

Chương VI

ÔTÔMÁT ĐẨY XUỐNG

Nội dung chính: Trong chương này, chúng ta khảo sát một dạng mô hình ôtômát khác, có khả năng nhận diện được lớp ngôn ngữ mà văn phạm phi ngữ cảnh sinh ra - ôtômát đẩy xuống (PDA) - với sự bổ sung thêm của Stack đóng vai trò như một bộ nhớ trong quá trình ôtômát thực hiện các phép chuyển để nhận dạng ngôn ngữ. Tiếp theo đó, mối quan hệ tương đương giữa hai cơ chế - ôtômát đẩy xuống và CFG- dùng biểu diễn cho lớp văn phạm phi ngữ cảnh cũng sẽ được nêu ra và chứng minh chặt chẽ.

Mục tiêu cần đạt : Cuối chương này, sinh viên có thể:

- Thiết kế PDA chấp nhận một ngôn ngữ phi ngữ cảnh cho trước bằng Stack rỗng hay trạng thái kết thúc.
- Chuyển một PDA chấp nhận ngôn ngữ bằng trạng thái kết thúc sang PDA chấp nhận ngôn ngữ bằng Stack rỗng và ngược lại.
- Xây dựng NPDA chấp nhận ngôn ngữ sinh ra từ một CFG.
- Viết văn phạm phi ngữ cảnh sinh ra lớp ngôn ngữ được chấp nhận bởi một NPDA cho trước.

Kiến thức cơ bản: Để tiếp thu tốt nội dung của chương này, sinh viên cần nắm vững các tính chất của lớp ngôn ngữ phi ngữ cảnh; cơ chế đoán nhận ngôn ngữ của dạng máy trừu tượng ôtômát và các thành phần bắt buộc của chúng; ...

Tài liệu tham khảo :

[1] V.J. Rayward-Smith – *A First course in Formal Language Theory (Second Editor)* – McGraw-Hill Book Company Europe – 1995 (**Chapter 6 : Pushdown Automata**)

[2] John E. Hopcroft, Jeffrey D.Ullman – *Introduction to Automata Theory, Languages and Computation* – Addison – Wesley Publishing Company, Inc – 1979 (**Chapter 5 : Pushdown Automata**)

[3] Hồ Văn Quân – *Giáo trình lý thuyết ôtômát và ngôn ngữ hình thức* – Nhà xuất bản Đại học quốc gia Tp. Hồ Chí Minh – 2002.

[4] Copy right by **David Matuszek** - *NPDA's and CFG's*:

<http://www.netaxs.com/people/nerp/automata/npda-cfg0.html>

I. ÔTÔMÁT ĐẨY XUỐNG (PDA : PUSHDOWN AUTOMATA)

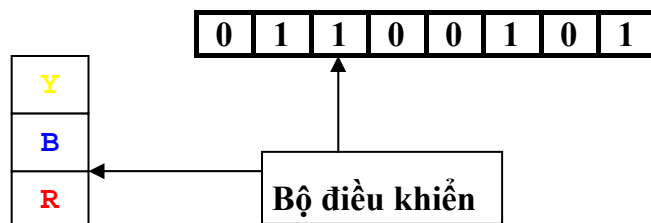
Như ta đã biết, lớp các ngôn ngữ chính quy được sinh từ văn phạm chính quy, đồng thời cũng được đoán nhận bởi các ôtômát hữu hạn (đơn định hoặc không đơn định). Trong phần này, chúng ta lại thấy một điều tương tự là lớp ngôn ngữ phi ngữ cảnh, được sinh ra từ văn phạm phi ngữ cảnh, cũng được đoán nhận bởi một loại ôtômát khác - gọi là ôtômát đẩy xuống (PDA).

Có một điều khác biệt là ở đây, chỉ có dạng ôtômát đẩy xuống không đơn định (NPDA) mới có thể đủ mạnh để đoán nhận lớp ngôn ngữ phi ngữ cảnh, còn dạng đơn định (DPDA) chỉ cho phép đoán nhận một tập con thực sự của lớp ngôn ngữ này. Tuy nhiên, tập con đó cũng bao gồm phần lớn các ngôn ngữ lập trình.

1.1. Mô tả PDA

Ôtômát đẩy xuống thực chất là một ôtômát với sự điều khiển cả hai: băng nhập và Stack (bộ đẩy xuống). Về cơ bản, nó vẫn giữ tất cả các thành phần của một ôtômát hữu hạn, với sự bổ sung thêm một ngăn xếp làm việc (Stack) đóng vai trò như một bộ giữ nhớ, nhờ đó mà khả năng ghi nhớ của ôtômát được tăng thêm. Stack xem như là một chồng đĩa, vì vậy cái đặt vào sau sẽ lấy ra trước (LIFO).

Với sự mở rộng này ôtômát đẩy xuống có thể chấp nhận cả các biểu thức không chính quy. Chẳng hạn tập hợp $L = \{wcw^R \mid w \in (0+1)^*\}$ (với w^R là chuỗi đảo ngược của chuỗi w) là một ngôn ngữ phi ngữ cảnh sinh bởi văn phạm $S \rightarrow 0S0 \mid 1S1 \mid c$ và nó không thể được chấp nhận bởi bất kỳ một ôtômát hữu hạn nào.



Hình 6.1 - Mô tả một PDA

Để chấp nhận ngôn ngữ L như trên ta dùng bộ điều khiển có hai trạng thái q_1, q_2 và một Stack trên đó ta đặt các đĩa xanh (B), vàng (Y), đỏ (R). Thiết bị sẽ thao tác theo các quy tắc sau đây:

- 1) Máy sẽ bắt đầu với một đĩa đỏ ở trên Stack và bộ điều khiển ở trạng thái q_1 .

2) Nếu 0 được đưa vào thiết bị thì ta đặt một đĩa xanh vào Stack. Nếu đưa 1 vào thiết bị ở trạng thái q_1 thì ta đặt một đĩa vàng vào Stack. Cả hai trường hợp thiết bị không thay đổi trạng thái.

3) Nếu c được đưa vào thiết bị ở trạng thái q_1 thì thiết bị đổi trạng thái sang q_2 và không thay đổi Stack.

4) Nếu 0 được đưa vào thiết bị ở trạng thái q_2 và đỉnh Stack là đĩa màu xanh thì đĩa được lấy ra. Nếu 1 đưa vào thiết bị ở trạng thái q_2 và đĩa vàng tại đỉnh Stack thì ta loại bỏ đĩa này. Trạng thái q_2 không thay đổi.

5) Nếu thiết bị ở trạng thái q_2 và đĩa đỏ tại đỉnh Stack ta loại bỏ đĩa này không cần ký hiệu nhập.

6) Ngoài các trường hợp trên thì thiết bị không thay đổi.

