BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP HÀ NỘI

PGS.TS. NGUYỄN VĂN ĐỊNH

GIÁO TRÌNH

AUTOMAT VÀ NGÔN NGỮ HÌNH THỨC

Chương 4

OTOMAT ĐẨY XUỐNG VÀ NGÔN NGỮ PHI NGỮ CẢNH

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 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:

4.1. Cây suy dẫn trong văn phạm phi ngữ cảnh.

4.2. Rút gọn các văn phạm phi ngữ cảnh

4.3. Dạng chuẩn Chomsky và dạng chuẩn Greibach

4.4. Otomat đẩy xuống

4.1. CÂY SUY DẪN TRONG VĂN PHẠM PHI NGỮ CẢNH

4.1.1. Cây suy dẫn đầy đủ trong văn phạm phi ngữ cảnh

Định nghĩa 4.1

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$. 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 $\Sigma \cup \Delta$. 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. Một đỉnh trong được gán nhãn là $A \in \Delta$, còn các đỉnh con của đỉnh này theo thứ tự từ trái sang phải được gán nhãn $B_1, B_2, \ldots, B_k$ nếu và chỉ nếu có quy tắc $A \rightarrow B_1B_2\ldots B_k$ trong văn phạm G.

Nếu đọc tất cả nhãn ở các lá theo thứ tự từ trái sang phải, ta sẽ nhận được một xâu nào đó gồm toàn ký hiệu chính, xâu đó sẽ là một từ trong L(G) và được gọi là kết quả của cây suy dẫn đầy đủ trong G.

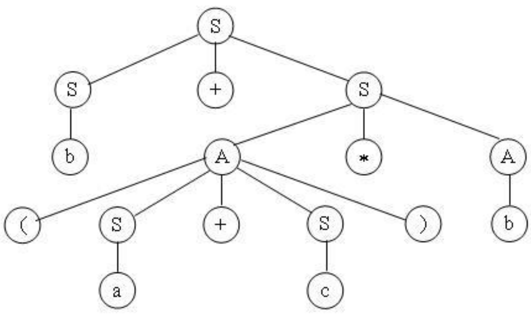
Thí dụ 4.1: Cho các văn phạm phi ngữ cảnh:

1. Cho $G_1 = <\{a, b, c, +, *, (,)\}, \{S, A\}, S, P_1>$

Với tập quy tắc:

$$P_1: \left| \begin{array}{l} S \rightarrow S+S \mid A*A \mid a \mid b \mid c, \\ A \rightarrow (S+S) \mid a \mid b \mid c. \end{array} \right.$$

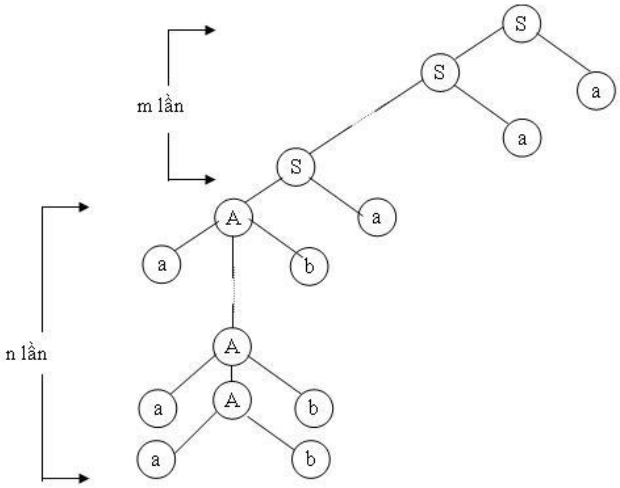
Cây suy dẫn của xâu b+(a+c)*b trong G_1 là:



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

2. Cho $G_2 = <\{a, b\}, \{S, A\}, S, \{S \to Sa \mid Aa, A \to aAb \mid ab\}>$,

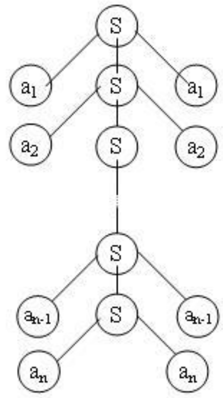
Cây suy dẫn của từ $\omega = a^n b^n a^m$ trong G_2 được cho trong hình 4.2 dưới đây:



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

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

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



Hình 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 4.1

Cho $G = <\Sigma, \Delta, S, P>$ là văn phạm phi ngữ cảnh và $\omega \in \Sigma^+$. 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 $S \to \varepsilon \notin P$. 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.

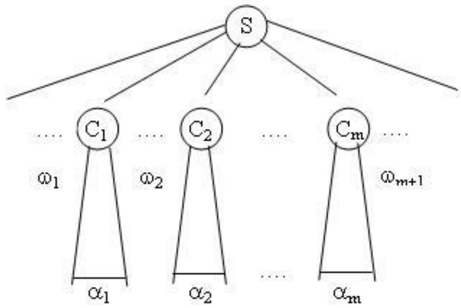
Giả sử ω là kết quả của một cây suy dẫn trong G_A và n là số ký hiệu không kết thúc trong cây. Bằng quy nạp theo n, ta sẽ chỉ ra rằng $\omega \in L(G_A)$.

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 $b_1, b_2, \ldots, b_k$. Theo định nghĩa của cây suy dẫn, ta có $A \rightarrow b_1b_2\ldots b_k$ hay $A \models \omega$.

Giả sử mệnh đề đúng với mọi cây suy dẫn có số ký hiệu không kết thúc không quá 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 $B_1, B_2, \ldots, B_k$. Nếu các đỉnh này đều là lá thì cây gốc A chỉ có một đỉnh có ký hiệu không kết thúc, điều này không thể xảy ra vì cây suy dẫn G_A có n ký hiệu không kết thúc. Giả sử trong các đỉnh này có các đỉnh trong là $C_1, C_2, ..., C_m$. Xét các cây con mà gốc của nó là $C_1, C_2, \ldots, C_m$, các cây con này có số ký hiệu không kết thúc không quá n−1. 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 $C_1 \models \alpha_1$, $C_2 \models \alpha_2, \ldots, C_m \models \alpha_m$. Sử dụng các suy dẫn này và quy tắc $A \rightarrow B_1B_2\ldots B_k$, ta nhận được:

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

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



Hình 4.4. Cây suy dẫn có kết quả là ω

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 $A \models \omega = b_1b_2 \ldots b_k$ (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 các nhãn là $b_1, b_2, \ldots, b_k$.

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 $A \rightarrow B_1B_2\ldots B_k$ và $C_1, C_2, \ldots, C_m$ là các ký hiệu không kết thúc trong các B_i ($1 \leq i \leq k$), tức là $B_1B_2\ldots B_k = \omega_1C_1\omega_2C_2\ldots \omega_mC_m\omega_{m+1}$, 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_{C_i} 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 hình 4.4. (trong hình 4.4, ta vẽ cây suy dẫn với A = S)

4.1.2. Văn phạm phi ngữ cảnh đa nghĩa

Định nghĩa 4.2

Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>. Ta nói văn phạm G là nhập nhằng (Ambiguity) 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 văn phạm không nhập nhằng hay đơn nghĩa.

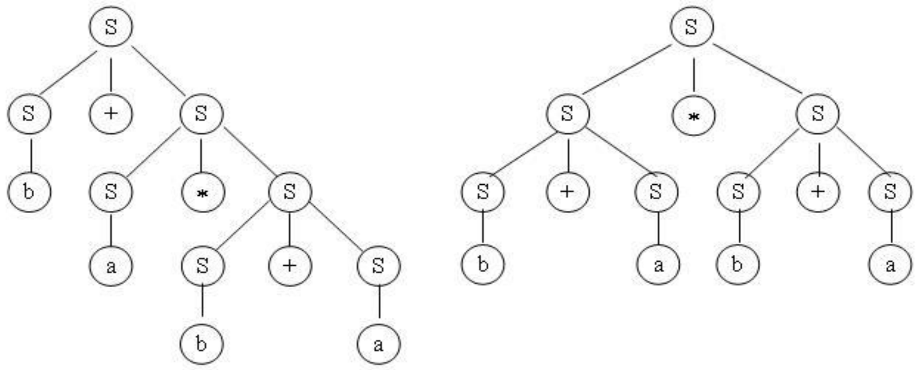
Một văn phạm phi ngữ cảnh được gọi là nhập nhằng vĩnh cửu nếu không tồn tại văn phạm phi ngữ cảnh đơn nghĩa nào tương đương với nó.

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

Thí dụ 4.2: Văn phạm phi ngữ cảnh sau là nhập nhằng:

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

vì xâu ω = b+a∗b+a là kết quả của hai cây suy dẫn khác nhau trong G, được cho trong hình 4.5:



Hình 4.5. Hai cây suy dẫn khác nhau cho từ **ω = b+a∗b+a**

Tương đương với văn phạm G ở trên, văn phạm:

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

là đơn nghĩa và L(G’) = L(G).

4.2. RÚT GỌN CÁC VĂN PHẠM PHI NGỮ CẢNH

Trong một văn phạm phi ngữ cảnh có thể có nhiều yếu tố dư thừa, chẳng hạn có những ký hiệu không hề tham gia vào quá trình sinh các ngôn ngữ, hoặc có những quy tắc dạng $A \rightarrow B$ chỉ là một phép đổi tên, hay các quy tắc có dạng $A \rightarrow \varepsilon$ chỉ sinh ra từ rỗng ε… Những yếu tố dư thừa đó làm mất thời gian trong quá trình hình thành các từ 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 ngôn ngữ sinh bởi văn phạm. 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. Dưới đây ta sẽ lần lượt xem xét các yếu tố dư thừa trong văn phạm phi ngữ cảnh cùng với những quy tắc để loại bỏ chúng.

