Chương V:

VĂN PHẠM PHI NGỮ CẢNH

Giảng viên:

Trần Nguyễn Dương Chi

Nội dung

- Văn phạm phi ngữ cảnh
- Giản lược văn phạm phi ngữ cảnh
- Chuẩn hóa văn phạm phi ngữ cảnh
- Các tính chất của văn phạm phi ngữ cảnh

2

Văn phạm phi ngữ cảnh (1)

- Văn phạm phi ngữ cảnh hay CFG (Context Free Grammar)
- Văn phạm phi ngữ cảnh hữu ích trong:
  - Mô tả các biểu thức số học có nhiều dấu ngoặc lồng nhau
  - Mô tả cấu trúc khối trong ngôn ngữ lập trình (begin – end)

Văn phạm phi ngữ cảnh (2)

<u>Định nghĩa</u>: là hệ thống gồm 4 thành phần **G(V, T, P, S)**

- V : tập hữu hạn các biến (ký tự chưa kết thúc)
- T : tập hữu hạn các ký tự kết thúc ($V \cap T = Ø$)
- P : tập hữu hạn các luật sinh dạng $A \to \alpha$ **(**$\alpha \in (V \cup T)$**)*)**
- S : ký hiệu bắt đầu của văn phạm

<u>Quy ước</u>:

- V: chữ in hoa (A, B, C, ..);
- T: chữ in thường (a, b, c, .., w, x, y..)
- α, β, γ, .. biểu diễn chuỗi ký hiệu kết thúc và biến

<u>Ví dụ</u>: G = ({S, A, B}, {a, b}, P, S) với P gồm các luật sinh:

$S \to AB$
$A \to aA$
$A \to a$
$B \to bB$
$B \to b$

hay

$S \to AB$
$A \to aA \mid a$
$B \to bB \mid b$

Dẫn xuất và ngôn ngữ

Dẫn xuất:

- Nếu $A \rightarrow \beta$ là luật sinh trong văn phạm G và α, γ là 2 chuỗi bất kỳ, thì khi áp dụng luật sinh $A \rightarrow \beta$ vào chuỗi $\alpha\mathbf{A}\gamma$ ta sẽ thu được chuỗi $\alpha\beta\gamma$:

$$\alpha\mathbf{A}\gamma \Rightarrow_G \alpha\beta\gamma$$

- Giả sử: $\alpha_1 \Rightarrow_G \alpha_2$, $\alpha_2 \Rightarrow_G \alpha_3$, ..., $\alpha_{m-1} \Rightarrow_G \alpha_m$, ta có:

$$\alpha_1 \Rightarrow^*_G \alpha_m$$

- Ta có: $\alpha \Rightarrow^*_G \alpha$ với mọi chuỗi α
- Thông thường, ta sẽ dùng $\Rightarrow$ và $\Rightarrow^*$ thay cho $\Rightarrow_G$ và $\Rightarrow^*_G$

Ngôn ngữ sinh bởi CFG: cho CFG G(V, T, P, S)

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

(chuỗi w gồm toàn ký hiệu kết thúc và được dẫn ra từ S)

5

Cây dẫn xuất (1)

Định nghĩa: cây dẫn xuất (hay cây phân tích cú pháp) của một văn phạm G(V, T, P, S) có đặc điểm

(1) Mỗi nút có một nhãn, là một ký hiệu $\in (V \cup T \cup \{\varepsilon\})$

(2) Nút gốc có nhãn là S (ký hiệu bắt đầu)

(3) Nếu nút trung gian có nhãn A thì $A \in V$

(4) Nếu nút n có nhãn A và các đỉnh n_1, n_2, ..., n_k là con của n theo thứ tự từ trái sang phải có nhãn lần lượt là X_1, X_2, ..., X_k thì $A \rightarrow X_1X_2...X_k$ là một luật sinh trong P

(5) Nếu nút n có nhãn là ε thì n phải là nút lá và là nút con duy nhất của nút cha của nó

6

Cây dẫn xuất (2)

