật toán sau đây, được gọi là thuật toán đơn định hóa otomat.

Thuật toán đơn định hóa:

Input: Otomat hữu hạn không đơn định A = <Q, Σ, δ, q_0, F>

Output: Otomat hữu hạn đơn định M = <Q', Σ, δ', s_0, F'>

Phương pháp:

Bước 1: Xây dựng hàm hai biến T: $2^Q \times \Sigma \rightarrow 2^Q$ thỏa mãn các điều kiện:

1/. $\forall$ q $\in$ Q, $\forall$ a $\in$ Σ thì T(q, a) = {q' $\in$ Q | q' = δ(q, a) }

2/. $\forall$ B $\subseteq$ Q mà δ(q, a) = B, $\forall$ a $\in$ Σ thì T(B, a) = $\bigcup_{p \in B} T(p, a)$

Bước 2: Xác định tập trạng thái mới Q' = {s_0, s_1, …, s_k | k $\leq 2^{|Q|}$-1}:

1/. Đặt s_0 = {q_0}, s_1 = {q_1}, … s_i = {q_i} $\forall$ {q_0}, {q_1}, …, {q_i} $\in$ Q,

2/. Đặt s_{i+1} = B_1, s_{i+2} = B_2,… $\forall$ B_1, B_2 … $\subseteq$ Q mà $\delta(q_j, a) = B_j$.

3/. Nếu otomat A là không đầy đủ, đặt $s_k = \varnothing$ và thêm vào hàm chuyển δ' các giá trị $\delta'(s_k, a) = s_k$ $\forall$ a $\in$ Σ để otomat M là otomat đầy đủ.

4/. Trạng thái khởi đầu của otomat M là s_0.

5/. Tập trạng thái kết thúc của otomat M là F' = {s $\in$ Q' | s $\cap$ F $\neq \varnothing$ }.

Bước 3: Xác định hàm chuyển δ': Q' $\times$ Σ $\rightarrow$ Q' của otomat M:

$$\forall s \in Q', \forall a \in \Sigma \text{ thì } \delta'(s, a) = T(s, a)$$

Việc chứng minh T(A) = T(M) là khá dễ dàng, dành cho sinh viên như là bài tập.

Thí dụ 2.3

Cho otomat A = <{p_0, p_1, p_2}, {a, b, c}, δ , p_0, {p_1, p_2}> với hàm chuyển δ cho bởi bảng sau:

δ	a	b	c
p_0	$\{p_1\}$	$\{p_1, p_2\}$	$\{p_2\}$
p_1	$\{p_2\}$		$\{p_0, p_2\}$
p_2	$\{p_1\}$	$\{p_1\}$	$\{p_2\}$

H. 3.15 Bảng chuyển của otomat A trong thí dụ 2.3

Hãy xây dựng otomat M = <Q', {a, b, c}, δ', s_0, F'> đơn định và đầy đủ, tương đương với otomat A.

1/. Xây dựng hàm T: $2^Q \times \Sigma \rightarrow 2^Q$

+ $T(p_0, a) = \{p_1\}$, $T(p_0, b) = \{p_1, p_2\}$, $T(p_0, c) = \{p_2\}$,

+ $T(p_1, a) = \{p_2\}$, $T(p_1, b) = \varnothing$, $T(p_1, c) = \{p_0, p_2\}$,

+ $T(p_2, a) = \{p_1\}$, $T(p_2, b) = \{p_1\}$, $T(p_2, c) = \{p_2\}$,

+ $T(\{p_1, p_2\}, a) = T(p_1,a) \cup T(p_2,a) = \{p_2\} \cup \{p_1\} = \{p_1, p_2\}$, $T(\{p_1, p_2\}, b) = \varnothing \cup \{p_1\} = \{p_1\}$, $T(\{p_1, p_2\}, c) = \{p_0, p_2\} \cup \{p_2\} = \{p_0, p_2\}$,

+ $T(\{p_0, p_2\}, a) = \{p_1\}$, $T(\{p_0, p_2\}, b) = \{p_1, p_2\}$, $T(\{p_0, p_2\}, c) = \{p_2\}$

2/. Đặt $s_0 = \{p_0\}$, $s_1 = \{p_1\}$, $s_2 = \{p_2\}$, $s_3 = \{p_1, p_2\}$, $s_4 = \{p_0, p_2\}$, $s_5 = \varnothing$ ta có:

+ Tập trạng thái mới Q' = $\{s_0, s_1, s_2, s_3, s_4, s_5\}$.

+ Trạng thái khởi đầu của M là s_0,

+ Tập trạng thái kết mới: F' = $\{s_1, s_2, s_3, s_4\}$.

3/. Hàm chuyển mới δ': $Q' \times \Sigma \rightarrow Q'$ được xác định như sau:

δ'	a	b	c
s_0	s_1	s_3	s_2
s_1	s_2	s_5	s_4
s_2	s_1	s_1	s_2
s_3	s_3	s_1	s_4
s_4	s_1	s_3	s_2
s_5	s_5	s_5	s_5

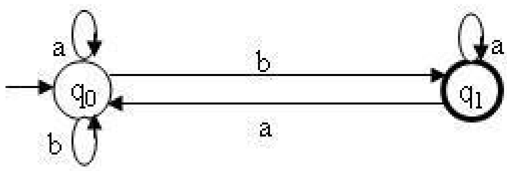
H. 3.16 Bảng chuyển của otomat đơn định M trong thí dụ 2.3

Rõ ràng otomat M = <$\{s_0, s_1, s_2, s_3, s_4, s_5\}$, {a, b, c}, δ', s_0, $\{s_1, s_2, s_3, s_4\}$> với hàm chuyển δ' cho bởi bảng trên là otomat hữu hạn đơn định và đầy đủ. Có thể thây rằng otomat M là tương đương với otomat A.

Thí dụ 2.4 Cho otomat không đơn định: A = <$\{q_0, q_1\}$, $\{a, b\}$, δ, q_0, $\{q_1\}$>,

trong đó $\delta(q_0, a) = \{q_0\}$, $\delta(q_0, b) = \{q_0, q_1\}$, $\delta(q_1, a) = \{q_0, q_1\}$, $\delta(q_1, b) = \varnothing$.

Đồ thị chuyển của A là:



H. 3.17 Đồ thị chuyển của otomat A trong thí dụ 2.4

Ta xây dựng otomat M = <Q', {a, b}, δ', t_0, F'> tương đương với A theo thuật toán đơn định hóa, ta có:

+ Q' = $\{t_0, t_1, t_2, t_3\}$, với $t_0 = \{q_0\}$, $t_1 = \{q_1\}$, $t_2 = \{q_0, q_1\}$, $t_3 = \varnothing$.

+ $\delta'(t_0, a) = t_0$, $\delta'(t_0, b) = t_2$, $\delta'(t_1, a) = t_2$, $\delta'(t_1, b) = t_3$, $\delta'(t_2, a) = \{q_0\}\cup\{q_0, q_1\} = t_2$, $\delta'(t_2, b) = \{q_0, q_1\} \cup \varnothing = t_2$, $\delta'(t_3, a) = t_3$, $\delta'(t_3, b) = t_3$.

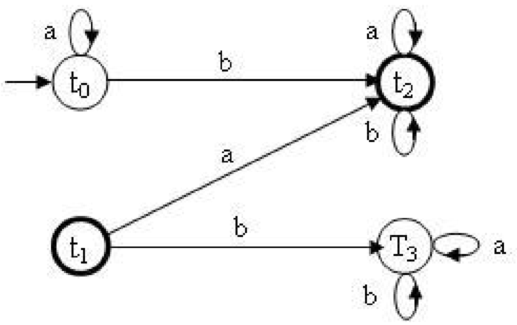
Ta có bảng chuyển của M:

δ'	a	b
t_0	t_0	t_2
t_1	t_2	t_3
t_2	t_2	t_2
t_3	t_3	t_3

H. 3.18 Bảng chuyển của otomat đơn định M trong thí dụ 2.4

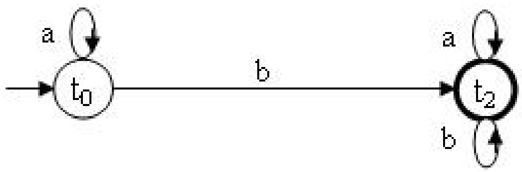
+ Do $t_1 \cap F = \{q_1\} \neq \varnothing$, $t_2 \cap F = \{q_1\} \neq \varnothing$ nên F' = $\{t_1, t_2\}$.

Rõ ràng otomat M là đơn định và có đồ thị chuyển như sau:



H. 3.19 Đồ thị chuyển của otomat M trong thí dụ 2.4

Nhìn vào bảng chuyển và đồ thị chuyển của M, ta thấy ngay rằng không có đường đi nào từ t_0 đến được đỉnh kết thúc t_1, vì vậy otomat M sẽ tương đương với otomat M' có đồ thị chuyển như sau:



H. 3.19 Đồ thị chuyển của otomat M' trong thí dụ 2.4

và ta có $T(A) = T(M) = T(M') = \{a^n b\omega \mid n \geq 0, \omega \in \{a, b\}^*\}$.

2.4 Sự tương đương giữa otomat đơn định và otomat không đơn định

Cá định lý dưới đây sẽ cho ta thấy sự tương đương giữa otomat hữu hạn đơn định và không đơn định.

Định lý 2.1 Nếu ngôn ngữ L được đoán nhận bởi một otomat hữu hạn không đơn định thì tồn tại một otomat hữu hạn đơn định đoán nhận L.

Việc chứng minh định lý này được suy từ thuật toán đơn định hóa các otomat.

Định lý 2.2 Lớp ngôn ngữ được sinh bởi otomat hữu hạn đơn định là trùng với lớp ngôn ngữ được sinh bởi otomat hữu hạn không đơn định.

Chứng minh: Ta gọi L_N là lớp ngôn ngữ sinh bởi các otomat hữu hạn không đơn định, L_D là lớp ngôn ngữ sinh bởi các otomat hữu hạn đơn định, ta cần chứng minh $L_N = L_D$. Ta sẽ chứng minh hai bao hàm thức:

- $L_N \subseteq L_D$. Giả sử L là một ngôn ngữ tùy ý thuộc lớp L_N, tức là tồn tại một otomat không đơn định A đoán nhận L, tức là ta có T(A) = L. Theo định lý 2.1, tồn tại một otomat đơn định M sao cho L = T(M), vậy L thuộc lớp L_D, hay $L_N \subseteq L_D$.

- $L_D \subseteq L_N$. Giả sử L là một ngôn ngữ tùy ý thuộc lớp L_D, tức là tồn tại một otomat đơn định M đoán nhận L, ta có T(M) = L. Tuy nhiên, ta luôn luôn có thể xem hàm chuyển đơn định $\delta(q, a) = p \in Q$ trong otomat đơn định như là một trường hợp đơn giản của hàm chuyển không đơn định $\delta(q, a) = \{p\} \in 2^Q$ trong otomat không đơn định. Như vậy, một otomat đơn định có thể được xem là một trường hợp đặc biệt của otomat không đơn định. Và vì thế, ngôn ngữ L nói trên có thể xem là được đoán nhận bởi otomat không đơn định. Do đó $L_D \subseteq L_N$.

Từ đó ta có $L_D = L_N$.

Định lý được chứng minh.

§3. Ngôn ngữ chính quy và biểu thức chính quy

Trong chương trước, ta đã định nghĩa các ngôn ngữ chính quy thông qua các văn phạm chính quy. Trong phần này, ta sẽ định nghĩa các ngôn ngữ chính quy trực tiếp từ các khái niệm về ngôn ngữ, ta cũng sẽ chỉ ra rằng các định nghĩa này là tương đương. Đồng thời với các ngôn ngữ chính quy, chúng ta đưa ra các khái niệm về biểu thức chính quy, là công cụ để biểu diễn các ngôn ngữ chính quy.

3.1 Ngôn ngữ chính quy và biểu thức chính quy

Định nghĩa 3.1 Cho bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_n\}$, khi đó ngôn ngữ chính quy *(regular languages)* được định nghĩa đệ quy như sau:

1/. Các ngôn ngữ $\varnothing$ và $\{a_i\}$ (i = 1, 2, …, n) được gọi là các ngôn ngữ chính quy trên bảng chữ cái Σ.

2/. Nếu R và S là hai ngôn ngữ chính quy trên bảng chữ cái Σ thì $R \cup S$; R.S; R^+ (hay S^+) là các ngôn ngữ chính quy trên bảng chữ cái Σ.

3/. Không có các ngôn ngữ chính quy nào khác trên bảng chữ cái Σ ngoài các ngôn ngữ chính quy được định nghĩa như trên.

Có thể thấy rằng định ngghĩa 3.1 trên đây là tương đương với định nghĩa ngôn ngữ chính quy thông qua các văn phạm chính quy. Thật vậy, có thể chỉ ra các văn phạm chính quy sinh ra các ngôn ngữ $\varnothing$ và ngôn ngữ {a} (xem thí dụ 4.7, §4, Ch. 1). Ngoài ra, trong chương 1 cũng đã chỉ ra rằng lớp các ngôn ngữ chính quy là đóng đối với các phép toán hợp, nhân ghép và lặp trên các ngôn ngữ. Như vậy lớp ngôn ngữ chính quy được định nghĩa theo định nghĩa trên đây là trùng với lớp ngôn ngữ chính quy đã được định nghĩa theo văn phạm.

Như vậy, từ định nghĩa 3.1, ta có định lý sau:

Định lý 3.1 Mọi ngôn ngữ chính quy trên bảng chữ cái Σ đều nhận được từ các ngôn ngữ hữu hạn bằng cách áp dụng một số hữu hạn lần các phép toán hợp, nhân ghép và phép lặp.

Chú ý:

1/. Các văn phạm chính quy không chứa các quy tắc sinh từ rỗng (còn gọi là các quy tắc rỗng, là các quy tắc có dạng $A \to \varepsilon$, với A là ký hiệu phụ), vì vậy các ngôn ngữ chính quy cũng không chứa từ rỗng ε.

2/. Ngôn ngữ chính quy suy rộng là các ngôn ngữ chính quy có chứa từ rỗng ε, văn phạm chính quy có chứa quy tắc rỗng được gọi là văn phạm chính quy suy rộng

Để diễn đạt các ngôn ngữ chính quy, ta đưa vào khái niệm biểu thức chính quy, được định nghĩa như sau:

Định nghĩa 3.2 Cho bảng chữ cái $\Sigma = \{a_1, a_2, \ldots, a_n\}$, khi đó biểu thức chính quy *(regular expresions)* được định nghĩa đệ quy như sau:

1/. $\varnothing$ và a (với $a \in \Sigma$) là các biểu thức chính quy trên bảng chữ cái Σ biểu diễn ngôn ngữ $\varnothing$ và ngôn ngữ {a}

2/. Nếu r và s là hai biểu thức chính quy biểu diễn các ngôn ngữ chính quy R và S trên bảng chữ cái Σ thì:

- r + s là biểu thức chính quy trên bảng chữ cái Σ biểu diễn ngôn ngữ $R \cup S$
- r.s là biểu thức chính quy trên bảng chữ cái Σ biểu diễn ngôn ngữ R.S
- r^+ (hay s^+) là biểu thức chính quy trên bảng chữ cái Σ biểu diễn ngôn ngữ R^+ (hay S^+)

3/. Không có các biểu thức chính quy nào khác trên bảng chữ cái Σ ngoài các biểu thức chính quy được định nghĩa như trên.

Từ định nghĩa ngôn ngữ chính quy và biểu thức chính quy, ta có các kết quả sau về các ngôn ngữ chính quy:

Định lý 3.2 Một ngôn ngữ trên bảng chữ cái Σ là chính quy khi và chỉ khi nó được biểu diễn được bằng một biểu thức chính quy.

Chú ý:

1/. Biểu thức chính quy suy rộng chấp nhận ε là biểu thức chính quy biểu diễn ngôn ngữ {ε}, và chấp nhận phép toán lặp (*), tức là nếu r là biểu thức chính quy biểu diễn ngôn ngữ chính quy R thì r^* là biểu thức chính quy suy rộng biểu diễn ngôn ngữ chính quy suy rộng R^*. Trong hầu hết các trường hợp, khi không cần phân biệt, ta dùng khái niệm "biểu thức chính quy" chung cho cả "biểu thức chính quy" và "biểu thức chính quy suy rộng"

2/. Trong các biểu thức chính quy ta có thể bỏ qua các dấu ngoặc và quy ước thứ tự các phép toán là phép lặp, phép nhân ghép và cuối cùng là các phép hợp.

Nếu r, s, t là các biểu thức chính quy thì ta có các kết quả sau:

- r+s = s+r,
- (r+s)+t = r+(s+t),
- r+r = r,
- (rs)t = r(st),
- r(s+t) = rs+rt, (s+t)r = sr+tr,
- $\varnothing^* = \varepsilon$,
- $(r^*)^* = r^*$, $(r^+)^+ = r^+$

Có thể chứng minh các kết quả trên bằng cách chỉ ra rằng hai biểu thức chính quy ở hai vế của mỗi đẳng thức đều biểu diễn cùng một ngôn ngữ chính quy. Xin dành việc chứng minh này cho sinh viên như là bài tập.

Thí dụ 3.1 Xác định ngôn ngữ chính quy được biểu diễn bởi biểu thức $r = (01^*+02)1$.

Ta có:

$$r = (01^*+02)1 = 01^*1+021,$$

vậy ngôn ngữ chính quy biểu diễn bởi r là:

$$L(r) = L(01^*1+021) = L(01^*1) \cup L(021) = \{01^n, 021 \mid n \geq 1\}$$

3.2 Sự liên hệ giữa otomat hữu hạn và ngôn ngữ chính quy

Trong chương trước ta đã thấy rằng với mọi ngôn ngữ chính quy đều tồn tại một văn phạm chính quy sinh ra nó, và ngược lại ngôn ngữ sinh bởi văn phạm chính quy là ngôn ngữ chính quy.

Trong phần này, ta sẽ thấy có một sự liên hệ tương tự như vậy giữa otomat hữu hạn và ngôn ngữ chính quy

Định lý 3.3 Nếu L là một ngôn ngữ chính quy thì tồn tại một otomat hữu hạn không đơn định A đoán nhận L, tức là L = T(A).

Chứng minh: Giả sử L là ngôn ngữ bởi văn pham chính quy $G = <\Sigma, \Delta, S, P>$, tức là $L = L(G)$. Xét otomat hữu hạn không đơn định $A = < Q, \Sigma, \delta, q_0, F>$, trong đó:

+ $Q = \Delta \cup \{E\}$, với E là ký hiệu mới và $E \notin \Sigma\cup\Delta$,

+ $q_0 = S$;

+ $F = \{E\}$ nếu quy tắc $S \to \varepsilon \notin P$ và $F = \{E, S\}$ nếu $S \to \varepsilon \in P$,

+ $\forall A\in\Delta, \forall a\in\Sigma$ ta đặt $\delta(A, a) = \{B \in \Delta \mid A\to aB\in P\}\cup\{E \mid A\to a \in P\}$ và $\delta(E, a) = \varnothing$

Ta chứng minh L = T(A).

1/. Lấy $\omega \in L$:

- nếu $\omega = \varepsilon$: trong văn phạm G có quy tắc $S\to\varepsilon \in P$, do đó $S\in F$. Trong trường hợp này $\delta(S, \varepsilon)=\{S\}$ nên $\varepsilon\in T(A)$.
- nếu $\omega = a_1a_2 \ldots a_n \neq \varepsilon$: Ta có suy dẫn $S \vdash a_1A_1 \vdash a_1a_2A_2 \vdash\ldots \vdash a_1a_2\ldots a_{n-1}A_{n-1} \vdash a_1\ldots a_{n-1}a_n$

Do đó tồn tại dãy quy tắc $S\to a_1A_1$, $A_1\to a_2A_2$, …, $A_{n-1}\to a_n$ trong P. Từ định nghĩa của δ, ta có $A_1 \in \delta(S, a_1)$, $A_2 \in \delta(A_1, a_2)$, …, $A_{n-1} \in \delta(A_{n-2}, a_{n-1})$, $E \in \delta(A_{n-1}, a_n)$. Như vậy, $E \in \delta(S, a_1a_2\ldots a_n)$ hay $\omega \in T(A)$. Vậy $L \subseteq T(A)$

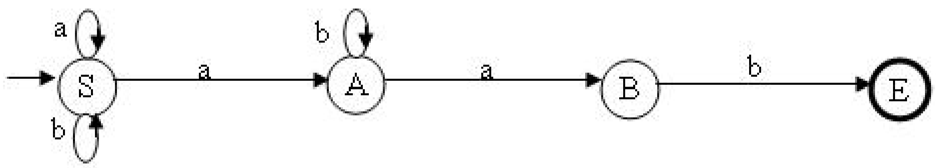
2/. Lấy $\omega\in T(A)$:

- nếu $\omega = \varepsilon$: $\delta(S, \varepsilon)\cap F \neq \varnothing$ hay $S\in F$, vậy có quy tắc $S\to\varepsilon \in P$, do đó $\varepsilon \in L(G)$
- nếu $\omega = a_1a_2 \ldots a_n \neq \varepsilon$: $\delta(S, \omega)\cap F \neq \varnothing$ với $\omega \neq \varepsilon$ hay $E \in \delta(S, \omega)$, do đó tồn tại các trạng thái $A_1, A_2, \ldots, A_{n-1}\in\Delta$ sao cho $A_1\in \delta(S, a_1)$, $A_2\in\delta(A_1, a_2)$,…, $A_{n-1}\in \delta(A_{n-2}, a_{n-1})$, $E\in\delta(A_{n-1}, a_n)$. Từ đó ta có $S\to a_1A_1$, $A_1\to a_2A_2$, …, $A_{n-1}\to a_n\in P$ hay trong G có một suy dẫn là $S \vdash a_1A_1 \vdash a_1a_2A_2 \vdash \ldots \vdash a_1a_2\ldots a_{n-1}A_{n-1} \vdash a_1\ldots a_{n-1}a_n = \omega$. Vì vậy $\omega \in L$. Hay $T(A) \subseteq L$

Vậy ta đã chứng minh L = T(A), tức là tồn tại một otomat hữu hạn không đơn định đoán nhận L.

