ĐẠI HỌC DÂN LẬP THĂNG LONG

BÀI GIẢNG

LÝ THUYẾT
VĂN PHẠM, NGÔN NGỮ và ÔTÔMAT

NGUYỄN QUỐC THẮNG - NGUYỄN LÂM TÙNG
Tổ bộ môn TOÁN

Nội dung chính

- **CHƯƠNG 1: NHẬP MÔN VỀ VĂN PHẠM VÀ NGÔN NGỮ HÌNH THỨC** **slide**
  - Ngôn ngữ và các phép toán trên ngôn ngữ
    - Định nghĩa bảng chữ cái và các phép toán trên từ 6
    - Các phép toán trên ngôn ngữ 11
  - Văn phạm và ngôn ngữ sinh bởi văn phạm
    - Định nghĩa văn phạm, dẫn xuất 17
    - Ngôn ngữ sinh bởi văn phạm 20
  - Phân loại và tính chất của văn phạm
    - Phân loại văn phạm và ngôn ngữ sinh bởi văn phạm 21
    - Một số tính chất của văn phạm 24
  - Một số ví dụ về văn phạm 39
  - Bài tập chương 1 46

Nội dung chính

CHƯƠNG 2: BIẾN ĐỔI VĂN PHẠM PHI NGỮ CẢNH THÀNH NHỮNG VĂN PHẠM ĐẶC BIỆT **slide**

- Giản lược văn phạm phi ngữ cảnh
  - Giản lược ký hiệu vô sinh 52
  - Giản lược ký hiệu không đến được 54
  - Giản lược ký hiệu thừa 56
  - Loại bỏ quy tắc rỗng 58
  - Loại bỏ quy tắc đơn 61
- Đưa văn phạm về dạng chuẩn Chomsky 63
- Khử quy tắc đệ quy trái 67
- Đưa văn phạm về dạng chuẩn Greiback 69
- Điều kiện cần của ngôn ngữ phi ngữ cảnh 73
- Tính đóng của ngôn ngữ phi ngữ cảnh 76
- Cây suy dẫn trong văn phạm phi ngữ cảnh 78
- Thuật toán phân tích từ trên xuống và từ dưới lên 82,88
- Bài tập chương 2 92

Nội dung chính

CHƯƠNG 3: NGUỒN VÀ ÔTÔMAT HỮU HẠN	slide
Nguồn và ngôn ngữ sinh bởi nguồn	96
Các phép toán trên nguồn	
Phép đơn định hóa và lấy phần bù	104,109
Phép hợp nguồn và phép giao nguồn	112,115
Phép tích ghép nguồn	119
Phép lặp, lặp cắt nguồn	122,124
Phép soi gương nguồn	126
Phép chia trái, chia phải nguồn	128,132
Ôtômat hữu hạn và ngôn ngữ đoán nhận bởi ôtômat hữu hạn	136
Sự tương đương của nguồn và ôtômat hữu hạn	146
Sự tương đương của nguồn và văn phạm chính quy	150
Sự tương đương của nguồn và biểu thức chính quy	154
Biểu thức chính quy tương đương với nguồn	160
Điều kiện cần và đủ của ngôn ngữ chính quy	167
Bài tập chương 3	169

Nội dung chính

- **CHƯƠNG 4: ÔTÔMAT ĐẨY XUỐNG** **slide**
  - Khái niệm về ôtômat đẩy xuống 180
  - Ngôn ngữ đoán nhận bởi ôtômat đẩy xuống 183
  - Ôtômat đẩy xuống đoán nhận ngôn ngữ phi ngữ cảnh 186
  - Xây dựng ôtômat đẩy xuống đoán nhận ngôn ngữ theo trạng thái kết và ngăn xếp rỗng 190
  - Các hàm $First$ và $Follow$ 194
  - Văn phạm LL_1 và thuật toán phân tích tất định 197
  - Bài tập chương 4 201

Chương 1 - Giới thiệu chung

- Phần đầu của chương này trình bày những khái niệm cơ bản nhất của ngôn ngữ hình thức. Bắt đầu từ những phép toán trên từ cho đến các phép toán trên ngôn ngữ.
- Phần hai trình bày nhưng khái niện về văn phạm và quá trình sinh ra ngôn ngữ của văn phạm. Sau khi nghiên cứu lớp ngôn ngữ sinh bởi văn phạm, ta được kết quả hết sức quan trọng là lớp ngôn ngữ này đóng với các phép toán hợp, giao, tích ghép, lặp, lặp cắt, soi gương, chia trái, chia phải. Ngoài ra phần này cũng phân loại ra một số lớp các văn phạm đặc biệt mà ta sẽ gặp trong các chương sau.
- Phần cuối chương trình bày một số ví dụ về văn phạm, những ví dụ này cho ta một số kỹ thuật để xây dựng văn phạm sinh ra một ngôn ngữ cho trước đồng thời cũng giới thiệu một số văn phạm đặc biệt sinh ra các ngôn ngữ phức tạp.

Một số khái niệm mở đầu- Bảng chữ cái

- **Định nghĩa .**
  Tập $\Sigma \neq \emptyset$ gồm hữu hạn hay vô hạn các đối tượng cụ thể được gọi là bảng chữ cái. Mỗi phần tử thuộc Σ được gọi là kí hiệu hay chữ cái.
- **Ví dụ.**
  $A = \{0, 1\}$ là bảng chữ cái nhị phân.
  $B = \{a, b, \ldots, z\}$ là bảng chữ cái tiếng Anh.
  $C = \{\square, 1, c\}$ là bảng chữ cái gồm hình vuông, số 1 và chữ c.

Một số khái niệm mở đầu-Từ-Ngôn ngữ

- **Định nghĩa**.
Một dãy hữu hạn các kí hiệu thuộc bảng chữ cái Σ được gọi là một từ hay một xâu trên bảng chữ cái Σ.
- **Ví dụ.** $\Sigma = \{a, b, c\}$
$\alpha = aabacc, \ \beta = abbca$ là hai xâu trên Σ
- Tổng số vị trí của tất cả các kí hiệu xuất hiện trong α được gọi là độ dài của từ α và ký hiệu là $|\alpha|$ hay $l(\alpha)$.
Trong ví dụ trên $|\alpha| = 6, \ |\beta| = 5$.
- Từ có độ dài 0 được gọi là từ rỗng hay từ trống, kí hiệu là $\wedge$ (hay ε hay e).
- Tập tất cả các từ trên bảng chữ cái Σ kí hiệu là Σ^* và $\Sigma^+ = \Sigma^* \setminus \{\wedge\}$.
- **Định nghĩa.** Mỗi tập con $A \subset \Sigma^*$ được gọi là một ngôn ngữ trên bảng chữ cái Σ.
- Tập $\emptyset$ được gọi là ngôn ngữ trống hay ngôn ngữ rỗng, do đó ngôn ngữ rỗng là ngôn ngữ trên bất cứ bảng chữ cái nào (Vì sao?)
- **Câu hỏi**: Ngôn ngữ rỗng khác với ngôn ngữ chỉ gồm một từ rỗng (Vì sao?)

Các phép toán trên từ- Tích ghép

- **Định nghĩa.** Tích ghép của các từ không rỗng $\alpha = a_1a_2\dots a_m$ và $\beta = b_1b_2\dots b_n$ là từ $c_1c_2\dots c_{m+n}$ trong đó:
  $c_1 = a_1, \dots c_m = a_m; c_{m+1} = b_1, \dots c_{m+n} = b_n.$
- Tích ghép của các từ α và β kí hiệu là $\alpha\beta$ hay $\alpha.\beta$
- Ta quy ước $\alpha\wedge = \wedge\alpha = \alpha$.
- Về thực chất tích ghép của từ α với từ β là xâu nhận được khi viết β ngay sau bên phải α.
  Ta viết $\omega^n = \omega\dots\omega$ (n lần) và $\omega^0 = \wedge$.
- Tích ghép có giao hoán, kết hợp không?
- **Ví dụ.**
  $\alpha = abbc$, $\beta = aacb$ thì $\alpha\beta = abbcaacb$
  $x = bac,\ y = caab$ thì $xy = baccaab$

Phép chia từ

- **Định nghĩa.** Nếu $\alpha = \beta\gamma$ thì
  - γ là thương của phép chia trái từ α cho từ β, ký hiệu là $\gamma = \beta\diagdown\alpha$
  - β là thương của phép chia phải từ α cho từ γ, ký hiệu là $\beta = \alpha\diagup\gamma$
- Với mọi từ α, β, γ nếu:
  - β không phải là phần đầu của α thì $\beta\diagdown\alpha$ không xác định.
  - γ không phải là phần cuối của α thì $\alpha\diagup\gamma$ không xác định.
- **Ví dụ.**
  Nếu $\alpha = 10101, \beta = 101$ thì $\quad \beta\diagdown\alpha = 01, \quad \alpha\diagup\gamma = 10$
  Nếu $x = abcbc,\ y = cbc$ thì $x\diagup y = ab, \quad y\diagdown x$ không xác định.
- **Tính chất.** Với mọi từ ω ta có:
  - $\Lambda\diagdown\omega = \omega, \quad \omega\diagup\Lambda = \omega$
  - $\omega^m\diagup\omega^n = \omega^{m-n} \quad (m \geq n)$

Phép soi gương

- **Định nghĩa.**
  Nếu $\alpha = a_1a_2\ldots a_m (m \geqslant 1)$ là từ khác trống thì từ $a_ma_{m-1}\ldots a_2a_1$ được gọi là từ soi gương hay từ ngược của α, kí hiệu là $\widetilde{\alpha}$.
- **Quy ước.** $\widetilde{\wedge} = \wedge$
- **Tính chất.** $\forall \alpha, \beta$ ta có $\widetilde{\widetilde{\alpha}} = \alpha$, $\widetilde{\alpha\beta} = \widetilde{\beta}.\widetilde{\alpha}$ và $|\widetilde{\alpha}| = |\alpha|$
- **Ví dụ.**
  $\alpha = abcc, \beta = bba$. Khi đó: $\widetilde{\alpha} = ccba$, $\widetilde{\beta} = abb$,
  $\widetilde{\alpha\beta} = \widetilde{abccbba} = abbccba = \widetilde{\beta}.\widetilde{\alpha}$
- **Bài tập.** Chứng minh rằng nếu $\alpha = \beta\gamma$ thì:
  $\beta\diagdown\alpha = \beta\diagdown\beta\gamma = \gamma = \widetilde{\widetilde{\gamma}} = \widetilde{\widetilde{\alpha}\diagup\widetilde{\beta}}$
  $\alpha\diagup\gamma = \beta\gamma\diagup\gamma = \beta = \widetilde{\widetilde{\beta}} = \widetilde{\widetilde{\gamma}\diagdown\widetilde{\alpha}}$

Các phép toán trên ngôn ngữ- Phép hợp,giao

- **Định nghĩa.** Tập các từ $\{x \mid x \in L_1 \text{ hoặc } x \in L_2\}$ được gọi là hợp của hai ngôn ngữ L_1 và L_2, ký hiệu là $L_1 \cup L_2$.
- **Ví dụ.**
  $L_1 = \{\wedge, a, b, ab, bc\}, \quad L_2 = \{a, c, ba, bc\}$
  Khi đó: $L_1 \cup L_2 = \{\wedge, a, b, c, ab, ba, bc\}$
- Chứng minh rằng phép hợp có tính giao hoán, kết hợp.
- Nếu L_1 và L_2 là hai ngôn ngữ hữu hạn thì khi nào ta có
  $$|L_1 \cup L_2| = |L_1| + |L_2| \,?$$
- **Định nghĩa.** Tập các từ $\{x \mid x \in L_1 \text{ và } x \in L_2\}$ được gọi là giao của các ngôn ngữ L_1 và L_2, ký hiệu là $L_1 \cap L_2$.
- **Ví dụ.**
  $L_1 = \{a, c, bc, ac, abc, aab\}, \qquad L_2 = \{b, c, bc, bca, abc, cab\}$
  Khi đó: $L_1 \cap L_2 = \{c, bc, abc\}$.
- Chứng minh rằng phép giao có tính giao hoán, kết hợp và phân phối với phép hợp.