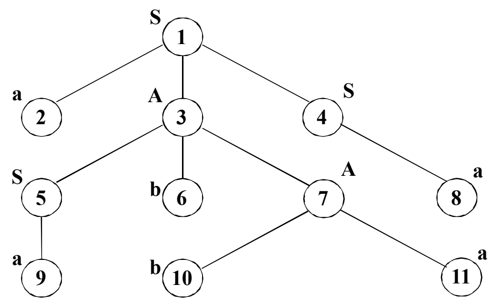
Ví dụ: xét văn phạm G({S, A}, {a, b}, P, S}, với P gồm:

$S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid SS \mid ba$

Một dẫn xuất của G:

$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbaa$



Định lý 5.1: nếu G(V, T, P, S) là một CFG thì $\mathbf{S} \Rightarrow^* \alpha$ nếu và chỉ nếu có cây dẫn xuất trong văn phạm sinh ra α.

Dẫn xuất trái nhất - Dẫn xuất phải nhất

Dẫn xuất trái nhất (phải nhất): nếu tại mỗi bước dẫn xuất, luật sinh được áp dụng vào biến bên trái nhất (phải nhất)

Ví dụ: xét văn phạm G với luật sinh:

$\mathbf{S \rightarrow AB}$

$\mathbf{A \rightarrow aA \mid a}$

$\mathbf{B \rightarrow bB \mid b}$

- Các dẫn xuất khác nhau cho từ **aaabb**:

  (a) $S \Rightarrow AB \Rightarrow aAB \Rightarrow aaAB \Rightarrow aaaB \Rightarrow aaabB \Rightarrow aaabb$

  (b) $S \Rightarrow AB \Rightarrow AbB \Rightarrow Abb \Rightarrow aAbb \Rightarrow aaAbb \Rightarrow aaabb$

  (c) $S \Rightarrow AB \Rightarrow aAB \Rightarrow aAbB \Rightarrow aAbb \Rightarrow aaAbb \Rightarrow aaabb$

  (d) $S \Rightarrow AB \Rightarrow aAB \Rightarrow aaAB \Rightarrow aaAbB \Rightarrow aaabB \Rightarrow aaabb$

- Dẫn xuất (a) là dẫn xuất trái nhất, (b) là dẫn xuất phải nhất
- Các dẫn xuất tuy khác nhau, nhưng có cùng một cây dẫn xuất

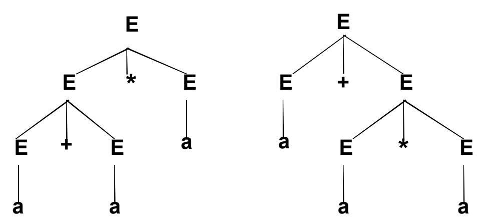
Văn phạm mơ hồ (1)

Khái niệm: một văn phạm phi ngữ cảnh G được gọi là văn phạm mơ hồ (ambiguity) nếu nó có nhiều hơn một cây dẫn xuất cho cùng một chuỗi w.

Ví dụ: xét văn phạm G với luật sinh:

$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

Với chuỗi **a + a * a**, ta có thể vẽ đến 2 cây dẫn xuất khác nhau



Điều này có nghĩa là biểu thức a + a * a có thể hiểu theo 2 cách khác nhau: **(a + a) * a** hoặc **a + (a * a)**

Văn phạm mơ hồ (2)

Khắc phục văn phạm mơ hồ:

- Quy định rằng các phép cộng và nhân luôn được thực hiện theo thứ tự từ trái sang phải (trừ khi gặp ngoặc đơn)

$$E \rightarrow E + T \mid E * T \mid T$$
$$T \rightarrow (E) \mid a$$

- Quy định rằng khi không có dấu ngoặc đơn ngăn cách thì phép nhân luôn được thực hiện ưu tiên hơn phép cộng

$$E \rightarrow E + T \mid T$$
$$T \rightarrow T * F \mid F$$
$$F \rightarrow (E) \mid a$$