Nói chung, mọi ký hiệu chính của bảng chữ cái vào Σ của văn phạm G đều được sử dụng trong các từ của ngôn ngữ L(G), (trong trường hợp trái lại, ký hiệu chính không được sử dụng sẽ được loại bỏ khỏi tập Σ). Vì vậy, để cho đơn giản ta chỉ xét các ký hiệu thừa đối với tập ký hiệu phụ Δ.

4.2.1. Rút gọn các ký hiệu thừa trong văn phạm phi ngữ cảnh

Định nghĩa 4.3

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$, ký hiệu $X \in \Delta$ 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 $\omega \in \Sigma^*$. Nói cách khác, ký hiệu X là có ích nếu nó tham gia vào quá trình sinh một từ nào đó của L(G).

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

Từ định nghĩa trên, ta suy ra X là ký hiệu thừa nếu X thuộc một trong các trường hợp sau:

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*. (*Non generating symboles*)

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*. (*Non reachable symboles*)

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

Thí dụ 4.3: Các văn phạm được xác định bởi các tập quy tắc dưới đây, đều chứa các ký hiệu thừa:

- $P_1 = \{S \rightarrow aSb \mid A \mid \varepsilon, A \rightarrow aA\}$, dễ thấy A là ký hiệu vô sinh.
- $P_2 = \{S \rightarrow A, A \rightarrow aA \mid \varepsilon, B \rightarrow bB \mid b\}$, dễ thấy B là ký hiệu không đến được.

Bổ đề 4.1 *(Quy tắc 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 mà trong G' không chứa các ký hiệu vô sinh.

Chứng minh: Ta sẽ xây dựng văn phạm G' theo quy tắc sau:

- Bước 1: Xây dựng tập Δ' không chứa ký hiệu vô sinh: Khởi đầu, cho $\Delta' = \emptyset$, nếu trong P có quy tắc dạng $A \rightarrow \omega$ với $A \in \Delta$, $\omega \in \Sigma^*$ thì kết nạp A vào Δ', tức là đặt $\Delta' := \Delta' \cup \{A\}$.
- Nếu $B \rightarrow X_1X_2\ldots X_n$ 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 B vào Δ', tức là đặt $\Delta' := \Delta' \cup \{B\}$.

Tiếp tục xét các quy tắc trong P, 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' không chứa các ký hiệu vô sinh.

- Bước 2: Xây dựng tập quy tắc P'gồm các quy tắc trong P không chứa các ký hiệu vô sinh.

Như vậy ta đã xây dựng được văn phạm G' không chứa các ký hiệu vô sinh, và vì các ký hiệu vô sinh không tham gia vào việc sinh ra các từ, cho nên rõ ràng L(G') = L(G).

Thí dụ 4.4: Loại ký hiệu vô sinh của văn phạm phi ngữ cảnh $G = <\{a, b\}, \{S, A, B, C\}, S, P>$, với tập quy tắc:

$$P : \left| \begin{array}{l} S \rightarrow aS \mid A \mid C, \\ A \rightarrow a, \\ B \rightarrow ab, \\ C \rightarrow aCb \end{array} \right.$$

Theo quy tắc trên, ta có $\Delta' = \{S, A, B\}$ và C là ký hiệu vô sinh, sau khi loại bỏ các quy tắc có chứa ký hiệu C, ta được văn phạm phi ngữ cảnh tương đương $G' = <\{a, b\}, \{S, A, B\}, S, P'>$, với tập quy tắc P':

$$P' : \left| \begin{array}{l} S \to aS \mid A, \\ A \to a, \\ B \to ab. \end{array} \right.$$

***Bổ đề 4.2** (Loại ký hiệu không đến được).*

Cho văn phạm phi ngữ cảnh G = <Σ, Δ, S, P>. Khi đó tồn tại văn phạm phi ngữ cảnh G' = <Σ, Δ', S, P'> tương đương với G, mà G' không chứa các ký hiệu không đến được, tức là $\forall X \in \Delta'$ có $\alpha, \beta \in (\Sigma \cup \Delta')^*$ để cho $S \models \alpha X \beta$.

Chứng minh: Ta luôn luôn giả thiết rằng bảng chữ cái vào Σ là không có ký hiệu thừa. Ta sẽ xây dựng văn phạm G' không chứa các ký hiệu không đến được theo quy tắc sau:

- Xây dựng Δ': Khởi đầu, cho Δ' = S, nếu một ký hiệu A đã được kết nạp vào Δ' và nếu trong P có $A \to \alpha$, với $\alpha \in (\Sigma \cup \Delta)^*$ thì ta kết nạp các ký hiệu phụ trong α vào Δ'. Thủ tục trên sẽ ngừng khi không còn bổ sung thêm được bất kỳ ký hiệu phụ nào vào tập Δ', và rõ ràng là Δ' chỉ chứa các ký hiệu đến được (tức là các ký hiệu phụ được dẫn xuất từ S)

- Tập P'chỉ gồm các quy tắc trong P mà chứa các ký hiệu đến được thuộc tập Δ'.

Như vậy ta đã xây dựng được văn phạm G' không chứa các ký hiệu không đến được, vì các ký hiệu không đến được không tham gia vào việc sinh ra các từ, cho nên L(G') = L(G).

***Chú ý*:** Ta có thể thực hiện việc xác định các ký hiệu không đến được bằng việc xây dựng một *đồ thị phụ thuộc* như sau:

- Tập đỉnh được gán nhãn là các ký hiệu phụ $A \in \Delta$

- Nếu trong P có quy tắc $A \to \alpha B \alpha'$, với $A, B \in \Delta$, α, α' là các xâu tùy ý thì có một cung từ A tới B.

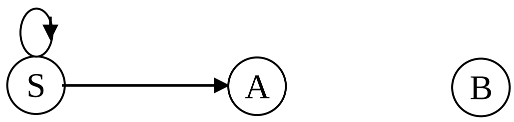
- Tất cả các đỉnh có đường đi từ S đến nó trong đồ thị phụ thuộc có nhãn là các ký hiệu đến được, các đỉnh còn lại ứng với các ký hiệu không đến được.

Sau khi loại bỏ các đỉnh không đến được, việc xây dựng tập quy tắc P' tương tự như trong định lý trên.

Thí dụ 4.5: Loại bỏ ký hiệu thừa của văn phạm phi ngữ cảnh G = <{a}, {S, A, B}, S, P>, với tập quy tắc:

$$
P: \begin{cases} S \rightarrow aS \mid A, \\ A \rightarrow a, \\ B \rightarrow aa. \end{cases}
$$

Vẽ đồ thị phụ thuộc với các đỉnh là S, A, B



Hình 4.6. Đồ thị phụ thuộc cho văn phạm G trong thí dụ 4.5

Từ đồ thị phụ thuộc, ta thấy không có đường đi nào từ S đến B, vậy B là ký hiệu không đến được. Loại bỏ ký hiệu không đến được B, và loại các quy tắc có chứa ký hiệu B, ta được văn phạm phi ngữ cảnh tương đương với văn phạm đã cho:

G' = <{a}, {S, A}, S, {S→ aS | A, A → a}>.

Dễ thấy rằng L(G) = L(G') = $\{a^n \mid n \geq 1\}$

Từ hai bổ đề trên ta có:

Định lý 4.2

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.

Thí dụ 4.6: Loại bỏ ký hiệu thừa của văn phạm phi ngữ cảnh G = <{a, b}, {S, A, B, C, D}, S, P>, với tập quy tắc:

$$
P: \begin{cases} S \rightarrow AB \mid CA, \\ B \rightarrow BC \mid AB, \\ A \rightarrow a, \\ C \rightarrow AB \mid b, \\ D \rightarrow aC. \end{cases}
$$

Bước 1: Trước hết, loại bỏ ký hiệu vô sinh B, ta được văn phạm G_1 với:

- tập ký hiệu phụ: Δ_1 = {S, A, C, D},
- và tập quy tắc: P_1 = {S→ CA, A→ a, C→ b, D→ aC}.

Bước 2: Loại bỏ ký hiệu không đến được của G_1:

Vẽ đồ thị phụ thuộc của G_1 như trên, không có một đường đi nào từ S đến D, từ đó xác định được D là ký hiệu không đến được, loại bỏ ký hiệu D khỏi tập ký hiệu phụ, loại bỏ các quy tắc có chứa ký hiệu D.

Hình 4.7. Đồ thị phụ thuộc cho văn phạm G_1 trong thí dụ 4.6

Cuối cùng ta có văn phạm G_2 không chứa các ký hiệu thừa và tương đương với văn phạm đã cho:

$$G_2 = <\{a, b\}, \{S, A, C\}, S, P_2>,$$

- với tập quy tắc: $P_2 = \{S\rightarrow CA, A\rightarrow a, C\rightarrow b\}$.

4.2.2. Rút gọn các quy tắc thừa trong văn phạm phi ngữ cảnh

Các quy tắc dạng $A\rightarrow \varepsilon$ hoặc $A\rightarrow B$ (với A, B là các ký hiệu phụ) được coi là các quy tắc thừa trong văn phạm phi ngữ cảnh, vì những quy tắc này chỉ kéo dài các quá trình sinh các từ trong văn phạm phi ngữ cảnh. Những quy tắc dạng $A\rightarrow B$ thực ra chỉ là một phép đổi tên, còn với quy tắc dạng $A\rightarrow \varepsilon$, thì nếu ngôn ngữ sinh bởi văn phạm G có chứa từ rỗng, thì chỉ cần một quy tắc $S\rightarrow \varepsilon$ là đủ, do đó những quy tắc trên cần được rút gọn.