Phép lấy phần bù

- **Định nghĩa.** Cho L là ngôn ngữ trên bảng chữ cái Σ, khi đó tập các từ $\{x \in \Sigma^* \mid x \notin L\}$ được gọi là ngôn ngữ bù của ngôn ngữ L, ký hiệu là $C_\Sigma L$ hay $C(L)$.
- **Ví dụ.** Cho $\Sigma = \{a\},\ L = \{a, a^3, a^5, \ldots\}$
Khi đó $\Sigma^* = \{a^n | n \in \mathbb{N}\}$ do vậy $C_\Sigma L = \{a^{2k} | k \in \mathbb{N}\}$.
- Chứng minh rằng nếu L_1, L_2 là hai ngôn ngữ trên bảng chữ cái Σ thì
$$\overline{L_1 \cup L_2} = \overline{L_1} \cap \overline{L_2}$$
$$\overline{L_1 \cap L_2} = \overline{L_1} \cup \overline{L_2}$$
Nếu L_1, L_2 là hai ngôn ngữ trên hai bảng chữ cái khác nhau thì hai đẳng thức trên có đúng không?

Phép tích ghép

- **Định nghĩa.** Tập các từ $\{x = yz | y \in L_1, z \in L_2\}$ được gọi là ngôn ngữ tích ghép của ngôn ngữ L_1 và L_2, ký hiệu là $L_1.L_2$ hay L_1L_2.
- **Quy ước.** $L^n = L.L \ldots L$ (n lần) và $L^0 = \{\wedge\}$.
- **Ví dụ.** $L_1 = \{\wedge, a, bc\}$, $\qquad L_2 = \{b, ca\}$
  Khi đó: $\quad L_1.L_2 = \{b, ca, ab, aca, bcb, bcca\}$
  $L_2.L_1 = \{b, ba, bbc, ca, caa, cabc\}$
- **Câu hỏi.** Nếu $|L_1| = n$, $\quad |L_2| = m$ thì $|L_1.L_2| = m.n$ có đúng không ?
- **Tính chất.**
  - $L_1.L_2 \neq L_2.L_1$
  - $(L_1.L_2).L_3 = L_1.(L_2.L_3)$
  - $L.\emptyset = \emptyset.L = \emptyset$
  - $L.\{\wedge\} = \{\wedge\}.L = L$

Phép lặp

- **Định nghĩa.**
  Ngôn ngữ $L^* = \bigcup_{i=0}^{\infty} L^i$ được gọi là ngôn ngữ lặp của ngôn ngữ L.
  Ngôn ngữ $L^+ = \bigcup_{i=1}^{\infty} L^i$ được gọi là ngôn ngữ lặp cắt của ngôn ngữ L.
- **Ví dụ.**
  $L = \{a, bc\}$
  $L^* = \{\wedge\} \cup \{a, bc\} \cup \{a^2, abc, bca, bcbc\} \cup \ldots$
  $L^+ = \{a, bc\} \cup \{a^2, abc, bca, bcbc\} \cup \ldots$
- **Tính chất.**
  - $(L^*)^* = L^*$
  - $\{\wedge\}^* = \{\wedge\}$
  - $(\emptyset)^* = \{\wedge\}$
  - $(\emptyset)^+ = \emptyset$
- **Câu hỏi.** Khi nào $L^* = L^+$?

Phép soi gương

- **Định nghĩa.**
  Ngôn ngữ $\widetilde{L} = \{\widetilde{x} \mid x \in L\}$ được gọi là ngôn ngữ soi gương hay ngôn ngữ ngược của ngôn ngữ L.
- **Ví dụ.**
  $L = \{\wedge, ab, abc, cbaa\}$ khi đó $\widetilde{L} = \{\wedge, ba, cba, aabc\}$
- **Tính chất.**
  - $\widetilde{\widetilde{L}} = L$
  - $\widetilde{\emptyset} = \emptyset$
- **Bài tập.** Chứng minh rằng $\widetilde{L \cup \widetilde{L}} = L \cup \widetilde{L}$.

Phép chia

- **Định nghĩa.** $L_2 \diagdown L_1 = \{\omega = \beta \diagdown \alpha \mid \alpha \in L_1, \beta \in L_2\}$ được gọi là thương của phép chia trái ngôn ngữ L_1 cho L_2.
  Tương tự $L_1 \diagup L_2 = \{\omega = \alpha \diagup \beta \mid \alpha \in L_1, \beta \in L_2\}$ được gọi là thương của phép chia phải của ngôn ngữ L_1 cho L_2.
- **Ví dụ.** $L_1 = \{a, b, abc, cab, bcaa\}, \quad L_2 = \{\wedge, c, ab\}$
  Khi đó:
  $L_2 \diagdown L_1 = \{a, b, abc, cab, bcaa, ab, c\}$,
  $L_1 \diagup L_2 = \{a, b, abc, cab, bcaa, ab, c\}$
  $L_1 \diagdown L_2 = \{b\}, \quad L_2 \diagup L_1 = \{a\}$
- **Tính chất.**
  - $\{\wedge\} \diagdown L = L \diagup \{\wedge\} = L, \quad L \diagup \emptyset = \emptyset \diagdown L = \emptyset$
  - $L \diagdown \Sigma^* = \Sigma^* \diagup L = \Sigma^*, \quad L \diagdown \Sigma^+ = \Sigma^+ \diagup L = \Sigma^* \quad (L \neq \emptyset, \{\wedge\})$
- Hai đẳng thức sau có đúng không?

$$L \diagup (L_1 \cap L_2) = (L \diagup L_1) \cap (L \diagup L_2)$$
$$(L_1 \cap L_2) \diagup L = (L_1 \diagup L) \cap (L_2 \diagup L)$$

Khái niệm về văn phạm

- **Định nghĩa.** Văn phạm hay văn phạm ngữ cấu là một bộ gồm bốn đối tượng

$$G = (\Sigma, V, \sigma, P) \quad \text{trong đó:}$$

  - Σ: Bảng chữ cái chính hay bảng chữ cái từ của văn phạm, phần tử thuộc Σ được gọi là ký hiệu cơ bản.
  - V: Bảng chữ cái phụ hay bảng chữ cái hỗ trợ của văn phạm, phần tử thuộc V được gọi là ký hiệu phụ hay ký hiệu bổ xung.
    Σ hữu hạn, khác rỗng và $\Sigma \cap V = \emptyset$.
    $\Sigma \cup V$ là bảng chữ cái hỗn hợp hay đầy đủ của văn phạm.
  - $\sigma \in V$: ký hiệu xuất phát của văn phạm.
  - $P = \{\varphi \to \psi | \varphi \in (\Sigma \cup V)^+, \psi \in (\Sigma \cup V)^*\}$, $\to \notin (\Sigma \cup V)$: tập quy tắc hay quy tắc thế của văn phạm.

- **Ví dụ.** Cho văn phạm $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c\}, \quad V = \{S, A, B\}, \sigma = S$
  $P = \{S \to AB,\ A \to aA,\ A \to c,\ B \to Bb,\ B \to a,\ Ab \to a,\ a \to c\}$

Vị trí của xâu trong từ

- **Định nghĩa.** Nếu $\omega = t_1\varphi t_2$ thì xâu $t_1^*\varphi^*t_2$ (với $* \notin \Sigma$) được gọi là vị trí của xâu φ trong từ ω.
- Nếu $|\varphi| = 1$ tức là $\varphi = b \in \Sigma$ thì $t_1^*b^*t_2$ được gọi là vị trí của ký hiệu b trong ω. Số vị trí của chữ cái a trong từ ω được kí hiệu là $l_a(\omega)$ hay $|\omega|_a$.
- **Ví dụ.** $\omega = abcbcbca$, $\varphi = bcb$, $\xi = bc$. Khi đó:
  - Có 2 vị trí của φ trong ω là a^*bcb^*cbca và abc^*bcb^*ca.
  - Có 3 vị trí của ξ trong ω là a^*bc^*bcbca, abc^*bc^*bca và $abcbc^*bc^*a$.
  - Có 3 vị trí của kí tự b trong ω là $a^*b^*cbcbca$, abc^*b^*cbca và $abcbc^*b^*ca$.
  - Có 2 vị trí của kí tự a trong ω là $*a^*bcbcbca$ và $abcbcbc^*a^*$.

Dẫn xuất

Giả sử $\varphi \to \psi$ là một quy tắc thế của văn phạm G và $\xi_1^*\varphi^*\xi_2$ là một vị trí của φ trong từ $\alpha = \xi_1\varphi\xi_2$ trên $\Sigma \cup V$.
Khi đó áp dụng quy tắc thế $\varphi \to \psi$ vào vị trí $\xi_1^*\varphi^*\xi_2$ ta được xâu $\beta = \xi_1\psi\xi_2$. Ta nói rằng xâu β dẫn được trực tiếp từ xâu α hay xâu α dẫn trực tiếp ra xâu β trong văn phạm G. Kí hiệu: $\alpha \Rightarrow_G \beta$ hay $\alpha \Rightarrow \beta$.

