Chương VIII

ÔTÔMÁT TUYẾN TÍNH GIỚI NỘI VÀ VĂN PHẠM CẢM NGỮ CẢNH

Nội dung chính : Trong chương này, chúng ta xét thêm một loại ôtômát, không mạnh bằng máy Turing, được gọi là ôtômát tuyến tính giới nội (Linear Bounded Automata – LBA). Đồng thời cũng xét thêm lớp văn phạm tương ứng với nó, là lớp văn phạm L_1 hay còn gọi là văn phạm cảm ngữ cảnh, lớp văn phạm nằm giữa lớp văn phạm L_0 và văn phạm phi ngữ cảnh L_2. Từ đó ta hoàn thành sự phân cấp các ngôn ngữ thành 4 cấp, gọi là sự phân cấp Chomsky.

Mục tiêu cần đạt: Cuối chương, sinh viên cần phải nắm vững:

- Khái niệm LBA, định nghĩa và các thành phần.
- Sự tương đương giữa LBA và văn phạm cảm ngữ cảnh.
- Mối tương quan giữa các lớp ngôn ngữ.

Kiến thức cơ bản: Để tiếp thu tốt nội dung của chương này, sinh viên cần hiểu rõ các dạng ôtômát đã được giới thiệu trong các chương trước, đặc biệt là mô hình máy Turing; nắm vững cơ cấu các lớp văn phạm…

Tài liệu tham khảo :

[1] Nguyễn Văn Ba – *Giáo trình ngôn ngữ hình thức* – Trường Đại học Bách khoa Hà nội – 1994.

[2] A. C. Fleck - Context Sensitive Languages:
http://www.cs.uiowa.edu/~fleck/PartIIIxpar

[3] Linear Bounder Automata:
http://cs.engr.uky.edu/~lewis/texts/theory/automata/lb-auto.pdf

I. ÔTÔMÁT TUYẾN TÍNH GIỚI NỘI (LBA)

Ta gọi Ôtômát tuyến tính giới nội (Linear Bounded Automata - LBA) là một máy Turing không đơn định và không có khả năng nới rộng vùng làm việc ra khỏi mút trái và mút phải của chuỗi nhập. Nó phải thỏa hai điều kiện sau :

1) Bộ chữ cái nhập của nó có chứa thêm hai ký hiệu đặc biệt ⊄ và $ dùng làm ký hiệu đánh dấu mút trái và mút phải.
2) LBA không thực hiện phép chuyển sang trái (L) từ ⊄ và không thực hiện phép chuyển sang phải (R) từ $, và cũng không viết các ký hiệu khác lên ⊄ và $.

LBA đơn giản là một máy Turing nhưng thay vì sử dụng một băng không giới hạn cho việc tính toán, nó bị hạn chế chỉ trong phạm vi băng chứa chuỗi nhập x với hai ô chứa các ký hiệu đánh dấu cận đầu mút. Sự giới hạn này làm cho việc tính toán phải thông qua một số các hàm tuyến tính trên độ dài chuỗi, do đó ta gọi mô hình này là *ôtômát tuyến tính giới nội*. LBA không dùng các ô trống ở trên băng về phía trái và phía phải của chuỗi nhập, vì vậy ký hiệu khoảng trắng B (Blank) như đã dùng ở máy Turing là không cần dùng ở đây. Trái lại, để LBA nhận biết được giới hạn bên trái và giới hạn bên phải của chuỗi nhập, ta phải đưa thêm vào bộ chữ cái nhập Σ hai ký hiệu đặc biệt ⊄, $ để đánh dấu mút trái và mút phải của chuỗi. Vậy, tại thời điểm bắt đầu, chuỗi nhập đưa vào ở trên băng sẽ có dạng ⊄ **w** $, trong đó w ∈ $(\Sigma - \{⊄, \$\})^*$ là chuỗi cần đoán nhận. Trong quá trình làm việc, khi đầu đọc đọc tới ô có chứa ⊄ hay $, thì phép chuyển tiếp theo sau đó chỉ có thể là đổi trạng thái, chuyển đầu đọc trở lại phía trong phạm vi băng (tức chuyển sang phải khi gặp ⊄ và chuyển sang trái khi gặp $), và không được phép viết ký hiệu gì khác trên băng tại ô đang đọc khi gặp ⊄ và $.