Định nghĩa 4.4

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$, quy tắc trong P có dạng $A\rightarrow \varepsilon$, với $A\in \Delta$, được gọi là ε-quy tắc (ε- productions) . Ký hiệu $B\in \Delta$ trong văn phạm G được gọi là ký hiệu triệt tiêu (nullable symbols) nếu B suy dẫn được ra ε, tức là $B \models \varepsilon$.

Trong hầu hết các định lý, các thuật toán trên các văn phạm phi ngữ cảnh G, người ta thường giả thiết rằng L(G) không chứa từ rỗng ε. 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 văn phạm mới chỉ sai khác với L(G) một từ rỗng ε. Nếu cần sinh ra từ rỗng, thì sau khi loại bỏ hết các ε-quy tắc, ta chỉ cần thêm vào văn phạm mới một ε-quy tắc duy nhất $S\rightarrow \varepsilon$.

Các ε-quy tắc cũng làm văn phạm phi ngữ cảnh trở nên cồng kềnh. 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 4.3 (*Loại bỏ các ε-quy tắc*)

Cho văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$, khi đó tồn tại một văn phạm phi ngữ cảnh $G' = <\Sigma, \Delta, S, P'>$ không chứa các ε-quy tắc sao cho $L(G') = L(G) \setminus \{\varepsilon\}$.

Chứng minh: Để cho đơn giản, ta luôn giả thiết văn phạm G là không chứa các ký hiệu thừa, (nếu trái lại, ta dùng các quy tắc 4.1 và 4.2 để loại bó các ký hiệu thừa).

Ta sẽ xây dựng văn phạm G' không chứa các ε-quy tắc theo các bước sau:

Bước 1: Xác định tập N các ký hiệu triệt tiêu

- Nếu $A \rightarrow \varepsilon$ là một quy tắc thuộc P thì A là ký hiệu triệt tiêu,
- Nếu $B \rightarrow \alpha \in P$ mà α là một xâu gồm toàn ký hiệu triệt tiêu ($\alpha \in N^*$) 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.

Bước 2: Xây dựng tập quy tắc P' từ tập quy tắc P theo 2 bước:

- Bước 1: loại tất cả các *ε-quy tắc* trong P để nhận được P'.
- Bước 2: Với mỗi quy tắc trong P' có dạng (p): $A \rightarrow X_1X_2\ldots X_n$ thêm vào P' các quy tắc mới nhận được từ quy tắc (p) trên đây bằng cách mỗi lần loại bỏ một tập con các ký hiệu triệt tiêu trong vế phải của (p).

Chú ý rằng, trong bước 2, có thể xuất hiện những quy tắc giống nhau và những ε-quy tắc mới trong P', hãy xóa bỏ các ε-quy tắc mới và với những quy tắc giống nhau chỉ giữ lại một quy tắc.

Tập quy tắc P' xây dựng như trên sẽ không còn các ε-quy tắc, và dễ thấy rằng L(G') = L(G).

Việc xác định tập các ký hiệu triệt tiêu N là rất quan trọng trong quá trình loại bỏ các ε-quy tắc. Ta có thủ tục tìm tập N như sau:

Procedure to find set of nullable symbols.

$i := 0;\ N^i := \varnothing;$

$N^i := N^i \cup \{A \mid A \in \Delta$, if any $A \rightarrow \varepsilon \in P\};$

Repeat

$N^{i+1} := N^i \cup \{B \mid B \in \Delta$, if any $B \rightarrow X_1X_2\ldots X_n \in P,\ X_i \in N^i\ \forall i\};$

$i := i+1;$

Until

$N^{i+1} = N^i;$

$N := N^i;$

Return (N).

Thí dụ 4.7: Loại bỏ các ε-quy tắc trong văn phạm phi ngữ cảnh G = <{a, b}, {S, A, B}, S, P> với tập quy tắc:

$$P: \left| \begin{array}{l} S \rightarrow ABA, \\ A \rightarrow aA \mid \varepsilon, \\ B \rightarrow bB \mid \varepsilon. \end{array} \right.$$

Ta thấy các ε-quy tắc là $A \rightarrow \varepsilon$ và $B \rightarrow \varepsilon$, theo thủ tục trên, ta tìm được tập các ký hiệu triệt tiêu là N = {A, B, S}. Loại bỏ các ε-quy tắc và thêm vào các quy tắc mới, ta được văn phạm mới G_1 với tập quy tắc P_1:

$$P_1: \left| \begin{array}{l} S \rightarrow ABA \mid BA \mid AA \mid AB \mid A \mid B, \\ A \rightarrow aA \mid a, \\ B \rightarrow bB \mid b. \end{array} \right.$$

Định nghĩa 4.5

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

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

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

Định lý 4.4 (*Loại bỏ các quy tắc đơn*)

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 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. Ta xây dựng văn phạm phi ngữ cảnh G’ = <Σ, Δ, S, P’> tương đương với G và không chứa quy tắc đơn theo các bước sau đây :

- Bước 1: Xây dựng tập quy tắc mới P’ bằng cách loại bỏ mọi quy tắc đơn trong P.
- Bước 2: Với mỗi quy tắc đơn $A \rightarrow B$ đã loại bỏ ở bước 1, thêm vào P’ quy tắc $A \rightarrow \alpha$, nếu có quy tắc $B \rightarrow \alpha$ trong P.
- Bước 3: Nếu trong P’ xuất hiện các quy tắc đơn mới, lặp lại quá trình trên.

Văn phạm G’ với tập quy tắc mới P’ sẽ không chứa quy tắc đơn, và có L(G’) = L(G).

Thí dụ 4.8: Trở lại thí dụ 4.7, với văn phạm phi ngữ cảnh G = <{a, b}, {S, A, B}, S, P> với tập quy tắc P có chứa các ε-quy tắc. Sau khi loại bỏ các ε-quy tắc, ta được văn phạm không chứa ε-quy tắc G_1 = <{a, b}, {S, A, B}, S, P_1>, với tập quy tắc P_1 :

$$P_1 : \left| \begin{array}{l} S \to ABA \mid BA \mid AA \mid AB \mid A \mid B, \\ A \to aA \mid a, \\ B \to bB \mid b. \end{array} \right.$$

Trong P_1 có các quy tắc đơn $S \to A$ và $S \to B$, ta xây dựng tập quy tắc P_2 không chứa quy tắc đơn:

$$P_2 : \left| \begin{array}{l} S \to ABA \mid BA \mid AA \mid AB \mid aA \mid a \mid bB \mid b, \\ A \to aA \mid a, \\ B \to bB \mid b. \end{array} \right.$$

Ta nhận được văn phạm phi ngữ cảnh G_2 = <{a, b}, {S, A, B}, S, P_2> không chứa các quy tắc đơn và các ε-quy tắc. Có thể kiểm tra được rằng:

- $L(G) = \{a^n, b^m, a^ib^j, b^ja^i, a^ib^ja^k \mid m, n, i, j, k \geq 0\}$
- $L(G_2) = L(G_1) = \{a^n, b^m, a^ib^j, b^ja^i, a^ib^ja^k \mid m, n, i, j, k \geq 1\}$

Sự sai khác giữa hai ngôn ngữ sau với ngôn ngữ đầu tiên chỉ là một từ rỗng {ε}, do trong văn phạm G có các ε-quy tắc, còn các văn phạm sau thì không có các quy tắc loại này. Để nhận được văn phạm hoàn toàn tương đương với văn phạm G mà không chứa các quy tắc đơn và ε-quy tắc, ta chỉ cần thêm vào văn phạm G_2 một ε-quy tắc duy nhất $S \to \varepsilon$.

Từ các định lý 4.2, 4.3 và 4.4 ta có kết quả sau:

Định lý 4.5

Cho văn phạm phi ngữ cảnh tùy ý G = <Σ, Δ, S, P>, luôn tồn tại một văn phạm phi ngữ cảnh tương đương G' = <Σ, Δ', S, P'> không chứa ε-quy tắc, không chứa các quy tắc đơn và không chứa các ký hiệu thừa.

Chứng minh: Áp dụng các quy tắc trong các định lý 4.2, 4.3 và 4.4, rút gọn văn phạm G theo thứ tự sau:

Bước 1. Từ văn phạm G, loại bỏ các ε-quy tắc (nếu có) để nhận được văn phạm G_1.

Bước 2. Từ văn phạm G_1, loại bỏ các quy tắc đơn (nếu có) để nhận được văn phạm G_2.

Bước 3. Từ văn phạm G_2, loại bỏ các ký hiệu thừa theo thứ tự: loại bỏ các ký hiệu vô sinh (nếu có) sau đó loại bỏ các ký hiệu không đến được (nếu có) để nhận được văn phạm G_3.

Nếu sau một bước nào đó, lại xuất hiện các quy tắc đơn hoặc các ký hiệu thừa, thì lại áp dụng quy tắc tương ứng để loại bỏ cho đến khi không còn các ε-quy tắc, các quy tắc đơn và các ký hiệu thừa trong văn phạm cuối cùng.

Thí dụ 4.9: Cho văn phạm phi ngữ cảnh G = <{a, b, c, d}, {S, A, B, C, D, S, P>, với tập quy tắc:

	$S\to a \mid aA \mid B \mid C$,	(1–4)
	$A\to aB \mid \varepsilon$,	(5–6)
P :	$B\to aA \mid b$,	(7–8)
	$C\to cCD$,	(9)
	$D\to ddd$.	(10)

Hãy rút gọn văn pham G để được văn phạm tương đương không chứa ε-quy tắc, không chứa các quy tắc đơn và không chứa các ký hiệu thừa.