Các quy tắc trên được tóm tắt như trong bảng sau:

INPUT				
Đỉnh Stack	Trạng thái	0	1	c
Xanh	q_1	Thêm đĩa xanh, giữ nguyên q_1	Thêm đĩa vàng, giữ nguyên q_1	Chuyển sang q_2
	q_2	Xoá đỉnh Stack, giữ nguyên q_2		
Vàng	q_1	Thêm đĩa xanh, giữ nguyên q_1	Thêm đĩa vàng, giữ nguyên q_1	Chuyển sang q_2
	q_2		Xoá đỉnh Stack giữ nguyên q_2	
Đỏ	q_1	Thêm đĩa xanh, giữ nguyên q_1	Thêm đĩa vàng, giữ nguyên q_1	Chuyển sang q_2
	q_2	Xoá đỉnh Stack không cần đọc input		

Hình 6.2 - Mô tả hoạt động của PDA chấp nhận ngôn ngữ $\{wcw^R \mid w \in (0+1)^*\}$

Một chuỗi được chấp nhận bởi thiết bị nếu nó đã đọc duyệt qua đến hết chuỗi đồng thời với Stack trở về trạng thái rỗng.

Nhận xét : Nhờ Stack có khả năng lưu giữ một số bất kỳ các ký hiệu mà PDA có thể nhớ nửa phần đầu của chuỗi (w) cho tới khi gặp ký hiệu phân cách c, cho dù chuỗi có độ dài lớn đến bao nhiêu. Và sau đó, các ký hiệu này được đem ra để so sánh dần với phần chuỗi ngược còn lại (w^R). Một ôtômát hữu hạn không có được khả năng ghi nhớ đó.

1.2. Định nghĩa

Ôtômát đẩy xuống có một bộ điều khiển hữu hạn và một Stack. Stack chứa một chuỗi các ký hiệu thuộc một bộ chữ cái nào đó. Ký hiệu bên trái nhất của chuỗi xem như ký

hiệu tại đỉnh Stack. PDA không đơn định nếu như có một số hữu hạn các lựa chọn phép chuyển trong mỗi lần chuyển.

Phép chuyển sẽ có hai kiểu:

- Kiểu thứ nhất phụ thuộc vào ký hiệu nhập, tức là với một trạng thái, một ký hiệu tại đỉnh Stack và một ký hiệu nhập; PDA sẽ lựa chọn trạng thái kế tiếp và một chuỗi các ký hiệu thay thế trên Stack, đầu đọc dịch đi sang phải một ký hiệu.

- Kiểu thứ hai không phụ thuộc vào ký hiệu nhập, gọi là ϵ - dịch chuyển : tương tự như kiểu thứ nhất, chỉ ngoại trừ là ký hiệu nhập không được dùng và đầu đọc không dịch chuyển sau khi chuyển trạng thái. Thực chất, bước chuyển đặc biệt này là một sự tạm ngừng quan sát trên băng nhập để sắp xếp lại các ký hiệu trong ngăn xếp.

Có hai cách để định nghĩa ngôn ngữ chấp nhận bởi ôtômát đẩy xuống:

- Ngôn ngữ được chấp nhận bởi Stack rỗng: gồm tất cả các input mà sau một chuỗi các phép chuyển trạng thái làm cho ôtômát dẫn tới Stack rỗng.

- Ngôn ngữ được chấp nhận bởi trạng thái kết thúc: ta thiết kế một số trạng thái kết thúc, khi đó ngôn ngữ chấp nhận bởi ôtômát có thể định nghĩa gồm tất cả các input mà có một chuỗi các phép chuyển làm cho ôtômát dẫn tới một trong những trạng thái kết thúc.

Ta có thể thấy hai cách định nghĩa cho sự chấp nhận chuỗi này là tương đương nhau trong mọi trường hợp, có nghĩa là nếu một tập hợp được chấp nhận bởi Stack rỗng của một PDA nào đó thì cũng sẽ được chấp nhận bằng trạng thái kết thúc trên một PDA khác, và ngược lại. Thiết kế PDA chấp nhận chuỗi bằng trạng thái kết thúc thường phổ biến hơn, nhưng sẽ dễ dàng hơn để chứng minh nguyên lý cơ bản của PDA khi thiết kế PDA chấp nhận chuỗi bằng Stack rỗng. Nguyên lý này được phát biểu như sau: **Một ngôn ngữ được chấp nhận bởi PDA khi và chỉ khi nó là một ngôn ngữ phi ngữ cảnh.**

Một cách hình thức, ta định nghĩa:

Định nghĩa : Một ôtômát đẩy xuống M là một hệ thống $M (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, trong đó :

- 1) Q là tập hữu hạn các trạng thái
- 2) Σ là bộ chữ cái gọi là bộ chữ cái nhập
- 3) Γ là bộ chữ cái gọi là bộ chữ cái Stack
- 4) δ : hàm chuyển ánh xạ từ $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow$ tập con hữu hạn của $Q \times \Gamma^*$
- 5) q_0 là trạng thái khởi đầu
- 5) Z_0 là một chữ cái riêng của Stack gọi là ký hiệu bắt đầu trên Stack
- 6) $F \subseteq Q$ là tập các trạng thái kết thúc

(Trong trường hợp PDA được thiết kế chấp nhận ngôn ngữ bằng Stack rỗng thì tập hợp $F = \emptyset$)

Trừ khi ta dùng các ký hiệu khác, ta quy ước dùng chữ nhỏ gần đầu bảng chữ cái để chỉ các ký hiệu nhập, các chữ nhỏ cuối bảng chữ cái để chỉ các chuỗi nhập. Các chữ hoa và chữ Hy Lạp chỉ ký hiệu và chuỗi ký hiệu Stack.

Câu hỏi :



So sánh các thành phần trong định nghĩa hình thức cho một ôtômát đẩy xuống PDA với ôtômát hữu hạn đã khảo sát trong chương 3 ? Nêu những khả năng mở rộng của PDA so với FA ?

Sự dịch chuyển

Hàm chuyển phụ thuộc ký hiệu nhập :

$$\delta(q, a, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \dots, (p_m, \gamma_m)\}$$

trong đó q và p_i , $1 \leq i \leq m$, là các trạng thái thuộc tập Q , $a \in \Sigma$, Z là một ký hiệu Stack và $\gamma_i \in \Gamma^*$, $1 \leq i \leq m$, là PDA ở trạng thái q với ký hiệu nhập a và ký hiệu Z tại đỉnh Stack, nó đi vào một trạng thái p_i nào đó thay Z bằng γ_i và dịch chuyển đầu đọc đi một ký hiệu. Ta quy ước rằng ký hiệu bên trái nhất của γ_i sẽ là ký hiệu được thay cao nhất trên Stack (nghĩa là nó nằm tại đỉnh Stack mới) và ký hiệu bên phải nhất của γ_i là ký hiệu được thay thấp nhất trong Stack. Chú ý rằng không được phép chọn p_i và γ_j với $i \neq j$ tại một bước chuyển nào đó.