Định nghĩa LBA

Một cách hình thức, LBA là một hệ thống $M(Q, \Sigma, \Gamma, \delta, q_o, ⊄, \$, F)$, trong đó các thành phần Q, Σ, Γ, q_o, F vẫn như đã định nghĩa ở máy Turing, còn ⊄, $ ∈ Σ và hàm chuyển :

$$\delta: Q \times \Gamma \rightarrow (Q \times \Gamma \times \{ L, R\})$$

phải thỏa mãn điều kiện:

- Nếu $(p, Y, E) \in \delta(q, ⊄)$ thì Y = ⊄ và E = R
- Nếu $(p, Y, E) \in \delta(q, \$)$ thì Y = $ và E = L

Ngôn ngữ được chấp nhận bởi LBA

Ta định nghĩa ngôn ngữ L(M) được đoán nhận bởi LBA M là tập hợp :

$$L(M) = \{ w \mid w \in (\Sigma - \{⊄, \$\})^* \text{ và } q_o⊄w\$ \vdash_M^* \alpha q \beta \text{ với } q \in F \text{ và } \alpha\beta \in \Gamma^* \}$$

Chú ý rằng các ký hiệu đánh dấu hai đầu mút ngay từ hình thái bắt đầu chúng đã có mặt trên băng nhập, nhưng chúng không được xem như thuộc một phần của chuỗi được chấp nhận hay không được chấp nhận bởi LBA. Vì đầu đọc của LBA không thể dịch chuyển ra ngoài phần chuỗi nhập nên chúng ta không cần định nghĩa các khoảng trống (ký hiệu Blank) phía bên phải của $.

II. VĂN PHẠM CẢM NGỮ CẢNH (CSG)

Ta gọi *văn phạm cảm ngữ cảnh* (**C**ontext **S**ensitive **G**rammar - CSG) là một hệ thống G (V, T, P, S), trong đó:

1) V là một tập hữu hạn các biến hay ký hiệu không kết thúc.
2) T là một tập hữu hạn các ký hiệu cuối, $V \cap T = \varnothing$
3) P là tập hữu hạn các luật sinh dạng $\alpha \to \beta$ trong đó $\alpha, \beta \in (V \cup T)^*$, chuỗi α phải có chứa biến và ràng buộc $|\alpha| \le |\beta|$
4) $S \in V$ là ký hiệu bắt đầu.

Ta định nghĩa ngôn ngữ do văn phạm cảm ngữ cảnh G sinh ra là

$$L(G) = \{ w \mid w \in \Sigma^* \text{ và } S \Rightarrow^* w\}$$

L(G) được gọi là *ngôn ngữ cảm ngữ cảnh* (**C**ontext **S**ensitive **L**anguage - CSL). Thuật ngữ "cảm ngữ cảnh" có xuất xứ từ một dạng chuẩn của văn phạm dạng này, trong đó mỗi luật sinh có dạng $\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$ với $\beta \neq \varepsilon$, cho thấy một biến A chỉ có thể được thay thế bởi một chuỗi β (khác rỗng) trong "ngữ cảnh" α_1 - α_2. Điều đó không giống như trong văn phạm phi ngữ cảnh, với các luật sinh có dạng $A \to \beta$ ($|\beta| \ge 0$), sự thay thế này không đòi hỏi ngữ cảnh.

Thí dụ 8.1 : Xét CSG G (V, T, P, S) với V ={ S, B, C}, ∑ ={a, b, c} và P gồm các luật sinh như sau :

1) $S \to aSBC$
2) $S \to aBC$
3) $CB \to BC$
4) $aB \to ab$
5) $bB \to bb$
6) $bC \to bc$
7) $cC \to cc$

Một cách phi hình thức, bằng cách áp dụng một số luật sinh cho các chuỗi dẫn xuất sinh ra ngôn ngữ, ta dễ thấy rằng văn phạm G sinh ra ngôn ngữ có dạng :

$$L = \{a^n b^n c^n \mid n \ge 1\}$$

Thật vậy, với luật sinh (1) và (2) ta có chuỗi dẫn xuất $S \Rightarrow^* a^n(BC)^n$. Sau đó, bằng cách áp dụng luật sinh (3), mọi biến B sẽ được thay thế lên trước các biến C trong chuỗi dẫn xuất : $a^n(BC) \Rightarrow^* a^n B^n C^n$. Bởi luật sinh (4) và (5), mọi biến B sẽ được thay thế thành các ký hiệu kết thúc b, và cuối cùng với (6) và (7), mọi biến C cũng sẽ được thay thế thành c. Tóm lại, ta có chuỗi dẫn xuất như sau :

$$S \Rightarrow^* a^n(BC)^n \Rightarrow^* a^n B^n C^n \Rightarrow^* a^n b^n c^n$$

Bài toán thành viên với CSG (Membership)

ĐỊNH LÝ 8.1 : Tồn tại giải thuật để xác định với mọi ngôn ngữ cảm ngữ cảnh CSG G(V, T, P, S) bất kỳ và một chuỗi nhập w ∈ T^*, liệu chuỗi w có thuộc ngôn ngữ L(G) hay không.