- **Định nghĩa**: Dãy từ $D = (\omega_0, \omega_1, \ldots, \omega_{m-1}, \omega_m)$ trên bảng chữ cái V' được gọi là một dẫn xuất trong G nếu: $\forall i\ (0 \leqslant i \leqslant m-1)$ thì $\omega_i \Rightarrow \omega_{i+1}$
  - Độ dài dẫn xuất D kí hiệu là $|D|$
  - D được gọi là dẫn xuất đầy đủ nếu $\omega_0 = \sigma, \omega_m \in \Sigma^*$
  - Dãy $(\omega_0', \omega_1', \ldots, \omega_m')$ trong đó ω_i' $(i = 0, 1, \ldots, m-1)$ là vị trí được thế tại bước thứ $i+1$ của dẫn xuất D được gọi là dẫn xuất đánh dấu của dẫn xuất D.
  - Dẫn xuất $D = (\omega_0, \ldots, \omega_m)$ được gọi là không lặp nếu $\omega_i \neq \omega_j (i \neq j)$.
- **Định nghĩa.** Xâu α dẫn được ra xâu β nếu tồn tại dẫn xuất: $D = (\alpha = \omega_0, \omega_1, \ldots, \omega_{m-1}, \omega_m = \beta)$ và kí hiệu là $\alpha \vdash_G \beta$ hay $\alpha \vdash \beta$.

Ngôn ngữ sinh bởi văn phạm

- **Định nghĩa.**
  $L(G) = \{x \in \Sigma^* \mid \sigma \vdash_G x\}$ được gọi là ngôn ngữ sinh bởi văn phạm G.
  Tóm lại, ngôn ngữ sinh bởi văn phạm G tập hợp tất cả các xâu mà σ có thể suy dẫn được.

- **Ví dụ.** Cho văn phạm $G = (\{a, b\}, \{A, B, C\}, A, P)$
  $P = \{A \to BBa, B \to Ba, B \to aB, A \to AC, B \to \wedge, C \to b\}$
  Dễ thấy rằng $A \vdash_G a^2b$, thật vậy, dẫn xuất đầy đủ trong G là:
  $D = (A, AC, BBaC, BaBaC, aBaC, a^2C, a^2b)$
  D có hai dẫn xuất đánh dấu khác nhau là:
  $D_1' = ({}^*A^*, {}^*A^*C, {}^*B^*BaC, {}^*B^*aBaC, a^*B^*aC, aa^*C^*, a^2b)$
  $D_2' = ({}^*A^*, {}^*A^*C, B^*B^*aC, Ba^*B^*aC, {}^*B^*aaC, aa^*C^*, a^2b)$
  Dễ dàng nhận thấy rằng $L(G) = \{a^nb^m | m \geqslant 0, n \geqslant 1\}$.

Phân loại văn phạm

- **Định nghĩa.** Văn phạm $G = (\Sigma, V, \sigma, P)$ được gọi là văn phạm cảm ngữ cảnh nếu mỗi quy tắc thế của nó đều có dạng $\xi_1 A \xi_2 \to \xi_1 \theta \xi_2$, trong đó $\xi_1, \xi_2 \in (\Sigma \cup V)^*, \quad A \in V, \quad \theta \in (\Sigma \cup V)^+$.
- **Ví dụ.** $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c\}, V = \{\sigma, A, B, F\}$
  $P = \{\sigma \to AF, A \to cb, aF \to ac, F \to Bbc, F \to \wedge, \ FBb \to Fcb\}$
- **Định nghĩa.** Văn phạm cảm ngữ cảnh mà mọi quy tắc thế đều có dạng $A \to \alpha$ trong đó $A \in V$ được gọi là văn phạm phi ngữ cảnh.
  Để mở rộng người ta cho phép văn phạm phi ngữ cảnh chứa quy tắc rỗng $A \to \wedge \quad (A \in V)$.
  Từ nay ta hiểu văn phạm phi ngữ cảnh là văn phạm phi ngữ cảnh mở rộng.
- **Ví dụ.** $G = (\{a, b, c\}, \{\sigma, S, T, Q\}, \sigma, P)$ trong đó:
  $P = \{\sigma \to aST, S \to bcT, S \to ab, T \to Qab, Q \to aQ, Q \to \wedge\}$

Văn phạm chính quy

- **Định nghĩa.** Văn phạm phi ngữ cảnh G được goi là văn phạm chính quy nếu mọi quy tắc thế thuộc một trong hai dạng sau:
  $A \rightarrow aB$, $A \rightarrow b$, với A, B là các kí hiệu phụ, a, b là kí hiệu cơ bản.
  Nếu bổ sung quy tắc rỗng thì ta gọi là văn phạm chính quy suy rộng hay văn phạm Saloma.
- **Ví dụ.** $G = (\{a, b, c\}, \{\sigma, A, B\}, \sigma, P\}$ trong đó:
  $P = \{\sigma \rightarrow aA, \sigma \rightarrow bB, A \rightarrow aB, B \rightarrow bA, A \rightarrow a, B \rightarrow cB\}$
- Ngôn ngữ do văn phạm ngữ cấu sinh ra gọi là ngôn ngữ ngữ cấu.
- Ngôn ngữ do văn phạm cảm ngữ cảnh sinh ra gọi là ngôn ngữ cảm ngữ cảnh.
- Ngôn ngữ do văn phạm phi ngữ cảnh sinh ra gọi là ngôn ngữ phi ngữ cảnh.
- Ngôn ngữ do văn phạm chính quy sinh ra gọi là ngôn ngữ chính quy.

Một số khái niệm bổ sung

- Quy tắc thế trung gian là quy tắc thế có dạng $\alpha \to \beta$ trong đó β chứa kí tự bổ sung.
- Quy tắc thế kết thúc là quy tắc thế có dạng $\alpha \to \beta$ trong đó $\beta \in \Sigma^*$.
- Trong dẫn xuất $D = (\omega_0, \omega_1, \ldots, \omega_n)$ thì ω_i được gọi là trạng thái của dẫn xuất D.
- Trạng thái kết thúc là một xâu thuộc $(\Sigma \cup V)^*$ trong một dẫn xuất nào đó mà từ đó có thể thực hiện liên tiếp các quy tắc thế kết thúc để đưa xâu này thành một xâu thuộc Σ^*.
- Trạng thái chết là một xâu thuộc $(\Sigma \cup V)^*$ trong một dẫn xuất nào đó mà từ đó không tồn tại một dẫn xuất để biến xâu này thành xâu trong Σ^*.
- Hai dẫn xuất trong văn phạm G được gọi là các dẫn xuất đồng lực nếu các từ đầu tiên và cuối cùng của chúng trùng nhau một cách tương ứng.
- Hai văn phạm G và G' được gọi là tương đương nếu $L(G) = L(G')$.

Một số tính chất của văn phạm

- **Định lý 1.**
  Đối với mỗi dẫn xuất D tùy ý trong một văn phạm bất kì luôn tồn tại dẫn xuất D' không lặp và đồng lực với nó.
- **Chứng minh.** Giả sử $D = (\omega_1, \omega_2, \ldots, \omega_n)$ là một dẫn xuất bất kỳ trong văn phạm G. Nếu trong D có hai trạng thái giống nhau là ω_j và ω_k với $j < k$ thì ta có dẫn xuất $D^{(1)} = (\omega_1, \omega_2, \ldots, \omega_j, \omega_{k+1}, \omega_{k+2}, \ldots, \omega_n)$ đồng lực với dẫn xuất D. Quá trình này tiếp tục cho đến khi ta được dẫn xuất $D^{(m)}$ không có hai trạng thái giống nhau, tức là ta có dẫn xuất không lặp đồng lực với D.
- **Định lý 2.**
  Đối với mỗi văn phạm G tùy ý luôn luôn có thể xây dựng văn phạm G' tương đương với nó mà vế trái của tất cả các quy tắc trong G' đều không chứa kí hiệu cơ bản.

Chứng minh

Ta xây dựng văn phạm $G' = (\Sigma, V \cup \overline{\Sigma}, \sigma, \overline{P} \cup \overline{\overline{P}})$ trong đó
$\overline{\Sigma} = \{\overline{a} | a \in \Sigma$ và xuất hiện ở vế trái của phép thế$\}$
($\overline{a}$ là kí hiệu phủ định mới tương ứng với a)
$\overline{P} = \{\overline{a} \rightarrow a | \overline{a} \in \overline{\Sigma}\}$, $\overline{\overline{P}}$ nhận được từ P bằng cách thay tất cả các vị trí của các ký hiệu cơ bản trong tất cả các phép thế bởi kí hiệu phủ định tương ứng(nếu có) .
Bây giờ ta chứng minh $L(G) \subseteq L(G')$ và $L(G') \subseteq L(G)$. Thật vậy, giả sử $\alpha \in L(G)$ khi đó tồn tại dẫn xuất $D = (\sigma = \omega_0, \omega_1, \ldots, \omega_m = \alpha)$
Do mỗi phép thế trong P đều ứng với một phép thế trong $\overline{\overline{P}}$ do đó ta có dẫn xuất tương ứng $D' = (\sigma = \omega'_0, \omega'_1, \ldots, \omega'_m = \alpha')$, trong đó ω'_i được tạo thành bằng cách thay tất cả các ký hiệu cơ bản trong ω_i bởi các kí hiệu phụ tương ứng trong G'. Đến đây xuất phát từ α' ta thực hiện liên tiếp các phép thế trong $\overline{P}$ đưa α' về α. Như vậy ta có dẫn xuất trong G' là
$D'' = (\sigma = \omega'_0, \omega'_1, \ldots, \omega'_m, \omega'_{m+1}, \ldots, \omega'_{m+n} = \alpha)$
Do vậy $\alpha \in L(G')$ và ta có $L(G) \subseteq L(G')$.

Chứng minh

Ngược lại giả sử $\beta \in L(G')$ khi đó tồn tại dẫn xuất
$D' = (\sigma = \omega'_0, \omega'_1, \ldots, \omega'_m = \beta)$
Thay ω_i bởi ω'_i ta có dẫn xuất:
$D = (\sigma = \omega_0, \omega_1, \ldots, \omega_m = \beta)$. Tuy nhiên nếu phép thế được dùng trong dẫn xuất trực tiếp $\omega'_i \Rightarrow \omega'_{i+1}$ nằm trong $\overline{P}$ thì $\omega_i = \omega_{i+1}$ do đó ta coi dẫn xuất trực tiếp $\omega_i \Rightarrow \omega_{i+1}$ là dẫn xuất dừng.
Loại bỏ trong D những dẫn xuất dừng ta được dẫn xuất trong G là:
$D = (\sigma = \omega_0, \omega_1, \ldots, \omega_n = \beta)$. Do vậy, $\beta \in L(G)$ suy ra $L(G') \subseteq L(G)$.

- **Nhận xét.** Từ chứng minh trên ta có thuật toán xây dựng văn phạm G':
  - $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma$ và xuất hiện ở vế trái của phép thế$\}$
    ($\overline{a}$ là kí hiệu phủ định mới tương ứng với a)
  - $\overline{P} = \{\overline{a} \rightarrow a \mid \overline{a} \in \overline{\Sigma}\}$
  - $\overline{\overline{P}}$ nhận được từ P bằng cách thay tất cả các vị trí của các ký hiệu cơ bản trong tất cả các phép thế bởi kí hiệu phủ định tương ứng(nếu có).