Hàm chuyển không phụ thuộc ký hiệu nhập :

$$\delta(q, \epsilon, Z) = \{(p_1, \gamma_1), (p_2, \gamma_2), \dots, (p_m, \gamma_m)\}$$

trong đó q là trạng thái mà PDA đang giữ, độc lập với ký hiệu nhập, PDA đi vào trạng thái p_i thay Z bởi γ_i với $1 \leq i \leq m$. Trong trường hợp này đầu đọc không dịch chuyển.

Thí dụ 6.1 : Mô tả dưới đây cho PDA chấp nhận ngôn ngữ $\{wcw^R \mid w \in (0+1)^*\}$ bằng Stack rỗng.

$$M((\{q_1, q_2\}, \{0, 1, c\}, \{R, B, Y\}, \delta, q_1, R, \emptyset))$$

- 1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
- 2) $\delta(q_1, 1, R) = \{(q_1, YR)\}$
- 3) $\delta(q_1, 0, B) = \{(q_1, BB)\}$
- 4) $\delta(q_1, 1, B) = \{(q_1, YB)\}$
- 5) $\delta(q_1, 0, Y) = \{(q_1, BY)\}$
- 6) $\delta(q_1, 1, Y) = \{(q_1, YY)\}$
- 7) $\delta(q_1, c, R) = \{(q_2, R)\}$
- 8) $\delta(q_1, c, B) = \{(q_2, B)\}$
- 9) $\delta(q_1, c, Y) = \{(q_2, Y)\}$
- 10) $\delta(q_2, 0, B) = \{(q_2, \epsilon)\}$
- 11) $\delta(q_2, 1, Y) = \{(q_2, \epsilon)\}$
- 12) $\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$

Hình 6.3 - Mô tả PDA chấp nhận wcw^R bằng Stack rỗng

Chú ý rằng mỗi phép chuyển PDA sẽ viết lên đỉnh Stack một chuỗi γ có độ dài 2. Chẳng hạn $\delta(q_1, 0, R) = \{(q_1, BR)\}$. Nếu γ có độ dài bằng 1 thì PDA đơn giản là thay

ký hiệu tại đỉnh Stack và không làm thay đổi độ dài Stack. Nếu γ bằng ε thì PDA loại bỏ (Pop) phần tử tại đỉnh Stack. Chẳng hạn $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$ nghĩa là PDA ở trạng thái q_2 với R ở đỉnh Stack thì PDA xóa R độc lập với ký hiệu nhập, trong trường hợp này đầu đọc không dịch chuyển, điều này có nghĩa là PDA không yêu cầu còn một ký hiệu nào trên chuỗi nhập.

Hình thái (ID : Instantaneous Descriptions)

Để hình thức hóa cấu hình của một PDA với một PDA cụ thể, ta định nghĩa một hình thái (ID). ID phải ghi nhớ trạng thái và nội dung của Stack. ID là một bộ ba (q, w, γ) , trong đó q là trạng thái, w là chuỗi nhập và γ là chuỗi các ký hiệu Stack.

Nếu $M(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ là một PDA, ta nói :

$$(q, aw, Z\alpha) \vdash_M (p, w, \beta\alpha) \text{ nếu } \delta(q, a, Z) \text{ chứa } (p, \beta)$$

Lưu ý a có thể là một ký hiệu trong input hoặc ε . Chẳng hạn với PDA mô tả như trên, ta có $(q_1, BY) \in \delta(q_1, 0, Y)$, suy ra rằng $(q_1, 011, YYR) \vdash (q_1, 11, BYYR)$.

Ta dùng ký hiệu \vdash_M^* cho bao đóng phản xạ và bắc cầu của \vdash_M , tức là : $I \vdash_M^* I$ đối với mỗi ID I , và $I \vdash_M J$ và $J \vdash_M^* K$ thì $I \vdash_M^* K$. Ta viết $I \vdash_M^i K$ nếu ID I trở thành K sau chính xác i bước chuyển. Chữ chỉ số dưới M trong các ký hiệu \vdash_M, \vdash_M^i và \vdash_M^* có thể được bỏ qua khi M đã được xác định.

Ngôn ngữ chấp nhận bởi PDA

Với PDA $M(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, ta định nghĩa :

Ngôn ngữ được chấp nhận bởi trạng thái kết thúc là:

$$L(M) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \gamma) \text{ với } p \in F \text{ và } \gamma \in \Gamma^*\}$$

Ngôn ngữ được chấp nhận bởi Stack rỗng là :

$$N(M) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon) \text{ với } p \in Q\}.$$

Khi có sự chấp nhận bằng Stack rỗng thì tập trạng thái kết thúc là không còn cần thiết vì vậy ta ký hiệu tập trạng thái kết thúc F là \emptyset .

Thí dụ 6.2 : Các phép chuyển hình thái của PDA chấp nhận chuỗi 001c100 thuộc ngôn ngữ $\{wcw^R \mid w \in (0+1)^*\}$ bằng Stack rỗng như sau :

$$(q_1, 001c100, R) \vdash (q_1, 01c110, YR) \vdash (q_1, 1c110, YYR) \vdash (q_1, c100, BYYR) \vdash \\ (q_2, 100, BYYR) \vdash (q_2, 00, YYR) \vdash (q_2, 0, YR) \vdash (q_2, \varepsilon, R) \vdash (q_2, \varepsilon, \varepsilon) : \text{Chấp nhận}$$

Nhận xét : Trong ví dụ thiết kế PDA chấp nhận ngôn ngữ $\{wcw^R \mid w \in (0+1)^*\}$ bằng Stack rỗng như trên, ta thấy các giá trị hàm chuyển của nó luôn là đơn trị. Tại mỗi thời điểm từ một trạng thái trong bộ điều khiển, có thể đọc vào hoặc không đọc một ký hiệu trên băng nhập, với một ký hiệu tại đỉnh Stack, chỉ có một giá trị xác định

bước chuyển kế tiếp. Vì thế, ta gọi dạng PDA này là ôtômat đẩy xuống đơn định - DPDA.

PDA không đơn định (NPDA)

Thí dụ 6.3 : Thiết kế PDA chấp nhận ngôn ngữ $\{ww^R \mid w \in (0+1)^*\}$ bằng Stack rỗng.

Câu hỏi :



Hãy nêu vai trò của ký hiệu c trong ngôn ngữ được cho bởi thí dụ 6.1 và cho nhận xét về sự khác biệt dạng chuỗi thuộc ngôn ngữ trong thí dụ 6.3 với thí dụ 6.1 đã nêu ở trên ?