Chứng minh

Giả sử | w | = n. Ta lập đồ thị mà mỗi đỉnh là một chuỗi thuộc $(V \cup T)^*$ có độ dài nhỏ hơn hoặc bằng n, có một cung từ đỉnh α đến đỉnh β nếu $\alpha \Rightarrow_G \beta$. Như vậy một đường trong đồ thị đó tương ứng với một suy dẫn trong G. Vậy w ∈ L(G) khi và chỉ khi có một đường đi từ đỉnh bắt đầu S tới đỉnh w trong đồ thị. Dùng bất cứ giải thuật nào cho phép tìm đường nối hai đỉnh trong đồ thị (đã có nhiều thuật toán như thế), ta sẽ xác định được phải chăng đã có đường đi từ đỉnh S tới đỉnh w.

Thí dụ 8.2 : Xét CSG G (V, T, P, S) với các luật sinh được cho như trong Thí dụ 8.1 trên và xét chuỗi nhập w = **abbc**. Ta cần xác định xem liệu chuỗi w ∈ L(G)?

Để tìm đường đi từ đỉnh **S** tới đỉnh **abbc** trong đồ thị nói trên ta có thể dùng phương pháp "vết dầu loang" như sau:

Lập các R(i), i = 0, 1, 2, … theo quy tắc sau:

$R(0) = \{ S \}$

$R(i) = R(i-1) \cup \{ \beta \mid \alpha \Rightarrow \beta \text{ với } \alpha \in R(i-1) \text{ và } |\beta| \le |w| \}$

Do $R(0) \subseteq R(1) \subseteq \ldots \subseteq R(i) \subseteq R(i+1) \subseteq \ldots \subseteq$ tập các đỉnh, vậy tồn tại số k nào đó sao cho:

$R(k) = R(k+1) = R(k+2) = \ldots$

Do đó quá trình thành lập các R(i) sẽ có thể ngừng sau k bước.

Và w ∈ L(G) khi và chỉ khi có i ≤ k để cho w ∈ R(i).

Trong thí dụ trên, giả sử khi ta xét | w |= 4, ta có:

R(0) = { S }
R(1) = {S, aSBC, aBC}
R(2) = {S, aSBC, aBC, abC}
R(3) = {S, aSBC, aBC, abC, abc}
R(4) = R(3)

Vậy chuỗi **abbc** không thuộc L(G).

III. SỰ TƯƠNG ĐƯƠNG GIỮA LBA VÀ CSG

Chúng ta chú ý rằng LBA có thể chấp nhận các chuỗi rỗng ε, còn CSG không thể sinh ra chuỗi rỗng. Ngoài trường hợp đó ra thì LBA sẽ chấp nhận chính xác tất cả các chuỗi được sinh ra từ CSG.

ĐỊNH LÝ 8.2 : Nếu L là một CSG thì L sẽ được chấp nhận bởi một LBA nào đó.

Chứng minh

Cách chứng minh định lý này cũng tương tự như cách chứng minh của định lý 7.9 ở chương trước về sự tương đương giữa lớp ngôn ngữ sinh từ văn phạm loại 0 với lớp ngôn ngữ mà máy Turing chấp nhận, chỉ khác là ở đây không cần dùng một băng nhập thứ hai để phát sinh các dạng câu theo chuỗi dẫn xuất lần lượt theo các suy dẫn của văn phạm, mà chỉ cần dùng rãnh thứ hai trên băng nhập của LBA vào việc đó.

Cho G = (V, T, P, S) là một CSG, ta xây dựng ôtômát LBA M như sau: Băng nhập của LBA gồm hai rãnh : rãnh 1 chứa chuỗi nhập w với các ký hiệu đánh dấu ⊄, $ ở hai đầu, rãnh 2 dùng để phát sinh các dạng câu α. Trạng thái bắt đầu, nếu w = ε thì M ngừng và không chấp nhận input, nếu không thì đầu đọc viết ký hiệu S ở rãnh 2, ngay dưới ký hiệu bên trái nhất của chuỗi w, tiếp đó M thực hiện quá trình sau:

1) Chọn trong số không đơn định một chuỗi con β của chuỗi α trên rãnh 2 sao cho $\beta \to \gamma$ là một luật sinh trong P.