Thí dụ 3.2 Cho ngôn ngữ $L = \{\omega ab^nab \mid n\geq 0, \omega\in \{a, b\}^*\}$. Ta có L = L(G) trong đó $G = <\{a, b\}, \{S, A, B\}, S, \{S\to aS, S\to bS, S\to aA, A\to bA, A\to aB, B\to b\}>$ là văn phạm chính quy.

Xây dựng otomat hữu hạn không đơn định $A = <\{S, A, B, E\}, \{a, b\}, \delta, S, \{E\}>$, trong đó $\delta(S, a) = \{S, A\}$, $\delta(S, b) = \{S\}$, $\delta(A, a) = \{B\}$, $\delta(A, b) = \{A\}$, $\delta(B, a) = \varnothing$, $\delta(B, b) = \{E\}$, $\delta(E, a) = \varnothing$, $\delta(E, b) = \varnothing$. Đồ thị chuyển của A được cho trong hình 3.20:



H. 3.20 Đồ thị chuyển của otomat A trong thí dụ 3.2

Theo định lý trên, otomat A đoán nhận ngôn ngữ chính quy L, thật vậy ta có:

$$T(A) = \{\omega ab^n ab \mid n \geq 0, \omega \in \{a, b\}^*\} = L$$

Định lý 3.4 Nếu L là ngôn ngữ được đoán nhận bởi một otomat hữu hạn đơn định thì L là một ngôn ngữ chính quy.

Chứng minh: Giả sử L = T(M), với M = $<Q, \Sigma, \delta, q_0, F>$ là một otomat hữu hạn đơn định. Xét văn phạm G = $<\Sigma, Q, q_0, P>$, trong đó P = $\{q \to ap \mid \delta(q, a) = p\} \cup \{q \to a \mid \delta(q, a) = p \in F\}$. Khi đó G là một văn phạm chính quy.

- Ta chứng minh L(G) = L, với giả thiết $\varepsilon \notin L$.

1/. Lấy $\omega = a_1a_2 \ldots a_n \in L(G)$, $\omega \neq \varepsilon$, trong G tồn tại suy dẫn $q_0 \models \omega$ hay $q_0 \vdash a_1q_1 \vdash a_1a_2q_2 \vdash \ldots \vdash a_1a_2 \ldots a_{n-1}q_{n-1} \vdash a_1 \ldots a_{n-1}a_n = \omega$.

Do đó $q_0 \to a_1q_1$, $q_1 \to a_2q_2$, …, $q_{n-1} \to a_{n-1}q_{n-1}$, $q_{n-1} \to a_n \in P$ hay ta có

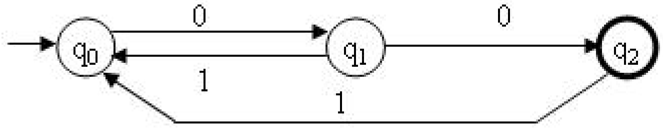
$p_1 = \delta(q_0, a_1)$, $p_2 = \delta(q_1, a_2)$, …, $q_{n-1} = \delta(q_{n-2}, a_{n-1})$, $q_n \in F$

tức là $\delta(q_0, \omega) = q_n \in F$ hay $\omega \in T(A) = L$

2/. Lấy $\omega = a_1a_2 \ldots a_n \in L$, $\omega \neq \varepsilon$, khi đó tồn tại dãy trạng thái $q_1, q_2, \ldots, q_n$ sao cho $\delta(q_0, a_1) = p_1$, $\delta(q_1, a_2) = q_2$, …, $\delta(q_{n-2}, a_{n-1}) = q_{n-1}$, $\delta(q_{n-1}, a_n) = q_n \in F$. Do đó trong G có các quy tắc $q_0 \to a_1q_1$, $q_1 \to a_2q_2$, …, $q_{n-1} \to a_{n-1}q_{n-1}$, $q_{n-1} \to a_n \in P$, ta có suy dẫn trong G: $q_0 \vdash a_1q_1 \vdash a_1a_2q_2 \vdash \ldots \vdash a_1a_2 \ldots a_{n-1}q_{n-1} \vdash a_1 \ldots a_{n-1}a_n = \omega$ hay $\omega \in L(G)$.

- Trong trường hợp $\varepsilon \in L$, ta xây dựng G' tương đương với G trong đó ký hiệu xuất phát không xuất hiện trong bất kỳ vế phải của quy tắc nào, đồng thời thêm vào G' quy tắc $q_0 \to \varepsilon$ để nhận được văn phạm chính quy G' sao cho $L(G') = L(G) \cup \{\varepsilon\}$. Vậy ta luôn có L(G) = L. Vậy định lý được chứng minh.

Thí dụ 3.3 Cho otomat hữu hạn đơn định A = $<\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_2\}>$, trong đó $\delta(q_0, 0) = q_1$, $\delta(q_1, 0) = q_2$, $\delta(q_1, 1) = q_0$, $\delta(q_2, 1) = q_0$. Đồ thị chuyển của A là:



H. 3.21 Đồ thị chuyển của otomat A trong thí dụ 3.3

Dễ thấy rằng $T(A) = \{\omega 00 \mid \omega \in \{01, 001\}^*\}$ là ngôn ngữ chính quy.

Kết luận Từ các định lý trên ta có kết luận về sự liên hệ giữa otomat hữu hạn và ngôn ngữ chính quy như sau:

1/. Gọi $\mathcal{D}$ là lớp các ngôn ngữ được đoán nhận bởi otomat hữu hạn đơn định, $\mathcal{N}$ là lớp các ngôn ngữ được đoán nhận bởi otomat hữu hạn không đơn định và $\mathcal{R}$ là lớp các ngôn ngữ chính quy.

Định lý 2.1 cho biết $\mathcal{D} = \mathcal{N}$.

Định lý 3.3 cho biết $\mathcal{R} \subset \mathcal{N}$.

Định lý 3.4 cho biết $\mathcal{D} \subset \mathcal{R}$

Vậy $\mathcal{D} = \mathcal{N} = \mathcal{R}$.

2/. Ngôn ngữ L là chính quy khi và chỉ khi:

a/. Tồn tại một biểu thúc chính quy biểu diễn L,

b/. Tồn tại một văn phạm chính quy sinh ngôn ngữ L,

c/. Tồn tại một otomat hữu hạn đoán nhận L

Thí dụ 3.4 Với ngôn ngữ chính quy $L = \{01^n, 021 \mid n \geq 1\}$, ta có:

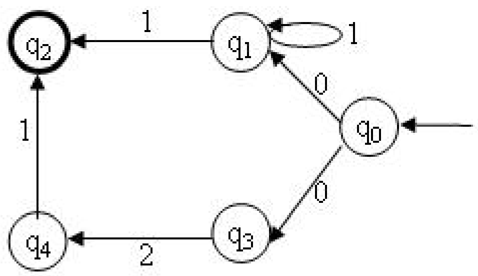
- Biểu thức chính quy biểu diễn L (xem thí dụ 3.2) là:

$$r = 01^*1+021$$

- Văn phạm chính quy sinh ngôn ngữ L:

$$G = <\{0, 1, 2\}, \{S, A, B, C\}, S, \{S\to 0A, A\to 1A, A\to 1, S\to 0B, B\to 2C, C\to 1\}>$$

- Otomat hữu hạn A đoán nhận L có đồ thị chuyển là:



H. 3.22 Đồ thị chuyển của otomat A trong thí dụ 3.4

§4. Điều kiện cần của ngôn ngữ chính quy

Khi một ngôn ngữ được đoán nhận bởi otomat hữu hạn, hoặc được sinh bởi một văn phạm chính quy, hoặc được xác định bởi một biểu thức chính quy thì nó là ngôn ngữ chính quy. Như vậy việc chứng minh một ngôn ngữ là chính quy là khá dễ dàng bằng cách chỉ ra rằng nó được xác định bằng một trong những cách trên. Tuy nhiên, để khẳng định một ngôn ngữ L không phải là ngôn ngữ chính quy thì lại không hề đơn giản. Dù ta không xây dựng được otomat hữu hạn, văn phạm chính quy hay biểu thức chính quy để xác định L, nhưng ta vẫn không thể kết luận được ngôn ngữ này không phải là ngôn ngữ chính quy, bởi vì ta không thể khẳng định được rằng không tồn tại những văn phạm chính quy hay những otomat hữu hạn sinh ra L. Như vậy, cần có một tiêu chuẩn để căn cứ vào đó có thể kết luận một ngôn ngữ không phải là ngôn ngữ chính quy, tiêu chuẩn đó là *điều kiện cần* của ngôn ngữ chính quy.

4.1 Otomat tối tiểu

Cùng một ngôn ngữ chính quy L, có thể có nhiều otomat hữu hạn đoán nhận nó. Tuy nhiên, trong số đó, trước hết chúng ta quan tâm đến các otomat có số trạng thái ít nhất cùng đoán nhận ngôn ngữ L.

Định nghĩa 4.1 Otomat có số trạng thái ít nhất trong các otomat hữu hạn cùng đoán nhận ngôn ngữ L được gọi là otomat tối tiểu của ngôn ngữ L.

Nhận xét: Dễ thấy rằng với mỗi ngôn ngữ L, otomat tối tiểu của nó có thể không duy nhất.

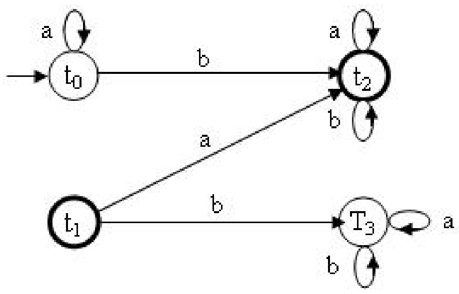
Thí dụ 4.1 Giả sử ta có otomat $M = <Q, \{a, b\}, \delta, t_0, F>$, với

+ $Q = \{t_0, t_1, t_2, t_3\}$, với $t_0 = \{q_0\}$, $t_1 = \{q_1\}$, $t_2 = \{q_0, q_1\}$, $t_3 = \varnothing$.

+ $\delta(t_0, a) = t_0$, $\delta(t_0, b) = t_2$, $\delta(t_1, a) = t_2$, $\delta(t_1, b) = t_3$, $\delta(t_2, a) = t_2$, $\delta(t_2, b) = t_2$, $\delta(t_3, a) = t_3$, $\delta'(t_3, b) = t_3$.

+ $F = \{t_1, t_2\}$.

otomat M là đơn định, có 4 trạng thái và có đồ thị chuyển như sau:

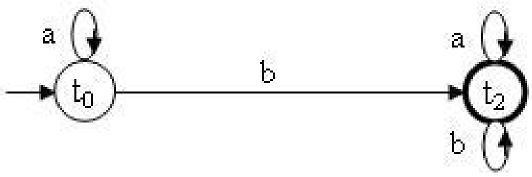


H. 3.23 Đô thị chuyên của otomat M trong thí dụ 4.1

Dễ thấy rằng otomat M đoán nhận ngôn ngữ:

$$L = T(M) = \{a^n b\omega \mid n \geq 0, \omega \in \{a, b\}^*\}.$$

Nhìn vào đồ thị chuyển của M, ta thấy ngay rằng không có đường đi nào từ t_0 đến được đỉnh kết thúc t_1, vì vậy otomat M sẽ tương đương với otomat M' có đồ thị chuyển như sau:



H. 3.24 Đồ thị chuyển của otomat tối tiểu M' trong thí dụ 4.1

Rõ ràng là otomat M' cũng đoán nhận ngôn ngữ $L = T(M') = \{a^n b\omega \mid n\geq 0, \omega\in\{a, b\}^*\}$, M' chỉ có hai trạng thái và là otomat tối tiểu của ngôn ngữ $L = \{a^n b\omega \mid n\geq 0, \omega\in\{a, b\}^*\}$.

4.2 Điều kiện cần của ngôn ngữ chính quy

Định lý 4.1 Nếu L là ngôn ngữ chính quy thì tồn tại số nguyên dương n sao cho với mọi $\omega \in L$ mà $|\omega| \geq n$ đều có thể phân tích được dưới dạng $\omega = uvw$, (với $|v| \geq 1$ hay $v \neq \varepsilon$) mà với mọi chỉ số i = 0, 1. 2,… ta có $uv^iw \in L$

Chứng minh: Vì L là một ngôn ngữ chính quy, khi đó tồn tại một otomat hữu hạn đoán nhận nó. Giả sử L = T(A) , với $A = <Q, \Sigma, \delta, q_0, F>$ là một otomat tối tiểu có n trạng thái, tức là $|Q| = n$. Ta chứng minh n là số tự nhiên cần tìm.

Giả sử $\omega = a_1a_2\ldots a_m \in L$ với $m \geq n$. Khi đó ta có $\delta(q_0, \omega) \in F$, tức là $\exists q_0, q_1\ldots, q_m \in Q$ sao cho $\delta(q_{i-1}, a_i) = q_i$, $1\leq i \leq m$ và $q_m \in F$. Do $m \geq n$ nên trong dãy $q_0, q_1, \ldots, q_m$ có ít nhất hai trạng thái trùng nhau, giả sử đó là $q_i = q_k$, $i < k \leq n$, (với k là số nhỏ nhất mà ta có $q_i = q_k$)

Đặt $u = a_1\ldots a_i$, $v = a_{i+1}\ldots a_k$, $w = a_{k+1}\ldots a_m$. Ta có $\omega = uvw$, $|v| = |a_{i+1}\ldots a_k| \geq 1$ (do i<k).

Ngoài ra ta có: $\delta(q_0, u) = q_i = q_k = \delta(q_0, uv)$,

theo bổ đề 1.1: $\delta(q_0, u) = \delta(q_0, uv) = \delta(\delta(q_0, u), v) = \delta(\delta(q_0, uv), v) = \delta(q_0, uv^2)$,

Tương tự, ta có: $\delta(q_0, u) = \delta(q_0, uv^2) = \delta(\delta(q_0, u), v^2) = \delta(\delta(q_0, uv), v^2) = \delta(q_0, uv^3)$,

tiếp tục ta được: $\delta(q_0, u) = \delta(q_0, uv^i)$, $\forall i\in N$.

Cuối cùng ta có: $\delta(q_0, uv^iw) = \delta(\delta(q_0, uv^i), w) = \delta(\delta(q_0, uv), w) = \delta(q_0, uvw) \in F$.

Vậy $uv^iw \in L$, $\forall i = 0, 1, 2, \ldots$.

Định lý được chứng minh.

Hệ quả 4.1 Cho A là otomat hữu hạn đơn định có n trạng thái và L là ngôn ngữ được đoán nhận bởi A. Khi đó $L \neq \varnothing$ khi và chỉ khi $\exists\ \omega \in L$ sao $|\omega| < n$.

Chứng minh: Điều kiện đủ là hiển nhiên. Bây giờ cho $L \neq \varnothing$. Giả sử mọi từ trong L đều có độ dài $\geq n$. Gọi α là từ có độ dài nhỏ nhất trong L, mà $|\alpha| \geq n$. Theo định lý 4.1, ta có $\alpha = uvw$, trong đó $|v| \geq 1$ và với mọi i = 0, 1, 2,… ta có $uv^iw \in L$. Với i = 0, $uw\in L$ mà $|uw| < |\alpha|$. Điều này mâu thuẫn với tính nhỏ nhất của $|\alpha|$. Vậy tồn tại $\omega\in L$ sao cho $|\omega| < n$.

Hệ quả 4.2 Tồn tại một ngôn ngữ phi ngữ cảnh mà không được đoán nhận bởi bất kỳ một otomat hữu hạn đơn định nào.

Chứng minh: Xét ngôn ngữ $L = \{a^nb^n \mid n \geq 1\}$ trên bảng chữ $\Sigma = \{a, b\}$. Ta có L = L(G), trong đó $G = <\Sigma, \{S\}, S, \{S\rightarrow aSb, S\rightarrow ab\}>$ là văn phạm phi ngữ cảnh. Giả sử L = T(A) với $A = <Q, \Sigma, \delta, q_0, F>$ là một otomat hữu hạn đơn định. Với n đủ lớn, $\alpha = a^nb^n$ có $|\alpha| \geq |Q|$. Theo định lý 1.1, ta có thể biểu diễn $a^nb^n = uvw$, trong đó $|v| \geq 1$, $uv^iw\in L$, $\forall i = 0, 1, 2,\ldots$ Ta hãy tập trung phân tích từ v và v^i:

– Nếu v chỉ chứa a ($|v|_a >0$ và $|v|_b = 0$) thì với i đủ lớn $|uv^iw|_a > |uv^iw|_b$.

– Nếu v chỉ chứa b ($|v|_b >0$ và $|v|_a = 0$) thì với i đủ lớn $|uv^iw|_b > |uv^iw|_a$

– Nếu $|v|_a > 0$ và $|v|_b > 0$ thì với i = 2 ta có $v = (ab)^2 = abab$, tức là a và b xen kẽ nhau trong uv^iw, khi đó uv^iw không thể có dạng a^nb^n.

Cả ba trường hợp đều mâu thuẫn với $uv^iw \in L$. Vậy không tồn tại một otomat hữu hạn đơn định nào đoán nhận L, tức là L không phải là một ngôn ngữ chính quy.

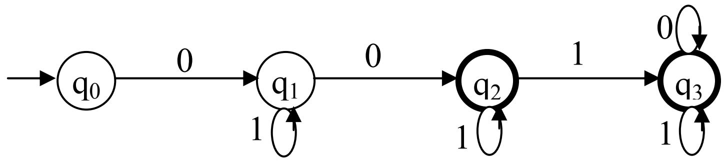
Hệ quả này là một ứng dụng điều kiện cần của ngôn ngữ chính quy, để chứng minh một ngôn ngữ không phải là chính quy nếu nó không thỏa mãn điều kiện cần này.

Hệ quả trên cho thấy rằng tồn tại một ngôn ngữ phi ngữ cảnh mà không phải là ngôn ngữ chính quy, tức là lớp các ngôn ngữ chính quy là tập con thực sự của lớp các ngôn ngữ phi ngữ cảnh.

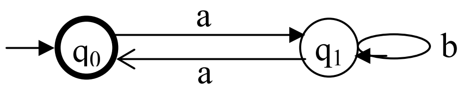
Bài tập chương 2

1. Hãy xây dựng các otomat hữu hạn có đồ thị chuyển sau và xác định các ngôn ngữ được đoán nhận bởi chúng.

a)



b)



2. Hãy xây dựng các otomat hữu hạn đơn định đoán nhận các ngôn ngữ sau:

a) $L = \{a^n b^m \mid n \geq 1, m \geq 1\}$.

b) $L = \{(01)^n, (010)^n \mid n \geq 0\}$.

c) $L = \{(aab)^n (baa)^m \mid n \geq 1, m \geq 1\}$.

3. Hãy xây dựng các otomat hữu hạn đơn định đoán nhận các ngôn ngữ sau:

a) $L_1 = \{\omega \in \{0, 1\}^* \mid \omega$ bắt đầu đúng 3 chữ số 0$\}$.

b) $L_2 = \{\omega \in \{0, 1\}^* \mid \omega$ không bắt đầu bởi hai chữ số 1 liên tiếp$\}$.

4. Hãy xây dựng các otomat hữu hạn đoán nhận các ngôn ngữ phần bù của các ngôn ngữ L_1 và L_2 trong bài tập 3.

Chú ý : Otomat huu han đơn định và đầy đủ thì phần bù của nó la otomat đổi két thành không kết và ngược lại.

5. Hãy xây dựng các otomat hữu hạn không đơn định đoán nhận các ngôn ngữ sau:

a) $L = \{\omega_1 aba \omega_2 \mid \omega_1, \omega_2 \in \{a, b\}^*\}$.

b) $L = \{\omega \in \{0, 1\}^* \mid \omega$ bắt đầu bằng luỹ thừa dương của 101$\}$.

c) $L = \{(1111)^n \omega \mid \omega \in \{0, 1\}^*, n \geq 0\}$.

6. Hãy thành lập các văn phạm chính quy sinh ra các ngôn ngữ mà được đoán nhận bởi các otomat hữu hạn không đơn định sau:

a) $A = <\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \delta, q_0, \{q_2, q_4\}>$, trong đó:

$\delta(q_0, a) = \{q_0, q_1\}$, $\delta(q_0, b) = \{q_0, q_1\}$, $\delta(q_1, a) = \varnothing$, $\delta(q_1, b) = \{q_2\}$, $\delta(q_2, a) = \{q_2\}$, $\delta(q_2, b) = \{q_2\}$, $\delta(q_3, a) = \{q_4\}$, $\delta(q_3, b) = \varnothing$, $\delta(q_4, a) = \{q_4\}$, $\delta(q_4, b) = \{q_4\}$.

b) $A = <\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\}>$, trong đó:

$\delta(q_0, 0) = \{q_0, q_1\}$, $\delta(q_0, 1) = \{q_1\}$, $\delta(q_1, 0) = \varnothing$, $\delta(q_1, 1) = \{q_0, q_1\}$.

7. Hãy xây dựng các otomat hữu hạn đơn định đoán nhận các ngôn ngữ mà được sinh bởi các văn phạm chính quy sau:

a) $G = <\{a, b, c\}, \{S, A, B, C, D\}, S, \{S \to aB, S \to aC, B \to aA, C \to bD,$
$D \to bC, D \to bA, A \to cA, A \to c\}>$.

b) $G = <\{0, 1\}, \{S, A, B, C, D, E\}, S, \{S \to \varepsilon, S \to 1A, B \to 0C, B \to 0D, A \to 1B, B \to 1D, D \to 0E, C \to 1B, C \to 0A, E \to 1A, D \to 1E, E \to 0B, A \to 0D, E \to 1, C \to 1\}>$.

8. Hãy xây dựng các otomat hữu hạn đơn định đoán nhận các ngôn ngữ được biểu diễn bởi các biểu thức chính quy sau:

a) $bba(a+b)^*$.
b) $(a+b)^*bab$.
c) $(bb+a)^*$
d) $(aa+b)^*$.
e) $(bb+a)^*(aa+b)^*$.