Rõ ràng ta thấy khi không có sự hiện diện của ký hiệu c ở giữa chuỗi nhập để xác định thời điểm bộ điều khiển có thể chuyển từ trạng thái q_1 sang trạng thái q_2 , thì vấn đề sẽ trở nên phức tạp hơn khi cần phải quyết định đâu là ký hiệu bắt đầu cho chuỗi ngược (w^R) ? Ở mỗi thời điểm mà bộ điều khiển đọc thấy hai ký hiệu liên tiếp giống nhau trong chuỗi nhập thì bắt buộc nó phải đoán thử cả hai khả năng cho ký hiệu thứ hai: hoặc vẫn giữ trạng thái q_1 và Push vào Stack nếu xem ký hiệu này vẫn thuộc chuỗi xuôi (w), hoặc chuyển sang trạng thái q_2 và Pop khỏi Stack nếu xem nó là ký hiệu bắt đầu cho chuỗi ngược (w^R).

Mô tả PDA không đơn định chấp nhận ngôn ngữ $\{ww^R \mid w \in (0+1)^*\}$ bằng Stack rỗng

$M (\{q_1, q_2\}, \{0, 1\}, \{R, B, Y\}, \delta, q_1, R, \emptyset)$

- 1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$
- 2) $\delta(q_1, 1, R) = \{(q_1, YR)\}$
- 3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \epsilon)\}$
- 4) $\delta(q_1, 0, Y) = \{(q_1, BY)\}$
- 5) $\delta(q_1, 1, B) = \{(q_1, YB)\}$
- 6) $\delta(q_1, 1, Y) = \{(q_1, YY), (q_2, \epsilon)\}$
- 7) $\delta(q_2, 0, B) = \{(q_2, \epsilon)\}$
- 8) $\delta(q_2, 1, Y) = \{(q_2, \epsilon)\}$
- 9) $\delta(q_1, \epsilon, R) = \{(q_2, \epsilon)\}$
- 10) $\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$

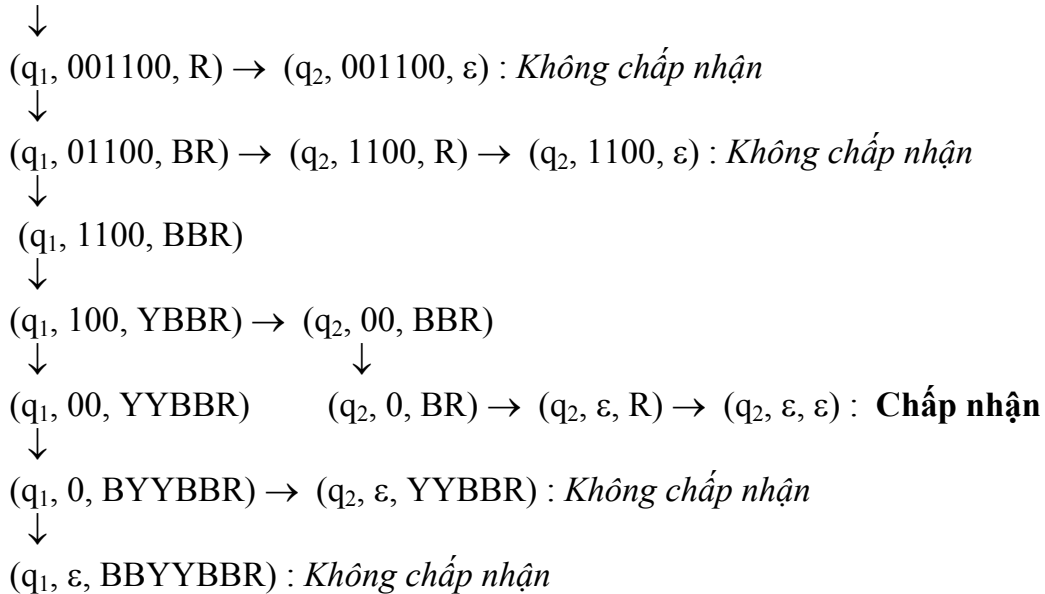
Hình 6.4 - Mô tả PDA không đơn định chấp nhận ww^R bằng Stack rỗng

Quy tắc (1) đến (3) cho phép M lưu trữ input trên Stack, quy tắc (3) và (6) cho phép M lựa chọn một trong hai phép chuyển. M có thể quyết định (đoán) đã đi đến giữa chuỗi nó chuyển sang phép chuyển thứ 2: M chuyển sang q_2 và thử sự thích hợp của phần chuỗi còn lại với các ký hiệu đang ở trên Stack. Nếu M đoán đúng và nếu chuỗi nhập có dạng ww^R thì M sẽ làm rỗng Stack của nó và chấp nhận chuỗi nhập.

Cũng như NFA một PDA không đơn định (NPDA) M chấp nhận một input nếu có một chuỗi các lựa chọn mà M làm rỗng Stack của nó. Nghĩa là M luôn luôn "đoán đúng", đoán sai không phải là nguyên nhân để loại bỏ input. Một input bị loại bỏ **nếu và chỉ nếu không có sự lựa chọn nào để làm rỗng Stack** (hay là không thể "đoán đúng" vì không tồn tại cách đúng).

Thí dụ 6.4 : Các phép chuyển hình thái của PDA chấp nhận chuỗi **001100** thuộc ngôn ngữ $\{ww^R | w \in (0+1)^*\}$ bằng Stack rỗng như sau :

Khởi đầu



Hình 6.5 - Hình thái của PDA với input 001100

PDA đơn định (DPDA)

Một PDA $M (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ được gọi là đơn định nếu:

- 1) $\forall q \in Q$ và $Z \in \Gamma$: nếu $\delta(q, \varepsilon, Z) \neq \emptyset$ thì $\delta(q, a, Z) = \emptyset, \forall a \in \Sigma$
- 2) Không có $q \in Q, Z \in \Gamma$ và $a \in (\Sigma \cup \{\varepsilon\})$ mà $\delta(q, a, Z)$ chứa nhiều hơn một phần tử.

Điều kiện 1 không cho phép khả năng chọn lựa giữa phép chuyển không xác định ký hiệu nhập (ϵ - dịch chuyển) và phép chuyển trên một ký hiệu input. Điều kiện 2 không cho phép chọn lựa một vài phép chuyển nào đó (q, a, Z) hay (q, ϵ, Z). Không như ô tô máy hữu hạn FA, một PDA thì thông thường được xét là không đơn định trừ khi ta có ghi chú cụ thể.

Đối với ô tô máy hữu hạn, dạng đơn định và không đơn định là tương đương nhau về phương diện chấp nhận ngôn ngữ. Tuy nhiên, điều này không đúng với ô tô máy đẩy xuống, PDA không đơn định và PDA đơn định là không tương đương nhau. Thực tế ngôn ngữ ww^R được chấp nhận bởi một PDA không đơn định nhưng không được chấp nhận bởi bất kỳ một PDA đơn định nào.