Bước 1. Từ văn phạm G, loại bỏ ε-quy tắc: có A là ký hiệu triệt tiêu, ta loại bỏ quy tắc $A\to \varepsilon$ và thêm vào các quy tắc $S\to a$, $B\to a$, ta được tập quy tắc mới

	$S\to a \mid aA \mid B \mid C$,	(1–4)
P_1 :	$A\to aB$,	(5)
	$B\to aA \mid b \mid a$,	(6–8)
	$C\to cCD$,	(9)
	$D\to ddd$.	(10)

Bước 2. Từ văn phạm P_1, loại bỏ các quy tắc đơn: $S\to B$, $S\to C$:

- Loại bỏ $S\to B$, thêm vào $S\to aA \mid b \mid a$, (do có quy tắc $B\to aA \mid b \mid a$)
- Loại bỏ $S\to C$, thêm vào $S\to cCD$, (do có quy tắc $C\to cCD$)

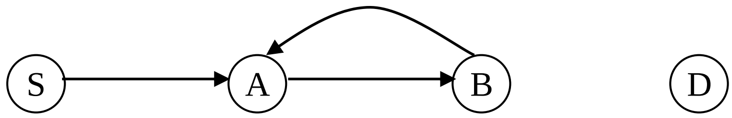
Ta được văn phạm G_2 tập quy tắc mới:

	$S\to a \mid aA \mid b \mid cCD$,
	$A\to aB$,
P_2 :	$B\to aA \mid b \mid a$,
	$C\to cCD$,
	$D\to ddd$.

Bước 3. Từ văn phạm G_2, loại bỏ các ký hiệu thừa: loại ký hiệu vô sinh C cùng với các quy tắc chứa C, ta được văn phạm G_3 tập quy tắc mới:

$$P_3 : \left| \begin{array}{l} S \rightarrow a \mid aA \mid b, \\ A \rightarrow aB, \\ B \rightarrow aA \mid b \mid a, \\ D \rightarrow ddd. \end{array} \right.$$

Từ G_3 với tập quy tắc P_3, có đồ thị phụ thuộc:



Hình 4.8. Đồ thị phụ thuộc cho văn phạm G_3 trong thí dụ 4.9

Rõ ràng D là ký hiệu không đến được vì không có đường đi nào từ đỉnh xuất phát S đến được D. Ta loại bỏ ký hiệu D cùng với các quy tắc chứa D, ta được văn phạm G_4 tập quy tắc mới:

$$P_4 : \left| \begin{array}{ll} S \rightarrow a \mid aA \mid b, & (1\text{–}3) \\ A \rightarrow aB, & (4) \\ B \rightarrow aA \mid b \mid a. & (5\text{–}7) \end{array} \right.$$

Cuối cùng, ta có văn phạm $G_4 = <\{a, b, c\}, \{S, A, B\}, S, P_4>$ tương đương với văn phạm G và G_4 không chứa ε-quy tắc, không chứa các quy tắc đơn và không chứa các ký hiệu thừa, thỏa mãn các yêu cầu của đầu bài.

4.3. MỘT SỐ DẠNG CHUẨN CỦA VĂN PHẠM PHI NGỮ CẢNH

Lớp văn phạm phi ngữ cảnh có rất nhiều ứng dụng và cũng là lớp văn phạm rất rộng. Văn phạm mà tất cả các quy tắc đều có vế trái chỉ gồm một ký hiệu phụ, có dạng $A \rightarrow \alpha$, đều là các văn phạm phi ngữ cảnh. Để thuận tiện cho việc nghiên cứu và ứng dụng lớp văn phạm này, người ta cần chia nó thành các lớp nhỏ hơn, theo những tiêu chuẩn nào đó. Lớp văn phạm chính quy cũng là một lớp con của lớp văn phạm phi ngữ cảnh. Chomsky và Greibach đã đưa ra các tiêu chuẩn để phân chia lớp văn phạm phi ngữ cảnh theo các dạng chuẩn khác nhau, gọi là dạng chuẩn Chomsky và dạng chuẩn Greibach.

Người ta cũng đã chứng minh được mọi văn phạm phi ngữ cảnh đều đưa được về văn phạm tương đương ở dạng chuẩn Chomsky hoặc Greibach.

4.3.1. Dạng chuẩn Chomsky

Định nghĩa 4.6

Văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> được gọi là văn phạm ở dạng chuẩn Chomsky (Chomsky Normal Form-CNF), nếu mọi quy tắc đều có dạng $A \rightarrow BC$ hoặc $A \rightarrow a$, với $A, B, C \in \Delta$, $a \in \Sigma$. (gọi là các quy tắc chuẩn Chomsky).

Từ định nghĩa trên, ta có nhận xét sau:

Nhận xét: Các văn phạm ở dạng chuẩn Chomsky sẽ thỏa mãn các điều kiện sau:

a/. Không chứa các ε-quy tắc,

b/. Không chứa các quy tắc đơn, dạng $A \rightarrow B$, với $A, B \in \Delta$,

c/. Không chứa các quy tắc mà vế phải có cả ký hiệu chính và ký hiệu phụ.

d/. Không chứa 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.

4.3.2. Đưa văn phạm phi ngữ cảnh về dạng chuẩn Chomsky

Bổ đề 4.3 *(Convert CFG to CNF).*

Với mọi văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> đều tồn tại văn phạm phi ngữ cảnh ở dạng chuẩn Chomsky G' = <Σ, Δ', S, P'> tương đương với G.

Để đơn giản trong chứng minh, ta có thể giả thiết văn phạm G là đã được rút gọn, tức là G không chứa các ký hiệu thừa, không chứa các ε-qui tắc và không chứa các qui tắc đơn. (Nếu trái lại, ta có thể áp dụng các định lý 4.2, 4.3 và 4.4 ở phần trên để rút gọn G). 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 phụ. Việc loại bỏ các quy tắc không chuẩn này cần đảm bảo nhận được văn phạm mới tương đương, và được 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:

$$A \rightarrow X_1 X_2 \dots X_m, \text{ với } X_i \in \Sigma \cup \Delta, \qquad (1)$$

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 X_a và thay thế cho ký hiệu chính $a \in \Sigma$ trong quy tắc (1), và thêm vào quy tắc $X_a \to a$. Lặp lại quá trình trên cho đến khi mọi ký hiệu trong vế phải của quy tắc (1) gồm toàn ký hiệu phụ.

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 hiệu phụ 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 mọi quy tắc đều có vế phải chỉ gồm một ký hiệu chính, hoặc vế phải chỉ gồm các ký hiệu phụ. Ta 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:

$$A \to Y_1Y_2\ldots Y_m, \text{ với } m>2, Y_i \in \Delta_1 \quad (2)$$

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:

$$A \to Y_1Z_1;\ Z_1 \to Y_2Z_2;\ Z_2 \to Y_3Z_3; \ldots;\ Z_{m-2} \to Y_{m-1}Y_m.$$

Ta nhận được văn phạm mới $G_2 = <\Sigma, \Delta_2, S, P_2>$ chỉ chứa các quy tắc ở dạng chuẩn Chomsky. 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.

Định lý 4.6

Với mọi ngôn ngữ phi ngữ cảnh L không chứa từ rỗng ε, luôn tồn tại một văn phạm ở dạng chuẩn Chomsky (CNF) sinh ra L.

Chứng minh: Do L là ngôn ngữ phi ngữ cảnh, nên tồn tại một văn pham phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$ sao cho L = L(G). Do L không chứa từ rỗng ε, nên có thể loại hết các ε-quy tắc, loại hết các quy tắc đơn và các ký hiệu thừa, sau đó áp dụng bổ đề 4.3 để được văn phạm tương đương G' ở dạng chuẩn Chomsky mà L(G') = L..

Thí dụ 4.10: Cho văn phạm phi ngữ cảnh G = <{a, b}, {S, A, B}, S, P>, với tập quy tắc P = {S→ ABA, A→ aA, A→ a, A→ B, B→ bB, B→ b}. Hãy xây dựng văn phạm ở dạng chuẩn Chomsky tương đương với G.

Trước hết, 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 với P_1 = {S→ ABA, A→ aA, A→ a, B→ bB, B→ b, A→bB, A→ b}.

Bây giờ áp dụng bổ đề 4.3 cho văn phạm G_1:

Bước 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 $S \to aA$, $A \to aA$ ta thay a bởi X_a, thêm X_a vào tập ký hiệu phụ mới, thêm quy tắc $X_a \to a$ vào tập quy tắc mới.

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

Ta nhận được G_2 với tập quy tắc mới $P_2 = \{S \to ABA, A \to X_aA, A \to a, B \to X_bB, B \to b, A \to X_bB, A \to b, X_a \to a, X_b \to b\}$. Tập P_2 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 tắc có hơn hai ký hiệu: thay quy tắc $S \to ABA$ bởi hai quy tắc $S \to AC$, $C \to BA$, và thêm C vào tập ký hiệu phụ.

Ta được tập quy tắc $P_3 = \{S \to ABA, A \to X_aA, A \to a, B \to X_bB, B \to b, A \to X_bB, A \to b, X_a \to a, X_b \to b, S \to AC, C \to BA\}$

Cuối cùng, ta được văn phạm $G_3 = <\{a, b\}, \{S, A, B, X_a, X_b, C\}, S, P_3>$ ở dạng chuẩn Chomsky, Theo cách xây dựng, dễ thấy rằng $L(G_3) = L(G)$. (Sinh viên tự chứng minh, xem như bài tập).

4.3.3. Dạng chuẩn Greibach

Định nghĩa 4.7

Văn phạm phi ngữ cảnh $G = <\Sigma, \Delta, S, P>$ được gọi là văn phạm ở dạng chuẩn Greibach (Greibach Normal Form-GNF), nếu mọi quy tắc của G đều có dạng $A \to a$ hoặc $A \to a\delta$, với $a \in \Sigma$, $\delta \in \Delta^*$ (gọi là các quy tắc chuẩn Greibach).