9. Cho ôtomat $A = (S, \Sigma , s_1, \delta, F)$, với tập trạng thái $S = \{s_1, s_2, s_3, s_4\}$, bảng chữ cái $\Sigma = \{ a, b\}$, $F = \{s_4\}$, và hàm chuyển trạng thái cho bởi bảng sau:

δ	s_1	s_2	s_3	s_4
a	$\{ s_2, s_4\}$	s_3		s_3
b	$\{ s_3, s_4\}$		s_4	

a. Viết đồ thị chuyển của ôtomat A.
b. Tìm otomat M đơn định và đầy đủ tương đương với otomat A
c. Viết đồ thị chuyển của ôtomat M

10. Cho otomat $A = (S, \Sigma , s_1, \delta, F)$, với tập trạng thái $S = \{s_1, s_2, s_3\}$, $\Sigma = \{ a, b, c\}$, $F = \{s_3\}$, và hàm chuyển trạng thái cho bởi bảng sau:

δ	s_1	s_2	s_3
a	s_2	s_3	$\{s_1, s_2\}$
b	s_3	s_1	
c	s_3	s_1	

a. Viết đồ thị chuyển của otomat A.

b. Tìm otomat $M = (Q, \Sigma , q_1, f, P)$,đơn định và đầu đủ, tương đương với otomat A

c. Viết đồ thị chuyển của otomat M.

Chương 3

OTOMAT ĐẨY XUỐNG VÀ NGÔN NGỮ PHI NGỮ CẢNH

Mở đầu

Đối với các lớp văn phạm được phân loại theo Chomsky, lớp văn phạm phi ngữ cảnh có vai trò quan trọng nhất trong việc ứng dụng để xây dựng các ngôn ngữ lập trình và các chương trình dịch.

Trong quá trình dịch từ chương trình nguồn ra chương trình đích, người ta sử dụng cấu trúc cú pháp của văn phạm phi ngữ cảnh để phân tích các xâu vào. Cấu trúc cú pháp của một xâu vào được xác định từ dãy các quy tắc suy từ xâu đó. Dựa vào dãy các quy tắc đó, bộ phân tích cú pháp của chương trình dịch sẽ cho biết xâu vào đang xét có thuộc vào xâu do văn phạm phi ngữ cảnh sinh ra hay không. Nói cách khác là với xâu vào ω và một văn phạm phi ngữ cảnh G, cần trả lời câu hỏi: $\omega \in L(G)$ hay không? Nếu có thì hãy tìm cách biểu diễn ω bằng văn phạm, tức là tìm các quy tắc sinh của văn phạm G để sinh ra xâu ω.

Trong chương này, chúng ta sẽ nghiên cứu sâu hơn về ngôn ngữ phi ngữ cảnh cùng với những cơ chế để sinh lớp ngôn ngữ này, đõ là các văn phạm phi ngữ cảnh và các otomat có bộ nhớ đẩy xuống (pushdown otomata). Chương này gồm các nội dung chủ yếu sau:

§ 1. Văn phạm phi ngữ cảnh và cây suy dẫn của nó.

1.1 Cây suy dẫn đầy đủ trong văn phạm phi ngữ cảnh

1.2 Rút gọn các văn phạm phi ngữ cảnh

§ 2. Dạng chuẩn Chomsky

2.1 Văn phạm chuẩn Chomsky

2.2 Đưa văn phạm phi ngữ cảnh về dạng chuẩn Chomsky

§ 3. Otomat đẩy xuống

3.1 Mô tả otomat đẩy xuống

3.2 Định nghĩa otomat đẩy xuống

3.3 Ngôn ngữ được đoán nhận bởi otomat đẩy xuống

§1. Văn phạm phi ngữ cảnh và cây suy dẫn của nó

1.1 Cây suy dẫn đầy đủ trong văn phạm phi ngữ cảnh

Định nghĩa 1.1 Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>. Cây suy dẫn đầy đủ trong văn phạm G là một đồ thị hữu hạn có hướng, không có chu trình và thoả mãn bốn điều kiện sau:

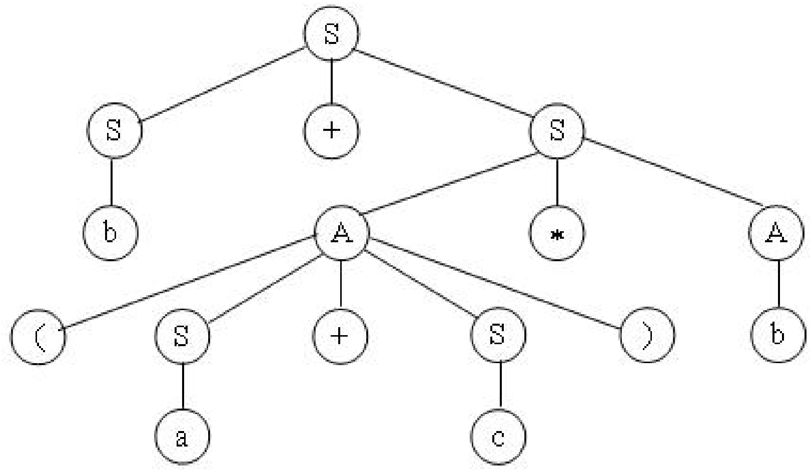
1. Mỗi đỉnh của cây được gán một nhãn là các ký hiệu trong tập $\Sigma \cup \Delta \cup \{\varepsilon\}$. Gốc của cây được gán nhãn là S.
2. Mỗi đỉnh trong được gán nhãn là một ký hiệu nào đó trong Δ.
3. Mỗi đỉnh ngoài (lá của cây) được gán nhãn là một ký hiệu trong tập Σ.
4. Nếu đỉnh m được gán nhãn là $A \in \Delta$, còn các đỉnh $n_1, n_2, \ldots, n_k$ là các con của đỉnh m theo thứ tự từ trái sang phải và được gán nhãn $B_1, B_2, \ldots, B_k$ tương ứng thì $A \rightarrow B_1B_2\ldots B_k$ là một quy tắc trong P của văn phạm G.

Nếu đọc tất cả nhãn ở các lá theo thứ tự từ trái sang phải, ta sẽ nhận được một từ nào đó. Từ đó sẽ là một phần tử trong L(G) và được gọi là kết quả của cây suy dẫn trong G.

Thí dụ 1.1 Cho các văn phạm phi ngữ cảnh:

- $G_1 = <\{a, b, c, +, *, (,)\}, \{S, A\}, S, \{S \rightarrow S+S \mid A*A \mid a \mid b \mid c, A \rightarrow (S+S) \mid a \mid b|c\}>$

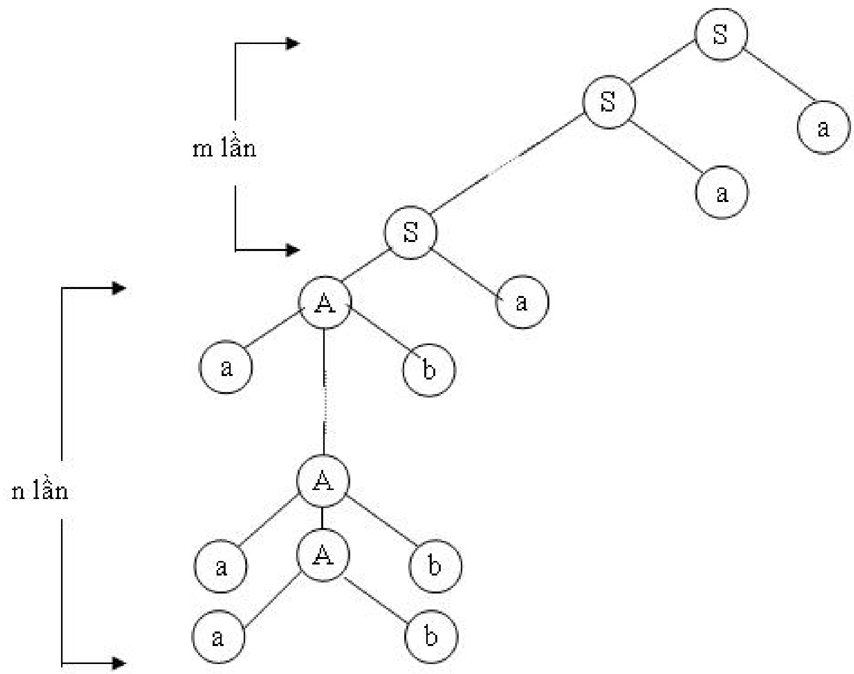
Cây suy dẫn của từ b+(a+c)∗b trong G_1 là:



H. 4.1 Cây suy dẫn của từ ω = b+(a+c)∗b trong G_1

- $G_2 = <\{a, b\}, \{S, A\}, S, \{S \rightarrow Sa \mid Aa, A \rightarrow aAb \mid ab\}>$,

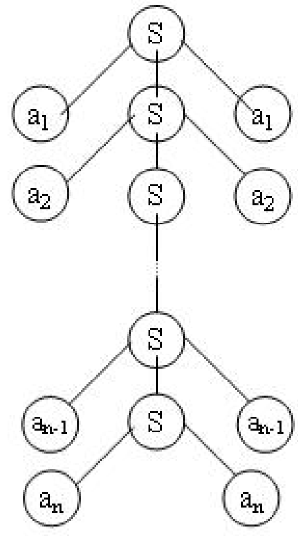
Cây suy dẫn của từ $\omega = a^nb^na^m$ trong G_2 được cho trong hình 4.2 dưới đây:



H. 4.2 Cây suy dẫn của từ $\omega = a^n b^n a^m$ trong G_2

- $G_3 = <\{\Sigma=\{a_1, a_2, \ldots, a_n\}, \{S\}, S, \{S\rightarrow aSa, S\rightarrow aa \mid a\in\Sigma\}>$.

Cây suy dẫn của từ $\omega\omega^R = a_1a_2\ldots a_na_n\ldots a_2a_1$ trong G_3 là:



H. 4.3 Cây suy dẫn của từ $\omega\omega^R = a_1a_2\ldots a_na_n\ldots a_2a_1$ trong G_3

Định lý 1.1 Cho $G = <\Sigma, \Delta, S, P>$ là văn phạm phi ngữ cảnh và $\omega\in\Sigma^* \setminus \{\varepsilon\}$. Khi đó $\omega \in L(G)$ khi và chỉ khi tồn tại một cây suy dẫn đầy đủ trong G có kết quả là ω.

Chứng minh: Do $\omega \neq \varepsilon$ nên ta có thể giả thiết rằng $S\rightarrow\varepsilon \notin P$. Bây giờ với mọi $A\in\Delta$, đặt $G_A = <\Sigma, \Delta, A, P>$, ta có G_A là văn phạm phi ngữ cảnh. Ta sẽ chứng tỏ rằng $\omega\in L(G_A)$ khi và chỉ khi tồn tại một cây suy dẫn trong G_A có kết quả là ω.

Giả sử ω là kết quả của một cây suy dẫn trong G_A và n là số ký hiệu không kết thúc trong cây. Bằng quy nạp theo n, ta sẽ chỉ ra rằng $\omega \in L(G_A)$.

Nếu tổng số ký hiệu không kết thúc trong cây là 1, ký hiệu này phải là A và là gốc của cây, do đó các con của A phải là các đỉnh được gán bởi các ký hiệu kết thúc, chẳng hạn b_1, b_2, …, b_k. Theo định nghĩa của cây suy dẫn, ta có $A \to b_1b_2 \dots b_k$ hay $A \models \omega$.

Giả sử mệnh đề đúng với mọi cây suy dẫn có số ký hiệu không kết thúc là $n-1$. Xét một cây suy dẫn trong G_A có kết quả là ω và trong cây có n ký hiệu không kết thúc. Gọi các con của A theo thứ tự từ trái sang phải là $B_1, B_2, \dots, B_k$. Nếu các đỉnh này đều là lá thì cây gốc A chỉ có một đỉnh có ký hiệu không kết thúc. Giả sử trong các đỉnh này có các đỉnh trong là $C_1, C_2, \dots, C_m$. Xét các cây con mà gốc của nó là $C_1, C_2, \dots, C_m$. Gọi α_i là kết quả của cây suy dẫn gốc C_i. Theo giả thiết quy nạp, $\alpha_i \in L(G_i)$. Vì tập các quy tắc trong G_{Ci} chứa trong tập các quy tắc trong G_A nên ta có các suy dẫn trong G_A là $C_1 \models \alpha_1$, $C_2 \models \alpha_2$, …, $C_m \models \alpha_m$. Sử dụng các suy dẫn này và quy tắc $A \to B_1B_2 \dots B_k$, ta nhận được:

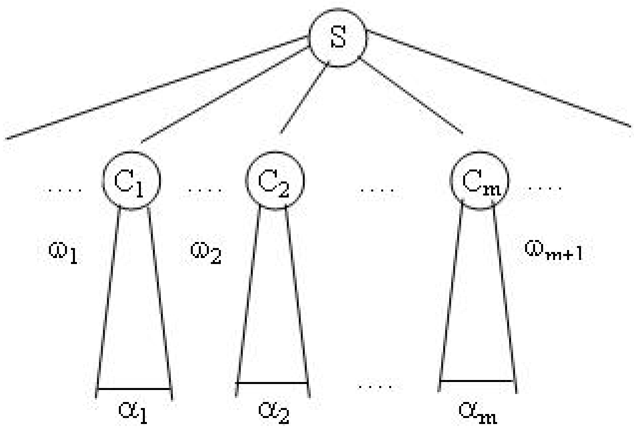
$A \models B_1B_2 \dots B_k \models \omega_1C_1\omega_2C_2 \dots \omega_mC_m\omega_{m+1} \models \dots \models \omega_1\alpha_1\omega_2\alpha_2 \dots \omega_m\alpha_m\omega_{m+1}$.

Do kết quả của cây suy dẫn trong G_A là ω nên $\omega = \omega_1\alpha_1\omega_2\alpha_2 \dots \omega_m\alpha_m\omega_{m+1}$ hay $\omega \in L(G_A)$.

Đảo lại, ta cần chứng minh rằng nếu có suy dẫn $A \models \omega$ ($\omega \neq \varepsilon$) trong G_A thì có thể xây dựng một cây suy dẫn trong G_A có kết quả là ω. Mệnh đề này được chứng minh bằng quy nạp theo độ dài của suy dẫn.

Trước hết, nếu $A \models \omega = b_1b_2 \dots b_k$ (suy dẫn một bước) thì có thể xây dựng một cây có gốc là A và các con từ trái sang phải lần lượt được gán cac nhãn là $b_1, b_2, \dots, b_k$.

Giả sử mệnh đề đúng với mọi suy dẫn có độ dài không lớn hơn n. Cho suy dẫn trong G_A là $A \models \omega$ có độ dài n+1. Giả sử quy tắc đầu tiên trong suy dẫn này là $A \to B_1B_2 \dots B_k$ và $C_1, C_2, \dots, C_m$ là các ký hiệu không kết thúc trong các B_i ($1 \leq i \leq k$), có nghĩa là $B_1B_2 \dots B_k = \omega_1C_1\omega_2C_2 \dots \omega_mC_m\omega_{m+1}$, ở đây $C_i \models \alpha_i$ có độ dài không vượt quá n. Theo giả thiết quy nạp, tồn tại các cây T_i của G_{Ci} mà kết quả của nó là α_i và do đó ta có thể xây dựng trong G_A cây suy dẫn có kết quả là ω như sau:



H. 4.4 Cây suy dẫn có kết quả là ω

Định nghĩa 1.2 Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$. Ta nói văn phạm G là nhập nhằng hay đa nghĩa nếu tồn tại một xâu ω là kết quả của hai cây suy dẫn khác nhau trong G.

Trong trường hợp ngược lại, ta nói G là không nhập nhằng hay đơn nghĩa.

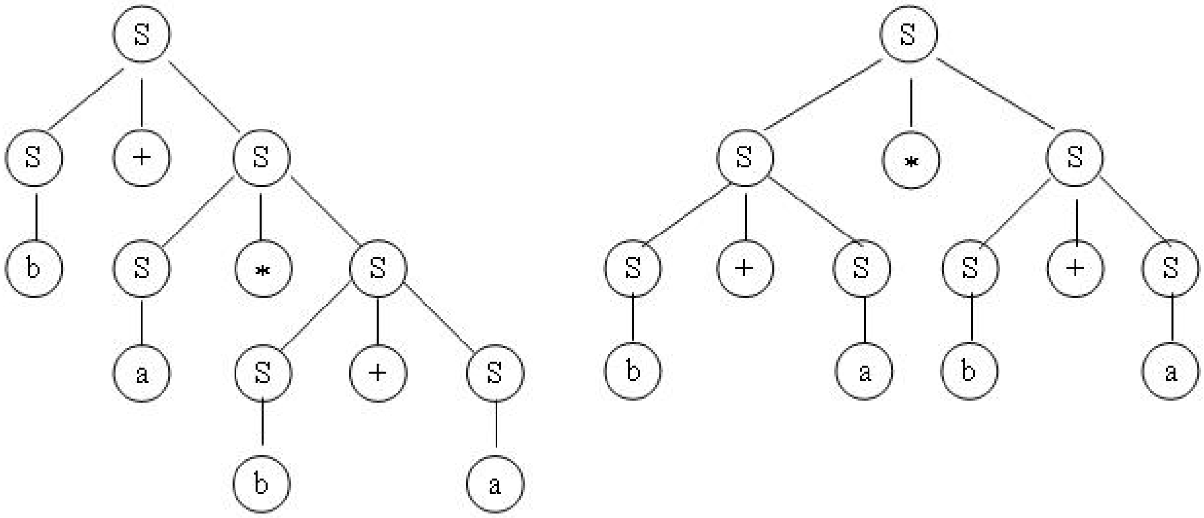
Một văn phạm phi ngữ cảnh được gọi là nhập nhằng vĩnh cửu nếu không tồn tại văn phạm phi ngữ cảnh đơn nghĩa nào tương đương với nó.

Ngôn ngữ do văn phạm G sinh ra gọi là ngôn ngữ nhập nhằng nếu G là văn phạm nhập nhằng.

Thí dụ 1.2 Văn phạm phi ngữ cảnh sau là nhập nhằng:

$G = <\{a, b, +, *\}, \{S\}, S, \{S \to S+S, S \to S*S, S \to a, S \to b\}>$,

vì xâu $\omega = b+a*b+a$ có hai suy dẫn trái khác nhau trong G được cho trong hình 4.5:



H. 4.5 Hai cây suy dẫn khác nhau cho từ $\omega = b+a*b+a$

Cùng với văn phạm G ở trên, văn phạm

$G' = <\{a, b, +, *\}, \{S, A\}, S, \{S \to A, S \to A+S, A \to A*A, A \to a, A \to b\}>$

là đơn nghĩa và $L(G') = L(G)$.

1.2 Rút gọn các văn phạm phi ngữ cảnh

Trong một văn phạm phi ngữ cảnh có thể có nhiều yếu tố thừa, chẳng hạn có những ký hiệu không hề tham gia vào quá trình sinh các ngôn ngữ, hoặc có những quy tắc dạng $A \to B$ chỉ làm mất thời gian trong quá trình hình thành các xâu của ngôn ngữ. Vì lẽ đó cần loại bỏ những yếu tố dư thừa không có ích trong việc sinh ngôn ngữ, sao cho việc loại bỏ đó không làm ảnh hưởng tới quá trình sinh ngôn ngữ. Điều đó có nghĩa là chỉ cần giữ lại các ký hiệu và các quy tắc có ích trong văn phạm G mà chúng thực sự là cần thiết trong quá trình sinh ngôn ngữ mà thôi.

Định nghĩa 1.3 Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$. X được gọi là ký hiệu có ích nếu tồn tại suy dẫn $S \models \alpha X \beta \models \omega$, trong đó $\alpha, \beta \in (\Sigma \cup \Delta)^*$, $X \in \Sigma \cup \Delta$ và $\omega \in \Sigma^*$.

Nếu ký hiệu X không thoả mãn điều kiện trên thì X được gọi là ký hiệu thừa.

Như vậy X là ký hiệu thừa nếu:

1/. Từ X không thể dẫn ra một xâu $\omega \in \Sigma^*$. Ký hiệu X có tính chất như thế được gọi là *ký hiệu vô sinh*.

2/. Từ ký hiệu xuất phát S không thể dẫn được một xâu nào có chứa ký hiệu X. Khi đó ta nói ký hiệu X là ký hiệu *không đến được*.

Như vậy một ký hiệu là thừa nếu nó hoặc là ký hiệu vô sinh hoặc là ký hiệu không đến được.

Bổ đề 1.1 *(loại ký hiệu vô sinh)* Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$ với $L(G) \neq \emptyset$. Khi đó tồn tại văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta', S, P'>$ tương đương với G sao cho $\forall A \in \Delta'$ có một xâu $\omega \in \Sigma^*$ để $A \models \omega$.

Chứng minh: Từ tập quy tắc P của G, ta xây dựng Δ' như sau:

+ Nếu trong P có quy tắc dạng $A \rightarrow \omega$ với $A \in \Delta$, $\omega \in \Sigma^*$ thì kết nạp A vào Δ'.

+ Nếu $A \rightarrow X_1X_2 \ldots X_n$ là quy tắc trong P mà $X_i \in \Sigma$ hoặc X_i là biến đã được kết nạp vào Δ' thì ta kết nạp A vào Δ'.

Cứ tiếp tục xét các quy tắc trong P, ta sẽ xây dựng các ký hiệu cho tập Δ'. Vì P là hữu hạn nên quá trình sẽ được dừng lại sau một số hữu hạn bước. Khi đó ta xây dựng được tập Δ'.

Ta xây dựng tiếp tập quy tắc P' gồm các quy tắc trong P mà các ký hiệu có mặt trong đó đều thuộc tập $\Sigma \cup \Delta'$.

Bổ đề 1.2 *(loại ký hiệu không đến được)* Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$. Khi đó tồn tại văn phạm phi ngữ cảnh $G' = <\Sigma', \Delta', S, P'>$ tương đương với G sao cho $\forall X \in \Sigma' \cup \Delta'$ có $\alpha, \beta \in (\Sigma' \cup \Delta')^*$ để cho $S \models \alpha X \beta$.

Chứng minh: Xây dựng tập Σ' và Δ' như sau:

Đưa ký hiệu S vào Δ'. Nếu một ký hiệu A đã được kết nạp vào Δ' và $A \rightarrow \alpha$, ở đây $\alpha \in (\Sigma' \cup \Delta')^*$ thì ta kết nạp các ký hiệu phụ trong α vào Δ', còn các ký hiệu kết thúc trong α thì kết nạp vào Σ'.