II. PDA VÀ VẤN PHẠM PHI NGỮ CẢNH

2.1. Tương đương của việc chấp nhận chuỗi bởi trạng thái kết thúc và bởi Stack rỗng

ĐỊNH LÝ 6.1: Nếu L là $L(M_2)$ với PDA M_2 thì L là $N(M_1)$ với PDA M_1 nào đó.

Chứng minh

Ta sẽ xây dựng M_1 tương tự như M_2 nhưng M_1 sẽ xóa rỗng Stack của nó khi M_2 đi vào trạng thái kết thúc. Ta dùng một trạng thái q_e của M_1 để xóa Stack của nó và dùng ký hiệu đánh dấu đáy Stack M_1 bằng ký hiệu X_0 , vì vậy M_1 không thể làm rỗng Stack của nó khi M_2 chưa đi vào trạng thái kết thúc.

Đặt $M_2(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ là PDA sao cho $L = L(M_2)$.

Đặt $M_1(Q \cup \{q_e, q_0'\}, \Sigma, \Gamma, \delta', q_0', X_0, \emptyset)$ trong đó δ' định nghĩa như sau:

- 1) $\delta'(q_0', \epsilon, X_0) = \{(q_0, Z_0X_0)\}$
- 2) $\delta'(q, a, Z)$ chứa mọi phần tử của $\delta(q, a, Z)$, $\forall q \in Q, a \in \Sigma$ hoặc $a = \epsilon$ và $Z \in \Gamma$
- 3) $\forall q \in F$ và $Z \in \Gamma \cup \{X_0\}$, $\delta'(q, \epsilon, Z)$ chứa (q_e, ϵ)
- 4) $\forall Z \in \Gamma \cup \{X_0\}$, $\delta'(q_0', \epsilon, Z)$ chứa (q_e, ϵ)

Quy tắc 1 làm cho PDA M_1 đi vào trạng thái khởi đầu của M_2 trừ việc thêm X_0 vào đáy Stack. Quy tắc 2 cho phép M_1 chuyển tương tự như M_2 . Quy tắc 3 và 4 cho phép M_1 chọn việc đi vào trạng thái q_e và xóa Stack hay là tiếp tục mô phỏng M_2 . Chú ý rằng M_2 có thể xóa rỗng Stack của nó khi chưa tới trạng thái kết thúc vì vậy M_1 phải được đánh dấu đáy Stack bằng X_0 . Vì nếu không làm như vậy thì khi M_1 chuyển tương tự như M_2 , M_1 sẽ xóa rỗng Stack và chấp nhận input trong khi M_2 chưa đi vào trạng thái kết thúc nghĩa là input chưa được chấp nhận.

Đặt $x \in L(M_2)$ thì $(q_0, x, Z_0) \vdash_{M_2}^* (q, \epsilon, \gamma)$ với $q \in F$. Ta xét M_1 với input x .

Theo quy tắc 1 : $(q_0', x, X_0) \vdash_{M_1}^* (q_0, x, Z_0X_0)$

Theo quy tắc 2 mỗi phép chuyển của M_2 là một phép chuyển trong M_1 , vậy:

$$(q_0, x, Z_0) \vdash_{M_1}^* (q, \epsilon, \gamma)$$

Nếu một PDA có thể thực hiện một chuỗi các phép chuyển từ một ID đã cho thì nó có thể làm một chuỗi các phép chuyển đó từ một ID bất kỳ thu được từ ID đầu tiên bằng cách thêm các chuỗi ký hiệu Stack vào dưới chuỗi Stack ban đầu (vì các ký hiệu ở phía dưới của Stack không làm ảnh hưởng gì).

Vậy $(q_0', x, X_0) \vdash_{M_1} (q_0, x, Z_0X_0) \vdash_{M_1}^* (q, \epsilon, \gamma X_0)$.

Theo quy tắc 3 và 4 : $(q, \epsilon, \gamma X_0) \vdash_{M_1}^* (q_e, \epsilon, \epsilon)$.

Vì vậy $(q_0', x, X_0) \vdash_{M_1}^* (q_e, \varepsilon, \varepsilon)$ và M_1 chấp nhận chuỗi x bằng Stack rỗng.

Ngược lại, nếu M_1 chấp nhận x bằng Stack rỗng thì dễ dàng chỉ ra rằng chuỗi các phép chuyển phải bắt đầu bằng một phép chuyển theo quy tắc 1, sau đó bằng một chuỗi phép chuyển theo quy tắc 2, trong khi thực hiện các phép chuyển này M_1 chuyển tương tự như M_2 , sau đó xóa Stack của M_1 bằng quy tắc chuyển 3 và 4.

Vậy $x \in L(M_2)$.

ĐỊNH LÝ 6.2 : Nếu L là $N(M_1)$ với PDA M_1 nào đó thì L là $L(M_2)$ với một PDA M_2 nào đó.

Chứng minh

Ta sẽ xây dựng M_2 tương tự M_1 và M_2 đi vào trạng thái kết thúc khi và chỉ khi M_1 làm rỗng Stack của nó.

Đặt $M_1 (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ là PDA sao cho $L = N(M_1)$.

Đặt $M_2 (Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_f\})$ trong đó δ' được định nghĩa như sau:

- 1) $\delta'(q_0', \varepsilon, X_0) = \{(q_0, Z_0 X_0)\}$
- 2) $\forall q \in Q, a \in \Sigma \cup \{\varepsilon\}, \text{ và } Z \in \Gamma : \delta'(q, a, Z) = \delta(q, a, Z)$
- 3) $\forall q \in Q, \delta'(q, \varepsilon, X_0)$ chứa (q_f, ε)

Quy tắc 1 cho phép M_2 đi vào hình thái khởi đầu ID của M_1 , trừ việc M_2 sẽ có chứa ở dưới đáy Stack của nó ký hiệu X_0 , ký hiệu này sẽ nằm bên dưới tất cả các ký hiệu Stack của M_1 . Quy tắc 2 cho phép M_2 chuyển tương tự như M_1 . Khi M_1 làm rỗng Stack của nó, thì M_2 khi chuyển tương tự như M_1 sẽ xóa toàn bộ Stack của nó trừ ký hiệu X_0 nằm dưới đáy Stack. Quy tắc 3 làm cho M_2 sau đó khi gặp X_0 xuất hiện thì đi vào trạng thái kết thúc và chấp nhận input x .

Chứng minh $L(M_2) = N(M_1)$ cũng tương tự như định lý 6.1

2.2. Tương đương giữa PDA và CFL

ĐỊNH LÝ 6.3: Nếu L là ngôn ngữ phi ngữ cảnh thì tồn tại PDA M sao cho $L = N(M)$.