Ta được văn phạm $G' = (\Sigma, V \cup \overline{\Sigma}, \sigma, \overline{P} \cup \overline{\overline{P}})$

Ví dụ

- **1.** Cho $G = (\Sigma, V, \sigma, P)$ trong đó: $\Sigma = \{0, 1\},\ V = \{A, B, S\},\ \sigma = S$
  $P = \{S \to 1AB,\ A \to 0A,\ 1B \to 0A,\ A \to 0,\ B \to 1\}$
  Ta thấy 1 ở vế trái của phép thế nên: $\overline{\Sigma} = \{\overline{1}\},\quad \overline{P} = \{\overline{1} \to 1\}$
  $\overline{\overline{P}} = \{S \to \overline{1}AB,\ A \to 0A,\ \overline{1}B \to 0A,\ A \to 0,\ B \to \overline{1}\}$
  Văn phạm $G' = (\Sigma', V', \sigma', P')$ xây dựng như sau:
  $\Sigma' = \Sigma = \{0, 1\},\ V' = \{A, B, S, \overline{1}\},\ \sigma' = S$
  $P' = \{S \to \overline{1}AB,\ A \to 0A,\ \overline{1}B \to 0A,\ A \to 0,\ B \to \overline{1}\} \cup \{\overline{1} \to 1\}$
- **2.** Cho văn phạm $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c\},\ V = \{A, B, T\},\ \sigma = T$
  $P = \{T \to AB,\ A \to aA,\ Ba \to bB,\ bB \to c\}$
  Ta thấy a, b nằm ở vế trái của phép thế nên:
  $\overline{\Sigma} = \{\overline{a}, \overline{b}\},\quad \overline{P} = \{\overline{a} \to a,\ \overline{b} \to b\}$
  $\overline{\overline{P}} = \{T \to AB,\ A \to \overline{a}A,\ B\overline{a} \to \overline{b}B,\ \overline{b}B \to c\}$
  Văn phạm $G' = (\Sigma', V', \sigma', P')$ được xây dựng như sau:
  $\Sigma' = \Sigma = \{a, b, c\},\ V' = \{A, B, T, \overline{a}, \overline{b}\},\ \sigma' = T$
  $P' = \{T \to AB,\ A \to \overline{a}A,\ B\overline{a} \to \overline{b}B,\ \overline{b}B \to c\} \cup \{\overline{a} \to a,\ \overline{b} \to b\}$

Một số tính chất của văn phạm

Giả sử $\mathcal{G}$ là một lớp văn phạm nào đó, $\mathcal{L}(\mathcal{G})$ là lớp ngôn ngữ do tất cả các văn phạm thuộc $\mathcal{G}$ sinh ra, ϕ là phép toán k-ngôi trên lớp các ngôn ngữ. Ta nói rằng lớp ngôn ngữ $\mathcal{L}(\mathcal{G})$ đóng đối với phép toán ϕ nếu với k văn phạm tùy ý $G_1, G_2, \ldots, G_k \in \mathcal{G}$ có thể xây dựng được văn phạm $G \in \mathcal{G}$ sao cho $L(G) = \phi(L(G_1), L(G_2), \ldots, L(G_k))$

Định lí 3: Lớp ngôn ngữ được sinh ra bởi tất cả các văn phạm đóng đối với các phép toán: hợp, giao, tích ghép, lặp, lặp cắt, soi gương, chia trái, chia phải.

Chứng minh tính đóng với phép hợp

- Giả sử $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$, $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ là hai văn phạm
- Xây dựng văn phạm G sinh ra ngôn ngữ $L(G_1) \cup L(G_2)$ như sau:
  - Nếu $V_1 \cap V_2 \neq \emptyset$ thì đổi ký hiệu của V_1 hoặc V_2 để $V_1 \cap V_2 = \emptyset$.
  - Biến đổi G_1, G_2 thành hai văn phạm không chứa ký hiệu cơ bản ở vế trái.
  - Xác định $G = (\Sigma, V, \sigma, P)$ trong đó :
    $\Sigma = \Sigma_1 \cup \Sigma_2$
    $V = V_1 \cup V_2 \cup \{\sigma\}$
    $P = P_1 \cup P_2 \cup \{\sigma \to \sigma_1, \sigma \to \sigma_2\}$

Tính đóng với phép hợp- Ví dụ

$G_1 = (\Sigma_1, V_1, \sigma_1, P_1), \quad \Sigma_1 = \{a, b\}, \quad V_1 = \{\sigma_1, A, B\}$
$P_1 = \{\sigma_1 \to a\sigma_1 bb, \sigma_1 \to Ab, A \to aB, B \to a, a \to b\}$
$G_2 = (\Sigma_2, V_2, \sigma_2, P_2)\ , \quad \Sigma_1 = \{a, b\}, \quad V_2 = \{\sigma_2, A, C\}$
$P_2 = \{\sigma_2 \to a\sigma_2 b, \sigma_2 \to \wedge, \sigma_2 \to Aa, A \to b, C \to aA\}$

- Do $V_1 \cap V_2 = \{A\} \neq \emptyset$ nên đổi A trong V_1 thành A_1, đổi A trong V_2 thành A_2, khi đó P_1 và P_2 thay đổi thành:
  $P_1 = \{\sigma_1 \to a\sigma_1 bb, \sigma_1 \to A_1 b, A_1 \to aB, B \to a, a \to b\}$
  $P_2 = \{\sigma_2 \to a\sigma_2 b, \sigma_2 \to \wedge, \sigma_2 \to A_2 a, A_2 \to b, C \to aA_2\}$
- Ta thấy a ở vế trái của phép thế trong P_1 nên ta bổ xung $\overline{a}$ vào P_1 và biến đổi P_1 thành:
  $P_1 = \{\sigma_1 \to \overline{a}\sigma_1 bb, \sigma_1 \to A_1 b, A_1 \to \overline{a}B, B \to \overline{a}, \overline{a} \to b, \overline{a} \to a\}$
- Tóm lại ta có $G = (\Sigma, V, \sigma, P)$ với $\Sigma = \{a, b\}$,
  $V = \{\sigma, \sigma_1, \sigma_2, A_1, A_2, B, C, \overline{a}\}$
  $P = \{\sigma \to \sigma_1, \sigma \to \sigma_2, \sigma_2 \to a\sigma_2 b, \sigma_2 \to \wedge, \sigma_2 \to A_2 a, A_2 \to b$
  $C \to aA_2, \sigma_1 \to A_1 b, A_1 \to \overline{a}B, B \to \overline{a}, \overline{a} \to b, \overline{a} \to a\}$

Chứng minh tính đóng với phép giao

- Giả sử $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$, $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ là hai văn phạm
- Xây dựng văn phạm G sinh ra ngôn ngữ $L(G_1) \cap L(G_2)$
  - Nếu $V_1 \cap V_2 \neq \emptyset$ thì đổi ký hiệu của V_1 hoặc V_2 để $V_1 \cap V_2 = \emptyset$.
  - Loại bỏ ký hiệu cơ bản ở vế trái của mỗi phép thế trong P_1 và P_2.
  - Đặt tương ứng mỗi kí hiệu cơ bản $a \in \Sigma_1$ với kí hiệu phủ định $\overline{a}$ và mọi kí hiệu cơ bản $b \in \Sigma_2$ với kí hiệu phủ định kép $\overline{\overline{b}}$.
    $\overline{\Sigma_1} = \{\overline{a} \mid a \in \Sigma_1\}, \overline{\overline{\Sigma_2}} = \{\overline{\overline{b}} \mid b \in \Sigma_2\}$. Chú ý rằng $\overline{\Sigma_1} \cap \overline{\overline{\Sigma_2}} = \emptyset$
    - $\overline{P_1}$ là tập các quy tắc nhận được từ P_1 bằng cách thay ký hiệu cơ bản trong Σ_1 bằng ký hiệu phủ định tương ứng.
    - $\overline{P_2}$ là tập các quy tắc nhận được từ P_2 bằng cách thay kí hiệu cơ bản trong Σ_2 bằng kí hiệu phủ định kép tương ứng.
  - $P' = \{\overline{a}\overline{\overline{b}} \to \overline{\overline{b}}\overline{a} \mid a \in \Sigma_1, b \in \Sigma_2\}$, $\quad P'' = \{\overline{a}\,\overline{\overline{a}} \to a \mid a \in \Sigma_1 \cap \Sigma_2\}$
- Văn phạm giao là $G = (\Sigma, V, \sigma, P)$ với: $\Sigma = \Sigma_1 \cap \Sigma_2$
  $V = V_1 \cup V_2 \cup \overline{\Sigma_1} \cup \overline{\overline{\Sigma_2}} \cup \{\sigma\}$, $P = \overline{P_1} \cup \overline{P_2} \cup P' \cup P'' \cup \{\sigma \to \sigma_1\sigma_2\}$

Tính đóng của phép giao - Ví dụ

$$G_1 = (\Sigma_1, V_1, \sigma_1, P_1), \quad \Sigma_1 = \{a, b\}, \quad V_1 = \{\sigma_1, A\}$$
$$P_1 = \{\sigma_1 \to a\sigma_1 bb, \sigma_1 \to a, \sigma_1 \to aA, A \to b\}$$
$$G_2 = (\Sigma_2, V_2, \sigma_2, P_2)\ , \quad \Sigma_1 = \{a, b\}, \quad V_2 = \{\sigma_2, A\}$$
$$P_2 = \{\sigma_2 \to a\sigma_2 b, \sigma_2 \to \wedge, \sigma_2 \to bA, A \to ab\}$$

- Do $V_1 \cap V_2 = \{A\} \neq \emptyset$ nên đổi A trong V_1 thành A_1, đổi A trong V_2 thành A_2, khi đó P_1 và P_2 thay đổi thành:
  $$P_1 = \{\sigma_1 \to a\sigma_1 bb, \sigma_1 \to a, \sigma_1 \to aA_1, A_1 \to b\}$$
  $$P_2 = \{\sigma_2 \to a\sigma_2 b, \sigma_2 \to \wedge, \sigma_2 \to bA_2, A_2 \to ab\}$$
- Văn phạm G sinh ra ngôn ngữ $L(G_1) \cap L(G_2)$ là $G = (\Sigma, V, \sigma, P)$:
  $$\Sigma = \{a, b\}$$
  $$V = \{\sigma, \sigma_1, \sigma_2, \overline{a}, \overline{\overline{a}}, \overline{b}, \overline{\overline{b}}\}$$
  $$P = \{\sigma \to \sigma_1\sigma_2, \sigma_1 \to \overline{a}\sigma_1\overline{b}\,\overline{b}, \sigma_1 \to \overline{a}, , \sigma_1 \to \overline{a}A_1, A_1 \to \overline{b}, \sigma_2 \to \overline{\overline{a}}\sigma_2\overline{\overline{b}}$$
  $$\sigma_2 \to \wedge, \sigma_2 \to \overline{\overline{b}}A_2, A_2 \to \overline{\overline{a}}\overline{\overline{b}}, \overline{a}\overline{\overline{b}} \to \overline{\overline{b}}\overline{a}, \overline{a}\,\overline{\overline{a}} \to a, \overline{b}\,\overline{\overline{b}} \to b, \overline{b}\overline{\overline{a}} \to \overline{\overline{a}}\overline{b}$$
  $$\overline{a}\,\overline{\overline{a}} \to \overline{\overline{a}}\,\overline{a}, \overline{b}\,\overline{\overline{b}} \to \overline{\overline{b}}\,\overline{b}\}$$