Giản lược văn phạm phi ngữ cảnh

Trong CFG có thể chứa các yếu tố thừa:

- Các ký hiệu không tham gia vào quá trình dẫn xuất ra chuỗi ký hiệu kết thúc
- Luật sinh dạng $A \rightarrow B$ (làm kéo dài chuỗi dẫn xuất)

$\Rightarrow$ *giản lược văn phạm nhằm loại bỏ những yếu tố vô ích, nhưng không được làm thay đổi khả năng sản sinh ngôn ngữ của văn phạm*

- Mỗi biến và mỗi ký hiệu kết thúc của văn phạm đều xuất hiện trong dẫn xuất của một số chuỗi trong ngôn ngữ
- Không có luật sinh $A \rightarrow B$ (với A, B đều là biến)
- Nếu ngôn ngữ không chấp nhận chuỗi rỗng ε thì không cần luật sinh $A \rightarrow \varepsilon$.

11

Các ký hiệu vô ích (1)

Khái niệm: một ký hiệu X được gọi là có ích nếu có một dẫn xuất

$$\mathbf{S} \Rightarrow^* \alpha\mathbf{X}\beta \Rightarrow^* \mathbf{w}$$

với α, β là các chuỗi bất kỳ và $w \in T^*$.

$\Rightarrow$ *có 2 đặc điểm cho ký hiệu có ích*

- X phải dẫn ra chuỗi ký hiệu kết thúc
- X phải nằm trong dẫn xuất từ S

12

Các ký hiệu vô ích (2)

<u>Bổ đề 1</u>: *(loại bỏ các biến không dẫn ra chuỗi ký hiệu kết thúc)*

Cho CFG G(V, T, P, S) với L(G) ≠ Ø, có một CFG G'(V', T', P', S) tương đương sao cho mỗi **A** $\in$ **V'** tồn tại **w** $\in$ **T*** để **A** $\Rightarrow$* **w**

<u>Giải thuật tìm V'</u>:

Begin

(1) OldV' := ∅;

(2) NewV' := { A | A → w với w ∈ T* };

(3) *While* OldV' ≠ NewV' *do*

begin

(4) OldV' := NewV';

(5) NewV' := OldV' ∪ {A | A → α với α ∈ (T ∪ OldV')* }

end;

(6) V' := NewV';

End;

Các ký hiệu vô ích (3)

<u>Bổ đề 2</u>: *(loại bỏ các biến không được dẫn ra từ ký hiệu bắt đầu)*

Cho CFG G(V, T, P, S), ta có thể tìm được CFG G'(V', T', P', S) tương đương sao cho mỗi **X** $\in$ **(V'** $\cup$ **T')** tồn tại α, β $\in$ **(V'** $\cup$ **T')*** để **S** $\Rightarrow$* α**X**β

<u>Cách tìm</u>:

- Đặt V' = {S}
- Nếu A $\in$ V' và A $\rightarrow \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_n$ là các luật sinh trong P thì
  - Thêm các biến của $\alpha_1, \alpha_2, \ldots, \alpha_n$ vào **V'**, các ký hiệu kết thúc vào T'
- Lặp lại cho đến khi không còn biến nào được thêm vào nữa

Các ký hiệu vô ích (4)

Định lý 5.2: mỗi ngôn ngữ phi ngữ cảnh (CFL) không rỗng được sinh ra từ một văn phạm phi ngữ cảnh (CFG) không có ký hiệu vô ích

Ví dụ: xét văn phạm

$S \rightarrow AB \mid a$

$A \rightarrow a$

- ***Áp dụng bổ đề 1***:
  $V' = \{S, A\}$
  $S \rightarrow a$
  $A \rightarrow a$

- ***Áp dụng bổ đề 2***:
  $V' = \{S\}$
  $S \rightarrow a$

Luật sinh ε (1)

Định lý 5.3: *(loại bỏ luật sinh $A \rightarrow \varepsilon$)*

Cho CFG G(V, T, P, S) và L là ngôn ngữ sinh ra bởi G. Khi đó $L - \{\varepsilon\}$ là ngôn ngữ sinh ra bởi CFG G'(V, T, P', S) không có ký hiệu vô ích và không có luật sinh ε.