Chứng minh

Giả sử ε không thuộc $L(G)$ (có thể sửa đổi lý luận cho trường hợp ngôn ngữ $L(G)$ có chứa ε). Đặt $G (V, T, P, S)$ là văn phạm phi ngữ cảnh có dạng chuẩn Greibach sinh ra L . Đặt $M (\{q\}, T, V, \delta, q, S, \emptyset)$, trong đó $\delta(q, a, A)$ chứa (q, γ) khi và chỉ khi $A \rightarrow a\gamma$ là một luật sinh trong P .

PDA M mô phỏng chuỗi dẫn xuất trái của G . Vì G là dạng chuẩn Greibach nên mỗi dạng câu trong dẫn xuất trái gồm một chuỗi các ký hiệu kết thúc x sau đó là một chuỗi các biến α . M lưu trữ phần hậu tố α của dạng câu bên trái trên Stack của nó sau khi xử lý phần tiền tố x .

Một cách hình thức ta chỉ ra rằng :

$$S \Rightarrow^* x\alpha \text{ bằng dẫn xuất trái khi và chỉ khi } (q, x, S) \vdash_M^* (q, \varepsilon, \alpha) \quad (1)$$

Trước tiên, chúng ta giả sử $(q, x, S) \vdash^i (q, \varepsilon, \alpha)$ và sẽ chỉ ra bằng quy nạp theo số lần i rằng $S \Rightarrow^* x\alpha$.

Với $i = 0$, điều đó hiển nhiên đúng vì $x = \varepsilon$ và $\alpha = S$.

Giả sử $i \geq 1$ và đặt $x = ya$.

Xét bước chuyển hình thái trước bước cuối :

$$(q, ya, S) \vdash^{i-1} (q, a, \beta) \vdash (q, \varepsilon, \alpha) \quad (2)$$

Nếu loại bỏ ký hiệu a ở cuối chuỗi input trong hình thái đầu tiên của (2), ta có:

$(q, y, S) \vdash^{i-1} (q, \varepsilon, \beta)$ (vì a không ảnh hưởng đến các phép chuyển của M).

Theo giả thiết quy nạp $S \Rightarrow^* y\beta$. Phép chuyển $(q, a, \beta) \vdash (q, \varepsilon, \alpha)$ sẽ suy ra $\beta = A\gamma$, với $A \in V$ và $A \rightarrow a\eta$ là một luật sinh trong G và $\alpha = \eta\gamma$.

Vậy $S \Rightarrow^* y\beta \Rightarrow ya\eta\gamma = x\alpha$

Ta đã chứng minh xong "nếu" của giả thiết (1)

Ngược lại, ta giả sử $S \Rightarrow^i x\alpha$ bằng dẫn xuất trái. Ta sẽ chứng minh quy nạp theo số bước dẫn xuất i rằng: $(q, x, S) \vdash^* (q, \varepsilon, \alpha)$

Với $i = 0$: phép chuyển hiển nhiên đúng

Xét $i \geq 1$ và giả sử $S \Rightarrow^{i-1} yA\gamma \Rightarrow ya\eta\gamma$, trong đó $x = ya$ và $\alpha = \eta\gamma$. Theo giả thiết quy nạp : $(q, y, S) \vdash^* (q, \varepsilon, A\gamma)$. Vậy $(q, ya, S) \vdash^* (q, a, A\gamma)$

Vì $A \rightarrow a\eta$ là một luật sinh nên $\delta(q, a, A)$ chứa (q, η) . Vậy :

$$(q, x, S) \vdash^* (q, a, A\gamma) \vdash^* (q, \varepsilon, \alpha)$$

Hay phần "chỉ nếu" của giả thiết (1) cũng đã được chứng minh xong.

Để kết thúc việc chứng minh, ta chú ý rằng giả thiết (1) với $\alpha = \varepsilon$ thì $S \Rightarrow^* x$ nếu và chỉ nếu $(q, x, S) \vdash^* (q, \varepsilon, \varepsilon)$. Tức là $x \in L(G)$ khi và chỉ khi $x \in N(M)$.

Thí dụ 6.3 : Xây dựng NPDA chấp nhận ngôn ngữ sinh bởi CFG G có các luật sinh như sau :

$$S \rightarrow aAA$$

$$A \rightarrow aS \mid bS \mid a$$

Ta có : CFG $G (\{S, A\}, \{a, b\}, P, S)$

NPDA tương đương $M (\{q\}, \{a, b\}, \{S, A\}, \delta, q, S, \emptyset)$ với δ như sau :

$$1) \quad \delta(q, a, S) = \{(q, AA)\}$$

$$2) \quad \delta(q, a, A) = \{(q, S), (q, \varepsilon)\}$$

$$3) \quad \delta(q, b, A) = \{(q, \varepsilon)\}$$

ĐỊNH LÝ 6.4 : Nếu L là $N(M)$ với PDA M thì L và ngôn ngữ phi ngữ cảnh.

Chứng minh

Gọi PDA $M (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$. Đặt $G (V, \Sigma, P, S)$ là CFG, trong đó :

- . V là tập các đối tượng dạng $[q, A, p]$ với $p, q \in Q; A \in \Gamma$
- . S là ký hiệu chưa kết thúc mới thêm vào.
- . P là tập các luật sinh có dạng :
 - 1) $S \rightarrow [q_0, Z_0, q], \forall q \in Q.$
 - 2) $[q, A, q_{m+1}] \rightarrow a[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]$
 $\forall q, q_1, q_2, \dots, q_{m+1} \in Q, a \in \Sigma \cup \{\varepsilon\}$ và $A, B_1, B_2, \dots, B_m \in \Gamma$
 sao cho $\delta(q, a, A)$ có chứa $(q_1, B_1 B_2 \dots B_m).$

Nếu $m = 0$ thì luật sinh có dạng $[q, A, q_1] \rightarrow a.$

Để nắm được chứng minh, cần phải lưu ý rằng các biến và luật sinh trong G được xác định sao cho dẫn xuất trái trong G của x mô phỏng PDA khi cho x nhập vào. Cụ thể hơn, các biến xuất hiện tại một bước bất kỳ trong G tương đương với các ký hiệu trên Stack của M. Nói cách khác $[q, A, p]$ dẫn ra x nếu và chỉ nếu x là nguyên nhân làm M xoá rỗng Stack của nó bằng chuỗi các phép chuyển từ trạng thái q đến trạng thái p.