Từ định nghĩa trên, ta có nhận xét sau:

Nhận xét: Các văn phạm ở dạng chuẩn Greibach có vế phải của mọi quy tắc được bắt đầu bằng một ký hiệu chính, tiếp theo là dãy gồm không hoặc nhiều ký hiệu phụ.

Dưới đây ta sẽ xem xét một thí dụ về văn phạm ở dạng chuẩn Greibach:

Thí dụ 4.11: Xét các văn phạm phi ngữ cảnh:

▪ Xét văn phạm $G = < \{a, b\}, \{S\}, S, \{S \to aSb \mid ab >$, .

Dễ thấy rằng G là văn phạm phi ngữ cảnh, và $L(G) = \{a^nb^n \mid n \geq 1\}$.

▪ Xét văn phạm $G' = < \{a, b\}, \{S, A\}, S, \{S \to aSA \mid aA, A \to b >$.

Rõ ràng là $S \vdash aSA ... \vdash a^{n-1}SA^{n-1} \vdash a^nA^n \models a^n b^n$. hay $L(G') = \{a^nb^n \mid n \geq 1\}$, như vậy văn phạm G' là tương đương với văn phạm G, hơn nữa G' là dạng chuẩn Greibach, vì vậy có thể coi G’ là kết quả của việc biến đổi văn phạm phi ngữ cảnh G về dạng chuẩn Greibach.

Người ta đã chứng minh được rằng với mọi văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> đều có thể đưa được về dạng chuẩn Greibach, tức là từ văn phạm G có thể xây dựng được văn phạm G’ ở dạng chuẩn Greibach, sao cho L(G’) = L(G). Tuy nhiên việc xây dựng văn phạm G_1 như vậy khá phức tạp, chúng ta không trình bày ở đây.

Các văn phạm ở dạng chuẩn Greibach có rất nhiều ứng dụng trong việc xây dựng các otomat đẩy xuống, mà ta sẽ nghiên cứu ở phần cuối của chương này.

4.4. 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).

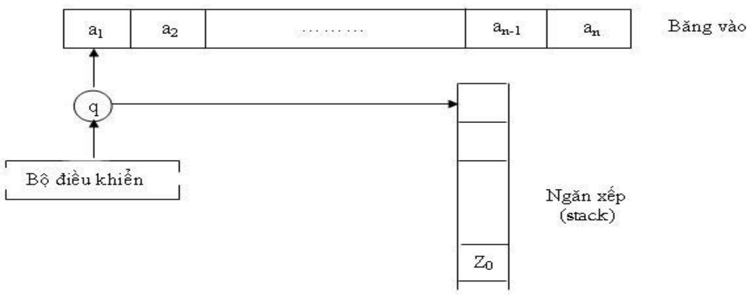
Tương tự, ta sẽ thấy trong phần này là lớp các ngôn ngữ phi ngữ cảnh do các văn phạm phi ngữ cảnh sinh ra sẽ trùng với lớp các ngôn ngữ được đoán nhận bởi otomat đẩy xuống không đơn định (Nondeteministic PushDown Automata).

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 NPDA, hay gọn hơn là PDA. Dưới đây, nếu không có chú thích gì thêm, thì các otomat được đề cập đến được hiểu là các otomat không đơn định (PDA). Khi cần chỉ rõ các otomat đẩy xuống là đơn định ta sẽ dùng ký hiệu DPDA-Deteministic PushDown Automata, hay gọn hơn là DPA.

4.4.1. 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. Tuy nhiên, với các otomat hữu hạn thì nó chỉ lưu trữ các trạng thái cố định của tập trạng thái và một trạng thái hiện tại, mà không lưu trữ được các kết quả của quá trình đọc xâu. Nhưng các otomat đẩy xuống còn có thêm bộ nhớ 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, vì vậy, các PA có thể nhớ và xử lý được các thông tin trong quá trình đọc xâu vào. 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 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 đọc 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 4.9 dưới đây:



Hình 4.9. 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, 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 và các trạng thái.

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 4.11: 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 ω = 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 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. Ký hiệu trên cùng của ngăn xếp khác với ký hiệu vừa đọc được, chứng tỏ nửa bên phải *x* không phải từ ngược của nửa bên trái xâu *x*, otomat sẽ dừng và không chấp nhận xâu *x*.

3. Otomat đọc hết xâu *x* nhưng ngăn xếp chưa bị loại hết, hoặc ngăn xếp rỗng trước khi đọc hết xâu *x*, chứng tỏ độ dài của hai xâu ở bên trái và bên phải ký hiệu ‘c’ là khác nhau, otomat sẽ dừng và không chấp nhận xâu *x*.

Nhờ có ngăn xếp mà otomat đẩy xuống có khả năng nhớ được nửa đầu của xâu x = $\omega c \omega^R$ và sau đó nó so sánh dần các ký hiệu phải nhất của nửa đầu (lưu trữ trong ngăn xếp) vói các ký hiệu trái nhất của nửa cuối ω^R của x. Otomat hữu hạn thông thường không có khả năng này.

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

4.4.2. Định nghĩa otomat đẩy xuống

Định nghĩa 4.7

Một otomat đẩy xuống không đơn định (PDA) là một bộ bảy:

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

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 $\Sigma \cap Q = \varnothing$;*

- *Γ 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 ngăn xếp;*

- *$z_0 \in \Gamma$ là ký hiệu đặc biệt, gọi là ký hiệu xuất phát của ngăn xếp (còn gọi là ký hiệu đáy của ngăn xếp);*

- *$q_0 \in Q$ gọi là trạng thái khởi đầu của otomat;*

- *$F \subset Q$ gọi là tập các trạng thái kết thúc, (F có thể là tập rỗng).*

- *Ánh xạ δ: $Q\times(\Sigma \cup \{\varepsilon\})\times\Gamma \rightarrow 2^{(Q \times \Gamma^*)}$ gọi là hàm chuyển của M, ký hiệu $2^{(Q \times \Gamma^*)}$ là tập tất cả các tập con của $Q\times\Gamma^*$.*

Vì otomat M là không đơn định, nên hàm chuyển có dạng $\delta(q, a, z) = \{<q_i, \gamma_i>, i = 1, 2,\ldots, m\}$ với kết quả là một tập các cặp giá trị $<q_i, \gamma_i>$ (với $i = 1, 2,\ldots, m$).

Với otomat đẩy xuống đơn định thì kết quả là duy nhất một cặp giá trị $<q_k, \gamma_k>$ nào đó, cũng có thể được xem như một tập chỉ gồm một cặp giá trị là $\{<q_k, \gamma_k>\}$, như vậy, otomat đẩy xuống đơn định có thể xem như một trường hợp đặc biệt của otomat đẩy xuống không đơn định. Tất cả những kết quả đã đúng với otomat đẩy xuống không đơn định cũng sẽ đúng với các otomat đẩy xuống đơn định.

Định nghĩa 4.8

Một đẳng thức dạng:

$$\boldsymbol{\delta(q, a, z) = \{<q_i, \gamma_i>, i = 1,2,\ldots,m\}}$$

trong đó $q, q_i \in Q$, $a \in \Sigma$, $z \in \Gamma$, $\gamma_i \in \Gamma^*$, được gọi là một bước chuyển của otomat đẩy xuống M.

Bước chuyển này được mô tả như sau: otomat M 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, nào đó, thay ký hiệu z ở đỉnh ngăn xếp bởi xâu γ_i, ($i = 1,\ldots, m$), đồng thời chuyển đầu đọc sang bên phải một ký hiệu. Quy ước rằng khi đưa γ_i vào ngăn xếp, ký hiệu bên trái nhất của γ_i (ký hiệu đầu tiên của γ_i) sẽ được đưa vào trươc và nằm ở phía dưới, còn ký hiệu bên phải nhất của γ_i sẽ nằm ở ô trên cùng của ngăn xếp.

- *Một đẳng thức dạng:*

$$\boldsymbol{\delta(q, \varepsilon, z) = \{<q_i, \gamma_i>, i = 1,2,\ldots,m\}}$$

gọi là một bước chuyển rỗng (hay ε-chuyển, bước chuyển "nhắm mắt") của otomat đẩy xuống M.

Thực chất của bước chuyển này là hàm ba biến $\delta(q, a, z)$ chấp nhận biến thứ hai là một ký hiệu rỗng, tương tự như *ε-chuyển* trong otomat hữu hạn. Khi đó otomat đẩy xuống vẫn chuyển sang trạng thái q_i và thay $z \in \Gamma$ ở đỉnh ngăn xếp bởi xâu γ_i, 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 $\gamma \in \Gamma^*$ trong ngăn xếp (quy ước là ký hiệu bên trái nhất của γ nằm ở đáy ngăn xếp).
- Trạng thái $q \in Q$.
- Phần xâu vào $\alpha \in \Sigma^*$ chưa được đọc đến trên băng vào (với quy ước ký hiệu bên trái nhất của α là ký hiệu sẽ được đọc tiếp).

Ba yếu tố trên lập nên một hình trạng của otomat M, được định nghĩa như sau:

Định nghĩa 4.9

Cho otomat đẩy xuống $M = \langle Q, \Sigma, \Gamma, \delta, q_0, z_0, F\rangle$. Một hình trạng của otomat M là một bộ ba:

$$\mathbf{K = \langle q, \alpha, \gamma \rangle}$$

trong đó $q \in Q$ là trạng thái hiện thời của M, $\alpha \in \Sigma^*$ là xâu còn lại chưa đọc trên băng vào, $\gamma \in \Gamma^*$ là xâu trong ngăn xếp, với ký hiệu trái nhất nằm ở đáy ngăn xếp.