Thủ tục kết nạp trên sẽ ngừng khi không còn bổ sung thêm được bất kỳ ký hiệu nào nữa vào các tập Σ' và Δ'.

Tập quy tắc P' được xây dựng như sau:

P' bao gồm mọi quy tắc trong P mà chứa các ký hiệu thuộc tập $\Sigma' \cup \Delta'$. Với cách xây dựng đó, ta có $L(G) = L(G')$, trong đó G' chỉ gồm các ký hiệu đến được.

Từ hai bổ đề trên ta có:

Định lý 1.2 Mọi ngôn ngữ phi ngữ cảnh khác rỗng đều có thể được sinh ra từ một văn phạm phi ngữ cảnh không có ký hiệu thừa.

Định nghĩa 1.4 Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>. Quy tắc trong P có dạng $A \to B$, ở đây $A, B \in \Delta$, được gọi là quy tắc đơn hay phép đổi tên.

Quy tắc đơn có tác dụng làm kéo dài quá trình sinh ra ngôn ngữ, vì vậy ta sẽ tìm cách loại quy tắc đơn mà không làm ảnh hưởng tới quá trình sinh ra ngôn ngữ của văn phạm đã cho.

Lưu ý rằng quy tắc $A \to a$, với $A \in \Delta$ và $a \in \Sigma$ không phải là quy tắc đơn.

Định lý 1.3 Đối với mọi văn phạm phi ngữ cảnh mà trong tập các quy tắc của nó có quy tắc đơn thì tồn tại một văn phạm phi ngữ cảnh tương đương với nó mà trong tập các quy tắc của nó không chứa quy tắc đơn.

Chứng minh: Giả sử G = <Σ, Δ, S, P> là văn phạm phi ngữ cảnh có chứa quy tắc đơn (và không chứa ký hiệu thừa). Ta xây dựng văn phạm phi ngữ cảnh G' = <Σ, Δ, S, P'> tương đương với G và không chứa quy tắc đơn.

Đưa tất cả các quy tắc không đơn của P vào P'. Nếu trong P có quy tắc $A \to B$, với $A, B \in \Delta$, thì tồn tại suy dẫn $S \models \alpha A\beta \models \alpha B\beta \models \alpha\omega\beta$, ở đây $\alpha, \beta \in (\Sigma \cup \Delta)^*$, $\omega \in \Sigma^*$ do Δ gồm các ký hiệu không thừa.

Vậy thay cho $A \to B$, ta đưa vào P' quy tắc $S \to \alpha A\beta$ và $A \to \omega$ đều là các quy tắc không đơn nhưng chức năng sinh ngôn ngữ tương đương với quy tắc $A \to B$.

Thí dụ 1.3 Văn phạm phi ngữ cảnh

$G = <\{a, +, *\}, \{S, A, B\}, S, \{S \to S+A, S \to A, A \to A*B, A \to B, B \to a\}>$

tương đương với văn phạm phi ngữ cảnh sau không còn các quy tắc đơn:

$G' = <\{a, +, *\}, \{S, A, B\}, S, \{S \to S+A, A \to A*B, B \to a, S \to A*B, A \to a, S \to a\}>$.

Định nghĩa 1.5 Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>, nếu trong P có quy tắc $A \to \varepsilon$, $A \in \Delta$, thì ta nói G có ε-quy tắc.

Chú ý rằng nếu L(G) không chứa từ rỗng ε thì có thể loại hết các ε-quy tắc trong P để được một văn phạm mới tương đương với G; còn nếu trong L(G) có chứa từ rỗng ε, thì không thể loại hết các ε-quy tắc khỏi G (ít nhất trong G phải chứa quy tắc $S \to \varepsilon$).

Các ε-quy tắc cũng làm văn phạm phi ngữ cảnh trở nên cồng kềnh, thiếu chính xác. Định lý dưới đây cho phép loại bỏ các ε-quy tắc trong văn phạm phi ngữ cảnh để được một văn phạm mới tương đương, chỉ sai khác một từ rỗng.

Định lý 1.4 Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>, giả sử L = L(G). Khi đó tồn tại một văn phạm phi ngữ cảnh G = <Σ', Δ', S, P'> không chứa các ε-quy tắc sao cho L(G') = L(G) \ {ε}

Chứng minh: Theo định lý 1.2, ta luôn giả thiết văn phạm G là không chứa các ký hiệu thừa. Ta sẽ xây dựng văn phạm G' không chứa các quy tắc rỗng theo các bước sau:

1/. Tìm tất cả các ký hiệu triệt tiêu (*nullable symbol*) theo thủ tục:

- Nếu $A \to \varepsilon \in P$ thì A là ký hiệu triệt tiêu
- Nếu $B \to \alpha \in P$ mà α là một xâu gồm toàn ký hiệu triệt tiêu thì B là ký hiệu triệt tiêu
- Lặp lại các bước trên cho đến khi không tìm thêm được ký hiệu triệt tiêu nào nữa.

2/. Xây dựng tập quy tắc P'

- Loại tất cả các quy tắc rỗng trong P (có dạng $A \to \varepsilon$),
- Tập quy tắc mới P' được xác định như sau: Nếu $A \to X_1X_2\dots X_n \in P$, $X_i \in (\Sigma \cup \Delta)^*$, thì đưa vào P' tất cả các quy tắc dạng $A \to \alpha_1\ \alpha_2 \dots \alpha_n$ (*) sao cho:

  a/. Nếu X_i không phải ký hiệu triệt tiêu thì $\alpha_i = X_i$, (giữ nguyên X_i)

  b/. Với các X_i là ký hiệu triệt tiêu thì mỗi lần thay một tập con của các ký hiệu triệt tiêu này bởi các ký hiệu rỗng ε để được một quy tắc mới.

  c/. Không thay tất cả các α_i bởi các ký hiệu rỗng, dù mọi X_i đều là ký hiệu triệt tiêu.

Thí dụ 1.4 Cho văn phạm phi ngữ cảnh G = <{a, b}, {I, A, B}, I, P > với tập quy tắc P = $\{I \to AB, A \to aA, A \to \varepsilon, B \to bB, B \to \varepsilon\}$. Hãy xây dựng văn phạm G' không có các ε-quy tắc, không có các ký hiệu thừa, sao cho L(G') = L(G) \ {ε}.

+ Dễ thấy G là không có các ký hiệu thừa

+ Các ký hiệu triệt tiêu là A và B.

+ Tập quy tắc P' = $\{ I \to AB, I \to A, I \to B, A \to aA, A \to a, B \to bB, B \to b \}$.

Vậy ta có G' = <{a, b}, {I, A, B}, I, P' > là văn phạm không chứa các ε-quy tắc. Dễ dàng nhận thấy $L(G') = \{a^n, b^m, a^ib^j \mid m, n, i, j \geq 1\}$, còn $L(G) = \{a^n, b^m, a^ib^j \mid m, n, i, j \geq 0\}$.

§2. Dạng chuẩn Chomsky

2.1 Văn phạm chuẩn của Chomsky

Định nghĩa 2.1 Văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> được gọi là văn phạm ở dạng chuẩn Chomsky, nếu mọi quy tắc đều có dạng $A \to BC$ hoặc $A \to a$, với $A, B, C \in \Delta$, $a \in \Sigma$.

Như vậy, có thể nhận xét rằng các văn phạm ở dạng chuẩn Chomsky sẽ không có các quy tắc thuộc các loại sau:

a/. Các ε-quy tắc (các quy tắc rỗng),

b/. Các quy tắc đơn, dạng $A \to B$, $A, B \in \Delta$,

c/. Các quy tắc mà vế phải có cả ký hiệu chính và ký hiệu phụ.

d/. Các quy tắc có vế phải nhiều hơn hai ký hiệu, Ta sẽ chứng tỏ rằng có thể đưa một văn phạm phi ngữ cảnh bất kỳ về văn phạm ở dạng chuẩn Chomsky.

2.2 Đưa văn phạm phi ngữ cảnh về dạng chuẩn Chomsky

Định lý 2.1 Đối với văn phạm phi ngữ cảnh tùy ý $G = <\Sigma, \Delta, S, P>$, luôn tồn tại một văn phạm phi ngữ cảnh ở dạng chuẩn Chomsky $G' = <\Sigma, \Delta', S, P'>$ tương đương với nó, tức là $L(G) = L(G')$.

Chứng minh: Theo các định lý 1.2, 1.3 và 1.4 ở phần trên, ta có thể giả thiết văn phạm G là không chứa các ký hiệu thừa, không chứa các ε-qui tắc và không chứa các qui tắc đơn. Để xây dựng văn phạm mới G' ở dạng chuẩn Chomsky, ta chỉ cần loại bỏ các quy tắc mà vế phải có chứa cả ký hiệu chính và ký hiệu phụ hoặc các quy tắc mà vế phải nhiều hơn hai ký hiệu. Việc loại bỏ các quy tắc không hợp lệ này tiến hành theo hai bước sau:

Bước 1: Với các quy tắc vế phải có chứa cả ký hiệu chính và ký hiệu phụ, tức là các quy tắc có dạng: $A \rightarrow X_1 X_2 \ldots X_m$, với $X_i \in \Sigma \cup \Delta$, (1)

Xét tất cả các X_i trong quy tắc (1), nếu $X_i \in \Delta$ thì giữ nguyên X_i, nếu $X_i = a \in \Sigma$, ta thêm vào ký hiệu phụ A_a, thay X_i trong quy tắc (1) bởi A_a, và thêm vào quy tắc $A_a \rightarrow a$. Lặp lại quá trình trên với tất cả các X_i trong quy tắc (1), quy tắc (1) trở thành : $A \rightarrow Y_1 Y_2 \ldots Y_m$, với Y_i là các ký hiệu phụ ($Y_i = X_i$ nếu $X_i \in \Delta$, $Y_i = A_i$ nếu $X_i = a \in \Sigma$).

Sau bước 1, ta nhận được văn phạm $G_1 = <\Sigma, \Delta_1, S, P_1>$ với Δ_1 và P_1 nhận được từ Δ và P sau khi thêm vào các ký hiệ ơhụ mới và các quy tắc mới như trên. Rõ ràng $L(G_1) = L(G)$ mà G_1 không chứa các quy tắc mà vế phải có cả ký hiệu chính và ký hiệu phụ.

Bước 2: Bây giờ trong G_1 cần loại bỏ các quy tắc mà vế phải có độ dài lớn hơn 2, gồm toàn ký hiệu phụ, là các quy tắc dạng: $A \rightarrow Y_1Y_2 \ldots Y_m$, với $m>2$, $Y_i \in \Delta_1$ (2)

Ta thêm m-2 ký hiệu phụ $Z_1, Z_2, \ldots Z_{m-2}$ vào tập Δ_1, và thêm vào m-2 quy tắc sau đây:
$A \rightarrow Y_1Z_1$; $Z_1 \rightarrow Y_2Z_2$; $Z_2 \rightarrow Y_3Z_3$; …; $Z_{m-2} \rightarrow Y_{m-1}Y_{m.}$

Ta nhận được văn phạm mới $G_2 < \Sigma, \Delta_2, S, P_2>$ không chứa các quy tắc có vế phải nhiều hơn 2 ký hiệu, không chứa các quy tác vế phải gồm cà ký hiệu chính và ký hiệu phụ. Dễ dàng chỉ ra rằng $L(G_2) = L(G_1) = L(G)$, vậy G_2 chính là văn phạm G' ở dạng chuẩn Chomsky cần tìm.

Thí dụ 2.1 Cho văn phạm phi ngữ cảnh $G = <\{a, b\}, \{S, A, B\}, S, P>$, với tập quy tắc $P = \{S \rightarrow A, S \rightarrow ABA, A \rightarrow aA, A \rightarrow a, A \rightarrow B, B \rightarrow bB, B \rightarrow b\}$. Hãy xây dựng văn phạm ở dạng chuẩn Chomsky tương đương với G.

Áp dụng định lý 1.3, ta có thể loại hết các quy tắc đơn trong G, để được văn phạm tương đương mà không chứa quy tắc đơn $G_1 = <\{a, b\}, \{S, A, B\}, S, P_1>$, với $P_1 = \{S \rightarrow ABA, S \rightarrow aA, S \rightarrow a, S \rightarrow bB, S \rightarrow b, A \rightarrow aA, A \rightarrow a, A \rightarrow bB, A \rightarrow b, B \rightarrow bB, B \rightarrow b\}$.

Bây giờ áp dụng định lý 3.1 cho văn phạm G_1, ta thay tất cả các quy tắc mà vế phải có chứa cả ký hiệu chính và ký hiệu phụ:

+ Với các quy tắc $S \rightarrow aA$, $A \rightarrow aA$ ta thay a bởi A_a, thêm A_a vào tập ký hiệu phụ mới Δ_2, thêm quy tắc $A_a \rightarrow a$ vào tập quy tắc mới P_2.

+ Với các quy tắc $S \rightarrow bB$, $A \rightarrow bB$, $B \rightarrow bB$ ta thay b bởi A_b, thêm A_b vào tập ký hiệu phụ mới Δ_2, thêm quy tắc $A_b \rightarrow b$ vào tập quy tắc mới P_2.

Ta nhận được $G_2 = <\{a, b\}, \{S, A, B, A_a, A_b\}, S, P_2>$, với $P_2 = \{S \rightarrow ABA, S \rightarrow A_aA, S \rightarrow a, S \rightarrow A_bB, S \rightarrow b, A \rightarrow A_aA, A \rightarrow a, A \rightarrow A_bB, A \rightarrow b, B \rightarrow A_bB, B \rightarrow b\}$ không có quy tắc nào mà vế phải có cả ký hiệu chính và ký hiệu phụ, và rõ ràng $G_2 \sim G_1 \sim G$.

+ Với G_2, ta giảm độ dài vế phải các quy tác có hơn hai ký hiệu: thay quy tắc $S \rightarrow ABA$ bởi hai quy tắc $S \rightarrow AC$, $C \rightarrow BA$, và thêm C vào tập ký hiệu phụ. Cuối cùng, ta được văn phạm $G_3 = <\{a, b\}, \{S, A, B, A_a, A_b, C\}, S, P_3>$, với $P_3 = \{S \rightarrow AC, C \rightarrow BA, S \rightarrow A_aA, S \rightarrow a, S \rightarrow A_bB, S \rightarrow b, A \rightarrow A_aA, A \rightarrow a, A \rightarrow A_bB, A \rightarrow b, B \rightarrow A_bB, B \rightarrow b\}$.
Văn phạm $G_3 \sim G_2 \sim G_1 \sim G$, mà G_3 là ở dạng chuẩn Chomsky.

§3. Otomat đẩy xuống

Như ta đã biết, lớp các ngôn ngữ chính quy do văn phạm chính quy sinh ra cũng trùng với lớp các ngôn ngữ được đoán nhận bởi otomat hữu hạn (đơn định hoặc không đơn định).

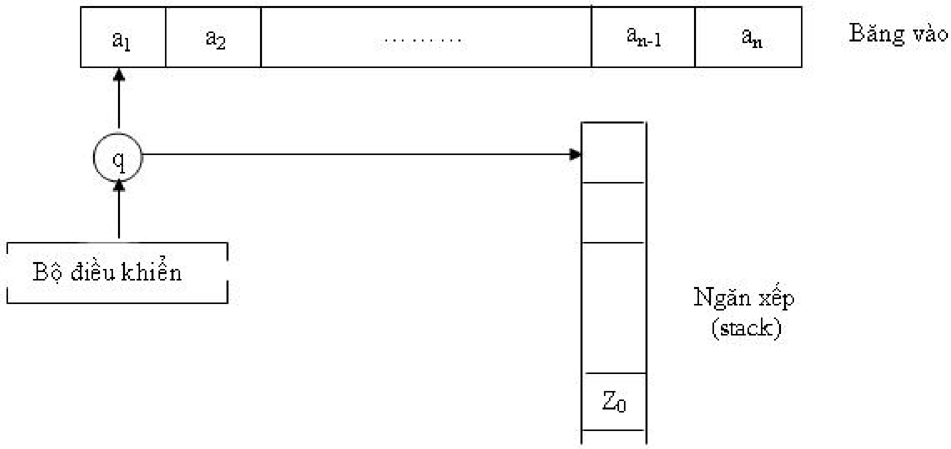
Tương tự, ta sẽ thấy trong phần này là lớp các ngôn ngữ phi ngữ cảnh do các văn phạm phi ngữ cảnh sinh ra sẽ trùng với lớp các ngôn ngữ được đoán nhận bởi otomat đẩy xuống không đơn định (Nondeteministic Pushdown Automata).

Đáng lưu ý, chỉ có lớp otomat đẩy xuống không đơn định mới có thể đoán nhận hết lớp ngôn ngữ phi ngữ cảnh. Còn otomat đẩy xuống đơn định chỉ có khả năng đoán nhận được lớp con thực sự của lớp ngôn ngữ phi ngữ cảnh mà thôi. Tuy vậy, lớp ngôn ngữ được đoán nhận bởi lớp các otomat đẩy xuống đơn định là khá rộng, nó bao gồm phần lớn các ngôn ngữ lập trình hiện nay. Ở đây, ta chỉ đề cập tới các otomat đẩy xuống không đơn định. (thường ký hiệu là NPA, hay gọn hơn là PA và cũng được hiểu là các otomat dẩy xuống không đơn định. Khi cần chỉ rõ các otomat đẩy xuống đơn định ta sẽ ký hiệu là DPA-Deteministic Pushdown Automata).

3.1 Mô tả otomat đẩy xuống

Otomat đẩy xuống cũng như otomat hữu hạn có bộ điều khiển là tập hữu hạn các trạng thái. Đầu đọc của otomat cho phép đọc lần lượt các ký hiệu trên băng vào từ trái sang phải. Ngoài ra, otomat đẩy xuống còn có thêm băng làm việc (ngăn xếp hay stack), nhờ có nó mà bộ nhớ của otomat đẩy xuống được tăng lên so với khả năng nhớ của otomat hữu hạn. Ngăn xếp được tổ chức theo nguyên tắc ký hiệu vào sau thì ra trước (LIFO), giống như ổ nạp đạn. Khi đưa ký hiệu vào ngăn xếp thì ký hiệu đó được trở thành ký hiệu đầu của ngăn xếp. Khi ngăn xếp đọc thì ký hiệu đó là ký hiệu trên cùng và khi ký hiệu đó được xong thì nó sẽ bị loại khỏi ngăn xếp. Một ngăn xếp như vậy còn được gọi là một danh sách đẩy xuống.

Một otomat đẩy xuống bao gồm một băng vào, một bộ nhớ Stack và một bộ điều khiển như hình 4.6 dưới đây:



H. 4.6 Mô hình làm việc của một otomat đẩy xuống.

Căn cứ vào trạng thái hiện tại của bộ điều khiển, ký hiệu vào mà đầu đọc đang quan sát và ký hiệu trên cùng của ngăn xếp, otomat đẩy xuống sẽ chuyển sang một trạng thái mới nào đó và đồng thời đầu đọc có thể được chuyển sang ô bên phải. Nếu đầu đọc chuyển sang ô bên phải thì ta gọi quá trình trên là một bước chuyển. Ngược lại, nếu ký hiệu vào không ảnh hưởng tới bước chuyển thì ta gọi đó là bước chuyển "nhắm mắt" và trong bước chuyển đó đầu đọc vẫn đứng yên tại chỗ. Thực chất của bước chuyển "nhắm mắt" là sự tạm ngừng quan sát băng vào để chấn chỉnh lại ngăn xếp.

Có hai cách đoán nhận xâu vào của otomat đẩy xuống:

Cách 1: Xâu vào được đọc xong và otomat đẩy xuống chuyển được về một trạng thái kết thúc nào đó.

Cách 2: Xâu vào được đọc xong và ngăn xếp trở thành rỗng.

Sau này ta sẽ chỉ ra hai cách đoán nhận trên là tương đương.

Thí dụ 3.1 Cho văn phạm phi ngữ cảnh:

$$G = <\{0, 1, c\}, \{S\}, S, \{S\to 0S0, S\to 1S1, S\to c\}>.$$

Dễ dàng thấy rằng $L(G) = \{\omega c\omega^R \mid \omega\in \{0, 1\}^*\}$ (ω^R là xâu viết các ký hiệu của xâu ω theo một thứ tự ngược lại). Chẳng hạn, đối với xâu $\omega = 010011$ thì xâu $\omega c\omega^R = 010011c110010$ sẽ có dẫn xuất sau đây: (S, 0S0, 01S10, 010S010, 0100S0010, 01001S10010, 010011S110010, 010011c110010).

Mặt khác xâu $\omega c\omega^R$ có thể được đoán nhận bởi otomat đẩy xuống như sau:

Trước hết đặt xâu $x = \omega c\omega^R$ lên băng vào. Đầu đọc dịch chuyển từ trái sang phải. Ban đầu ngăn xếp rỗng. Otomat đẩy xuống có hai trạng thái p, q trong đó p là trạng thái đầu. Khi ở trạng thái đầu p, đầu đọc đọc ký hiệu trên băng vào là 0 hoặc 1 và nó đưa ký hiệu đó vào ngăn xếp. Trạng thái của otomat đẩy xuống lúc đó vẫn là p. Đầu đọc dịch chuyển sang bên phải một ô và đọc ký hiệu trên ô mới này. Nếu ký hiệu này là 0 hoặc 1 thì otomat đẩy xuống đưa ký hiệu đó vào ngăn xếp, trạng thái của otomat vẫn là p và đầu đọc lại được chuyển sang phải một ô.