Để chứng minh $L(G) = N(M)$, ta quy nạp theo số bước dẫn xuất của G hoặc số bước chuyển trạng thái của M rằng $[q, A, p] \Rightarrow_G^* x$ nếu và chỉ nếu $(q, x, A) \vdash_M^* (p, \varepsilon, \varepsilon)$

Thí dụ 6.4 : Xây dựng CFG G tương đương sinh ra ngôn ngữ được chấp nhận bởi PDA sau :

$$M (\{q_0, q_1\}, \{0, 1\}, \{Z_0, X\}, \delta, q_0, Z_0, \emptyset)$$

với δ được cho như sau :

- 1) $\delta (q_0, 0, Z_0) = \{(q_0, XZ_0)\}$
- 2) $\delta (q_0, 0, X) = \{(q_0, XX)\}$
- 3) $\delta (q_0, 1, X) = \{(q_1, \varepsilon)\}$
- 4) $\delta (q_1, 1, X) = \{(q_1, \varepsilon)\}$
- 5) $\delta (q_1, \varepsilon, X) = \{(q_1, \varepsilon)\}$
- 6) $\delta (q_1, \varepsilon, Z_0) = \{(q_1, \varepsilon)\}$

Ta xây dựng CFG G $(V, \{0, 1\}, P, S)$ sinh ra $N(M)$ với các thành phần như sau :

$$V = \{ S, [q_0, X, q_0], [q_0, X, q_1], [q_1, X, q_0], [q_1, X, q_1], \\ [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1] \}$$

Tập luật sinh P chứa các luật sinh có dạng :

$$\text{Các luật sinh cho ký hiệu bắt đầu } S : S \rightarrow [q_0, Z_0, q_0] \mid [q_0, Z_0, q_1]$$

Các luật sinh cho các biến khác trong V được xây dựng từ các hàm chuyển của PDA như sau :

- $\delta_1) \quad [q_0, Z_0, q_0] \rightarrow 0 [q_0, X, q_0][q_0, Z_0, q_0]$
 $\quad \quad \quad \mid 0 [q_0, X, q_1][q_1, Z_0, q_0]$
 $\quad [q_0, Z_0, q_1] \rightarrow 0 [q_0, X, q_0][q_0, Z_0, q_1]$
 $\quad \quad \quad \mid 0 [q_0, X, q_1][q_1, Z_0, q_1]$
- $\delta_2) \quad [q_0, X, q_0] \rightarrow 0 [q_0, X, q_0][q_0, X, q_0]$
 $\quad \quad \quad \mid 0 [q_0, X, q_1][q_1, X, q_0]$
 $\quad [q_0, X, q_1] \rightarrow 0 [q_0, X, q_0][q_0, X, q_1]$
 $\quad \quad \quad \mid 0 [q_0, X, q_1][q_1, X, q_1]$
- $\delta_3) \quad [q_0, X, q_1] \rightarrow 1$
- $\delta_4) \quad [q_1, Z_0, q_1] \rightarrow \varepsilon$

$$\delta_5) [q_1, X, q_1] \rightarrow \varepsilon$$

$$\delta_6) [q_1, X, q_1] \rightarrow 1$$

Nhận xét rằng không có luật sinh nào cho các biến $[q_1, X, q_0]$ và $[q_1, Z_0, q_0]$. Vì tất cả các luật sinh cho biến $[q_0, X, q_0]$ và $[q_0, Z_0, q_0]$ đều có chứa $[q_1, X, q_0]$ hoặc $[q_0, Z_0, q_0]$ ở vế phải, nên sẽ không thể có chuỗi ký hiệu kết thúc nào có thể được dẫn ra từ các biến $[q_0, X, q_0]$ hoặc $[q_0, Z_0, q_0]$. Loại bỏ 4 biến này ra khỏi tập biến V và xóa các luật sinh có liên quan đến chúng trong tập P , ta thu được văn phạm có dạng như sau:

$$\begin{aligned} S &\rightarrow [q_0, Z_0, q_1] \\ [q_0, Z_0, q_1] &\rightarrow 0 [q_0, X, q_1][q_1, Z_0, q_1] \\ [q_0, X, q_1] &\rightarrow 0 [q_0, X, q_1][q_1, X, q_1] \\ [q_0, X, q_1] &\rightarrow 1 \\ [q_1, Z_0, q_1] &\rightarrow \varepsilon \\ [q_1, X, q_1] &\rightarrow \varepsilon \\ [q_1, X, q_1] &\rightarrow 1 \end{aligned}$$

Câu hỏi :



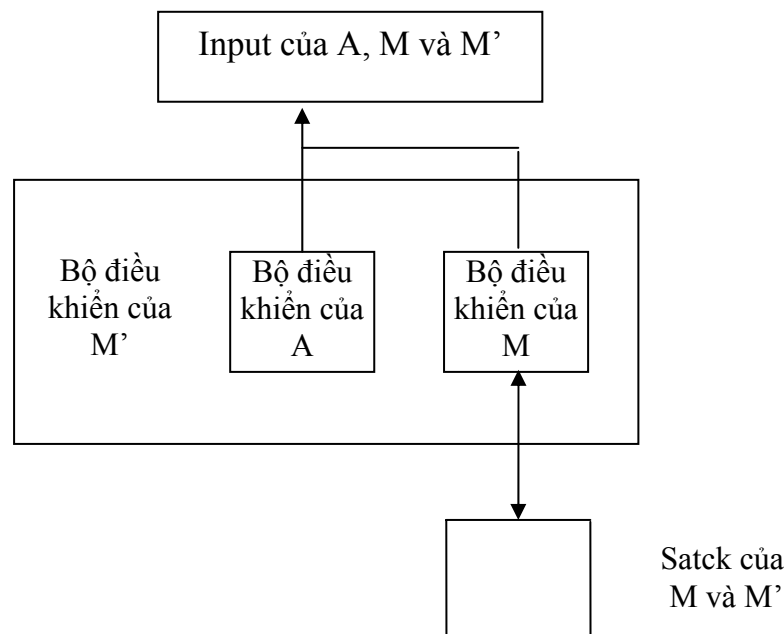
Sinh viên hãy dùng các kiến thức đã học trong chương trước (ĐỊNH LÝ 5.2) để viết một văn phạm tương đương với văn phạm trên không có chứa các ký hiệu vô ích ?

2.3. Quan hệ giữa CFL và tập hợp chính quy

ĐỊNH LÝ 6.5 : Nếu L là CFL và R là tập chính quy thì $L \cap R$ là CFL.

Chứng minh

Đặt L là $L(M)$ với PDA $M (Q_M, \Sigma, \Gamma, \delta_M, q_0, Z_0, F_M)$ và đặt R là $L(A)$ với DFA $A (Q_A, \Sigma, \delta_A, p_0, F_A)$. Ta xây dựng PDA M' cho $L \cap R$ bằng cách cho M và A cùng “chạy song song”. Tức là với một ký hiệu nhập a thì M và A thực hiện các phép chuyển độc lập nhau. M' chấp nhận input nếu cả M và A cùng chấp nhận.