Giả sử otomat đang ở trạng thái q, xâu $\alpha = a_1a_2\ldots a_k \in \Sigma^*$, $\gamma = x_1x_2\ldots x_m \in \Gamma^*$, khi đó hình trạng hiện thời của otomat là bộ ba $K = \langle q, a_1a_2\ldots a_k, x_1x_2\ldots x_m\rangle$. 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 $\langle p, \gamma\rangle \in \delta(q, a_1, x_m)$ thì hình trạng $K = \langle q, a_1a_2\ldots a_k, x_1x_2\ldots x_m\rangle$ có thể chuyển sang hình trạng $K' = \langle p, a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma\rangle$ và ký hiệu:

$\langle q, a_1a_2\ldots a_k, x_1x_2\ldots x_m\rangle \vdash \langle p, a_2\ldots a_k, x_1x_2\ldots x_{m-1}\gamma\rangle$, hay $K \vdash K'$.

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

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

3/. Nếu có dãy hữu hạn các hình trạng $K_0, K_1, \ldots, K_n$ sao cho: $K_0 \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_n$ thì ta ký hiệu: $K_0 \models K_n$.

4.4.3 Ngôn ngữ được đoán nhận bởi otomat đẩy xuống

Định nghĩa 4.10

Cho otomat đẩy xuống $M = <Q, \Sigma, \Gamma, \delta, q_0, z_0, F>$ 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: $K_0 \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_n$, với

1. Hình trạng đầu $K_0 = <q_0, \omega, z_0>$,

2. Hình trạng kết thúc $K_n = <p, \varepsilon, \gamma>$, với $p \in F$ (p- trạng thái kết thúc, còn γ tùy ý).

- Ta gọi ngôn ngữ được đoán nhận bởi otomat đẩy xuống M theo tập trạng thái kết thúc, là tập tất cả các từ được đoán nhận bởi M theo tập trạng thái kết thúc, ký hiệu là T(M). Vậy:

$$\mathbf{T(M) = \{\omega \in \Sigma^* \mid <q_0, \omega, z_0> \models <p, \varepsilon, \gamma>, p \in F\}}$$

Định nghĩa 4.11

Cho otomat đẩy xuống $M = <Q, \Sigma, \Gamma, \delta, q_0, z_0, F>$ và xâu $\omega \in \Sigma^*$. Ta nói rằng otomat M đoán nhận xâu ω theo ngăn xếp rỗng nếu tồn tại một dãy hữu hạn các hình trạng $K_0, K_1 \ldots, K_n$ sao cho: $K_0 \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_n$, với

1. Hình trạng đầu $K_0 = <q_0, \omega, z_0>$,

2. Hình trạng kết thúc $K_n = <p, \varepsilon, \varepsilon>$, với p là trạng thái tùy ý thuộc Q.

- Ta gọi ngôn ngữ được đoán nhận bởi otomat đẩy xuống M theo ngăn xếp rỗng, là tập tất cả các từ được đoán nhận bởi M theo ngăn xếp rỗng, ký hiệu là N(M). Vậy:

$$\mathbf{N(M) = \{\omega \in \Sigma^* \mid <q_0, \omega, z_0> \models <p, \varepsilon, \varepsilon>, p \in Q\}}$$

Thí dụ 4.12 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:

(1). $\delta(q_0, \varepsilon, z_0) = \{<q_0, \varepsilon>\}$, (4). $\delta(q_0, a, z_0) = \{<q_1, z_0z_1>\}$,

(2). $\delta(q_1, a, z_1) = \{<q_1, z_1z_1>\}$, (5). $\delta(q_1, b, z_1) = \{<q_2, \varepsilon>\}$,

. (3). $\delta(q_2, \varepsilon, z_0) = \{<q_0, \varepsilon>\}$, (6). $\delta(q_2, b, z_1) = \{<q_2, \varepsilon>\}$.

(giá trị của hàm chuyển $\delta(q, a, z)$ là không xác định tại các bộ ba khác)

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

Với α = aabb, ta có hình trạng khởi đầu $K_0 = \langle q_0, aabb, z_0\rangle$, dãy các hình trạng liên tiếp, bắt đầu từ K_0 là: $\langle q_0, aabb, z_0\rangle \vdash \langle q_1, abb, z_0z_1\rangle \vdash \langle q_1, bb, z_0z_1z_1\rangle \vdash \langle q_2, b, z_0z_1\rangle \vdash \langle q_2, \varepsilon, z_0\rangle \vdash \langle q_0, \varepsilon, \varepsilon\rangle$.

Vậy ta có: $\langle q_0, aabb, z_0\rangle \models \langle q_0, \varepsilon, \varepsilon\rangle$, theo định nghĩa, xâu α = aabb được đoán nhận bởi otomat M theo ngăn xếp rỗng.

Với β = abaab, ta có hình trạng khởi đầu $K_0 = \langle q_0, abaab, z_0\rangle$, dãy hình trạng liên tiếp, bắt đầu từ K_0 là: $\langle q_0, abaab, z_0\rangle \vdash \langle q_1, baab, z_0z_1\rangle \vdash \langle q_2, aab, z_0\rangle$ đến đây, hàm chuyển $\delta(q_2, a, z_0)$ là không xác định, otomat dừng và không đoán nhận xâu β.

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

Từ hình trạng $K_{n-1} = \langle q_2, \varepsilon, z_0\rangle$, nếu xem đây là hình trạng cuối cùng, có thể suy ra rằng otomat M đoán nhận xâu α = aabb theo tập trạng thái kết thúc.

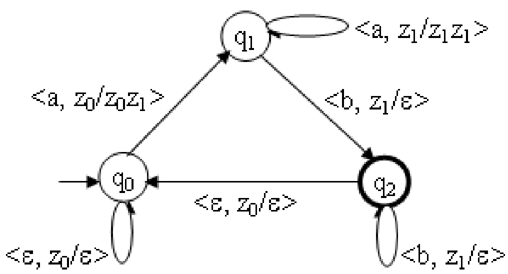
Tổng quát, ta có thể chứng minh được rằng $N(M) = T(M) = \{a^nb^n \mid n \geq 0\}$.

Chú ý

1/. Nếu otomat đoán nhận ngôn ngữ theo ngăn xếp rỗng thì có thể không cần đến tập trạng thái kết thúc, trong trường hợp này, có thể đặt $F = \emptyset$.

2/. 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 tập trạng thái Q. Các cung được xác định như sau: Với $a \in \Sigma \cup \{\varepsilon\}$; $p, q \in Q$, $z \in \Gamma$, $\gamma \in \Gamma^*$, nếu $\langle p, \gamma\rangle \in \delta(q, a, z)$ (tức là $\langle p, \gamma\rangle$ là một trong những giá trị của hàm chuyển δ) thì sẽ có một cung từ q tới p được gán nhãn là $(a, z/\gamma)$.

Chẳng hạn, đồ thị chuyển của otomat đẩy xuống M trong thí dụ 4.12 là:



Hình 4.10. Đồ thị chuyển trạng thái của PDA trong thí dụ 4.12

Việc vẽ đồ thị chuyển của otomat đẩy xuống là khó khăn hơn trường hợp otomat hữu hạn, tuy nhiên đây cũng là một cách biểu diễn khá thuận lợi cho các otomat đẩy xuống.

Thí dụ 4.13: Cho otomat đẩy xuống $M = <\{q_0, q_1\}, \{0, 1\}, \{z_0, z_1, z_2\}, \delta, q_0, z_0>$, trong đó:

$\delta(q_0, 0, z_0) = \{<q_0, z_0z_1>\}$, $\delta(q_0, 0, z_1) = \{<q_0, z_1z_1>, <q_1, \varepsilon>\}$,

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

$\delta(q_0, 1, z_1) = \{<q_0, z_1z_2>\}$, $\delta(q_0, 1, z_2) = \{<q_0, z_2z_2>, <q_1, \varepsilon>\}$,

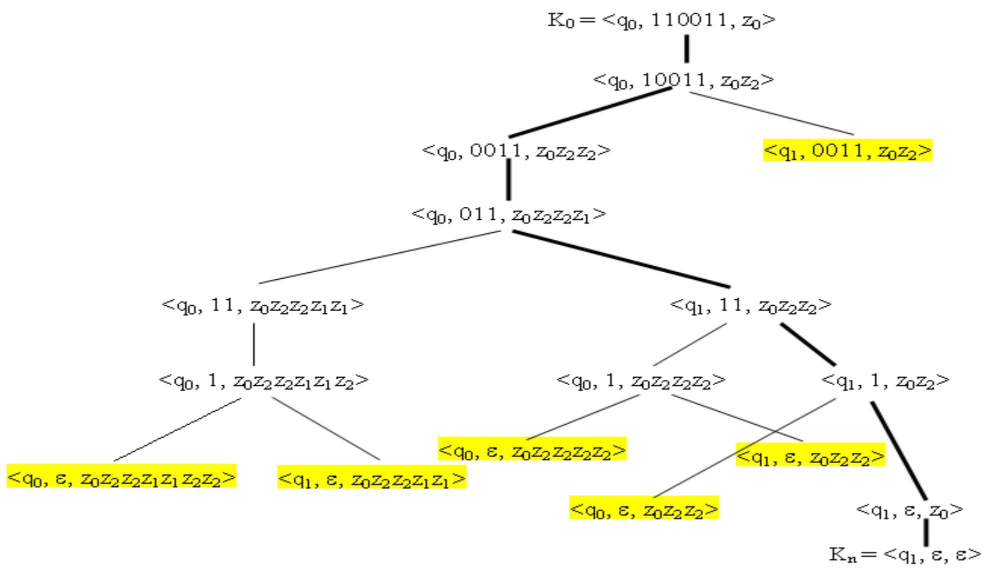
$\delta(q_1, 0, z_1) = \{<q_1, \varepsilon>\}$, $\delta(q_1, 1, z_2) = \{<q_0, z_2z_2>, <q_1, \varepsilon>\}$,

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

(giá trị của hàm chuyển $\delta(q, a, z)$ là không xác định tại các bộ ba khác)

Với $\omega = 110011$, ta có thể vẽ cây biểu diễn các khả năng biến đổi của các hình trạng như hình H.4.11. dưới đây.