Quá trình đó vẫn tiếp tục cho tới khi đầu đọc gặp ký hiệu c. Khi đó otomat sẽ chuyển trạng thái p sang trạng thái q và đầu đọc chuyển sang phải một ô. Tại thời điểm này ngăn xếp xuất hiện xâu ω. Ký hiệu bên phải nhất của ω nằm trên cùng của ngăn xếp, còn trạng thái của otomat là q. Đầu đọc đang chỉ ô bên phải ký hiệu c. Nếu ký hiệu của ô này bằng ký hiệu của ô trên cùng của ngăn xếp thì otomat đẩy xuống loại ký hiệu trên cùng ra khỏi ngăn xếp, ký hiệu dưới nó lại lên vị trí trên cùng của ngăn xếp, otomat đẩy xuống chuyển đầu đọc sang phải một ô, còn trạng thái vẫn là q. Quá trình đó cứ tiếp tục và có hai khả năng xảy ra:

1/. Otomat đẩy xuống đọc hết xâu x và ngăn xếp trở thành rỗng thì otomat dừng lại và đoán nhận được xâu $x = \omega c\omega^R$.

2/. Các ký hiệu ở ngăn xếp chưa bị loại hết thì đầu đọc gặp ký hiệu trên xâu vào khác với ký hiệu trên cùng của ngăn xếp. Trong trường hợp này otomat đẩy xuống không đoán nhận xâu x.

Nhờ có ngăn xếp mà otomat đẩy xuống có khả năng nhớ được nửa đầu của xâu $x = \omega c\omega^R$ với ω có độ dài tuỳ ý và sau đó nó so sánh dần với nửa cuối ω^R của x. Otomat hữu hạn không có khả năng này.

Bây giờ ta định nghĩa một cách hình thức otomat đẩy xuống như sau:

3.2 Định nghĩa otomat đẩy xuống

Định nghĩa 3.1 Một otomat đẩy xuống là một bộ bảy:

$$M = <Q, \Sigma, \Delta, \delta, q_0, z_0, F>,$$

trong đó,

+ Σ là tập hữu hạn khác rỗng các ký hiệu, gọi là bảng chữ cái vào, mỗi ký hiệu trong Σ gọi là ký hiệu vào;

+ Q là một tập hữu hạn, khác rỗng các trạng thái sao cho $\Sigma \cap Q = \varnothing$;

+ Δ là tập hữu hạn khác rỗng các ký hiệu, gọi là bảng chữ cái ngăn xếp, mà các ký hiệu của nó gọi là các ký hiệu của ngăn xếp;

+ $z_0 \in \Delta$ là ký hiệu đặc biệt, gọi là ký hiệu đáy của ngăn xếp (còn gọi là ký hiệu đầu tiên của danh sách đẩy xuống);

+ $q_0 \in Q$ gọi là trạng thái khởi đầu;

+ $F \subset Q$ gọi là tập các trạng thái kết thúc;

+ δ: $Q\times(\Sigma\cup\{\varepsilon\})\times\Delta \rightarrow 2^{(Q \times \Delta^*)}$ gọi là hàm chuyển của M, $2^{(Q \times \Delta^*)}$ là tập các tập con của $Q\times\Delta^*$.

Định nghĩa 3.2 Một đẳng thức dạng:

$$\delta(q, a, z) = \{<q_1, \gamma_1>, <q_2, \gamma_2>, \ldots, <q_m, \gamma_m>\},$$

trong đó q, $q_i \in Q$, $a \in \Sigma$, $z \in \Delta$, $\gamma_i \in \Delta^*$, (i = 1, …, m) được gọi là một bước chuyển của otomat đẩy xuống M. Nó đang ở trạng thái q, đọc ký hiệu a ở băng vào và z là ký hiệu ở đỉnh ngăn xếp. Khi đó nó chuyển sang trạng thái q_i, thay ký hiệu z ở đỉnh ngăn xếp bởi xâu γ_i, (i = 1, …, m), đồng thời chuyển đầu đọc sang bên phải một ô. Quy ước rằng khi đưa γ_i vào ngăn xếp, ký hiệu bên trái nhất của γ_i sẽ nằm ở phần dưới, còn ký hiệu bên phải nhất của γ_i sẽ nằm ở ô trên cùng của ngăn xếp.

Một đẳng thức dạng:

$$\delta(q, \varepsilon, z) = \{<q_1, \gamma_1>, <q_2, \gamma_2>, \ldots, <q_m, \gamma_m>\}$$

diễn tả một bước chuyển "nhắm mắt" của otomat đẩy xuống M: Otomat ở trạng thái q, ký hiệu z ở đỉnh ngăn xếp. Khi đó otomat đẩy xuống chuyển trạng thái q về q_i và thay $z\in\Delta$ ở đỉnh ngăn xếp bởi xâu γ_i ($1\le i \le m$), còn đầu đọc thì không dịch chuyển.

Như vậy, ở một thời điểm, tình huống tức thời của otomat đẩy xuống xác định bởi ba yếu tố sau:

– Xâu $\gamma\in\Delta^*$ trong ngăn xếp (với quy ước là ký hiệu bên trái nhất của γ nằm ở đáy ngăn xếp).

– Trạng thái $q\in Q$.

– Phần xâu vào $x\in\Sigma^*$ chưa được đọc đến trên băng vào (với quy ước ký hiệu bên trái nhất của x là ký hiệu sẽ được đọc tiếp).

Ta có định nghĩa cho mỗi tình huống tức thời của PDA như sau:

Định nghĩa 3.3 Cho otomat đẩy xuống $M = <Q, \Sigma, \Delta, \delta, q_0, z_0, F>$. Một hình trạng của otomat M là một bộ ba $K = <q, \alpha, \beta>$, trong đó $q\in Q$, $\alpha\in\Sigma^*$, $\beta\in\Delta^*$.

Giả sử $\alpha = a_1a_2\ldots a_k \in \Sigma^*$, $\beta = x_1x_2\ldots x_m \in \Delta^*$. Dưới tác động của ánh xạ chuyển trạng thái δ, otomat M có thể chuyển từ hình trạng này sang hình trạng khác theo nguyên tắc sau:

1/. Nếu $<p, \gamma>\in\delta(q, a_1, x_m)$ thì hình trạng $<q, a_1a_2\ldots a_k, x_1x_2\ldots x_m>$ có thể chuyển sang hình trạng $<p, a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma>$ và ký hiệu:

$$<q, a_1a_2\ldots a_k, x_1x_2\ldots x_m> \vdash <p, a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma>.$$

2/. Nếu $<p, \gamma>\in \delta(q, \varepsilon, x_m)$ thì hình trạng $<q, a_1a_2\ldots a_k, x_1x_2\ldots x_m>$ có thể chuyển sang hình trạng $<p, a_1a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma>$ và ký hiệu:

$$<q, a_1a_2\ldots a_k, x_1x_2\ldots x_m> \vdash <p, a_1a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma>.$$

3.3 Ngôn ngữ được đoán nhận bởi otomat đẩy xuống

Định nghĩa 3.4 Cho otomat đẩy xuống $M = <Q, \Sigma, \Delta, \delta, q_0, z_0, F>$ và $\omega\in\Sigma^*$. Ta nói rằng otomat M đoán nhận từ ω theo tập trạng thái kết thúc nếu tồn tại một dãy hữu hạn các hình trạng $K_0, K_1, \ldots, K_n$ sao cho:

1/. $K_0 = <q_0, \omega, z_0>$, gọi là hình trạng đầu;

2/. $K_n = <p, \varepsilon, \gamma>$, $p \in F$, gọi là hình trạng kết thúc;

3/. $K_0 \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_n$.

Ký hiệu $T(M) = \{\omega \in \Sigma^* \mid \omega$ được đoán nhận bởi M theo tập trạng thái kết thúc$\}$.

T(M) được gọi là ngôn ngữ được đoán nhận bởi otomat đẩy xuống M theo tập trạng thái kết thúc.

Định nghĩa 3.5 Cho otomat đẩy xuống $M = <Q, \Sigma, \Delta, \delta, q_0, z_0, F>$ và $\omega \in \Sigma^*$. Ta nói rằng otomat M đoán nhận từ ω theo ngăn xếp rỗng nếu tồn tại một dãy hữu hạn các hình trạng K_0, K_1, …, K_n sao cho:

1/. $K_0 = <q_0, \omega, z_0>$, gọi là hình trạng đầu;

2/. $K_n = <p, \varepsilon, \varepsilon>$, p tùy ý thuộc Q, gọi là hình trạng kết thúc;

3/. $K_0 \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_n$.

Ký hiệu $N(M) = \{\omega \in \Sigma^* \mid \omega$ được đoán nhận bởi M theo ngăn xếp rỗng$\}$, được gọi là ngôn ngữ được đoán nhận bởi otomat đẩy xuống M theo ngăn xếp rỗng.

Thí dụ 3.2 Cho otomat đẩy xuống $M = <\{q_0, q_1, q_2\}, \{a, b\}, \{z_0, z_1\}, \delta, q_0, z_0, \{q_2\}>$, trong đó $\delta(q_0, \varepsilon, z_0) = \{<q_0, \varepsilon>\}$, $\delta(q_0, a, z_0) = \{<q_1, z_0z_1>\}$, $\delta(q_1, a, z_1) = \{<q_1, z_1z_1>\}$, $\delta(q_1, b, z_1) = \{<q_2, \varepsilon>\}$, $\delta(q_2, b, z_1) = \{<q_2, \varepsilon>\}$, $\delta(q_2, \varepsilon, z_0) = \{<q_0, \varepsilon>\}$ (ở đây, ảnh của các bộ ba khác qua δ được hiểu là $\emptyset$).

Xét các từ α = aabb và β = abaab. Ta có:

$<q_0, aabb, z_0> \vdash <q_1, abb, z_0z_1> \vdash <q_1, bb, z_0z_1z_1> \vdash <q_2, b, z_0z_1> \vdash <q_2, \varepsilon, z_0> \vdash <q_0, \varepsilon, \varepsilon>$.

$<q_0, abaab, z_0> \vdash <q_1, baab, z_0z_1> \vdash <q_2, aab, z_0>$.

Do đó $\alpha \in N(M)$, $\beta \notin N(M)$, $\beta \notin T(M)$.

Tổng quát, ta có thể chứng minh được rằng $N(M) = T(M) = \{a^nb^n \mid n \geq 0\}$.

Thí dụ 3.3 Cho otomat đẩy xuống $M = <\{q_0, q_1\}, \{0, 1\}, \{z_0, z_1, z_2\}, \delta, q_0, z_0>$, trong đó $\delta(q_0, 0, z_0) = \{<q_0, z_0z_1>\}$, $\delta(q_0, 0, z_1) = \{<q_0, z_1z_1>, <q_1, \varepsilon>\}$,

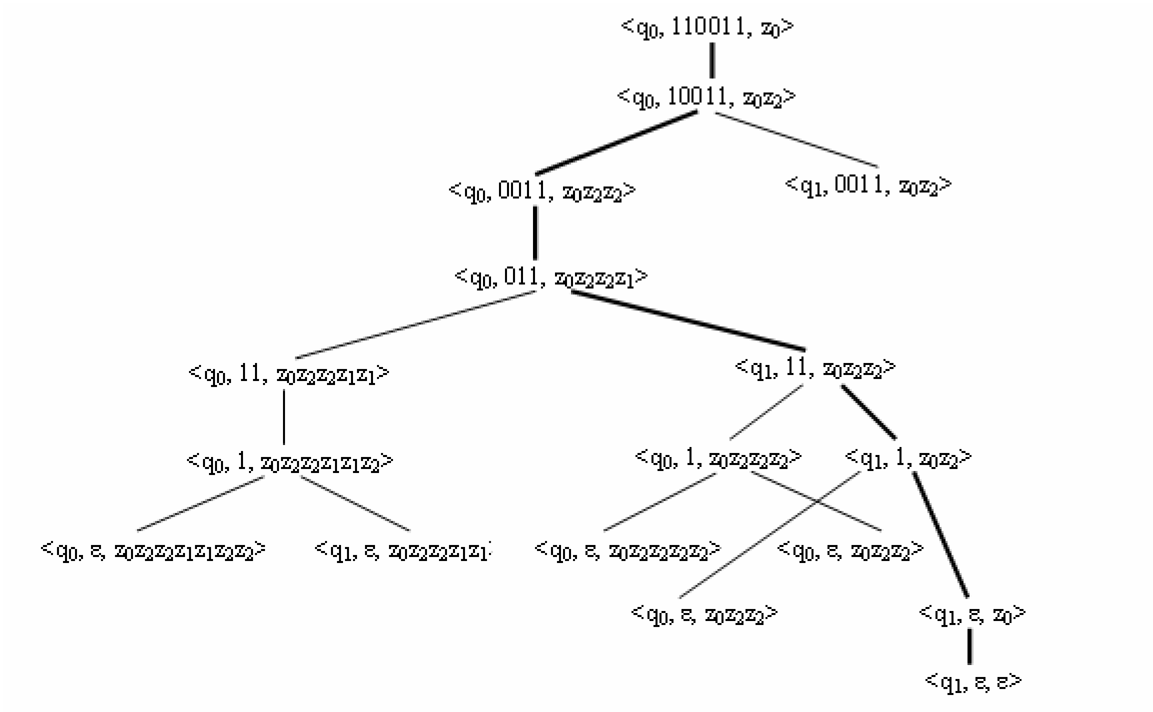
$\delta(q_0, 0, z_2) = \{<q_0, z_2z_1>\}$, $\delta(q_0, 1, z_0) = \{<q_0, z_0z_2>\}$, $\delta(q_0, 1, z_1) = \{<q_0, z_1z_2>\}$,

$\delta(q_0, 1, z_2) = \{<q_0, z_2z_2>, <q_1, \varepsilon>\}$, $\delta(q_1, 0, z_1) = \{<q_1, \varepsilon>\}$,

$\delta(q_1, 1, z_2) = \{<q_0, z_2z_2>, <q_1, \varepsilon>\}$, $\delta(q_0, \varepsilon, z_0) = \{<q_1, \varepsilon>\}$, $\delta(q_1, \varepsilon, z_0) = \{<q_1, \varepsilon>\}$.

Xét ω = 110011, ta có thể vẽ cây biểu diễn các khả năng biến đổi của các hình trạngủtong hình 4.7.

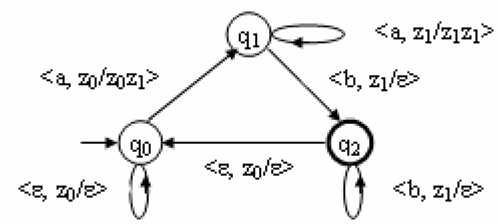
Đường in đậm là dãy dịch chuyển từ hình trạng đầu $<q_0, \omega, z_0>$ đến hình trạng cuối $<q_1, \varepsilon, \varepsilon>$ theo ngăn xếp rỗng.



H. 4.7 Cây biểu diễn các khả năng biến đổi của các hình trạng trong thí dụ 3.3

Chú ý Cũng như đối với otomat hữu hạn, ta có thể mô tả otomat đẩy xuống bằng đồ thị chuyển. Đó là một đa đồ thị có hướng, có khuyên G với tập đỉnh của G là Q. Với $a \in \Sigma \cup \{\varepsilon\}$, $p, q \in Q$, $z \in \Delta$, $\gamma \in \Delta^*$, nếu $<p, \gamma> \in \delta(q, a, z)$ thì sẽ có một cung từ q tới p được gán nhãn là (a, z/γ).

Chẳng hạn, đồ thị chuyển của otomat đẩy xuống M trong Thí dụ 3.2 là:



H. 4.8 Đồ thị chuyển của otomat đẩy xuống trong thí dụ 3.2

Định lý 3.1 Cho L là một ngôn ngữ phi ngữ cảnh. Khi đó tồn tại một otomat đẩy xuống M đoán nhận L theo tập trạng thái kết thúc.

Chứng minh: Giả sử $G = <\Sigma, \Delta, S, P>$ là văn phạm phi ngữ cảnh sinh ra ngôn ngữ L. Ta xây dựng otomat đẩy xuống $M = <Q, \Sigma, \Gamma, \delta, q_0, z_0, F>$ đoán nhận L với:

+ $Q = \{q_0, q_1, q_2\}$ là tập các trạng thái,

+ $\Gamma=\Sigma\cup\Delta\cup\{\%\}$ là tập các ký hiệu ngăn xếp, ký hiệu % là không thuộc $\Sigma\cup\Delta$,

+ $z_0 = S$ là ký hiệu đầu của ngăn xếp,

+ $q_0 \in Q$ là trạng thái đầu,

+ $F = \{q_2\}$ là tập các trạng thái kết thúc,

+ Hàm chuyển δ: $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to P(Q \times \Gamma^*)$ được định nghĩa qua các biểu thức sau:

1/. $\delta(q_1, \varepsilon, z) = \{<q_1, \alpha^R> \mid z \to \alpha \in P, z \in \Delta, \alpha \in \Gamma^*\}$.

2/. $\delta(q_1, a, a) = \{<q_1, \varepsilon>\}$, với mọi $a \in \Sigma$.

3/. $\delta(q_1, \varepsilon, \%) = \{<q_2, \%>\}$.

4/. $\delta(q_0, \varepsilon, S) = \{<q_1, \%S>\}$.

Ta sẽ chứng minh $L(G) = T(M)$.

Giả sử $\omega \in L(G)$. Khi đó tồn tại dãy suy dẫn đầy đủ trong G là:

$D = (S, u_1z_1v_1, u_1u_2z_2v_2, \ldots, u_1\ldots u_{n-1}z_{n-1}v_{n-1}, u_1u_2\ldots u_n = \omega)$, ở đây $z_i \in \Delta$, $u_i \in \Sigma$, $v_i \in \Sigma \cup \Delta$ $(1 \le i \le n-1)$ và $u_n \in \Sigma^*$.

Dựa vào các quy tắc của G sử dụng trong dãy suy dẫn đầy đủ D của xâu ω, otomat đẩy xuống M đoán nhận ω theo nguyên tắc sau:

Áp dụng hàm chuyển xây dựng ở trên, trong các biểu thức 4, 1, 2 ta có:

$<q_0, \omega, z_0> \vdash <q_1, \omega, \%S> \vdash <q_1, u_1u_2\ldots u_n, \%v_1^Rz_1u_1^R> \vdash <q_1, u_2u_3\ldots u_n, \%v_1^Rz_1>$.

Giả sử các quy tắc tiếp theo (sau quy tắc $S \to u_1z_1v_1$) trong D là $z_i \to x_i \in P$ $(i = 1, 2, \ldots, n-2)$ với $z_i \in \Delta$, $x_i \in \Sigma \cup \Delta$ sao cho $x_iv_i = u_{i+1}z_{i+1}v_{i+1}$. Khi đó M sau thời điểm thực hiện quy tắc $S \to u_1z_1v_1$ nó có hình trạng $<q_1, u_2u_3\ldots u_n, \%v_1^Rz_1>$. Hình trạng của M khi áp dụng quy tắc $z_i \to x_i$ biến đổi như sau:

$<q_1, u_2u_3\ldots u_n, \%v_1^Rz_1> \vdash <q_1, u_3\ldots u_n, \%v_1^Rx_1^R> =$

$<q_1, u_3\ldots u_n, \%v_2^Rz_2u_2^R> \vdash <q_1, u_3\ldots u_n, \%v_2^Rz_2> \vdash \ldots \vdash <q_1, u_n, \%v_{n-1}^Rz_{n-1}>$.

Trong D, sử dụng quy tắc $z_{n-1} \to x_{n-1} \in P$ với $x_{n-1}v_{n-1} = u_n$, ta có:

$<q_1, u_n, \%v_{n-1}^Rz_{n-1}> \vdash <q_1, u_n, \%v_{n-1}^Rx_{n-1}^R> \vdash <q_1, \varepsilon, \%> \vdash <q_2, \varepsilon, \%>$.

Từ đó ta có dãy suy dẫn các hình trạng trong M: $<q_0, \omega, z_0> \vdash <q_2, \varepsilon, \%>$ mà $q_2 \in F$ nên M đoán nhận được xâu ω theo tập trạng thái kết thúc.

Thí dụ 3.4 Cho văn phạm phi ngữ cảnh

$G=<\{a, b\}, \{S, A\}, S, \{S \to a, S \to bSA, A \to b, S \to bA, A \to aS\}>$.

Theo chứng minh của Định lý 3.1, ta có thể xây dựng otomat đẩy xuống đoán nhận L(G) như sau:

M = <$\{q_0, q_1, q_2\}$, {a, b}, {a, b, S, A, %}, δ, q_0, S, $\{q_2\}$>, trong đó $\delta(q_1, \varepsilon, S)$ = {<q_1, a>, <q_1, Asb>, <q_1, Ab>}, $\delta(q_1, \varepsilon, A)$ = {<q_1, b>,<q_1, Sa>}, $\delta(q_1, a, a)$ = {<q_1, ε>}, $\delta(q_1, b, b)$ = {<q_1, ε>}, $\delta(q_1, \varepsilon, \%)$ = {<q_2, %>},$\delta(q_0, \varepsilon, S)$ = {<q_1, %S>}.

Định lý 3.2 Cho otomat đẩy xuống M. Khi đó tồn tại otomat đẩy xuống M' sao cho N(M') = T(M).

Chứng minh: Giả sử M = <Q, Σ, Γ, δ, q_0, z_0, F> là otomat đẩy xuống nào đó. Ta xây dựng otomat đẩy xuống M' = <Q', Σ', Γ', δ', q'_0, z'_0, F'> sao cho (M') = T(M).

Muốn vậy ta đưa thêm vào ký hiệu trạng thái mới $q_1, q_2 \notin Q$ và ký hiệu ngăn xếp mới $\% \notin \Gamma$ và đặt:

$\Sigma' = \Sigma$, $Q' = Q \cup \{q_1, q_2\}$, $\Gamma' = \Gamma \cup \{\%\}$, $q'_0 = q_1$, $z'_0 = \%$, $F' = \emptyset$,

δ': Q' x ($\Sigma \cup \{\varepsilon\}$) x $\Gamma' \rightarrow$ P(Q' x Γ'^*) được định nghĩa như sau:

1/. $\delta'(q_1, \varepsilon, \%)$ = {<q_0, %z_0>},

2/. $\delta'(q, x, z) = \delta(q, x, z)$ với $x \in \Sigma$, $q \in Q$, $z \in \Gamma$,

3/. $\delta'(q, \varepsilon, z) = \delta(q, \varepsilon, z)$ nếu $q \in Q \setminus F$, $z \in \Gamma$ và $\delta'(q, \varepsilon, z) = \delta(q, \varepsilon, z) \cup$
$\cup$ {<q_2,ε>} nếu $q \in F$, $z \in \Gamma$,

4/. $\delta'(q, \varepsilon, \%)$ = {<q_2, ε>} với $q \in F$,

5/. $\delta'(q_2, \varepsilon, z)$ = {<q_2, ε>} với $z \in \Gamma \cup \{\%\}$.