Hình 6.6 - Chạy một FA và PDA song song

Một cách hình thức, đặt $M' (Q_A \times Q_M, \Sigma, \Gamma, \delta, [p_0, q_0], Z_0, F_A \times F_M)$, trong đó hàm chuyển δ được xác định như sau :

$$\delta([p, q], a, X) \text{ chứa } ([p', q'], \gamma) \Leftrightarrow \delta_A(p, a) = p' \text{ và } \delta_M(q, a, X) \text{ chứa } (q', \gamma).$$

Chú ý rằng a có thể bằng ε , khi đó $p' = p$.

Dễ dàng chứng minh quy nạp theo i rằng : $([p_0, q_0], w, Z_0) \vdash_M^i . ([p, q], \varepsilon, \gamma) \Leftrightarrow (q_0, w, Z_0) \vdash_M^i (q, \varepsilon, \gamma)$ và $\delta(p_0, w) = p$ (1)

Với $i = 0$: thì (1) hiển nhiên đúng vì $p = p_0, q = q_0, \gamma = Z_0$ và $w = \varepsilon$.

Giả sử (1) đúng tới $i - 1$ ($i > 0$).

Xét $([p_0, q_0], xa, Z_0) \vdash_M^{i-1} . ([p', q'], a, \beta) \vdash_M . ([p, q], \varepsilon, \gamma)$, trong đó $w = xa$ và a là ε hoặc là một ký hiệu $\in \Sigma$.

Theo giả thiết quy nạp, $\delta_A(p_0, x) = p'$ và $(q_0, x, Z_0) \vdash_M^{i-1} (q', \varepsilon, \beta)$.

Theo định nghĩa của δ , thực tế $([p', q'], a, \beta) \vdash_M . ([p, q], \varepsilon, \gamma)$ nên có thể suy ra $\delta_A(p', a) = p$ và $(q', a, \beta) \vdash_M (q, \varepsilon, \gamma)$. Vậy $\delta_A(p_0, w) = p$ và $(q_0, w, Z_0) \vdash_M^ (q, \varepsilon, \gamma)$.*

Tương tự, ta có thể chứng minh rằng : Nếu $(q_0, w, Z_0) \vdash_M^i (q, \varepsilon, \gamma)$ và $\delta_A(p_0, w) = p$ thì $([p_0, q_0], w, Z_0) \vdash_M^ . ([p, q], \varepsilon, \gamma)$ (xem phần này như bài tập).*

Tổng kết chương VI: Đến chương này, chúng ta đã có thể nắm bắt được một vài ý tưởng cơ bản liên quan đến các khái niệm về ngôn ngữ chính quy, ngôn ngữ phi ngữ cảnh, và mối quan hệ của chúng với các dạng ôtômát hữu hạn và đẩy xuống. Những khảo sát chứng tỏ ngôn ngữ chính quy thực sự là một tập hợp con của ngôn ngữ phi ngữ cảnh, và do đó, ôtômát đẩy xuống xét về một mặt nào đó có khả năng nhận dạng ngôn ngữ mạnh hơn rất nhiều so với ôtômát hữu hạn. Điều này gợi cho chúng ta một ý tưởng có thể mở rộng hơn nữa về khả năng đoán nhận ngôn ngữ của cơ chế ôtômát. Nếu so sánh ôtômát hữu hạn và ôtômát đẩy xuống, ta thấy rằng bản chất của sự khác

biệt thể hiện ở bộ lưu trữ tạm thời dùng Stack. Nếu không có bộ lưu trữ, chúng ta có dạng ôtômát hữu hạn, nếu bộ lưu trữ là Stack, ta có dạng ôtômát đẩy xuống mạnh hơn. Từ suy luận này, chúng ta hoàn toàn có thể mong đợi để định nghĩa ngay cả những họ ngôn ngữ rộng lớn hơn nếu có thể cung cấp cho cơ chế ôtômát một bộ nhớ với khả năng lưu trữ linh hoạt hơn. Điều này dẫn đến khái niệm cơ bản về máy Turing sẽ được giới thiệu trong chương sau, một cơ chế ôtômát có tính máy móc hay tính giải thuật.

BÀI TẬP CHƯƠNG VI

6.1. Xây dựng PDA chấp nhận các ngôn ngữ :

- a) $\{0^m 1^m 2^n \mid m, n \geq 1\}$
- b) $\{a^k b^l c^n d^m \mid m = k + l + n\}$
- c) $\{w \mid w \in \{a, b\}^* \text{ và } \#a(w) = \#b(w)\}$
- d) $\{w \mid w \in \{a, b\}^* \text{ và } \#a(w) = 2\#b(w)\}$

6.2. Xây dựng PDA tương đương với văn phạm :

- a) $S \rightarrow + SS \mid *SS \mid a$
- b) $\begin{cases} S \rightarrow aS \mid bS \mid aA \\ A \rightarrow bB \mid b \\ B \rightarrow aC \\ C \rightarrow b \end{cases}$

6.3. Xây dựng văn phạm CFG tương đương với các PDA sau :

- a) $M(\{q_0, q_1\}, \{0, 1\}, \{Z_0, X\}, \delta, q_0, Z_0, \emptyset)$, trong đó δ được cho như sau:

$$\begin{aligned}\delta(q_0, 1, Z_0) &= \{(q_0, XZ_0)\} \\ \delta(q_0, 0, X) &= \{(q_0, XX)\} \\ \delta(q_0, 1, X) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, 1, X) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, \varepsilon, X) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, \varepsilon, Z_0) &= \{(q_1, \varepsilon)\}\end{aligned}$$

- b) $M(\{q_0, q_1\}, \{0, 1\}, \{Z_0, X\}, \delta, q_0, Z_0, \emptyset)$, trong đó δ được cho như sau:

$$\begin{aligned}\delta(q_0, 1, Z_0) &= \{(q_0, XZ_0)\} \\ \delta(q_0, 1, X) &= \{(q_0, XX)\} \\ \delta(q_0, 0, X) &= \{(q_1, X)\} \\ \delta(q_0, \varepsilon, Z_0) &= \{(q_0, \varepsilon)\} \\ \delta(q_1, 1, X) &= \{(q_1, \varepsilon)\} \\ \delta(q_1, 0, Z_0) &= \{(q_0, Z_0)\}\end{aligned}$$

- c) $M(\{q_0, q_1\}, \{a, b, c\}, \{Z_0, X\}, \delta, q_0, Z_0, \emptyset)$, trong đó δ được cho như sau:

$$\begin{aligned}\delta(q_0, a, Z_0) &= \{(q_0, X)\} \\ \delta(q_0, a, X) &= \{(q_0, XX)\} \\ \delta(q_0, c, X) &= \{(q_1, X)\}\end{aligned}$$

$$\begin{aligned}\delta(q_0, b, Z_0) &= \{(q_0, X)\} \\ \delta(q_0, b, X) &= \{(q_0, XX)\} \\ \delta(q_1, c, X) &= \{(q_1, \varepsilon)\}\end{aligned}$$