Cách tìm:

- *Bước 1*: xác định tập biến rỗng Nullable
  i. $A \rightarrow \varepsilon$ $\Rightarrow A \in$ Nullable
  ii. $B \rightarrow X_1X_2...X_n$, $\forall X_i \in$ Nullable $\Rightarrow B \in$ Nullable
- *Bước 2:* xây dựng tập luật sinh P'
  Với mỗi luật sinh $A \rightarrow X_1X_2...X_n$ trong P, ta xây dựng luật sinh

$A \rightarrow \alpha_1\alpha_2...\alpha_n$ với điều kiện:

i. Nếu $X_i \notin$ Nullable thì $\alpha_i = X_i$

ii. Nếu $X_i \in$ Nullable thì $\alpha_i = X_i \mid \varepsilon$

iii. Không phải tất cả α_i đều bằng ε

Luật sinh ε (2)

Ví dụ: loại bỏ luật sinh ε trong văn phạm sau:

$S \to AB$
$A \to aA \mid \varepsilon$
$B \to bB \mid \varepsilon$

- *Bước 1*: xác định tập biến rỗng Nullable
  - i. $A \to \varepsilon \quad \Rightarrow A \in$ Nullable
  - ii. $B \to \varepsilon \quad \Rightarrow B \in$ Nullable
  - iii. $S \to AB \quad \Rightarrow S \in$ Nullable

- *Bước 2:* xây dựng tập luật sinh P'

  $S \to AB \mid A\varepsilon \mid \varepsilon B$
  $A \to aA \mid a\varepsilon$
  $B \to bB \mid b\varepsilon$

Chú ý: văn phạm G' không chấp nhận chuỗi rỗng ε như văn phạm G. Để G' tương đương G, ta cần thêm luật sinh $S \to \varepsilon$ vào G'.

17

Luật sinh đơn vị (1)

Định lý 5.4: *(loại bỏ luật sinh $A \to B$)*
Mỗi CFL không chứa ε được sinh ra bởi CFG không có ký hiệu vô ích, không có luật sinh ε hoặc luật sinh đơn vị.

Cách tìm: đặt L= L(G) là CFL không chứa ε và được sinh ra bởi văn phạm G(V, T, P, S). Theo định lý 3, ta có thể loại bỏ tất cả luật sinh ε trong G.

Để loại bỏ luật sinh đơn vị, ta xây dựng tập P' mới theo giải thuật:

```
For (mỗi biến A ∈ V) do
  Begin
    Tính Δ_A = { B | B ∈ V và A ⇒* B (luật sinh đơn vị)} ;
    For (mỗi biến B ∈ Δ_A) do
      For (mỗi luật sinh B → α thuộc P) do
        If (B → α không là luật sinh đơn vị) then
          Thêm luật sinh A → α vào P'
  End ;
```

18

Luật sinh đơn vị (2)

Ví dụ: loại bỏ luật sinh đơn vị trong văn phạm

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid a$

Ta có: $\Delta_E = \{E, T, F\} \Rightarrow$ thêm vào P' các luật sinh

$E \to E + T \mid T * F \mid (E) \mid a$

Tương tự:

$\Delta_T = \{T, F\} \Rightarrow$ thêm vào P' : $T \to T * F \mid (E) \mid a$

$\Delta_F = \{F\} \Rightarrow$ thêm vào P' : $F \to (E) \mid a$

19

Dạng chuẩn Chomsky (CNF) (1)

Định lý 5.5: một ngôn ngữ phi ngữ cảnh bất kỳ không chứa ε đều được sinh ra bằng một văn phạm nào đó mà các luật sinh có dạng $A \to BC$ hoặc $A \to a$, với A, B, C là biến và a là ký hiệu kết thúc.