Bây giờ ta chỉ ra N(M') = T(M).

Giả sử $w \in N(M')$. Khi đó theo cách làm việc của M' thì ta có một dãy các hình trạng sau:

<q'_0, w, z'_0> = <q_1, w, %> $\vdash K_1 \vdash K_2 \vdash \ldots \vdash K_t$ = <q, ε, ε>,

với $t \geq 2$, K_i = <q_i, w_i, z_i>, $w_i \in \Sigma^*$, $q_i \in Q'$, $z_i \in \Gamma'^*$ (i = 1, 2, …, t).

Theo cách xây dựng của M' thì

K_1 = <q_0, w, %z_0>, K_t = <q_2, ε, ε> và $w \in N(M')$. Từ đó $\exists i < t$ sao cho K_i = <q_i, ε, %z'_i> $\vdash$ <q_2, ε, %z'_{i+1}>, ở đây $q_i \in F$, $z'_i, z'_{i+1} \in \Gamma^*$ và K_j = <q_j, w_j, %z'_j> $\vdash$ <q_{j+1}, w_{j+1}, %z'_{j+1}>, ở đây $w_j \in \Sigma^*$, $q_j \in Q$, $z'_j \in \Gamma^*$, $(1 \leq j \leq i)$. Cũng như K_j = <q_2, ε, %z'_i>, ở đây $z'_j \in \Gamma^*$ $(1<j<t)$. Từ đó <q_0, w, z_0> $\vdash$ <q_i, ε, %z_i> và do $q_i \in F$ suy ra $w \in T(M)$.

Tóm lại ta có bao hàm thức $N(M') \subset T(M)$. Bao hàm thức ngược lại $T(M) \subset N(M')$ được suy trực tiếp từ cách xây dựng M' từ M.

Định lý 3.3 Cho M là một otomat đẩy xuống. Khi đó tồn tại một văn phạm phi ngữ cảnh G sao cho L(G) = N(M).

Chứng minh: Giả sử M = <Q, Σ, Γ, δ, q_0, z_0, F> là một otomat đẩy xuống. Theo định lý 3.2, ta cần chỉ ra có văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> sao cho L(G) = N(M). Không mất tính chất tổng quát, giả sử $\varepsilon \notin N(M)$. Ta xây dựng văn phạm G ở trên như sau:

+ $\Delta = \{S\} \cup \{[q_1, z, q_2] \mid q_1, q_2 \in Q, z \in \Gamma \cup \Sigma\}$,

+ $P = P_1 \cup P_2 \cup P_3$, ở đây $P_1 = \{S \rightarrow [q_0, z, q] \mid q \in Q, z \in \Gamma\}$,

$P_2 = \{[t_1, z, t_2] \rightarrow x[t_3, z_n, q_{n-1}][q_{n-1}, z_{n-1}, q_{n-2}]\ldots[q_2, z_2, q_1][q_1, z_1, t_2] \mid n \geq 1, q_i \in Q, (1 \leq i \leq n), t_1, t_2, t_3 \in Q, x \in \Sigma \cup \{\varepsilon\}, \langle t_3, z_1 z_2 \ldots z_n \rangle \in \delta(t_1, x, z)\}$,

$P_3 = \{[t_1, z, t_2] \rightarrow x \mid \langle t_2, \varepsilon \rangle \in \delta(t_1, x, z)$ với $z \in \Gamma, t_1, t_2 \in Q, x \in \Sigma \cup \{\varepsilon\}\}$.

Người ta chỉ ra được với G định nghĩa như trên là văn phạm phi ngữ cảnh mà L(G) = N(M).

Nếu ta gọi gọi $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$ lần lượt là lớp các ngôn ngữ phi ngữ cảnh, lớp các ngôn ngữ được đoán nhận bởi otomat đẩy xuống theo tập trạng thái kết thúc, lớp các ngôn ngữ được đoán nhận bởi otomat đẩy xuống theo ngăn xếp rỗng, ta có:

- $\mathcal{P}_1 \subset \mathcal{P}_2$ (theo định lý 3.1),
- $\mathcal{P}_2 \subset \mathcal{P}_3$ (theo định lý 3.2),
- $\mathcal{P}_3 \subset \mathcal{P}_1$ (theo định lý 3.3).

Vì vậy, $\mathcal{P}_1 = \mathcal{P}_2 = \mathcal{P}_3$, tức là lớp các ngôn ngữ phi ngữ cảnh là trùng với lớp các ngôn ngữ được đoán nhận bởi các otomat đẩy xuống theo tập trạng thái kết hay theo ngăn xếp rỗng.

Bài tập chương 3

1. Cho văn phạm phi ngữ cảnh: $G = <\{x, +, *, (,)\}, \{S, A, B\}, S, \{S \to A \mid S+A, A \to A*B \mid B, B \to x \mid (S)\}>$, và $\omega = (x+x*x)*(x+x*x*x)$. Hãy tìm một suy dẫn từ S của ω và vẽ cây suy dẫn đầy đủ có kết quả là ω.

2. Chứng tỏ các văn phạm phi ngữ cảnh sau là nhập nhằng:

a/. $G = <\{a, b\}, \{S, A\}, S, \{S \to A, A \to AbA, A \to a\}>$.

b/. $G = <\{a, b\}, \{S, A, B\}, S, \{S \to A, A \to Bb, A \to Ab, B \to Bb, A \to a, B \to b\}>$.

3. Xây dựng các văn phạm phi ngữ cảnh không có ký hiệu thừa, tương đương với các văn phạm sau đây:

a/. $G_1 = <\Sigma_1, \Delta_1, I, P_1>$ với $P_1 = \{I \to ABC, A \to B, B \to a, B \to b, I \to cab\}$

b/. $G_2 = <\Sigma_2, \Delta_2, I, P_2>$ với $P_2 = \{I \to B, B \to C, C \to c, I \to ACI, I \to ab\}$

c/. $G_3 = <\Sigma_3, \Delta_3, I, P_3>$ với $P_3 = \{I \to AB, B \to a, B \to C, I \to b\}$

4. Xây dựng các văn phạm phi ngữ cảnh không có ký hiệu thừa, không có ε-quy tắc, tương đương với các văn phạm sau đây:

a/. $G_1 = <\Sigma_1, \Delta_1, I, P_1>$ với $P_1 = \{I \to aIb, I \to aABb, A \to B, B \to \varepsilon, A \to c\}$

b/. $G_2 = <\Sigma_2, \Delta_2, I, P_2>$ với $P_2 = \{I \to aIbI, I \to bIaI, I \to \varepsilon\}$

5. Xây dựng các văn phạm ở dạng chuẩn Chomsky tương đương với các văn phạm phi ngữ cảnh sau đây:

a/. $G_1 = <\{a, +, *\}, \{I, A, B\}, I, P_1>$ với $P_1 = \{I \to I+A, I \to A, A \to A*B, A \to B, B \to a\}$

b/. $G_2 = <\{a, b, +, *\}, \{I, A, B, C\}, I, P_2>$ với $P_2 = \{I \to A+B, A \to B*C, A \to B, B \to a, B \to C, C \to b\}$

c/. $G_3 = <\{0, 1\}, \{I, B, C, D\}, I, P_3>$ với $P_3 = \{I \to B, I \to C, B \to 0B, B \to 1B, B \to 011, C \to 0D, C \to 1C, C \to \varepsilon, D \to 0C, D \to 1D\}$

6. Hãy xây dựng các otomat đẩy xuống đoán nhận các ngôn ngữ phi ngữ cảnh được sinh bởi các văn phạm sau:

a/. $G_1 = <\{a, b\}, \{S\}, S, \{S \to aSb, S \to ab\}>$.

b/. $G_2 = <\{a, b\}, \{S, A, B\}, S, \{S \to AB, A \to aAa, B \to bBb, A \to a, B \to b\}>$.

7. Hãy xây dựng các otomat đẩy xuống đoán nhận các ngôn ngữ sau:

a) $L = \{a^n b^{2n} \mid n \geq 0\}$.

b) $L = \{\omega c \omega^R \mid \omega \in \{0,1\}^*$, kí hiệu ω^R chỉ xâu ngược của xâu ω }

Chương 4
MÁY TURING

Mở đầu

Khi thiết kế và cài đặt một phần mềm tin học cho một vấn đề nào đó, ta cần phải đưa ra phương pháp giải quyết mà thực chất đó là thuật toán giải quyết vấn đề này. Rõ ràng rằng, nếu không tìm được một phương pháp giải quyết thì không thể lập trình được. Chính vì thế, thuật toán là khái niệm nền tảng của hầu hết các lĩnh vực của tin học. Ta có thể hiểu khái niệm thuật toán như sau. Nếu cho trước một bài toán thì một cách giải bài toán được phân định ra thành một số hữu hạn bước, có kết thúc cuối cùng và đạt được kết quả mong muốn gọi là thuật toán.

Về mặt lịch sử, trong những năm 30 của thế kỷ trước, khi khoa học và công nghệ phát triển, nhân loại đã nêu ra nhiều bài toán mà không tìm thấy lời giải. Có nghĩa là không tìm được thuật toán để giải chúng. Người ta đã phải tìm cách định nghĩa chính xác khái niệm thuật toán. Năm 1936, A. Turing đã đưa ra một công cụ rất tốt để mô tả thuật toán mà ngày nay người ta gọi là máy Turing. Để ghi nhớ công lao này, Hội Tin học Mỹ (ACM) đã đặt ra giải thưởng Turing trong tin học. Cho đến nay, giải thưởng Turing là giải thưởng tin học lớn nhất thế giới. Tiếp theo Turing, một số nhà khoa học khác đã đưa ra các công cụ chính xác hoá khái niệm thuật toán. Đó là các khái niệm hàm đệ quy, thuật toán Marcop, văn phạm sinh của N. Chomsky. Những khái niệm này là cơ sở phát triển của việc nghiên cứu và ứng dụng thuật toán. Mặt khác chính nhờ các khái niệm này, người ta cũng xác định được những bài toán không thể giải được bằng thuật toán.

A. Turing đã đề xuất khái niệm máy Turing nhằm chính xác hoá khái niệm thuật toán. Thực tế đã chứng tỏ rằng máy Turing là một công cụ rất tốt để mô tả thuật toán. Trải qua nhiều thập niên, lý thuyết về máy Turing đã phát triển không ngừng bởi sự đóng góp công sức của nhiều nhà khoa học, trong đó có những công trình nền tảng của Hartmanis, Lewis, Stearns, Minsky, Blum, Hopcroft, Ullman.

Thực chất, máy Turing là một mô hình máy. Nó phân rã toàn bộ quá trình hoạt động ra thành các bước thao tác rất đơn giản. Bản thân máy Turing là một mô hình khái quát và đơn giản có thể mô hình hoá một quá trình tính toán bất kỳ.

Máy Turing có thể xem là một máy với bộ nhớ ngoài có dung lượng được xem như vô hạn. Trong bộ nhớ ngoài, các giá trị được bố trí sao cho có thể truy cập, đọc và sửa đổi được.

Ta có thể xem máy Turing như là một máy đoán nhận một ngôn ngữ gọi là ngôn ngữ đếm được đệ quy. Đồng thời được sử dụng để mô tả một lớp hàm quan trọng, gọi là các hàm có thể tính được. Chương này cũng mô tả một máy Turing phổ dụng mà nó có thể bắt chước hoạt động của tất cả các máy Turing khác. Từ đó ta đi đến khái niệm bài toán không giải được bằng thuật toán.

§1. Máy Turing và lớp các hàm có thể tính được

1.1 Máy Turing

Định nghĩa 1.1 **:** Máy Turing đơn định là một bộ bảy

$$M = \langle Q, \Sigma, \Delta, \delta, s_0, B, F\rangle,$$

trong đó,

+ Q là tập hữu hạn khác rỗng, gọi là tập các trạng thái;

+ Σ là một bảng chữ cái, gọi là bảng chữ vào hay bảng chữ trong;

+ Δ là một bảng chữ cái, $\Delta \supset \Sigma$, gọi là bảng chữ ngoài hay tập các ký hiệu có thể ghi được lên băng;

+ δ: $D \rightarrow Q \times \Delta \times \{R, L\}$, với $D \subset Q \times \Delta$ và $R, L \notin Q \times \Delta$, gọi là ánh xạ chuyển;

+ $s_0 \in Q$, gọi là trạng thái khởi đầu;

+ $B \in \Delta \setminus \Sigma$, gọi là ký hiệu trắng;

+ $F \subset Q$, gọi là tập các trạng thái kết thúc.

Trong trường hợp miền giá trị của δ là $P(Q \times \Delta \times \{R, L\})$ thì máy Turing được gọi là không đơn định và lớp các ngôn ngữ được đoán nhận bởi máy Turing đơn định và không đơn định sẽ trùng nhau. Ngoài ra, có nhiều dạng máy Turing; chẳng hạn, máy Turing với băng vô hạn một đầu hoặc băng vô hạn hai đầu. Ở đây, ta chỉ xét lớp các máy Turing đơn định với băng vô hạn hai đầu.

Định nghĩa 1.2 Cho máy Turing $M = \langle Q, \Sigma, \Delta, \delta, s_0, B, F\rangle$. Bộ ba $\langle\varphi, s, a\psi\rangle$, trong đó $\varphi, \psi \in \Delta^*$, $s \in Q$, $a \in \Delta$, φ không được bắt đầu và ψ không được kết thúc bởi B, được gọi là một hình trạng của M. $\varphi a \psi$ được gọi là từ ứng với hình trạng đã cho.

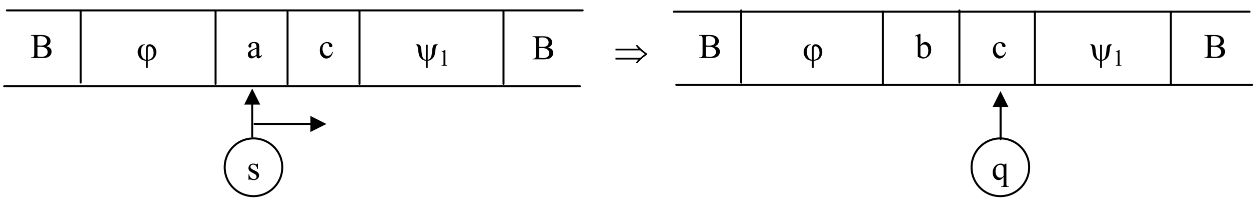
Bộ ba $\langle\varepsilon, s_0, a\psi\rangle$, trong đó $a \in \Delta$, $\psi \in \Delta^*$, được gọi là hình trạng đầu (có từ ứng với nó là $a\psi$).

Định nghĩa 1.3 Cho máy Turing $M = \langle Q, \Sigma, \Delta, \delta, s_0, B, F\rangle$. Ta nói hình trạng $\alpha = \langle\varphi, s, a\psi\rangle$ chuyển đến hình trạng β của M, ký hiệu $\alpha \vdash^M \beta$ hay đơn giản là $\alpha \vdash \beta$, nếu thoả mãn một trong các điều sau:

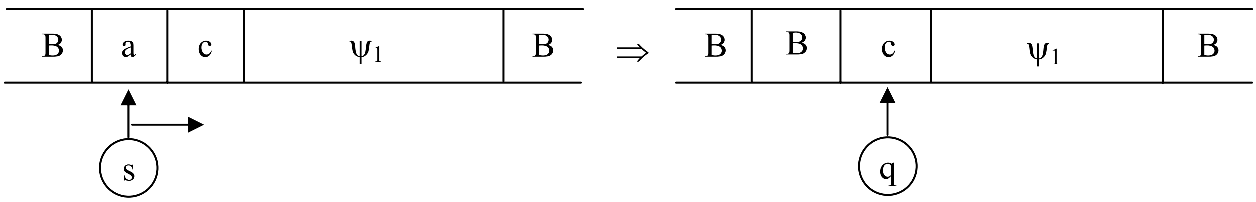
1/. $\delta(s, a) = \langle q, b, R\rangle$:

a/. $\psi = c\psi_1 \neq \varepsilon$, $c \in \Delta$, $\psi_1 \in \Delta^*$:

- $\alpha \vdash \langle\varphi b, q, c\psi_1\rangle$, nếu $\varphi b \notin \{B\}^*$,

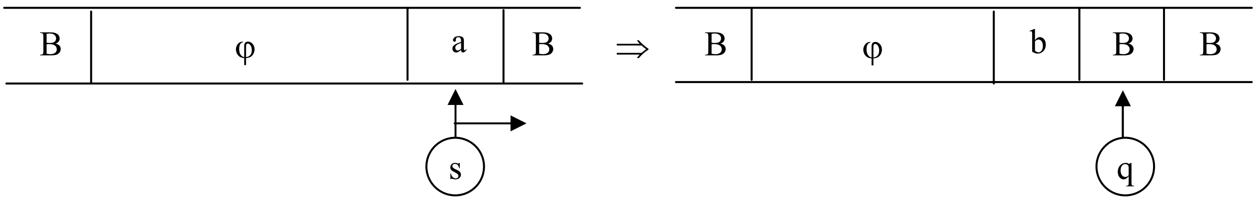


- $\alpha \vdash \langle \varepsilon, q, c\psi_1 \rangle$, nếu $\varphi b \in \{B\}^*$,

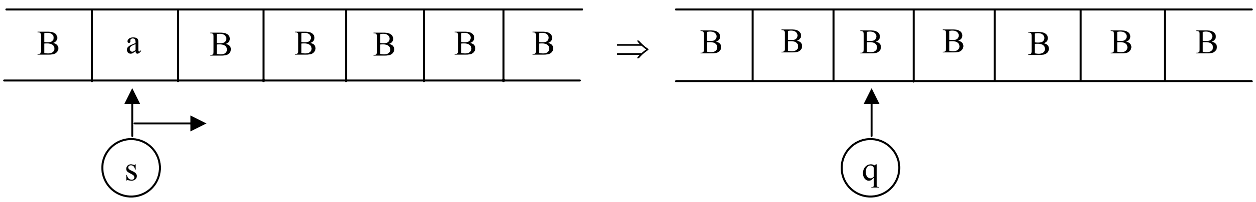


b/. $\psi = \varepsilon$:

- $\alpha = \langle \varphi, s, a \rangle \vdash \langle \varphi b, q, B \rangle$, nếu $\varphi b \notin \{B\}^*$,



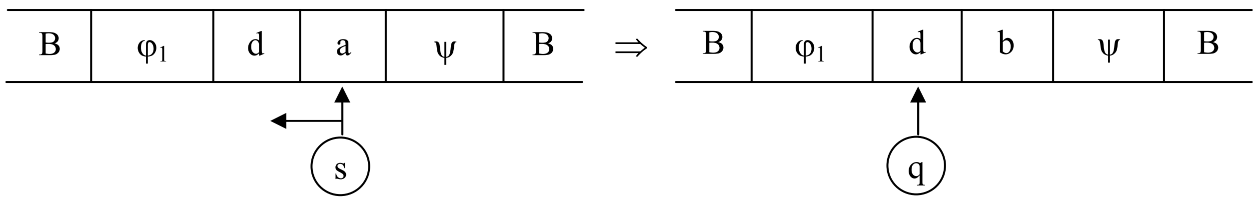
- $\alpha = \langle \varphi, s, a \rangle \vdash \langle \varepsilon, q, B \rangle$, nếu $\varphi b \in \{B\}^*$,



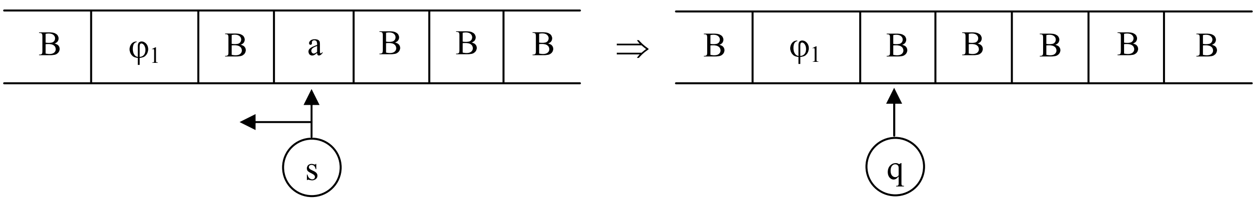
2/. $\delta(s, a) = \langle q, b, L \rangle$:

a/. $\varphi = \varphi_1 d \neq \varepsilon$, $d \in \Delta$, $\varphi_1 \in \Delta^*$:

- $\alpha \vdash \langle \varphi_1, q, db\psi \rangle$, nếu $db\psi \notin \{B\}^*$,

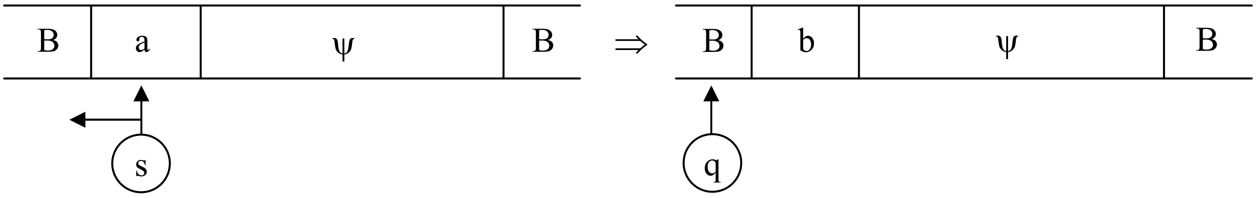


- $\alpha \vdash \langle \varphi_1, q, B \rangle$, nếu $db\psi \in \{B\}^*$,

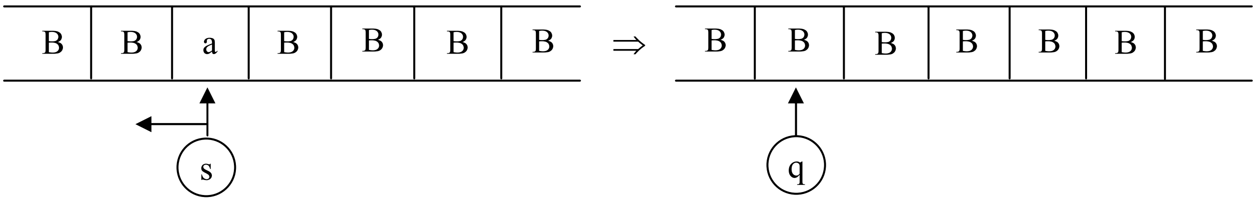


b/. $\varphi = \varepsilon$:

- $\alpha = <\varepsilon, s, a\psi> \vdash <\varepsilon, q, Bb\psi>$, nếu $Bb\psi \notin \{B\}^*$,



- $\alpha = <\varepsilon, s, a\psi> \vdash <\varepsilon, q, B>$, nếu $Bb\psi \in \{B\}^*$,



Dãy hình trạng α_i ($1 \le i \le n$) của máy Turing M sao cho $\alpha_i \vdash \alpha_{i+1}$ ($1 \le i \le n-1$) được gọi là quá trình tính toán trong M, ký hiệu $\alpha_1 \models^M \alpha_n$ hay đơn giản là $\alpha_1 \models \alpha_n$.