Đường in đậm là dãy dịch chuyển từ hình trạng đầu $<q_0, \omega, z_0>$ đến hình trạng cuối $<q_1, \varepsilon, \varepsilon>$ Vậy otomat M đã đoán nhận xâu $\omega = 110011$ theo ngăn xếp rỗng. Các hình trạng tô đậm là các hình trạng tại đó không có bước chuyển, vì giá trị của hàm chuyển $\delta(q, a, z)$ là không xác định.



Hình 4.11. Cây biểu diễn dãy hình trạng trong thí dụ 4.13

4.4.4. Sự tương đương giữa văn phạm phi ngữ cảnh và otomat đẩy xuống

Ta đã biết rằng mỗi văn phạm phi ngữ cảnh sẽ xác định một ngôn ngữ phi ngữ cảnh, mặt khác mỗi otomat đẩy xuống cũng đoán nhận một ngôn ngữ phi ngữ cảnh, theo ngăn xếp rỗng hoặc theo tập trạng thái kết thúc. Ta nói một văn phạm phi ngữ cảnh tương đương với một otomat đẩy xuống nếu chúng cùng xác định một ngôn ngữ. Trong phần này, ta sẽ nghiên cứu sâu hơn các mối liên hệ giữa các otomat đẩy xuống và các văn phạm phi ngữ cảnh.

Định lý 4.7 *<Convert CFG to PDA >*

Cho L là một ngôn ngữ phi ngữ cảnh. Khi đó tồn tại một otomat đẩy xuống không đơn định M đoán nhận L theo ngăn xếp rỗng.

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 $M = <Q, \Sigma, \Gamma, \delta, q, z_0, \varnothing >$ đoán nhận L với:

- $Q = \{q\}$ là tập trạng thái chỉ gồm một trạng thái q,
- $\Gamma = \Sigma \cup \Delta$ là tập các ký hiệu ngăn xếp,
- $z_0 = S$ là ký hiệu xuất phát của ngăn xếp,
- $q \in Q$ là trạng thái khởi đầu,
- Hàm chuyển δ: $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$ được xác định như sau:

1. Với mỗi ký hiệu $A \in \Delta$, có $\delta(q, \varepsilon, A) = \{<q, \alpha^R> \mid A \to \alpha \in P, \alpha \in (\Sigma \cup \Delta)^*\}$.
2. Với mỗi ký hiệu $a \in \Sigma$, có $\delta(q, a, a) = \{<q, \varepsilon>\}$,

Người ta đã chứng minh được với otomat M xây dựng như trên, ta có N(M) = L(G). Chúng ta có thể kiểm chứng điều này qua thí dụ sau:

Thí dụ 4.14. Cho ngôn ngữ phi ngữ cảnh $L = \{0^n1^n \mid n \geq 1\}$, hãy xây dựng otomat đẩy xuống đoán nhận L theo ngăn xếp rỗng.

Trước hết, dễ dàng xây dựng văn phạm PNC sinh ngôn ngữ L:

$$G = <\{0, 1\}, \{S\}, S, \{S \to 0S1, S \to 01\}>.$$

Theo định lý 4.7, từ văn phạm G, ta xây dựng được một otomat đẩy xuống:

$$M = < \{q\}, \{0, 1\}, \{0, 1, S\}, \delta, q, S, \varnothing >,$$

với hàm chuyển δ xác định bởi:

- $\delta(q, \varepsilon, S) = \{<q, 1S0>; <q, 10>\}$. (1)
- $\delta(q, 0, 0) = \{<q, \varepsilon>\}$, (2)
- $\delta(q, 1, 1) = \{<q, \varepsilon>\}$. (3)

Ta xét quá trình đoán nhận xâu $\omega = 0011 \in L$ của otomat M như sau (các chỉ số là số hiệu công thức của hàm chuyển được áp dụng trong bước chuyển tương ứng):

$K_0 = <q, 0011, S> \vdash_1 <q, 0011, 1S0> \vdash_2 <q, 011, 1S \vdash_1 <q, 011, 110> \vdash_2 <q, 11,11> \vdash_3 <q, 1, 1> \vdash_3 <q, \varepsilon, \varepsilon>$. Vậy otomat M đoán nhận xâu $\omega = 0011$ theo ngăn xếp rỗng.

Tổng quát, ta có T(N) = L = L(G).

Định lý 4.8 *<Convert CFG to PDA >*

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, với \$ là ký hiệu mới, không thuộc $\Sigma \cup \Delta$,
- $z_0 = S$ là ký hiệu xuất phát của ngăn xếp,
- $q_0 \in Q$ là trạng thái khởi đầu,
- $F = \{q_2\}$ là tập các trạng thái kết thúc,
- Hàm chuyển δ: $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ được định nghĩa qua các biểu thức sau:

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

  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 $z_i \in \Delta$, $u_i \in \Sigma$, $v_i \in \Sigma \cup \Delta$ $(1 \leq i \leq n-1)$ và $\omega \in \Sigma^*$.

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:

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

Giả sử các quy tắc tiếp theo (sau quy tắc $S \rightarrow u_1z_1v_1$) trong D là $z_i \rightarrow x_i \in P$ ($i = 1, 2, \ldots, n-2$) 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 $S \rightarrow u_1z_1v_1$ 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 $z_i \rightarrow x_i$ biến đổi như sau:

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

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

Trong D, sử dụng quy tắc $z_{n-1} \rightarrow x_{n-1} \in P$ 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: $<q_0, \omega, z_0> \models <q_2, \varepsilon, \$>$ mà $q_2 \in F$ nên M đoán nhận được xâu ω theo tập trạng thái kết thúc.

Thí dụ 4.15: Cho văn phạm phi ngữ cảnh:

$G = <\{a, b\}, \{S, A\}, S, \{S \rightarrow aSb, S \rightarrow aAb, A \rightarrow ab\}>$.

Theo chứng minh của Định lý 4.8, ta xây dựng otomat đẩy xuống đoán nhận L(G) như sau:

$M = <\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, S, A, \$\}, \delta, q_0, S, \{q_2\}>$,với hàm chuyển:

- $\delta(q_1, \varepsilon, S) = \{<q_1, bSa>, <q_1, bAa>\}$, (1)
- $\delta(q_1, \varepsilon, A) = \{<q_1, ba\}$, (2)
- $\delta(q_1, a, a) = \{<q_1, \varepsilon>\}$, (3)
- $\delta(q_1, b, b) = \{<q_1, \varepsilon>\}$, (4)
- $\delta(q_1, \varepsilon, \$) = \{<q_2, \$>\}$, (5)
- $\delta(q_0, \varepsilon, S) = \{<q_1, \$S>\}$. (6)

Chẳng hạn, otomat M đoán nhận từ $\alpha = aabb \in L(G)$ qua dãy hình trạng sau: (các chỉ số là số hiệu công thức của hàm chuyển đước áp dụng trong bước chuyển tương ứng):

$K_0 = <q_0, aabb, S> \vdash_6 <q_1, aabb, \$S> \vdash_1 <q_1, aabb, \$bAa> \vdash_3 <q_1, abb, \$bA> \vdash_2 <q_1, abb, \$bba> \vdash_3 <q_1, bb, \$bb> \vdash_4 <q_1, b, \$b> \vdash_4 <q_1, \varepsilon, \$> \vdash_5 <q_2, \varepsilon, \$>$. do q_2 là trạng thái kết thúc, nên xâu $\alpha = aabb \in T(M)$..

Định lý 4.9

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

Chứng minh: Giả sử $M = <Q, \Sigma, \Gamma, \delta, q_0, z_0, F>$ là otomat đẩy xuống nào đó. Ta xây dựng otomat đẩy xuống $M' = <Q', \Sigma, \Gamma', \delta', q'_0, z'_0, F'>$ sao cho $N(M') = T(M)$.

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:

$Q' = Q \cup \{q_1, q_2\}$, $\Gamma' = \Gamma \cup \{\$\}$, trạng thái khởi đầu mới $q'_0 = q_1$, $z'_0 = \$$, $F' = \emptyset$,

Hàm chuyển δ': $Q' \times (\Sigma \cup \{\varepsilon\}) \times \Gamma' \rightarrow 2^{(Q' \times \Gamma'^*)}$ 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 $\delta'(q, \varepsilon, z) = \delta(q, \varepsilon, z) \cup \{<q_2, \varepsilon>\}$ nếu $q \in F$, $z \in \Gamma$,

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

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

Bây giờ ta chỉ ra $N(M') = T(M)$.

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

$<q'_0, w, z'_0> = <q_1, w, \$> \vdash K_1 \vdash K_2 \vdash \ldots \vdash K_t = <q, \varepsilon, \varepsilon>$, với $t \geq 2$, $K_i = <q_i, w_i, z_i>$, $w_i \in \Sigma^*$, $q_i \in Q'$, $z_i \in \Gamma'^*$ $(i = 1, 2, \ldots, t)$.