Đóng với phép nhân (tích ghép)

Giả sử $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ và $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ là hai văn phạm bất kỳ. Khi đó văn phạm $G = (\Sigma, V, \sigma, P)$ sinh ngôn ngữ $L(G_1).L(G_2)$ được xây dựng như sau:

- Nếu $V_1 \cap V_2 \neq \emptyset$ thì đổi ký hiệu của V_1 hoặc V_2 để $V_1 \cap V_2 = \emptyset$.
- Biến đổi G_1 và G_2 thành hai văn phạm không chứa ký hiệu cơ bản ở vế trái trong mỗi phép thế (theo định lý 2).
- Các thành phần của G là:
  $\Sigma = \Sigma_1 \cup \Sigma_2$
  $V = V_1 \cup V_2 \cup \{\sigma\}$
  $P = P_1 \cup P_2 \cup \{\sigma \to \sigma_1\sigma_2\}$

Đóng với phép nhân-Ví dụ

- Cho hai văn phạm :
  $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ trong đó $\Sigma_1 = \{a, b, c\}, V_1 = \{\sigma, A, B\}$
  $P_1 = \{\sigma \to AB, A \to a^2A, AB \to bcBc, B \to \wedge\}$
  $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ trong đó $\Sigma_2 = \{a, b, c\}, V_2 = \{\sigma_2, C, D\}$
  $P_2 = \{\sigma_2 \to CD, C \to aC, D \to Db, CD \to \wedge, cC \to cacbC\}$
- Ta thấy rằng trong P_2 có c ở vế trái của phép thế nên ta bổ xung $\overline{c}$ vào V_2 và biến đổi P_2 thành:
  $P_2 = \{\sigma_2 \to CD, C \to aC, D \to Db, CD \to \wedge, \overline{c}C \to \overline{c}a\overline{c}bC, \overline{c} \to c\}$
- Khi đó văn phạm sinh ra ngôn ngữ $L(G_1)L(G_2)$ là $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c\}$
  $V = \{A, B, C, D, \sigma, \sigma_1, \sigma_2, \overline{c}\}$
  $P = \{\sigma \to \sigma_1\sigma_2, \sigma_1 \to AB, A \to a^2A, AB \to bcBc, B \to \wedge, \sigma_2 \to CD$
  $C \to aC, D \to Db, CD \to \wedge, \overline{c}C \to \overline{c}a\overline{c}bC, \overline{c} \to c\}$

Đóng với phép lặp và lặp cắt

Giả sử $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ là văn phạm tùy ý.

- Văn phạm $G = (\Sigma, V, \sigma, P)$ sinh ra ngôn ngữ $(L(G_1))^+$ được xây dựng như sau:
  - Biến đổi G_1 thành văn phạm không có ký hiệu cơ bản ở vế trái của mỗi phép thế (theo định lý 2).
  - Các thành phần của G là;
    $\Sigma = \Sigma_1,\ V = V_1,\ \sigma = \sigma_1$
    $P = P_1 \cup \{\sigma_1 \to \sigma_1\sigma_1\}$
- Muốn sinh ra ngôn ngữ $(L(G_1))^*$ ta bổ xung vào P quy tắc $\sigma_1 \to \wedge$
- **Ví dụ.** Cho văn phạm $G = (\{a, b, c\}, \{\sigma, A, B\}, \sigma, P)$ trong đó:

  $$P = \{\sigma \to BaA,\ A \to bA,\ B \to acA,\ cB \to Ab, B \to a\}$$

  Văn phạm sinh ngôn ngữ $L(G)^+$ là:

  $$G' = (\{a, b, c\}, \{\sigma, A, B, \overline{c}\}, \sigma, P') \text{ trong đó:}$$

  $$P' = \{\sigma \to \sigma\sigma,\ \sigma \to BaA,\ A \to bA,\ B \to a\overline{c}A,\ \overline{c}B \to Ab,$$
  $$B \to a,\ \overline{c} \to c\}$$

Đóng với phép soi gương

Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm tùy ý.

- Văn phạm sinh ra ngôn ngữ soi gương của ngôn ngữ $L(G)$ là:
  $G' = (\Sigma', V', \sigma', P')$ trong đó
  $\Sigma' = \Sigma,\ V' = V,\ \sigma' = \sigma$
  $P' = \{\widetilde{\alpha} \to \widetilde{\beta} \mid (\alpha \to \beta) \in P\}$
- **Ví dụ 1.** Cho văn phạm
  $G = (\{a, b, c\}, \{A, B\}, A, \{A \to AB, A \to aA, B \to Bb, AB \to c\})$
  Văn phạm sinh ngôn ngữ soi gương của ngôn ngữ $L(G)$ là:
  $G = (\{a, b, c\}, \{A, B\}, A, \{A \to BA, A \to Aa, B \to bB, BA \to c\})$
- **Ví dụ 2.** Cho văn phạm
  $G = (\{0, 1, 2\}, \{S, A\}, S, \{S \to A10, A \to 0A, S \to 2S1, AS1 \to 102\})$
  Văn phạm sinh ngôn ngữ soi gương của ngôn ngữ $L(G)$ là:
  $G = (\{0, 1, 2\}, \{S, A\}, S, \{S \to 01A, A \to A0, S \to 1S2, 1SA \to 201\})$

Tính đóng với phép chia

Giả sử $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ và $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ là hai văn phạm tùy ý.

- Văn phạm sinh ngôn ngữ thương bên phải của $L(G_1)$ cho $L(G_2)$ được xây dựng như sau:
  - Nếu $V_1 \cap V_2 \neq \emptyset$ thì đổi ký hiệu của V_1 hoặc V_2 để $V_1 \cap V_2 = \emptyset$.
  - Biến đổi G_1 và G_2 thành hai văn phạm không chứa ký hiệu cơ bản ở vế trái trong mỗi phép thế (theo định lý 2).
  - Đặt $\overline{\Sigma_2} = \{\overline{a} | a \in \Sigma_2\}$, $V' = \{a\overline{a} \to \wedge | a \in \Sigma_2\}$ và $\widetilde{P_2}$ có được từ quy tắc trong P_2 bằng cách thay ký hiệu cơ bản bằng ký hiệu phủ định tương ứng sau đó thay tất cả các phép thế có dạng $\alpha \to \beta$ thành $\tilde{\alpha} \to \tilde{\beta}$.
  - Văn phạm sinh ngôn ngữ $L(G_1)/L(G_2)$ là $G = (\Sigma, V, \sigma, P)$ với:
    $\Sigma = \Sigma_1 \cup \Sigma_2,\ V = V_1 \cup V_2 \cup \{\sigma\} \cup \overline{\Sigma_2}$
    $P = P_1 \cup \widetilde{P_2} \cup \{\sigma \to \sigma_1\sigma_2\} \cup V'$
- Văn phạm sinh ra ngôn ngữ thương bên trái của $L(G_1)$ cho $L(G_2)$ làm tương tự (coi như bài tập).

Tính đóng của phép chia - Ví dụ

Cho hai văn phạm:

$G_1 = (\Sigma_1, V_1, \sigma_1, P_1),\ \Sigma_1 = \{a, b\}, V_1 = \{\sigma_1, A\}$

$P_1 = \{\sigma_1 \to aA, A \to bA, A \to a\}$

$G_2 = (\Sigma_2, V_2, \sigma_2, P_2),\ \Sigma_2 = \{a, b\}, V_2 = \{\sigma_2, C, D\},$

$P_2 = \{\sigma_2 \to CD, C \to bC, D \to aD, C \to \wedge, D \to a\}$

- Văn phạm sinh ra $L(G_1)/L(G_2)$ là $G = (\Sigma, V, \sigma, P)$

  $\Sigma = \{a, b\}, V = \{\sigma, \sigma_1, \sigma_2, A, C, D, \overline{a}, \overline{b}\}$

  $P = \{\sigma \to \sigma_1\sigma_2, \sigma_1 \to aA, A \to bA, A \to a, \sigma_2 \to DC, C \to C\overline{b}$

  $D \to D\overline{a}, C \to \wedge, D \to \overline{a}, a\overline{a} \to \wedge, b\overline{b} \to \wedge\}$

- Văn phạm sinh ra ngôn ngữ $L(G_1)\backslash L(G_2)$ là $G' = (\Sigma', V', \sigma', P')$ trong đó:

  $\Sigma' = \{a, b\}, V' = \{\sigma, \sigma_1, \sigma_2, A, C, D, \overline{a}, \overline{b}\}$

  $P' = \{\sigma \to \sigma_1\sigma_2, \sigma_1 \to A\overline{a}, A \to A\overline{b}, A \to \overline{a}, \sigma_2 \to CD, C \to bC$

  $D \to aD, C \to \wedge, D \to a, \overline{a}a \to \wedge, \overline{b}b \to \wedge\}$

Một số ví dụ về văn phạm - Ví dụ 1

- Cho $\Sigma = \{a_1, a_2, \ldots, a_n\}$
  - Cho văn phạm $G = (\Sigma, \{\sigma\}, \sigma, P)$ trong đó $P = \{\sigma \to a_i\sigma, \sigma \to a_i\}$. Dễ thấy $L(G) = \Sigma^+$.
    Muốn G sinh ra Σ^* chỉ cần đổi P thành $P' = \{\sigma \to a_i\sigma, \sigma \to \wedge\}$
  - Văn phạm chính quy suy rộng $G = (\Sigma, \{\sigma_2, \sigma_4\}, \sigma_2, P)$ trong đó $P = \{\sigma_2 \to a_i\sigma_4, \sigma_4 \to a_i\sigma_2, \sigma_2 \to \wedge\}$ sinh ra ngôn ngữ $L(G) = \{\omega \in \Sigma^* | l(\omega) \text{ chẵn}\}$.
  - Văn phạm $G_1 = (\Sigma, \{\sigma_1, \sigma_3\}, \sigma_1, P_1)$ trong đó $P_l = \{\sigma_1 \to a_i\sigma_3, \sigma_3 \to a_i\sigma_1, \sigma_3 \to \wedge\}$ sinh ra ngôn ngữ gì?
  - Nếu mỗi số tự nhiên ta viết thành $n = \underbrace{||\ldots|}_{\text{n lần}}$ thì tất cả các số tự nhiên chia 3 dư 2 được sinh bởi văn phạm chính quy:
    $G = (\{|\}, \{\sigma, \sigma_1, \sigma_2, A\}, \sigma, P)$ trong đó
    $P = \{\sigma \to |\sigma_1, \sigma_1 \to |\sigma_2, \sigma_2 \to |\sigma, \sigma \to |A, A \to |\}$