Cách tìm: giả sử CFL L=L(G) với CFG G(V, T, P, S)

- *Bước 1*: thay thế tất cả các luật sinh có độ dài vế phải là 1
  - Áp dụng định lý 4.4 để loại bỏ luật sinh đơn vị và ε
- *Bước 2*: thay thế tất cả luật sinh có độ dài vế phải lớn hơn 1 và có chứa ký hiệu kết thúc

  $A \to X_1X_2...X_i...X_n$ (X_i = a) $\quad$ $A \to X_1X_2...C_a...X_n$, $C_a \to a$

- *Bước 3*: thay thế các luật sinh mà vế phải có nhiều hơn 2 ký hiệu chưa kết thúc

  $A \to B_1B_2...B_m$ (m>2) $\quad$ $A \to B_1 D_1$, $D_1 \to B_2 D_2$, ..., $D_{m-2} \to B_{m-1} B_m$

20

Dạng chuẩn Chomsky (CNF) (2)

<u>Ví dụ</u>: tìm văn phạm có dạng CNF tương đương văn phạm sau:

$S \to A \mid ABA$
$A \to aA \mid a \mid B$
$B \to bB \mid b$

<u>Bước 1</u>: $\Delta_s = \{S, A, B\}$, $\Delta_A = \{A, B\}$, $\Delta_B = \{B\}$

$S \to aA \mid a \mid bB \mid b \mid ABA$
$A \to aA \mid a \mid bB \mid b$
$B \to bB \mid b$

<u>Bước 2</u>: thay a bằng C_a và b bằng C_b trong các luật sinh có độ dài vế phải > 1:

$S \to C_aA \mid a \mid C_bB \mid b \mid ABA$
$A \to C_aA \mid a \mid C_bB \mid b$
$B \to C_bB \mid b$
$C_a \to a$
$C_b \to b$

Dạng chuẩn Chomsky (CNF) (3)

$S \to C_aA \mid a \mid C_bB \mid b \mid ABA$
$A \to C_aA \mid a \mid C_bB \mid b$
$B \to C_bB \mid b$
$C_a \to a$
$C_b \to b$

<u>Bước 3</u>: thay thế các luật sinh có độ dài vế phải > 2:

$\mathbf{S} \to C_aA \mid a \mid C_bB \mid b \mid A\mathbf{D_1}$
$A \to C_aA \mid a \mid C_bB \mid b$
$B \to C_bB \mid b$
$C_a \to a$
$C_b \to b$
$\mathbf{D_1 \to BA}$

Dạng chuẩn Greibach (GNF) (1)

Bổ đề 3: *(thay thế các luật sinh trực tiếp)*

Cho G(V, T, P, S) là một CFG, đặt $A \rightarrow \alpha_1 B \alpha_2$ là luật sinh trong P và $B \rightarrow \beta_1 \mid \beta_2 \mid ... \mid \beta_r$ là các B - luật sinh; văn phạm $G_1(V, T, P_1, S)$ thu được từ G bằng cách loại bỏ luật sinh $A \rightarrow \alpha_1 B \alpha_2$ và thêm vào luật sinh $A \rightarrow \alpha_1\beta_1\alpha_2 \mid \alpha_1\beta_2\alpha_2 \mid ... \mid \alpha_1\beta_r\alpha_2$ tương đương G

Bổ đề 4: *(dùng loại bỏ văn phạm đệ quy trái)*

Đặt G(V, T, P, S) là CFG; $A \rightarrow A\alpha_1 \mid A\alpha_2 \mid ... \mid A\alpha_r$ là tập các A – luật sinh có A là ký hiệu trái nhất của vế phải (luật sinh đệ quy trái). Đặt $A \rightarrow \beta_1 \mid \beta_2 \mid ... \mid \beta_s$ là các A - luật sinh còn lại; $G_1(V \cup \{B\}, T, P_1, S)$ là CFG được tạo thành bằng cách thêm biến mới B vào V và thay các A - luật sinh bằng các luật sinh dạng:

$$\left.\begin{array}{l} A \rightarrow \beta_i \\ A \rightarrow \beta_i B \end{array}\right\} (1 \le i \le s) \qquad \left.\begin{array}{l} B \rightarrow \alpha_i \\ B \rightarrow \alpha_i B \end{array}\right\} (1 \le i \le r)$$