2) Thay β bởi γ, nếu cần thiết ta phải dịch chuyển phần cuối chuỗi sang phải cho đủ chỗ, tuy nhiên nếu dịch chuyển ra ngoài $ thì LBA ngừng và không chấp nhận.

3) (Hình thái hiện tại ở rãnh 1 là ⊄ **w** $, còn ở rãnh 2 là chuỗi α, mà $S \Rightarrow_G \alpha$ và $|\alpha| \le |w|$). So sánh rãnh 1 và rãnh 2, nếu α = w thì LBA ngừng và chấp nhận w. Nếu không thì trở về bước (1).

Như vậy khi M chấp nhận chuỗi w, thì $S \Rightarrow_G^* w$. Ngược lại nếu $S \Rightarrow_G^* w$ thì mọi dạng câu α xuất hiện trong chuỗi dẫn xuất đó đều thoả mãn $|\alpha| \le |w|$, bởi vì mọi luật sinh $\beta \to \gamma$ trong văn phạm G đều thỏa $|\beta| \le |\gamma|$. Như vậy M có thể thực hiện chuỗi dẫn xuất đó trên rãnh 2, giữa hai ký hiệu đánh dấu đầu mút ⊄ và $. Vậy M chấp nhận chuỗi nhập w.

Tóm lại M sẽ chấp nhận mọi chuỗi sinh ra bởi văn phạm G.

<u>ĐỊNH LÝ 8.3</u> : Nếu L = L(M) với một LBA M (Q, Σ, Γ, δ, q_o, ⊄, $, F) thì L – {ε} là một ngôn ngữ cảm ngữ cảnh.

Chứng minh

Cách chứng minh định lý này cũng tương tự như cách chứng minh của định lý 7.10 ở chương trước, bằng cách ta xây dựng một CSG G thực hiện 3 giai đoạn:

- Giai đoạn 1: Văn phạm cho phép sinh ra một chuỗi w (chuỗi nhập của M), cũng được chứa trong ⊄, $ và q_0.
- Giai đoạn 2: Văn phạm lặp lại công việc của M.
- Giai đoạn 3: Khi xuất hiện trạng thái $q \in F$, ta thu về chuỗi w với lưu ý rằng các luật sinh $\alpha \to \beta$ đều có $|\alpha| = |\beta|$.

Quá trình mô phỏng lại các luật sinh đó bởi các luật sinh của CSG sẽ không có gì vướng mắc. Chỉ ở giai đoạn 3, việc xoá đi các ký hiệu đánh dấu hai đầu mút ⊄ và $, q không được phép làm rút ngắn chuỗi nhập lại. Để giải quyết vướng mắc này, ta gắn các ký hiệu ⊄, $, q kề bên với các ký hiệu của chuỗi nhập mà không để đứng rời ra như trước.

Cụ thể, giai đoạn 1 thực hiện bởi các luật sinh trong G sau:

$S_1 \to [a, q_0 \not\subset a]S_2$ $\qquad$ $S_1 \to [a, q_0 \not\subset a\$]$

$S_2 \rightarrow [a, a]S_2$, $S_2 \rightarrow [a,a\$]$

$\forall a \in \Sigma - \{\lhd, \$\}$

Các luật sinh trong G cho phép thực hiện giai đoạn 2, giống như LBA M thực hiện (sinh viên tự xây dựng xem như bài tập).

Cuối cùng, ở giai đoạn 3, các luật sinh sau đây sẽ được sử dụng, với $q \in F$:

$$[a, \alpha q \beta] \rightarrow a$$

$\forall a \in \Sigma - \{\lhd, \$\}$ và $\forall \alpha, \beta$ có thể có.

Chú ý rằng số luật sinh là hữu hạn, vì α và / hoặc β chỉ gồm $\lhd$, \$ và một ký hiệu nhập vào. Chúng ta cũng có thể xoá thành phần thứ hai của một biến nếu nó liền kề với ký hiệu kết thúc bằng cách dùng các luật sinh dạng:

$$[a, \alpha]b \rightarrow ab$$
$$b[a, \alpha] \rightarrow ba$$

$\forall a, b \in \Sigma - \{\lhd, \$\}$ và $\forall \alpha$ có thể có.