Ví dụ 2

- Lập văn phạm sinh ra ngôn ngữ $\{a^{n+1}b^{n+2} | n \in \mathbb{N}\}$
  Ta có bốn cách phân tích xâu $a^{n+1}b^{n+2}$ như sau:
  $a^{n+1}b^{n+2} = a^n ab^2 b^n = aa^n b^n b^2 = aa^n b^2 b^n = a^n ab^n b^2$
  Từ bốn cách phân tích trên ta có bốn văn phạm tương ứng:
  $G_1 = (\{a,b\},\{\sigma\},\sigma,P_1), \quad P_1 = \{\sigma \to a\sigma b,\ \sigma \to ab^2\}$
  $G_2 = (\{a,b\},\{\sigma,\sigma_1\},\sigma,P_2), \ P_2 = \{\sigma \to a\sigma_1 b^2,\ \sigma_1 \to a\sigma_1 b,\ \sigma_1 \to \wedge\}$
  $G_3 = (\{a,b\},\{\sigma,\sigma_1\},\sigma,P_3), \quad P_3 = \{\sigma \to a\sigma_1,\ \sigma_1 \to a\sigma_1 b,\ \sigma_1 \to b^2\}$
  $G_4 = (\{a,b\},\{\sigma,\sigma_1\},\sigma,P_4), \quad P_4 = \{\sigma \to \sigma_1 b^2,\ \sigma_1 \to a\sigma_1 b,\ \sigma_1 \to a\}$
- **Nhận xét.**
  - Một ngôn ngữ có thể được sinh ra bởi nhiều văn phạm khác nhau.
  - Trong ví dụ trên văn phạm G_1 là gọn nhất, điều này có được là do trong phân tích xâu $a^{n+1}b^{n+2}$ ta để phần lẻ ab^2 vào bên trong cụm đối xứng $a^n b^n$.
  - Cụm đối xứng tổng quát $a^{kn}b^{pn}$ với k,p cố định được sinh bởi văn phạm sau:
    $G = (\{a,b\},\{\sigma\},\sigma,P), \quad P = \{\sigma \to a^k\sigma b^p,\ \sigma \to \wedge\}$

Ví dụ 3

- Hãy xây dựng văn phạm phi ngữ cảnh sinh các ngôn ngữ sau:
  $L = \{a^{2n+1}c^{t+1}d^{2t+1}b^{n+2}c^{s+2}f^{p+1}a^{3s+2} | n, t, s, p \in \mathbb{N}\}$
  - **Phân tích**: Xâu $x \in L_1$ có dạng $a^{2n}ac^tcdd^{2t}b^2b^nc^sc^2f^pfa^2a^{3s}$
    có 4 cụm đối xứng là : (a^{2n}, b^n), (c^t, d^{2t}), (c^s, a^{3s}), (f^p)
  - **Quy tắc** : cụm bên ngoài được sinh trước, hai cụm cạnh nhau thì sinh cùng nhau theo đúng thứ tự.
  - Văn phạm là: $G = (\{a, b, c, d, f\}, \{\sigma, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}, \sigma, P)$ P gồm:
    $\sigma \rightarrow \sigma_1\sigma_3$ (để sinh hai cụm liền nhau (a^{2n}, b^n) và (c^s, a^{3s})
    $\sigma_1 \rightarrow a^2\sigma_1 b$ (lặp để sinh cụm đối xứng (a^{2n}, b^n)).
    $\sigma_1 \rightarrow a\sigma_2 b^2$ (chuyển sang σ_2 để chuẩn bị sinh cụm (c^t, d^{2t}).
    $\sigma_2 \rightarrow c\sigma_2 d^2$ (lặp để sinh cụm đối xứng (c^t, d^{2t})).
    $\sigma_2 \rightarrow cd$ (kết thúc việc sinh phần đầu thuộc cụm (a^{2n}, b^n)).
    $\sigma_3 \rightarrow c\sigma_3 a^3$ (lặp để sinh cụm đối xứng (a^s, a^{3s})).
    $\sigma_3 \rightarrow c^2\sigma_4 a^2$ (chuyển sang σ_4 để chuẩn bị sinh cụm f^p)
    $\sigma_4 \rightarrow f\sigma_4$, $\sigma_4 \rightarrow f$ (lặp để sinh cụm f^p và kết thúc sinh cụm f^p)

Ví dụ 4

- Cho bảng chữ cái $\Sigma = \{a, b, c, d, f\}$. Hãy xây dựng văn phạm phi ngữ cảnh sinh các ngôn ngữ sau trên bảng chữ cái Σ:
  - $L_1 = \{x \in \Sigma^* | x$ bắt đầu bằng từ con $abbb$,
    kết thúc bằng ccd chứa từ con $abbcd$ và x có độ dài chẵn $\}$
  - $L_2 = \{x \in \Sigma^* | x$ bắt đầu bằng từ con ab,
    kết thúc bằng cd, chứa từ con bc và x có độ dài chẵn $\}$
- Giải:
  - **Phân tích**:
    $x \in L_1$ suy ra $x = abbby_1abbcdz_1ccd$ hoặc $x = abbby_2abbcdz_2ccd$
    trong đó y_1, z_1 có độ dài lẻ y_2, z_2 có độ dài chẵn.
  - Từ đây ta có: $G = (\Sigma, \{\sigma, \sigma_1, \sigma_2, \sigma_3, \sigma_4\}, \sigma, P)$ với P gồm:
    $\sigma \to abbb\sigma_1abbcd\sigma_1ccd, \quad \sigma \to abbb\sigma_2abbcd\sigma_2ccd$
    $\sigma_1 \to a\sigma_3 | \; b\sigma_3 | \; c\sigma_3 | \; d\sigma_3 | \; f\sigma_3 |$
    $\sigma_3 \to a\sigma_1 | \; b\sigma_1 | \; c\sigma_1 | \; d\sigma_1 | \; f\sigma_1 | \wedge$
    $\sigma_2 \to a\sigma_4 | \; b\sigma_4 | \; c\sigma_4 | \; d\sigma_4 | \; f\sigma_4 | \wedge$
    $\sigma_4 \to a\sigma_2 | \; b\sigma_2 | \; c\sigma_2 | \; d\sigma_2 | \; f\sigma_2 |$

Ví dụ 5

- Cho văn phạm $G = (\Sigma, V, S, P)$ trong đó:
  $\Sigma = \{a_1, a_2, b\}$
  $V = \{S, A_1, A_2\}$
  $P = \{S \to \sigma A_1 a_1,\ S \to SA_2a_2, S \to b,\ a_1A_1 \to A_1a_1, a_1A_2 \to A_2a_1$
  $a_2A_1 \to A_1a_2, a_2A_2 \to A_2a_2, bA_1 \to a_1b, bA_2 \to a_2b\}$
  Có thể chứng minh được $L(G) = \{\omega b\omega \mid \omega \in \{a_1, a_2\}^*\}$.
  Dẫn xuất sinh ra xâu $a_1a_2a_2ba_1a_2a_2$ là:

$$D = (\mathbf{S},\ \mathbf{S}A_2a_2,\ \mathbf{S}A_2a_2A_2a_2,\ \mathbf{S}A_1a_1A_2a_2A_2a_2,\ bA_1\mathbf{a_1A_2}a_2A_2a_2,$$

$$bA_1A_2a_1\mathbf{a_2A_2}a_2,\ bA_1A_2\mathbf{a_1A_2}a_2a_2,\ \mathbf{bA_1}A_2A_2a_1a_2a_2,$$

$$a_1\mathbf{bA_2}A_2a_1a_2a_2,\ a_1a_2\mathbf{bA_2}a_1a_2a_2,\ a_1a_2a_2ba_1a_2a_2)$$

Ví dụ 6

- Cho văn phạm $G = (\Sigma, V, S, P)$ trong đó:
  $\Sigma = \{a, b\}$
  $V = \{S, A, B, C, D\}$
  $P = \{S \to ABC, AB \to iADi, Dij \to jDi, DiC \to BiC$
  $iB \to Bi, AB \to \wedge, C \to \wedge\}$ với $i, j \in \{a, b\}$
  Có thể chứng minh được $L(G) = \{\omega\omega \mid \omega \in \{a, b\}^*\}$
  Dẫn xuất sinh ra xâu $aabaab$ là:

$$D = (\mathbf{S},\ \mathbf{AB}C,\ aA\mathbf{DaC},\ a\mathbf{AB}aC,\ aaA\mathbf{Daa}C,$$

$$aaAa\mathbf{DaC},\ aaA\mathbf{aB}aC,\ aa\mathbf{AB}aaC,\ aabA\mathbf{Dba}aC,$$

$$aabAa\mathbf{Dba}C,\ aabAaa\mathbf{DbC},\ aabAa\mathbf{aB}bC,\ aabA\mathbf{aB}abC,$$

$$aab\mathbf{AB}aabC,\ aabaab\mathbf{C},\ aabaab).$$

Ví dụ 7

Cho văn phạm $G = (\Sigma, V, \sigma, P)$ trong đó

$\Sigma = \{|\}$

$V = \{\sigma, A, B, C, D, E, F\}$

$P = \{\sigma \to ABCDF$

$BD \to DCB$

$BC \to CB$

$AD \to AAE$

$EC \to AE$

$EB \to BE$

$EF \to BBDF$

$DF \to |, B \to |, A \to |, \sigma \to |\}$

Biết rằng $L(G) = \{|^{n^2}, n \in \mathbb{Z}^+\}$. Phân tích các quy tắc trong văn phạm trên, từ đó xây dựng văn phạm G_1 và G_2 sinh ra ngôn ngữ:

$L_1 = \{|^{2n^2+n+1}\}$ và $L_2 = \{|^{n^3}\}$

Bài tập chương 1

- **1.** Loại bỏ ký hiệu cơ bản ở vế trái của các phép thế trong văn phạm sau:

  a. $G_1 = (\{a, b, c, d\}, \{\sigma, A, B, F\}, \sigma, P)$

  $P = \{\sigma \to AcF,\ bA \to Aa,\ cB \to bc,\ aF \to ab,$
  $cF \to ad,\ aB \to bF\}$

  b. $G_2 = (\{a, b, c\}, \{\sigma, A, B, C\}, \sigma, P)$

  $P = \{\sigma \to Ab,\ bA \to AaC,\ AcB \to bc,\ Bc \to cb,$
  $c \to aC,\ bbC \to a\}$

  c. $G_3 = (\{0, 1, 2\}, \{\sigma, S, A, B\}, \sigma, P)$

  $P = \{\sigma \to 1ASB,\ 0B1 \to A00,\ A2B \to 01,\ B2 \to 102,$
  $2B \to 1,\ 2S1 \to 0\}$

- **2.** Tìm ngôn ngữ sinh ra bởi các văn phạm sau:

  a. $G = (\{a, b, c\}, \{\sigma, A, B\}, \sigma, P)$

  $P = \{\sigma \to AB,\ A \to aA,\ B \to Bb,\ AB \to c\}$

  b. $(\{0, 1, 2\}, \{S, T, K\}, S, P)$

  $P = \{S \to SS,\ S \to TK,\ T \to 10K,\ K \to 02T,\ KK \to 1\}$

Bài tập chương 1

- **3.** Cho hai văn phạm:

  $G_1 = (\{a, b, c\}, \{\sigma_1, A, B\}, \sigma_1, P_1)$

  $P_1 = \{\sigma_1 \rightarrow AB,\ A \rightarrow bA,\ B \rightarrow Ab,\ B \rightarrow cB,\ B \rightarrow BA,$
  $\qquad A \rightarrow a,\ B \rightarrow c\}$

  $G_2 = (\{a, b, c\}, \{\sigma_2, A, B\}, \sigma_2, P_2)$

  $P_2 = \{\sigma_2 \rightarrow a\sigma_2,\ \sigma_2 \rightarrow aA,\ A \rightarrow bA,\ B \rightarrow aB,\ B \rightarrow c,$
  $\qquad B \rightarrow bA,\ A \rightarrow ab\}$

  a. Tìm văn phạm sinh ra ngôn ngữ $L(G_1) \cup L(G_2)$.

  b. Tìm văn phạm sinh ra ngôn ngữ $L(G_1).L(G_2)$.

  c. Tìm văn phạm sinh ra ngôn ngữ $L(G_1)^*$.

  d. Tìm văn phạm sinh ra ngôn ngữ $L(G_1) \cap L(G_2)$.

  e. Tìm văn phạm sinh ra ngôn ngữ $\widetilde{L(G_2)}$.

  f. Tìm văn phạm sinh ra ngôn ngữ $L(G_1) \setminus L(G_2)$.

  g. Tìm văn phạm sinh ra ngôn ngữ $L(G_1) / L(G_2)$.

Bài tập chương 1

- **4.** Tìm văn phạm sinh ra ngôn ngữ sau:

  a. $L_1 = \{a^n b^n c^k,\ a^s b^k c^s \mid m, n, k \in \mathbb{N}^*\}$

  b. $L_2 = \{a^{n+1} b^{m+2} c d^{2m+1} c^n d^{2k+3} a^s b^k \mid m, n, k, s \in \mathbb{N}\}$

  c. $L_3 = \{x \in \{a, b, c\}^* \mid x$ bắt đầu bằng từ bc, chứa từ ac và kết thúc bằng từ $ba\}$

  d. $L_4 = \{x \in \{a, b, c\}^* \mid x$ bắt đầu bằng từ abc, chứa từ ca và kết thúc bằng từ bc và x có độ dài chẵn. $\}$

  e. $L_5 = \{x \in \{a, b, c\}^* \mid x$ không chứa từ bac và độ dài chia hết cho 3$\}$

  f. $L_6 = \{x \in \{a, b\}^* \mid l_a(x) = l_b(x)\}$

  g. $L_7 = \{a^n b^n c^n \mid n \in \mathbb{N}^*\}$

Bài tập chương 1

- **5.** Cho văn phạm $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a_1, a_2, b\}$
  $V = \{\sigma, A_1, A_2\}$
  $P = \{\sigma \to \sigma A_1 a_1,\ \sigma \to \sigma A_2 a_2, \sigma \to b,\ a_1 A_1 \to A_1 a_1, a_1 A_2 \to A_2 a_1$
  $a_2 A_1 \to A_1 a_2, a_2 A_2 \to A_2 a_2, b A_1 \to a_1 b, b A_2 \to a_2 b\}$
  Tìm dẫn xuất sinh ra xâu: $a_2 a_1 a_2 b a_2 a_1 a_2$
- **6.** Cho văn phạm $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b\}$
  $V = \{\sigma, A, B, C, D\}$
  $P = \{\sigma \to ABC, AB \to iADi, Dij \to jDi, DiC \to BiC$
  $iB \to Bi, AB \to \wedge, C \to \wedge\}$ với $i, j \in \{a, b\}$
  Tìm dẫn xuất sinh ra xâu $abbabb$

Chương 2 - Giới thiệu chung

Như ta đã biết trong văn phạm không phải ký hiệu hay phép thế nào cũng đóng vai trò sinh ra ngôn ngữ. Do vậy phần đầu của chương này sẽ đề cập đến vấn đề giản lược văn phạm phi ngữ cảnh, tức là loại bỏ các ký hiệu, phép thế không cần thiết nhưng vẫn giữ nguyên ngôn ngữ mà nó sinh ra. Cụ thể ta có 3 loại giản lược sau:

- Giản lược các ký hiệu thừa
- Giản lược quy tắc rỗng
- Giản lược quy tắc đơn

Phần hai của chương này sẽ trình bày thuật toán đưa văn phạm phi ngữ cảnh về những văn phạm đặc biệt sau:

- Dạng chuẩn Chomsky
- Văn phạm không đệ quy trái
- Dạng chuẩn Greiback

Phần cuối chương trình bày điều kiện cần và đủ của ngôn ngữ phi ngữ cảnh và thuật toán tìm cây suy dẫn ra xâu sinh bởi văn phạm.

Kí hiệu có ích và kí hiệu thừa

- **Định nghĩa.**
  Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$
  - $X \in \Sigma \cup V$ được gọi là kí hiệu có ích nếu tồn tại suy dẫn $\sigma \vdash \alpha X \beta \vdash \omega$, trong đó $\alpha, \beta \in (\Sigma \cup V)^*$, $\omega \in \Sigma^*$.
  - Nếu X không phải là ký hiệu có ích thì X được gọi là kí hiệu thừa.
- **Định nghĩa.**
  - Kí hiệu $X \in \Sigma \cup V$ được gọi là kí hiệu vô sinh nếu không tồn tại dẫn xuất $X \vdash \omega \in \Sigma^*$.
  - Kí hiệu $X \in \Sigma \cup V$ được gọi là kí hiệu không đến được nếu không tồn tại dẫn xuất $\sigma \vdash \alpha X \beta$, $\alpha, \beta \in (\Sigma \cup V)^*$.
- **Nhận xét.** Nếu X là kí hiệu vô sinh hoặc kí hiệu không đến được thì X là kí hiệu thừa. (Điều ngược lại có đúng không?)

Loại kí hiệu vô sinh

- **Định lý 1.** Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ với $L(G) \neq \emptyset$. Ta có thể xây dựng văn phạm phi ngữ cảnh $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ tương đương với G sao cho $\forall A \in V_1$ không phải là kí hiệu vô sinh.
- **Chứng minh.** Văn phạm $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ được xây dựng từ G bằng cách loại bỏ các kí hiệu vô sinh như sau:
  - Xây dựng V_1:
    - Nếu trong P có quy tắc $A \to \omega$ mà $\omega \in \Sigma^*$ thì đưa A vào V_1.
    - Nếu trong P có quy tắc $A \to X_1X_2 \ldots X_n$ mà $X_i \in \Sigma$ hoặc V_1 thì đưa A vào V_1.
    - Quá trình trên tiếp tục cho đến khi nào không đưa thêm được phần tử nào vào V_1 , khi đó V_1 được hình thành.
  - Xây dựng P_1:
    - P_1 gồm tất cả các quy tắc trong P mà kí hiệu có mặt trong đó đều thuộc tập $\Sigma \cup V_1$.
  - Lấy $\Sigma_1 = \Sigma$, $\sigma_1 = \sigma$.

Ví dụ

- Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c, d\}, \ V = \{\sigma, A, B, F, T\}$
  $P = \{\sigma \to aABb, \ A \to aB, \ B \to c\sigma, \ B \to d, \ \sigma \to bT,$
  $T \to aTb, \ F \to c, \ F \to aT\}$
- Ta xây dựng văn phạm $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ tương đương với G và không chứa ký hiệu vô sinh như sau:
  - Xây dựng V_1:
    - Do $B \to d$ nên $V_1 = \{B\}$
    - Do $A \to aB$ nên $V_1 = \{A, B\}$
    - Do $\sigma \to aABb$ nên $V_1 = \{\sigma, A, B\}$
    - Do $F \to c$ nên $V_1 = \{\sigma, A, B, F\}$
  - $P_1 = \{\sigma \to aABb, \ A \to aB, \ B \to c\sigma, \ B \to d, \ F \to c\}$
  - $\Sigma_1 = \Sigma = \{a, b, c, d\}$
  - $\sigma_1 = \sigma$

Loại kí hiệu không đến được

- **Định lý 2.** Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$, ta có thể xây dựng văn phạm phi ngữ cảnh $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ tương đương với G sao cho $\forall \delta \in \Sigma_2 \cup V_2$ đều là ký hiệu đến được tức là tồn tại $\alpha, \beta \in (\Sigma_2 \cup V_2)^*$ để cho $\sigma_2 \vdash \alpha\delta\beta$.
- **Chứng minh.** Văn phạm $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ được xây dựng từ G bằng cách loại các kí hiệu không đến được như sau:
  - Xây dựng V_2 và Σ_2:
    - Đưa σ vào V_2
    - Nếu có $A \in V_2$ và $A \to \alpha$ trong P thì đưa các kí hiệu phụ trong α vào V_2 còn các kí hiệu cơ bản trong α vào Σ_2.
    - Quá trình trên dừng khi không thêm được bất kỳ kí hiệu nào vào Σ_2 và V_2 khi đó Σ_2, V_2 được hình thành.
  - Xây dựng P_2 và σ_2:
    - P_2 gồm tất cả các quy tắc trong P mà các kí hiệu trong đó thuộc $\Sigma_2 \cup V_2$.
    - $\sigma_2 = \sigma$