Theo cách xây dựng của M' thì

$K_1 = <q_0, w, \$z_0>$, $K_t = <q_2, \varepsilon, \varepsilon>$ và $w \in N(M')$. Từ đó $\exists i < t$ sao cho $K_i = <q_i, \varepsilon, \$z'_i> \vdash <q_2, \varepsilon, \$z'_{i+1}>$, 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 $K_j = <q_2, \varepsilon, \$z'_i>$, ở đây $z'_j \in \Gamma^*$ $(1<j<t)$.

Từ đó ta có $<q_0, w, z_0> \models <q_i, \varepsilon, \$z_i>$ 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. Vậy $N(M') = T(M)$.

Định lý 4.10

<Convert PDA to CFG >

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, ta cần xây dựng văn phạm phi ngữ cảnh G = <Σ, Δ, S, P> sao cho L(G) = N(M). Không mất tính chất tổng quát, giả sử ε∉ N(M). Ta xây dựng văn phạm G như sau:

- Δ = {S, [pzq]}với p, q∈ Q, z∈ Γ, ở đây [pqz] xem như một ký hiệu phụ.
- Tập quy tắc P xây dựng như sau:

1. Với mọi trạng thái p∈ Q, trong P có các quy tắc:

$S\rightarrow [q_0z_0p]$

2. Với mỗi bước chuyển δ(q, a, z) = <p, γ>, với a∈ Σ ; q, p∈ Q; $\gamma = z_1z_2\ldots z_k$; z, $z_i\in \Gamma$, (i = 1, 2, .k) thêm vào P các quy tắc:

$[qzq_i]\rightarrow a[pz_1p_1][q_1z_2p_2]\ldots [q_{k-1}z_kp_k]$, với q_i, $p_i\in$ Q.

3. Với mỗi bước chuyển δ(q, a, z) = <p, ε>, với a∈ Σ; p, q ∈ Q; z ∈ Γ, thêm vào P quy tắc:

$[qzp]\rightarrow a$

4. Với mỗi bước chuyển δ(q, ε, z) = <p, ε >,với p, q∈ Q; z∈ Γ thêm vào P quy tắc:

$[qzp]\rightarrow \varepsilon$.

Người ta đã chứng minh được rằng văn phạm G được xây dựng như trên là văn phạm phi ngữ cảnh mà L(G) = N(M).

Thí dụ 4.15: Cho otomat đẩy xuống M = <{q, p}, {0, 1}, {Z_0 , Z_1}, δ, q¸ Z_0, ∅ >, với hàm chuyển :

- $\delta(q, 1, Z_0) = <q, Z_0Z_1>$
- $\delta(q, 1, Z_1) = <q, Z_1Z_1>$
- $\delta(q, \varepsilon, Z_1) = <q, \varepsilon>$
- $\delta(q, 0, Z_1) = < p, Z_1>$
- $\delta(p, 1, Z_1) = <p, \varepsilon >$
- $\delta(p, 0, Z_0) = <q, Z_0 >$

Hãy xây dựng văn phạm phi ngữ cảnh G sao cho L(G) = N(M).

Ta sẽ xây dựng văn phạm G theo chứng minh định lý 4.10. Từ các bước chuyển của M ta xây dựng tập quy tắc của văn phạm G như sau:

$S \to [qZ_0q] \mid [qZ_0p]$

Với $\delta(q, 1, Z_0) = < q, Z_0Z_1>$ ta có các quy tắc :

$[qZ_0q] \to 1[qZ_0q][qZ_1q]$

$[qZ_0q] \to 1[qZ_0p][pZ_1q]$

$[qZ_0p] \to 1[qZ_0q][qZ_1p]$

$[qZ_0p] \to 1[qZ_0p][pZ_1p]$

Với $\delta(q, 1, Z_1) = <q, Z_1Z_1>$ ta có các quy tắc :

$[qZ_1q] \to 1[qZ_1q][qZ_1q]$

$[qZ_1q] \to 1[qZ_1p][pZ_1q]$

$[qZ_1p] \to 1[qZ_1q][qZ_1p]$

$[qZ_1p] \to 1[qZ_1p][pZ_1p]$

Với $\delta(q, \varepsilon, Z_1) = <q, \varepsilon>$ ta có các quy tắc :

$[qZ_1q] \to \varepsilon$

Với $\delta(q, 0, Z_1) = <p, Z_1>$ ta có các quy tắc :

$[qZ_1q] \to 0[pZ_1q]$

$[qZ_1p] \to 0[pZ_1p]$

Với $\delta(p, 1, Z_1) = <p, \varepsilon>$ ta có các quy tắc :

$[pZ_1p] \to 1$

Với $\delta(p, 0, Z_0) = <q, Z_0>$ ta có các quy tắc :

$[pZ_0q] \to 0[qZ_0q]$

$[qZ_0p] \to 0[qZ_0p]$.

Cuối cùng, các ký hiệu phụ dạng [qZp] có thể được đổi tên cho gọn:

$[qZ_0q] = A$, $[qZ_0p] = B$, $[pZ_0q] = C$, $[pZ_0p] = D$,

$[qZ_1q] = E$, $[qZ_1p] = F$, $[pZ_1q] = G$, $[pZ_1p] = H$.

Văn phạm PNC nhận được tương đương với otomat đẩy xuống M là

$G = <\{0, 1\},\{S, A, B, C, D, E, F, G, H\}, S, P>$, với tập quy tắc P:

$$P: \left| \begin{array}{l} S \to A \mid B \\ A \to 1AE \mid 1BG \\ B \to 1AF \mid 1BH \mid 0B \\ C \to 0A \\ E \to 1EE \mid 1FG \mid 0G \mid \varepsilon \\ F \to 1EF \mid 1FH \mid 0H \\ H \to 1 \end{array} \right.$$

Kết luận: Nếu ta gọi P_1, P_2, 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, theo các định lý trên, ta có:

- $P_1 \subset P_2$ (theo định lý 4.8),
- $P_2 \subset P_3$ (theo định lý 4.9),
- $P_3 \subset P_1$ (theo định lý 4.10).

Vì vậy, $P_1 = P_2 = P_3$, 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 thúc hay theo ngăn xếp rỗng.

BÀI TẬP CHƯƠNG 4

1. Cho văn phạm phi ngữ cảnh: G = <{x, +, ∗, (,)}, {S, A, B}, S, {S→ A | S+A, A→ A∗B | B, B→ x | (S)}>, và ω = (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→ A, A→ AbA, A→ a}>.

b). G = <{a, b}, {S, A, B}, S, {S→ A, A→ Bb, A→ Ab, B→ Bb, A→ a, B→ b}>.

3. Xây dựng các văn phạm phi ngữ cảnh không có ký hiệu thừa, tương đương với các văn phạm sau đây: (viết văn phạm xây dựng được với đầy đủ các thành phần)

a). G_1 = <Σ_1, Δ_1, S, P_1> với P_1 = {S→ aAa, A→ Sb | bCC, C→ abb, E→ aC}

b). G_2 = <Σ_2, Δ_2, S, P_2> với P_2 = {S→ aBa | BC, A→ aC |BCC, C→ a, B→ bcc, D→ E, E→ d}

c). G_3 = <Σ_3, Δ_3, I, P_3> với P_3 = {S→ aAa, A→ bBB, B→ ab, C→ aB}

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 = <Σ_1, Δ_1, I, P_1> với P_1 = {I→ aIb, I→ aABb, A→ B, B→ ε, A→ c}

b). G_2 = <Σ_2, Δ_2, I, P_2> với P_2 = {I→ aIbI, I→ bIaI, I→ ε}

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→ I+A, I→ A, A→ A*B, A→ B, B→ a}

b). G_2 = <{a, b, +, *}, {I, A, B, C}, I, P_2> với P_2 = {I→ A+B, A→ B*C, A→ B, B→ a, B→ C, C→ b}

c). G_3 = <{0, 1}, {I, B, C, D}, I, P_3> với P_3 = {I→ B, I→ C, B→ 0B, B→ 1B, B→ 011, C→ 0D, C→ 1C, C→ ε, D→ 0C, D→ 1D}

6. Xây dựng các văn phạm ở dạng chuẩn Chomsky tương đương với các văn phạm phi ngữ cảnh có tập quy tắc dưới đây:

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

b). G_5 với tập quy tắc P_5 = {S→ SaS, S→ b}, c). G_6 với tập quy tắc $P_6 = \{S \to aSS, S \to b\}$,

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

7. 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 $\omega\}$

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

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

b) $L = \{\omega c^2 \omega^R \mid \omega \in \{0,1\}^*$, kí hiệu ω^R chỉ xâu ngược của xâu $\omega\}$

9. Hãy xây dựng otomat đẩy xuống đoán nhận ngôn ngữ L gồm các từ $\omega \in \{0,1\}^*$mà có số ký hiệu '1' đúng bằng số ký hiệu '0'. (Chẳng hạn $01001101 \in L$, $01001110 \in L$ nhưng $10011 \notin L$)

10. Hãy xây dựng otomat đẩy xuống đoán nhận ngôn ngữ L gồm các từ $\omega \in \{0,1\}^*$mà với mọi tiền tố của ω, có số ký hiệu '1' không vượt quá số ký hiệu '0'. (Chẳng hạn $01001101 \in L$, nhưng $01001110 \notin L$).

11. Cho otomat đẩy xuống $M = <\{q\}, \{i, e\}, \{X,Z\}, \delta, q, Z>$, với hàm chuyển cho bởi các công thức: $\delta(q, i, Z) = \{(q, XZ)\}$, $\delta(q, e, X) = \{(q, \varepsilon)\}$ and $\delta(q, \varepsilon, Z) = \{(q, \varepsilon)\}$.

Áp dụng định lý 4.10, hãy xây dựng văn phạm phi ngữ cảnh G sao cho L(G) = N(M).