Các hình trạng không thể chuyển đến hình trạng mới được gọi là hình trạng cuối. Quá trình tính toán được bắt đầu bởi hình trạng đầu và kết thúc bởi hình trạng cuối được gọi là một quá trình tính toán hoàn chỉnh.

1.2 Ngôn ngữ được đoán nhận bởi máy Turing

Định nghĩa 1.4 Cho máy Turing $M = <Q, \Sigma, \Delta, \delta, s_0, B, F>$ và $\omega \in \Sigma^*$. Ta nói M đoán nhận ω nếu tồn tại quá trình tính toán hoàn chỉnh $<\varepsilon, s_0, \omega> \models <\varphi, q, a\psi>$ với $q \in F$. Tập hợp các từ được đoán nhận bởi máy Turing M được gọi là ngôn ngữ được đoán nhận bởi M, ký hiệu T(M).

Ngôn ngữ được đoán nhận bởi máy Turing còn được gọi là ngôn ngữ đệ quy đếm được (Recursively Enumerable). Ngôn ngữ được đoán nhận bởi máy Turing mà nó sẽ dừng sau một số hữu hạn bước đối với mọi từ vào được gọi là ngôn ngữ đệ quy. Từ định nghĩa suy ra rằng mọi ngôn ngữ đệ quy đều là ngôn ngữ đếm được đệ quy.

Chú ý: Sự hoạt động của máy Turing được thể hiện ở ánh xạ chuyển. Ánh xạ này có thể được mô tả bằng bảng hoặc đồ thị chuyển.

Bảng gồm các cột được đánh dấu bằng các ký hiệu của Δ và các dòng được đánh dấu bằng các trạng thái. Nếu $\delta(s, a) = <q, b, c>$, với $a, b \in \Delta$, $s, q \in Q$, $C \in \{R, L\}$ thì bộ ba $<b, C, q>$ được ghi vào ô ứng với dòng s cột a.

Đồ thị chuyển là một đa đồ thị có hướng, có khuyên G với tập đỉnh của G là Q. Với a, $b \in \Delta$, s, $q \in Q$, $C \in \{R, L\}$, nếu $\delta(s, a) = <q, b, c>$ thì có một cung từ s đến q với nhãn là $<a/b, C>$.

Thí dụ 1.1 Cho máy Turing:

$$M = <\{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}, \{0, 1\}, \{B, 0, 1, X\}, \delta, s_0, B, \{s_0\}>,$$

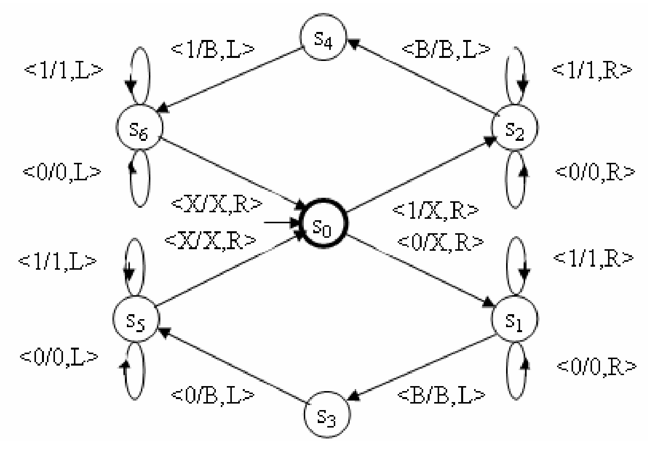
trong đó

$\delta(s_0, 0) = <s_1, X, R>$, $\delta(s_0, 1) = <s_2, X, R>$, $\delta(s_1, 0) = <s_1, 0, R>$,
$\delta(s_1, 1) = <s_1, 1, R>$, $\delta(s_1, B) = <s_3, B, L>$, $\delta(s_2, 0) = <s_2, 0, R>$,
$\delta(s_2, 1) = <s_2, 1, R>$, $\delta(s_2, B) = <s_4, B, L>$, $\delta(s_3, 0) = <s_5, B, L>$,
$\delta(s_4, 1) = <s_6, B, L>$, $\delta(s_5, 0) = <s_5, 0, L>$, $\delta(s_5, 1) = <s_5, 1, L>$,
$\delta(s_5, X) = <s_0, X, R>$, $\delta(s_6, 0) = <s_6, 0, L>$, $\delta(s_6, 1) = <s_6, 1, L>$,
$\delta(s_6, X) = <s_0, X, R>$.

Ánh xạ chuyển có thể cho bằng bảng sau:

	B	0	1	X
S_0		$<X, R, s_1>$	$<X, R, s_2>$	
S_1	$<B, L, s_3>$	$<0, R, s_1>$	$<1, R, s_1>$	
S_2	$<B, L, s_4>$	$<0, R, s_2>$	$<1, R, s_2>$	
S_3		$<B, L, s_5>$		
S_4			$<B, L, s_6>$	
S_5		$<0, L, s_5>$	$<1, L, s_5>$	$<X, R, s_0>$
S_6		$<0, L, s_6>$	$<1, L, s_6>$	$<X, R, s_0>$

Đồ thị chuyển của M là:



Ta hãy xem máy Turing M hoạt động như thế nào đối với các từ 001 và 1001.

Đối với từ 001, ta có dãy hình trạng:

$<\varepsilon, s_0, 001> \vdash <X, s_1, 01> \vdash <X0, s_1, 1> \vdash <X01, s_1, B> \vdash <X0, s_3, 1>$.

Rõ ràng $<X0, s_3, 1>$ hình trạng cuối, nhưng s_3 không phải là trạng thái kết thúc, do đó M không đoán nhận từ 001.

Đối với từ 1001, ta có dãy hình trạng:

$<\varepsilon, s_0, 1001> \vdash <X, s_2, 001> \vdash <X0, s_2, 01> \vdash <X00, s_2, 1> \vdash <X001, s_2, B>$
$\vdash <X00, s_4, 1> \vdash <X0, s_6, 0> \vdash <X, s_6, 00> \vdash <B, s_6, X00> \vdash <X, s_0, 00>$
$\vdash <XX, s_1, 0> \vdash <XX0, s_1, B> \vdash <XX, s_3, 0> \vdash <X, s_5, X> \vdash <XX, s_0, B>$.

Do $<XX, s_0, B>$ là hình trạng cuối và s_0 là trạng thái kết thúc nên từ 1001 được đoán nhận bởi máy Turing M.

Từ đồ thị chuyển dễ dàng thấy rằng M hoạt động với xâu vào ω như sau: M đọc xâu ω từ trái sang phải. Bắt đầu từ trạng thái s_0, thay ký hiệu đã đọc bởi ký hiệu X, đồng thời nếu ký hiệu vừa đọc là 0 thì chuyển sang trạng thái s_1 và nếu ngược lại thì chuyển sang trạng thái s_2. Tại các trạng thái s_1 hoặc s_2, máy M chuyển đầu đọc qua phải mà không thay đổi ký hiệu được đọc cho đến khi gặp ký hiệu B. Từ s_1 máy chuyển sang s_3 và từ s_2 máy chuyển sang s_4. Từ s_3 nếu gặp 0 thì xoá 0 và sang s_5, từ s_4 nếu gặp 1 thì xoá 1 và sang s_6. Ở đây, ta cần lưu ý rằng xoá 0 trong trường hợp xuất phát từ s_0, máy thay 0 bởi X và xoá 1 trong trường hợp xuất phát từ s_0, máy thay 1 bởi X. Tại các trạng thái s_5 và s_6, máy dịch chuyển qua trái mà không làm thay đổi các ký hiệu trên băng cho đến khi gặp ký hiệu X, máy quay trở lại s_0 và tiếp tục quá trình trên cho đến khi máy dừng ở các trường hợp sau:

- Máy ở trạng thái s_3 gặp 1 hoặc ở trạng thái s_4 gặp 0. Trong trường hợp này rõ ràng ω ban đầu không có dạng $\alpha\alpha^R$ và máy không đoán nhận từ này.
- Máy ở trạng thái s_0 và gặp ký hiệu B. Điều này có nghĩa là các ký hiệu 0, 1 trên băng đã được thay bằng X hoặc B. Điều này chỉ xảy ra khi xâu vào ω có dạng $\alpha\alpha^R$. Vậy $T(M)=\{\alpha\alpha^R \mid \alpha\in\{0, 1\}^*\}$.

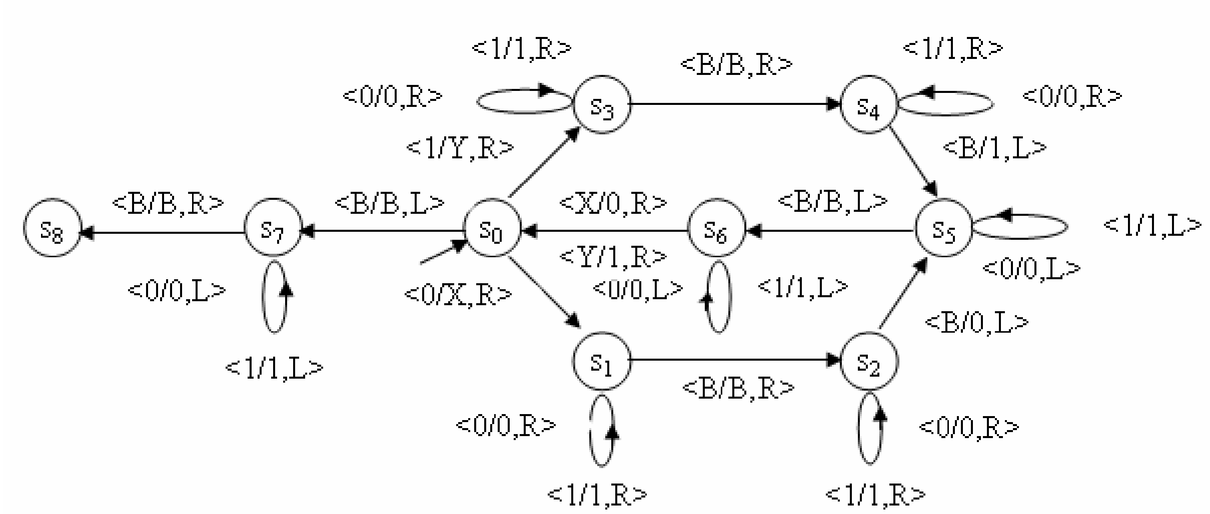
Định nghĩa 1.5 Cho máy Turing $M = <Q, \Sigma, \Delta, \delta, s_0, B, F>$. Hàm được xác định bởi máy Turing M là hàm:

$$f_M(\omega) = \begin{cases} \psi \text{ khi } <\varepsilon, s_0, a\omega'> \vDash <\psi', q, b\psi''> \text{ là một quá trình tính toán hoàn chỉnh} \\ \text{ở đây } a\omega' = \omega, \psi' b\psi'' = \psi; \\ \text{Không xác định khi không tồn tại quá trình như vậy.} \end{cases}$$

Thí dụ 1.2 Cho hàm $f(\omega) = \omega B\omega$ ($\omega\in\{0, 1\}^*$). Ta xây dựng máy Turing M xác định hàm f như sau:

$$M = <\{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, s_0, B, \varnothing>,$$

trong đó ánh xạ chuyển δ được chỉ ra trong đồ thị chuyển dưới đây:

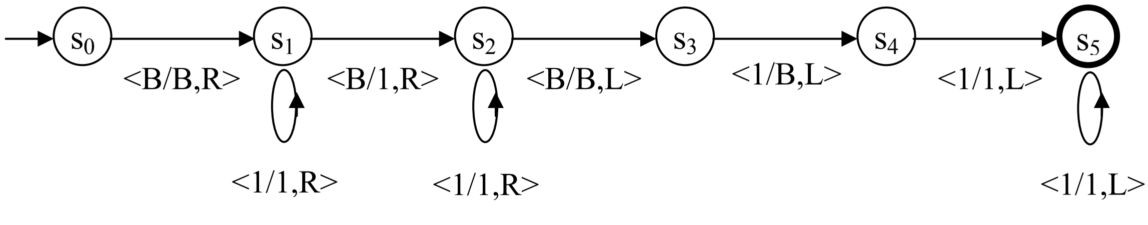


M hoạt động như sau: ký hiệu đầu tiên của ω được thay bởi X hoặc là Y tuỳ thuộc vào ký hiệu đó là 0 hay 1, sau đó đầu đọc/ghi chuyển sang phải để tìm ký hiệu B, thay ký hiệu B tiếp theo bằng 0 hoặc 1 tuỳ thuộc trước đó đã ghi x hay Y. Sau đó chạy ngược lại để tìm ký hiệu X hay Y và thay nó bởi 0 hoặc 1 tương ứng và lại chuyển sang phải. Nếu ký hiệu này là B thì tính toán kết thúc, ngược lại thì lặp lại quá trình trên. Dễ dàng thấy rằng, sau mỗi vòng thực hiện một ký hiệu của ω được ghi sang bên phải và khi quá trình tính toán kết thúc trên băng là ωBω hay $f_M = \omega B\omega$.

Định nghĩa 1.6 Cho hàm f: D→ N, với N là tập số tự nhiên, $D \subset N^m$ và m là một số nguyên dương. Ở đây, với mỗi số tự nhiên n, ký hiệu $\overline{n} = 1^{n+1}$. Ta nói hàm f có thể tính được bằng máy Turing nếu tồn tại máy Turing M xác định hàm sau:

$$h(B\overline{n}_1 B\overline{n}_2 \ldots \overline{n}_{m-1} B\overline{n}_m) = \begin{cases} \varphi B\overline{f(x_1, x_2, ..., x_m)} & \text{nếu } f(x_1, x_2, \ldots, x_m) \text{ tồn tại và} \\ & <\varepsilon, s_0, B\overline{n}_1 B\overline{n}_2 \ldots \overline{n}_{m-1} B\overline{n}_m> \vdash <\varphi, q, B\overline{f(x_1, x_2, ..., x_m)}> \\ & \text{là quá trình tính toán hoàn chỉnh;} \\ \text{Không xác định nếu } f(x_1, x_2, \ldots, x_m) \text{ không tồn tại.} & \end{cases}$$

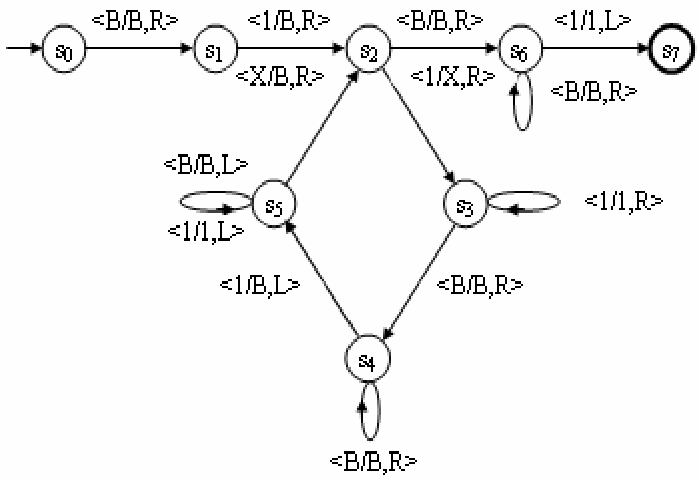
Thí dụ 1.3 Cho hàm $f(n_1, n_2) = n_1+n_2$. Ta xây dựng máy Turing M với đồ thị chuyển như sau:



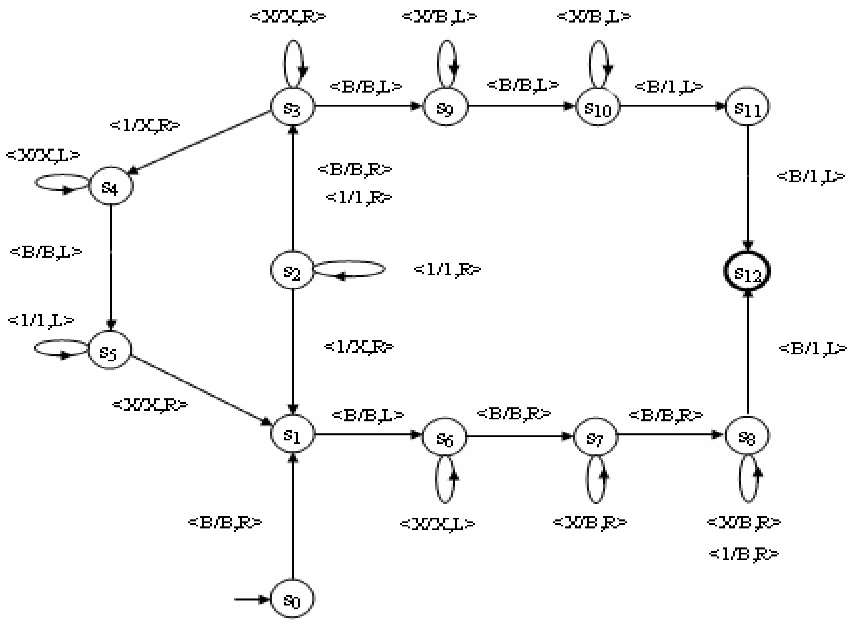
Đối với máy Turing M, với hình trạng đầu là $<\varepsilon, s_0, B\overline{n}_1 B\overline{n}_2>$ chỉ có các quá trình tính toán hoàn chỉnh với hình trạng kết thúc $<\varepsilon, s_5, B\overline{n_1 + n_2}>$. Do đó f là hàm có thể tính được bằng máy Turing.

Thí dụ 1.4 Cho hàm $f(n_1, n_2)=\begin{cases} n_2 - n_1 & \text{nếu } n_2 \geq n_1; \\ & \text{Không xác định nếu ngược lại.} \end{cases}$

Máy Turing M có đồ thị chuyển dưới đây với hình trạng đầu <ε, s_0, $B\overline{n_1}B\overline{n_2}$> và trong trường hợp xác định, hình trạng cuối là <φ, s_7, $B\overline{n_2 - n_1}$ > sẽ xác định hàm f.



Thí dụ 1.5 Cho hàm $f(n_1, n_2)=\begin{cases} 0 \; khi \; n_2 \geq n_1; \\ 1 \; khi \; n_2 < n_1. \end{cases}$ Khi đó hàm f có thể tính được bằng máy Turing M có đồ thị chuyển dưới đây. Hình trạng đầu là <ε, s_0, $B\overline{n_1}B\overline{n_2}$>, hình trạng cuối là sẽ là <ε, s_{12}, B1> trong trường hợp $n_2 \geq n_1$ và <ε, s_{12}, B11> trong trường hợp $n_1 > n_2$.



Thí dụ 1.6 Chúng ta đưa ra một ngôn ngữ chương trình thích hợp cho việc mô tả hoạt động của máy Turing. Ngôn ngữ này dựa trên các chỉ thị cơ sở sau:

k: print A,

k: move right,

k: move left,

k: if A then go to h,

k: stop,

ở đây, k và h là các số tự nhiên ký hiệu dòng chỉ thị, còn A là một ký hiệu trên băng.

Chương trình là một dãy chỉ thị. Việc thực hiện chương trình thông thường là theo thứ tự tự nhiên của cách viết trừ khi gặp lệnh if A then go to h (nếu ký hiệu hiện hành trên băng là A thì nhảy đến thực hiện lệnh có nhãn h). Sau đây là chương trình mô tả hoạt động của một máy Turing thực hiện phép toán nhân hai:

1: print B,

2: move left,

3: if 1 then go to 2,

4: print 1,

5: move left,

6: if 1 then go to 5,

7: print 1,

8: move right,

9: if 1 then go to 1,

10: stop,

Các máy Turing và các hàm có thể tính được bằng máy Turing đóng vai trò quan trọng trong lý thuyết thuật toán. Cơ sở cho việc xây dựng một số lý thuyết thuật toán từ máy Turing là định đề Church sau đây.

Định đề Church: Lớp các hàm có thể tính được bằng thuật toán trùng với lớp các hàm có thể tính được bằng máy Turring.

§2. Máy Turing phổ dụng

Mở đầu

Máy Turing phổ dụng là một máy Turing có thể bắt chước sự hoạt động của bất kỳ máy Turring nào. Máy Turing phổ dụng U có thể được hiểu như sau: với một cách mã hoá thích hợp ánh xạ chuyển trạng thái của máy Turing bất kỳ và từ ω trên bảng chữ vào. Với từ đã mã hoá này, máy Turing phổ dụng U dừng khi và chỉ khi máy Turing M dừng với từ ω. Ta có thể xem máy Turing phổ dụng như là mô hình toán học của máy tính điện tử ngày nay. Các máy này thực hiện công việc bằng cách mã hoá chương trình theo ngôn ngữ bên trong được gọi là ngôn ngữ máy.

Tập các ký hiệu ghi lên băng là hữu hạn nên ta có thể ký hiệu chúng như sau: S_0=B, S_1, S_2, …, S_m và có thể mã hoá bằng bộ các chữ số 1. Chẳng hạn, B−1, S_1−11, S_2−111, …, S_m−1^{m+1}.

Tương tự, tập các trạng thái là hữu hạn và ta cũng có thể mã hoá chúng: q_0−1, q_1−11, q_2−111, …, q_n−1^{n+1}.