Như vậy các luật sinh vừa được xây dựng mô tả văn phạm là CSG và có thể chứng minh $L(M) - \{\varepsilon\} = L(G)$.

IV. TƯƠNG QUAN GIỮA CÁC LỚP NGÔN NGỮ

Ngôn ngữ đoán nhận bởi các văn phạm cũng được phân loại theo tên của từng lớp văn phạm, ta gọi đó là sự phân cấp Chomsky về ngôn ngữ.

Có 4 lớp ngôn ngữ đã được giới thiệu – tập đệ quy liệt kê (r.e), ngôn ngữ cảm ngữ cảnh (CSL), ngôn ngữ phi ngữ cảnh (CFL) và tập chính quy (r) tương đương với 4 lớp ngôn ngữ loại 0, 1, 2 và 3.

Theo lý thuyết được xây dựng xuyên suốt trong giáo trình này, ta có thể tóm tắt lại như sau:

a) L là ngôn ngữ loại 0 khi và chỉ khi L được đoán nhận bởi một máy Turing.

b) L là ngôn ngữ loại 1 khi và chỉ khi L được đoán nhận bởi một ôtômát tuyến tính giới nội (sai khác chuỗi rỗng ε)

c) L là ngôn ngữ loại 2 khi và chỉ khi L được đoán nhận bởi một ôtômát đẩy xuống (không đơn định).

d) L là ngôn ngữ loại 3 khi và chỉ khi L được đoán nhận bởi một ôtômát hữu hạn (sai khác chuỗi rỗng ε).

Ta cũng cần lưu ý rằng sự phân cấp ngôn ngữ như trên là một bao hàm thức nghiêm ngặt, thể hiện quy luật sau:

a) Lớp các ngôn ngữ loại 3 là tập con thực sự của lớp ngôn ngữ loại 2. Thật vậy mọi văn phạm chính quy đều là văn phạm phi ngữ cảnh. Hơn nữa người ta có thể chứng minh rằng ngôn ngữ $\{0^n1^n \mid n \geq 1\}$ là một ngôn ngữ phi ngữ cảnh, nhưng không phải là ngôn ngữ chính quy.

b) Lớp các ngôn ngữ loại 2 không chứa các chuỗi rỗng là tập con thực sự của lớp ngôn ngữ loại 1. Thật vậy mọi văn phạm phi ngữ cảnh có dạng chuẩn Chomsky đều là văn phạm cảm ngữ cảnh. Hơn nữa người ta có thể chứng minh rằng ngôn ngữ $\{a^{2i} \mid i \geq 1\}$ là ngôn ngữ cảm ngữ cảnh nhưng không là ngôn ngữ phi ngữ cảnh.

c) Lớp các ngôn ngữ loại 1 là tập con thực sự của lớp các ngôn ngữ loại 0. Thật vậy, mọi văn phạm cảm ngữ cảnh đều là văn phạm cấu trúc không hạn chế. Mặt khác người ta cũng đề xuất được những ngôn ngữ là đệ quy liệt kê (loại 0), mà không cần làm ngữ cảnh (loại 1). Các thí dụ đó được xây dựng dựa trên các khái niệm "đệ quy" và "sự giải được", mà khuôn khổ giáo trình này không cho phép đề cập đến.

Tổng kết chương VIII: Với sự giới thiệu mô hình ôtômát tuyến tính giới nội LBA và lớp ngôn ngữ cảm ngữ cảnh mà nó đoán nhận, mô hình phân cấp ngôn ngữ theo Noam Chomsky đã được hoàn chỉnh.

BÀI TẬP CHƯƠNG VIII

8.1. Xây dựng văn phạm cảm ngữ cảnh sinh ra các ngôn ngữ sau:

a) $\{ ww \mid w \in (0+1)^+\}$
b) $\{ 0^k \mid k = i^2 \text{ và } i \geq 1\}$
c) $\{ 0^i \mid i \text{ không là số nguyên tố}\}$
d) $\{ a^i b^{2i} c^{3i} \mid i \geq 1\}$
e) $\{ a^i b^i c^k \mid i \geq 1, k \leq 1\}$

8.2. Thiết kế ôtômát tuyến tính giới nội LBA đoán nhận các ngôn ngữ sau:

a) $\{ a^n b^n c^n \mid n \geq 1\}$
b) $\{ ww \mid w \in (a + b + c)^*\}$