Thì ta có G_1 tương đương G, hay $L(G) = L(G_1)$

23

Dạng chuẩn Greibach (GNF) (2)

Định lý 5.6: mỗi CFL bất kỳ không chứa ε được sinh ra bởi một CFG mà mỗi luật sinh có dạng $A \rightarrow a\alpha$ với A là biến, a là ký hiệu kết thúc và α là một *chuỗi các biến* (có thể rỗng)

Đặt G là CFG sinh ra CFL không chứa ε

Bước 1: xây dựng G' có dạng CNF tương đương G

Bước 2: đổi tên các biến trong G' thành $A_1, A_2, ..., A_m$ (m ≥1) với A_1 là ký hiệu bắt đầu. Đặt $V = \{A_1, A_2, ..., A_m\}$

Bước 3: thay thế luật sinh sao cho nếu $A_i \rightarrow A_j\gamma$ thì $j > i$

- Nếu j<i : áp dụng bổ đề 3. Nếu i=j : áp dụng bổ đề 4 *(giải thuật)*
- Trong P chỉ chứa các luật sinh dạng: $A_i \rightarrow A_j\gamma$ (j > i), $A_i \rightarrow a\gamma$ hoặc $B_k \rightarrow \gamma$ với $\gamma \in (V \cup \{B_1, B_2, ..., B_{i-1}\})^*$

Bước 4: thay thế các A_i – luật sinh về đúng dạng *(áp dụng bổ đề 3)*

Bước 5: thay thế các B_k – luật sinh về đúng dạng *(bổ đề 3)*

24

Dạng chuẩn Greibach (GNF) (3)

Giải thuật : *(thay thế sao cho $A_i \rightarrow A_j\gamma$ thì $j > i$)*

Begin
(1) ***for*** k := 1 ***to*** m ***do begin***
(2) ***for*** j := 1 ***to*** k-1 ***do***
(3) ***for*** Mỗi luật sinh dạng $A_k \rightarrow A_j\alpha$ ***do***
begin
(4) ***for*** Tất cả luật sinh $A_j \rightarrow \beta$ ***do***
(5) Thêm luật sinh $A_k \rightarrow \beta\alpha$;
(6) Loại bỏ luật sinh $A_k \rightarrow A_j\alpha$
end;
(7) ***for*** Mỗi luật sinh dạng $A_k \rightarrow A_k\alpha$ ***do***
begin
(8) Thêm các luật sinh $B_k \rightarrow \alpha$ và $B_k \rightarrow \alpha B_k$;
(9) Loại bỏ luật sinh $A_k \rightarrow A_k\alpha$
end;
(10) ***for*** Mỗi luật sinh $A_k \rightarrow \beta$ trong đó β không bắt đầu bằng A_k ***do***
(11) Thêm luật sinh $A_k \rightarrow \beta B_k$
end;
end;

Dạng chuẩn Greibach (GNF) (4)

Ví dụ: tìm văn phạm có dạng GNF cho văn phạm G sau:

$$A_1 \rightarrow A_2A_1 \mid A_2A_3$$
$$A_2 \rightarrow A_3A_1 \mid a$$
$$A_3 \rightarrow A_2A_2 \mid b$$

Bước 1: G thỏa CNF

Bước 2: ta có V = $\{A_1, A_2, A_3\}$

Bước 3: ta cần sửa đổi luật sinh $A_3 \rightarrow A_2A_2$

- Áp dụng bổ đề 3: $A_3 \rightarrow A_3A_1A_2 \mid aA_2$
  $\Rightarrow \mathbf{A_3 \rightarrow A_3A_1A_2 \mid aA_2 \mid b}$
-

Dạng chuẩn Greibach (GNF) (5)

- Áp dụng bổ đề 4, ta thu được tập luật sinh:

$A_1 \rightarrow A_2A_1 \mid A_2A_3$
$A_2 \rightarrow A_3A_1 \mid a$
$A_3 \rightarrow aA_2 \mid b \mid aA_2B \mid bB$
$B \rightarrow A_1A_2 \mid A_1A_2B$

Bước 4: A_3 đã có dạng chuẩn. Thay thế A_3 vào A_2 :

$B \rightarrow A_1A_2 \mid A_1A_2B$
$A_3 \rightarrow aA_2 \mid b \mid aA_2B \mid bB$
$A_2 \rightarrow aA_2A_1 \mid bA_1 \mid aA_2BA_1 \mid bBA_1 \mid a$
$A_1 \rightarrow aA_2A_1A_1 \mid bA_1A_1 \mid aA_2BA_1A_1 \mid bBA_1A_1 \mid aA_1 \mid$
$aA_2A_1A_3 \mid bA_1A_3 \mid aA_2BA_1A_3 \mid bBA_1A_3 \mid aA_3$

Bước 5: thay thế các B_k – luật sinh

$B \rightarrow aA_2A_1A_1A_2 \mid bA_1A_1A_2 \mid aA_2BA_1A_1A_2 \mid bBA_1A_1A_2 \mid aA_1A_2 \mid$
$aA_2A_1A_3A_2 \mid bA_1A_3A_2 \mid aA_2BA_1A_3A_2 \mid bBA_1A_3A_2 \mid aA_3A_2 \mid$
$aA_2A_1A_1A_2B \mid bA_1A_1A_2B \mid aA_2BA_1A_1A_2B \mid bBA_1A_1A_2B \mid aA_1A_2B \mid$
$aA_2A_1A_3A_2B \mid bA_1A_3A_2B \mid aA_2BA_1A_3A_2B \mid bBA_1A_3A_2B \mid$
aA_3A_2B

27

Bổ đề bơm cho CFL (1)

- **Bổ đề bơm:** cho L là một CFL bất kỳ, tồn tại một số n chỉ phụ thuộc vào L sao cho nếu $z \in L$ và $|z| \geq n$ thì ta có thể viết $z=uvwxy$ sao cho: $|vx| \geq 1$, $|vwx| \leq n$ và $\forall i \geq 0$ ta có $uv^iwx^iy \in L$

28

Bổ đề bơm cho CFL (2)

Ví dụ: chứng minh L = $\{a^i b^i c^i \mid i \geq 1\}$ không là CFL

- Giả sử L là CFL, khi đó tồn tại số n theo bổ đề bơm
- Xét chuỗi $z = a^n b^n c^n$, $|z| \geq n$, ta có thể viết z=uvwxy thỏa bổ đề
- Ta có: $vwx \in a^n b^n c^n$, $|vwx| \leq n$ nên vwx không thể đồng thời chứa cả ký hiệu a và c (vì giữa a và c có n ký hiệu b) → vx cũng không thể chứa cả ký hiệu a và c.
- Do $|vx| \geq 1$ và trong uvwxy chứa số ký hiệu a, b, c bằng nhau:
  - Nếu vx có chứa ký hiệu a (nên không thể chứa ký hiệu c) thì khi bơm chuỗi vx, số ký hiệu c sẽ không đổi (luôn là n), nhưng số ký hiệu a sẽ thay đổi. Ví dụ: chuỗi $uv^0wx^0y \notin L$ vì có số ký hiệu a (ít hơn n) số ký hiệu c (luôn là n) không bằng nhau.
  - Nếu vx không chứa ký hiệu a thì khi bơm chuỗi vx, số ký hiệu a không đổi, nhưng số ký hiệu b (hoặc c) sẽ thay đổi.

29

Tính chất đóng của CFL

Định lý 5.7: CFL đóng với phép hợp, phép nối kết và phép bao đóng Kleen.

Định lý 5.8: CFL không đóng với phép giao

Hệ quả: CFL không đóng với phép lấy phần bù

30