Cuối cùng hai ký hiệu L và R cũng có thể mã hoá: L−1, R−11.

Bây giờ để mã hoá ánh xạ chuyển trạng thái của máy Turing, ta sử dụng bảng biểu diễn ánh xạ này. Trong bảng, các cột được ký hiệu bởi các ký hiệu có thể ghi lên băng, các dòng được ký hiệu bởi các trạng thái. Tiếp theo liệt kê các phần tử của bảng theo dòng cùng với các chỉ số dòng và cột tương ứng của chúng, thứ tự là các phần tử của dòng 1, dòng 2, …từ trái sang phải. Chẳng hạn, trên một dòng xuất hiện bộ $<q_i, S_j, S_k, C, q_h>$ có nghĩa là $\delta(q_i, S_j) = <q_h, S_k, C>$. Giữa các dãy mã của các bộ năm $<q_i, S_j, S_k, C, q_h>$ có chèn hai chữ số 0 và giữa mã các ký hiệu trong cùng một bộ năm được chèn bởi một chữ số 0. Máy Turing M được mã hoá như vậy ký hiệu là [M].

Ta chấp nhận mà không chứng minh ở đây là với máy Turing M bất kỳ tồn tại một máy Turing tương đương chỉ có một trạng thái kết thúc, vì vậy ta có thể xem q_0 là trạng thái đầu và q_1 là trạng thái kết thúc duy nhất của máy Turing M.

Thí dụ 2.1 Cho máy Turing M với ánh xạ chuyển được cho bởi bảng sau:

	B	S_1
q_0	$<S_1, R, q_1>$	$<S_1, R, q_0>$
q_1		$<S_1, L, q_2>$
q_2		$<S_1, L, q_2>$

Các phần tử của bảng được sắp xếp thành dãy dưới đây:

$<q_0, B, S_1, R, q_1>, <q_0, S_1, S_1, R, q_0>, <q_1, S_1, S_1, L, q_2>, <q_2, S_1, S_1, L, q_2>$.

Dãy này sẽ được mã hoá dưới dạng xâu nhị phân:

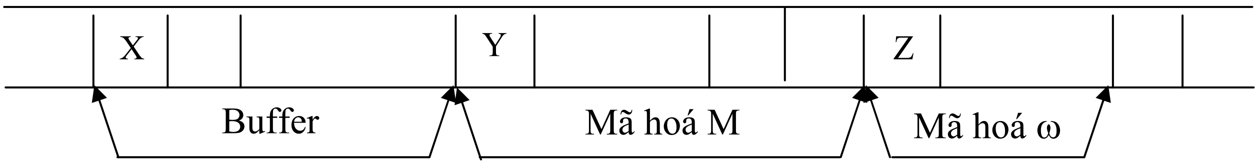
[M] = 10101101101100101101101101001101101101011100111011011010111.

Nếu trên băng vào có $\omega = S_1S_1BS_1$ thì mã tương ứng của nó là:

$[\omega]$ = 1101101011.

2.1. Hoạt động của máy Turing phổ dụng

Bây giờ giả sử máy Turing M có n trạng thái và bảng chữ ghi lên băng có m ký hiệu, thêm vào đó các ký hiệu, các trạng thái và ánh xạ chuyển của M được mã hoá như đã nói ở trên. Mô hình hoá hoạt động của máy Turing M bằng một máy Turing phổ dụng U có thể mô tả khái quát như sau: Trước hết [M] và [ω] cần phải được ghi lên băng của máy Turing phổ dụng U theo quy cách sau đây. Ký hiệu X được ghi lên băng chia băng thành hai nửa vô hạn. Nửa băng bên phải được dành ra ba đoạn kề nhau kể từ vị trí ký hiệu ngay sau X: Đoạn đầu tiên được gọi là Buffer gồm n+m+2 vị trí ký hiệu và tất cả được nhận ký hiệu 0; đoạn tiếp theo được gọi là vùng mã hoá của M, bắt đầu bởi ký hiệu Y, tiếp sau Y là [M] và được kết thúc bởi ba chữ số 0; đoạn sau cùng được gọi là đoạn mã của ω, bắt đầu bởi ký hiệu Z và tiếp theo là [ω]. Hình ảnh của băng lúc đầu là như sau:



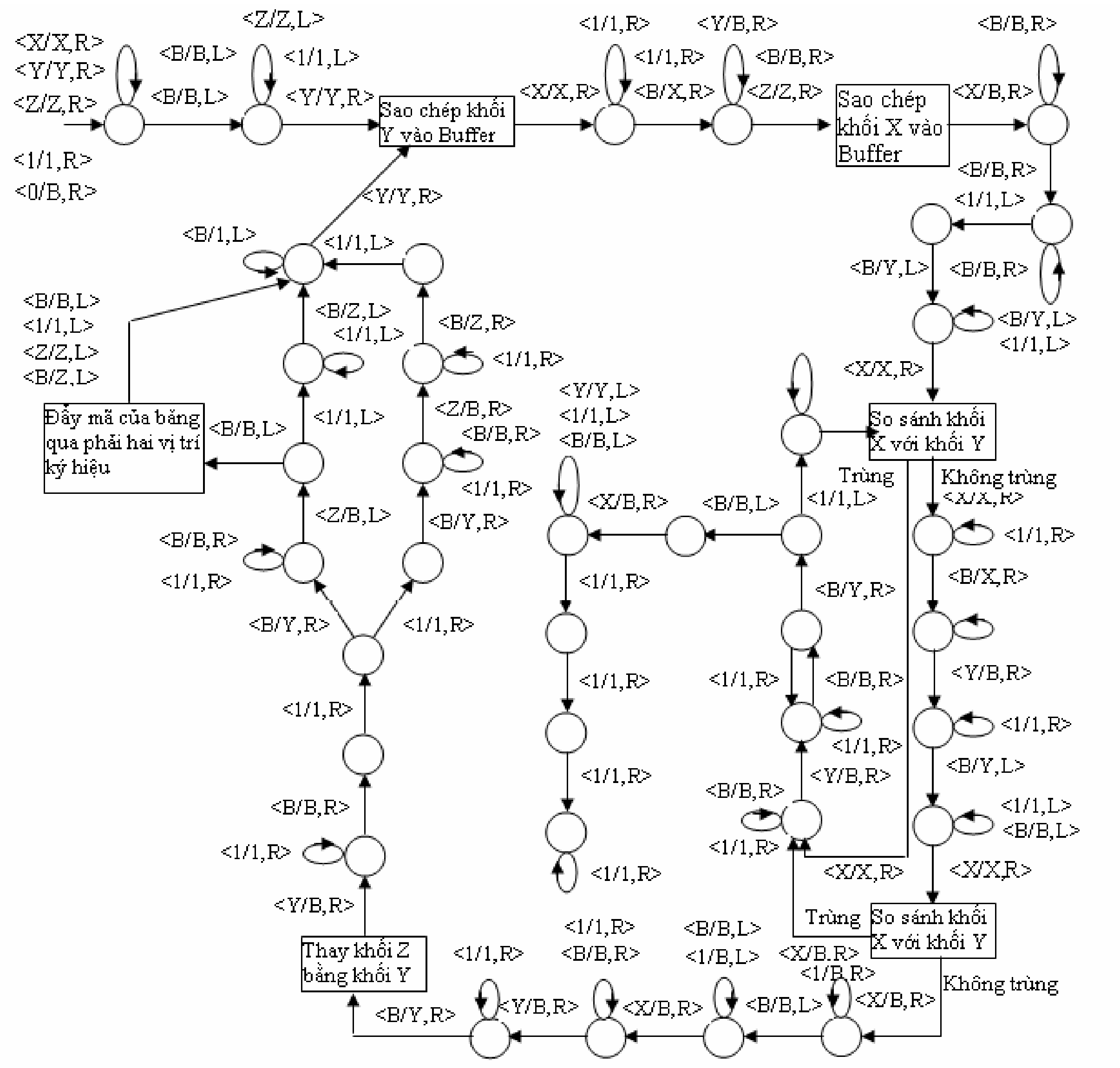
Buffer phục vụ cho việc ghi nhận hình trạng của M trong từng bước. Ta có thể sao chép vào vùng này trạng thái bên trong và mã hoá của ký hiệu đang đọc. Ký hiệu Y thường đứng trước bộ năm xác định trạng thái hiện hành của M, ký hiệu hiện hành trên băng, hướng chuyển động của đầu đọc trên băng. Z đánh dấu ký hiệu đang đọc trên băng của M.

Quá trình tính toán trong U mô phỏng hoạt động của máy Turing M với xâu vào ω được chia ra các pha thích hợp với việc dịch chuyển các hình trạng của M.

Một giai đoạn (pha) hoạt động của máy Turing phổ dụng U có thể tóm tắt như sau. Đầu tiên sao chép vào Buffer một khối các ký hiệu 1 nằm ngay sau Y (gọi là khối Y), sau đó ghi vào cuối khối vừa được chép một ký hiệu X, tiếp theo xoá ký hiệu Y, chạy sang phải tìm ký hiệu Z và sao chép khối ký hiệu 1 ngay sau Z (gọi là khối Z) vào Buffer ngay sau ký hiệu X rồi ghi lại ký hiệu Y trước [M]. Như vậy sau giai đoạn này trong Bufffer chứa mã của trạng thái và ký hiệu hiện hành của máy Turing M. Bước tiếp theo, máy Turing phổ dụng U so sánh hai khối ký hiệu 1 liên tiếp nhau sau Y với nội dung Buffer. Nếu trùng nhau thì tìm được bộ năm cần tìm. Nếu ngược lại thì tìm đến mã hoá của bộ năm tiếp theo sau Y và lại tiếp tục so sánh. Trong trường hợp giữa các bộ năm mô tả M không tìm thấy bộ nào thích hợp thì U dừng. Ngược lại, nếu tìm được bộ năm cần tìm thì xoá nội dung buffer rồi chuyển Y đến trước phần tử thứ ba trong bộ năm đó. Đổi nội dung của khối sau Z bởi nội dung của khối sau Y và chuyển Y đến trước phần tử thứ tư của bộ năm. Sau khi đã đọc xong phần tử thứ tư mà nó xác định hướng chuyển động của đầu đọc/ghi của M và U chuyển ký hiệu Y đến sau phần tử trước phần tử thứ năm. Tuỳ thuộc vào nội dung của khối thứ tư (một ký hiệu 1 hay hai ký hiệu 1) U chuyển Z qua phải hay qua trái một khối. Nếu Z lúc đầu nằm ở tận cùng trái của băng ghi và M cần dịch chuyển sang phải thì U đẩy mã của từ sang phải và ghi mã hoá của ký hiệu trắng vào sau Z. Nếu Z nằm tận cùng phải của băng và cần chuyển sang phải, khi đó U ghi mã của ký hiệu trắng vào cuối từ. Khi hoàn thành các công việc trên khối ký hiệu 1 đứng sau Y, ký hiệu trạng thái hiện hành của M, còn khối sau Z xác định ký hiệu M cần đọc tiếp theo. Như vậy, giai đoạn tiếp theo của việc mô phỏng bước tiếp theo của M có thể bắt đầu.

Các giai đoạn hoạt động của máy Turing phổ dụng U mô hình hoá hoạt động từng bước của máy Turing M như dã chỉ ở trên. Ngoài ra, U còn thực hiện công việc sau đây. Đầu tiên U thay tất cả các ký hiệu 0 trên ba đoạn của băng vào bằng các khoảng trắng, cuối công việc, khi M dừng máy U còn kiểm tra liệu trạng thái cuối của M có phải là trạng thái kết thúc hay không.

Các pha của một máy Turing phổ dụng U được chia thành chín phần, với đồ thị chuyển trạng thái phù hợp cho việc mô tả máy, được cho trong hình dưới đây:



– Phần 1: Thay các ký hiệu 0 bởi ký hiệu B và đầu đọc/ghi chuyển đến trước Y.

– Phần 2: Sao chép mã của trạng thái hiện hành vào buffer.

– Phần 3: Sao chép mã của ký hiệu cần đọc trên băng của M vào buffer.

– Phần 4: Đặt X và Y vào trước buffer và trước ký hiệu của [M].

– Phần 5: Tìm bộ năm có mã của trạng thái và ký hiệu trên băng trùng với buffer.

– Phần 6: Xoá buffer.

– Phần 7: Thay mã ký hiệu đã đọc bằng mã ký hiệu mới của M.

– Phần 8: Đẩy Z sang phải hay sang trái một khối mà mã ký hiệu của khối đó sẽ được đọc trong pha tiếp theo. Nếu cần thì ghi mã một khoảng trắng vào phải hoặc trái từ trên băng của M.

– Phần 9: Máy Turing phổ dụng U dừng ở trạng thái kết thúc khi và chỉ khi M dừng ở trạng thái kết thúc. Đồng thời trong vùng mã hoá của từ trên băng sẽ chứa mã của từ đáng ra còn lại trên băng của M, còn mã của trạng thái cuối của M có thể thấy trên buffer.

§3. Vấn đề không giải được bằng thuật toán

3.1 Mở đầu:

Trong toán học thường gặp vấn đề cần xác định liệu một phần tử của một lớp vô hạn nào đó có một số tính chất đã cho hay không. Chẳng hạn, ta có thể hỏi liệu hai số tự nhiên cho trước có nguyên tố cùng nhau hay không, hoặc là một máy Turing có dừng sau một số hữu hạn bước hay không, … Các vấn đề trên được gọi là vấn đề thuật toán và có thể quy về vấn đề là liệu một từ trên bảng chữ nào đó có thuộc vào một ngôn ngữ hình thức đã cho hay không.

Một bài toán được gọi là không giải được bằng thuật toán nếu không tồn tại một máy Turing nào sau một số hữu hạn bước xác định được liệu một sự mã hoá cụ thể nào đó của bài toán có thuộc ngôn ngữ hình thức đã cho hay không. Trong trường hợp ngược lại thì bài toán được gọi là giải được bằng thuật toán. Như vậy, một vấn đề giải được bằng thuật toán nếu và chỉ nếu ngôn ngữ hình thức mô tả nó là đệ quy.

Sau đây ta sẽ nghiên cứu một vài tính chất của ngôn ngữ đệ quy và đệ quy đếm được. Việc chứng minh các tính chất này khá phức tạp nên ta không đi sâu vào chi tiết.

3.2. Định lý 3.1 Phần bù của một ngôn ngữ đệ quy là ngôn ngữ đệ quy.

Chứng minh: Giả sử L là một ngôn ngữ đệ quy trên bảng chữ Σ. Điều này có nghĩa là tồn tại một máy Turing M mà với từ $\omega \in \Sigma^*$ tuỳ ý nó dừng sau một số hữu hạn bước và T(M) = L. Thay tập trạng thái kết thúc của M bằng phần bù của nó, ta được một máy Turing mới M'. Dưới tác động cùng một từ, M' dừng với trạng thái kết thúc khi và chỉ khi dưới tác động của từ đó M dừng với trạng thái không kết thúc. Điều này dẫn đến T(M') = $\overline{L}$. Do M' dừng sau một số hữu hạn bước với mọi từ vào nên T(M') = $\overline{L}$ cũng là ngôn ngữ đệ quy.

3.3. Định lý 3.2 Nếu L là ngôn ngữ đệ quy đếm được trên bảng chữ Σ và phần bù $\overline{L}$ của nó cũng là đệ quy đếm được thì L là đệ quy.

Chứng minh: Giả sử M_1, M_2 là các máy Turing sao cho $T(M_1) = L$ và $T(M_2) = \overline{L}$. Ta cần xây dựng máy Turing M’ mà nó mô hình hoá đồng thời sự hoạt động của M_1, M_2. Điều ta cần có là dưới tác động của từ ω, máy Turing M’ ngừng hoạt động khi và chỉ khi M_1 hoặc M_2 đoán nhận từ ω. Ta muốn M_1 dừng với trạng thái kết thúc, còn M_2 dừng với trạng thái không kết thúc. Việc xây dựng này là có thể thực hiện được và với mọi từ ω máy Turing M’ sau một số hữu hạn bước sẽ dừng. Nếu $\omega \in \Sigma^*$ thì $\omega \in L$ hoặc $\omega \in \overline{L}$. Nếu $\omega \in \overline{L}$ thì M_2 sẽ đoán nhận ω sau một số hữu hạn bước. Từ đó nếu M_1 không dừng sau một số hữu hạn bước thì M’ phải hoàn thành công việc của mình trong thời gian hữu hạn. Như vậy $T(M_1) = L$ là đệ quy.

3.4. Định lý 3.3 Tồn tại một ngôn ngữ đệ quy đếm được nhưng không đệ quy.

Chứng minh: Xét ngôn ngữ T(U) được đoán nhận bởi máy Turing U. Khi đó T(U) là đệ quy đếm được.

Giả sử T(U) là đệ quy. Điều này có nghĩa là tồn tại một máy Turing M (không đòi hỏi là trùng với U) sao cho T(M) = T(U) và sẽ dừng sau một số hữu hạn bước dưới tác động của mọi từ trên bảng chữ vào. Ta xây dựng máy Turing M’ như sau:

Giả sử ω là một từ trên bảng chữ vào của M’. Trước hết M’ cho mã [ω] của ω, đồng thời trên cơ sở sắp xếp các từ trên bảng chữ vào tìm chỉ số i của ω (số thứ tự của ω trong dãy là i). Tiếp theo ứng với các chỉ số i, máy Turing M’ thiết lập sự mô tả mã của máy Turing thứ i trong dãy các máy Turing M_1, M_2, …Cuối cùng M thiết lập sự mô tả chuẩn của từ ω_i và M_i, $<M_i\omega_i>$. M’ bắt chước sự hoạt động của máy Turing M với từ $<M_i\omega_i>$ mà theo giả thiết nó tồn tại, đoán nhận ngôn ngữ T(U) và dừng sau một số hữu hạn bước đối với mọi từ trên bảng chữ vào. Nếu M kết thúc sự hoạt động với trạng thái không kết thúc thì khi đó M’ chuyển sang một trạng thái kết thúc riêng và dừng. Nếu M dừng ở trạng thái kết thúc thì M’ bắt đầu bắt chước sự hoạt động của máy Turing mà nó không dừng với mọi từ của bảng chữ vào.

Rõ ràng rằng máy Turing M’ đoán nhận từ $\omega = \omega_i$ khi và chỉ khi không đoán nhận $<M_i\omega_i>$. Ta biết rằng $T(M) = T(U) =\{<M_i\varphi> \mid \varphi \in T(M_i)\}$. Đồng thời mọi máy Turing đều có mặt trong dãy M_1, M_2, … và do đó M’ cũng nằm trong dãy này, có nghĩa là tồn tại một số tự nhiên h sao cho $M’= M_h$.

Bây giờ ta xem máy Turing M_1 hoạt động như thế nào với từ ω_1. Ta nhận được $\omega_1 \in T(M')$ khi và chỉ khi $\omega_1 \in T(M_1)$. Điều này mâu thuẫn với giả thiết. Vậy T(U) không đệ quy.

Hệ quả 3.1 Tồn tại một ngôn ngữ hình thức nhưng không đệ quy đếm được.

Chứng minh: Như trong chứng minh của Định lý 4.3.4, T(U) là đệ quy đếm được nhưng không đệ quy. Do đó theo Định lý 4.3.3, phần bù của T(U) là không đệ quy đếm được.

3.5. Định lý 3.4 Một ngôn ngữ hình thức là loại 0 khi và chỉ khi nó là đệ quy đếm được. Điều này có nghĩa là lớp ngôn ngữ hình thức loại 0 chính là lớp ngôn ngữ đệ quy đếm được.

Chú ý: Nhờ vào Định lý 4.3.4, ta thấy rằng có những ngôn ngữ đệ quy đếm được nhưng không đệ quy. Với việc mã hoá thích hợp, ta chỉ ra rằng nhiều vấn đề không giải quyết được bằng thuật toán. Ta sẽ khảo sát một số vấn đề liên quan đến lớp các ngôn ngữ đệ quy đếm được. Chẳng hạn như một ngôn ngữ đệ quy đếm được có rỗng hay không, có hữu hạn hay không, có chính quy hay không, có là phi ngữ cảnh hay không và có là cảm ngữ cảnh hay không, … Tất cả các ngôn ngữ có tính chất này hình thành lên một lớp con của lớp các ngôn ngữ đệ quy đếm được. Khi ta khảo sát một ngôn ngữ có hay không một tính chất đã cho thì thực tế ta cần giải quyết vấn đề ngôn ngữ này có thuộc hay không lớp con đã cho của lớp các ngôn ngữ đệ quy đếm được.

Ta nói rằng một tính chất của các ngôn ngữ đệ quy đếm được là tầm thường nếu hoặc mọi ngôn ngữ đệ quy đếm được đều có tính chất này hoặc ngược lại mọi ngôn ngữ đệ quy đếm được không có tính chất này. Điều này có nghĩa là lớp con các ngôn ngữ xác định bởi tính chất này hoặc bằng rỗng hoặc bằng chính lớp các ngôn ngữ đệ quy đếm được. Như vậy các tính chất: một ngôn ngữ đã cho có rỗng hay không, có hữu hạn hay không, có chính quy hay không, … là không tầm thường. Có những ngôn ngữ đệ quy đếm được có những tính chất trên và có những ngôn ngữ đệ quy đếm được không có tính chất trên.

Từ Định lý 4.3.3, ta biết rằng muốn xác định bằng thuật toán việc thực hiện một tính chất nào đó thì vấn đề này được giải quyết bằng thuật toán khi và chỉ khi việc phủ định của tính chất này cũng được giải quyết bằng thuật toán.

3.6. Định lý 3.5 Cho trước một tính chất không tầm thường của lớp các ngôn ngữ đệ quy đếm được. Khi đó vấn đề xác định rằng một ngôn ngữ có tính chất này hay không là không giải quyết được bằng thuật toán.

Hệ quả 3.2 Việc xác định rằng một ngôn ngữ đệ quy đếm được có là:

a) Rỗng,
b) Hữu hạn,
c) Chính quy,
d) Phi ngữ cảnh,
e) Cảm ngữ cảnh,
f) Đệ quy

hay không là vấn đề không giải được bằng thuật toán.