Ví dụ

- Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c, d\}, \ V = \{\sigma, A, B, C, K\}$
  $P = \{\sigma \to bA, \ A \to aA, \ B \to c\sigma, \ B \to dB, \ \sigma \to bK,$
  $K \to aKb, \ C \to a, \ K \to aC\}$
- Ta xây dựng văn phạm $G_1 = (\Sigma_1, V_1, \sigma, P_1)$ tương đương với G và không chứa ký hiệu không đến được như sau:
  - Xây dựng V_1 và Σ_1:
    - Do $\sigma \to bA$ nên $V_1 = \{\sigma, A\}, \ \Sigma_1 = \{b\}$
    - Do $A \to aA$ nên $\Sigma_1 = \{a, b\}$
    - Do $\sigma \to bK$ nên $V_1 = \{\sigma, A, K\}$
    - Do $K \to aC$ nên $V_1 = \{\sigma, A, K, C\}$

    Đến đây ta được $V_1 = \{\sigma, A, K, C\}, \ \Sigma_1 = \{a, b\}$
  - $P_1 = \{\sigma \to bA, \ A \to aA, \ \sigma \to bK, \ K \to aKb, \ C \to a,$
    $K \to aC\}$
  - $\sigma_1 = \sigma$

Loại kí hiệu thừa

- **Định lý 3.** Mọi ngôn ngữ phi ngữ cảnh không rỗng đều có thể được sinh ra từ một văn phạm phi ngữ cảnh không có kí hiệu thừa.
- **Chứng minh.**
  Giả sử $L(G)$ là ngôn ngữ được sinh bởi văn phạm phi ngữ cảnh G, quá trình loại bỏ kí hiệu thừa được thực hiện theo hai bước sau:
  - Bước 1: Loại bỏ các kí hiệu vô sinh trong G được G_1.
  - Bước 2: Loại bỏ các kí hiệu không đến được trong G_1 được G_2.
    Khi đó văn phạm G_2 không có ký hiệu thừa và tương đương với G.
- Câu hỏi: Có thể đảo thứ tự của hai bước trên được không?

Ví dụ

- Loại bỏ các kí hiệu thừa trong văn phạm sau
  $G = (\Sigma, V, \sigma, P)$ trong đó
  $\Sigma = \{a, b, c\}$
  $V = \{\sigma, A, B\}$
  $P = \{\sigma \to ABa, \sigma \to ab, \sigma \to \sigma a, A \to a, B \to Bb, B \to Bc\}$
- Giải:
  - Trước hết ta thấy B là kí hiệu vô sinh, ta được $G_1 = (\Sigma_1, V_1, \sigma, P_1)$ trong đó $\Sigma_1 = \{a, b, c\}$, $V_1 = \{\sigma, A\}$
    $P_1 = \{\sigma \to ab, \sigma \to \sigma a, A \to a\}$
  - Ta thấy A và c là hai kí hiệu không đến được, ta có:
    $G_2 = (\Sigma_2, V_2, \sigma, P_2)$ trong đó
    $\Sigma_2 = \{a, b\}$, $V_2 = \{\sigma\}$
    $P_2 = \{\sigma \to ab, \sigma \to \sigma a\}$

Loại bỏ quy tắc rỗng

- **Định nghĩa.**
  Quy tắc $A \to \wedge$, $A \in V$ được gọi là quy tắc rỗng
- **Nhận xét.** Nếu $\wedge \in L(G)$ thì trong G luôn có quy tắc rỗng (ngược lại có đúng không?)
- **Định lý 4.** Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm phi ngữ cảnh, khi đó $L(G) \setminus \{\wedge\}$ được sinh bởi văn phạm phi ngữ cảnh G' không có quy tắc rỗng.
- **Nhận xét.** Định lý trên cho thấy rằng:
  Nếu $\wedge \notin L(G)$ thì ta có thể loại bỏ các quy tắc rỗng trong G để được văn phạm G' không chứa quy tắc rỗng tương đương với G.
- **Hệ quả.** Nếu $\wedge \in L(G)$ thì ta thể xây dựng được văn phạm G' tương đương với G mà chỉ có duy nhất một phép thế rỗng.

Chứng minh

Ta xây dựng G' theo ba bước như sau:

- Nếu $A \to \wedge$ là quy tắc rỗng trong G thì gọi A là kí hiệu triệt tiêu.
- Nếu $A \to X_1X_2\ldots X_n$ là quy tắc trong G mà trong đó $X_{i_1},\ldots X_{i_k}$ là các kí hiệu triệt tiêu thì bổ sung vào P' tất cả các quy tắc có dạng $A \to X_1X_2\ldots X_n$ trong đó một hoặc một số kí hiệu triệt tiêu $X_{i_1}, X_{i_2}, \ldots, X_{i_k}$ được thay bởi $\wedge$ (nếu quy tắc tạo thành không phải quy tắc rỗng).
- Bổ sung vào P' tất cả các quy tắc của P mà không phải quy tắc rỗng.

Văn phạm $G'' = (\Sigma'', V'', \sigma, P'')$ trong hệ quả trên được xây dựng như sau:

- Thay P' bởi P'' gồm các quy tắc trong P' mà mọi ký hiệu σ thay bằng σ_1, bổ xung vào P'' phép thế $\sigma \to \sigma_1$ và $\sigma \to \wedge$.
- $V'' = V' \cup \{\sigma_1\}, \quad \Sigma'' = \Sigma$

Quy tắc rỗng - Ví dụ

- Cho $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b\}, V = \{\sigma, A, B\}$
  $P = \{A \to \wedge, A \to a,\ B \to \wedge,\ B \to b,\ \sigma \to ABA, \sigma \to a\sigma\}.$
- Xây dựng văn phạm G' không chứa quy tắc rỗng và $L(G') = L(G) \setminus \{\wedge\}$
  Ta thấy A, B là ký hiệu triệt tiêu nên loại bỏ quy tắc rỗng $A \to \wedge,\ B \to \wedge$ khỏi P và bổ xung vào P' các quy tắc sinh ra từ phép thế $\sigma \to ABA$ là :
  $$\sigma \to AB, \sigma \to BA, \sigma \to AA,\ \sigma \to B$$
  Kết hợp với các quy tắc khác rỗng trong P ta có:
  $P' = \{A \to a, \sigma \to ABA, \sigma \to AB,\ B \to b,\ \sigma \to BA, \sigma \to B,$
  $\qquad \sigma \to AA,\ \sigma \to a\sigma\}.$
  Vậy $G' = (\Sigma, V, \sigma, P')$ là văn phạm cần tìm.
- Văn phạm G'' tương với G và có duy nhất một phép thế rỗng là:
  $P'' = \{A \to a, \sigma_1 \to ABA, \sigma_1 \to AB,\ B \to b,\ \sigma_1 \to BA,$
  $\qquad \sigma_1 \to B,\ \sigma_1 \to AA,\ \sigma_1 \to a\sigma_1,\ \sigma \to \sigma_1, \sigma \to \wedge\}$
  $V'' = \{\sigma, \sigma_1, A, B\}, \quad \Sigma'' = \Sigma = \{a, b\}$
  Vậy $G'' = (\Sigma'', V'', \sigma, P'')$ là văn phạm cần tìm.

Quy tắc đơn

- **Định nghĩa**:
  Quy tắc có dạng $A \to B$ với $A, B \in V$ được gọi là quy tắc đơn.
- **Định lý 5**: Mọi ngôn ngữ phi ngữ cảnh đều có thể sinh ra từ một văn phạm phi ngữ cảnh không có quy tắc đơn.
- **Chứng minh**: Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm phi ngữ cảnh sinh ra ngôn ngữ phi ngữ cảnh đã cho, ta xây dựng tập quy tắc P' không chứa quy tắc đơn từ P sao cho $L(G) = L(G')$ với $G' = (\Sigma, V, \sigma, P')$.
  - Nếu $A \to B$ là quy tắc đơn thì A gọi là kí hiệu đơn, B gọi là kí hiệu đơn tương ứng của A.
  - Nếu $X \to X_1 X_2 \ldots X_n$ là một quy tắc trong P và $X_{i_1}, \ldots X_{i_k}$ là các ký hiệu đơn thì sổ sung vào P' tất cả các quy tắc hình thành từ quy tắc trên bằng cách thay một hoặc một số kí hiệu đơn X_{i_j} bằng ký hiệu đơn tương ứng của X_{i_j}.
  - Bổ sung vào P' tất cả các quy tắc của P mà không phải là quy tắc đơn.

Quy tắc đơn - Ví dụ

- Cho văn phạm:
  $G = (\Sigma, V, \sigma, P)$
  $\Sigma = \{a, b, c\}, V = \{\sigma, A, B, C\}$
  $P = \{A \rightarrow B, A \rightarrow C, C \rightarrow a, A \rightarrow c, B \rightarrow b, \sigma \rightarrow AabA\}$
  - Dễ thấy A là ký hiệu đơn với hai quy tắc đơn là $A \rightarrow B,\ A \rightarrow C$
    Ta đưa vào P' các quy tắc sinh ra từ phép thế $\sigma \rightarrow AabA$ là:
    $\sigma \rightarrow BabA|AabB|BabB|CabA|AabC|CabC|BabC|CabB$
  - Bổ xung vào P' các quy tắc trong P mà không phải là quy tắc đơn ta được:
    $P' = \{A \rightarrow c, B \rightarrow b, C \rightarrow a$
    $\sigma \rightarrow AabA|BabA|AabB|BabB|CabA|AabC|CabC|BabC|CabB\}$

  Vậy $G' = (\Sigma, V, \sigma, P')$ và $L(G') = L(G)$

Văn phạm chuẩn của Chomsky

- **Định nghĩa.**
  Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ được gọi là văn phạm chuẩn của Chomsky nếu mọi quy tắc trong P đều có dạng $A \to BC$ hoặc $A \to a$, với $A, B, C \in V, a \in \Sigma$.
- **Ví dụ.**
  Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c\}, \quad V = \{\sigma, A, B, C, F\}$
  $P = \{\sigma \to AB,\ A \to BF,\ F \to B\sigma,\ A \to b,\ B \to c,\ C \to a\}$
  là văn phạm dạng chuẩn Chomsky.
- **Định lý 6.**
  Đối với mỗi văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ luôn tồn tại một văn phạm phi ngữ cảnh dạng chuẩn Chomsky $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ tương đương với G.

Chứng minh

- Bước 0: Loại bỏ quy tắc đơn theo các bước của định lý 5.
- Bước 1: Xây dựng văn phạm $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ từ G chỉ chứa quy tắc có dạng: $A \rightarrow a$ và $A \rightarrow B_1 B_2 \ldots B_n$, $B_i \in V_1$ $a \in \Sigma_1$.
  - Trước hết ta đặt $\Sigma_1 = \Sigma, V_1 = V, \sigma_1 = \sigma$.
  - Nếu $A \rightarrow X$ là quy tắc trong P, $|X| \geqslant 2$ và X chứa các kí hiệu cơ bản thì:
    - Ta bổ sung vào V_1 kí hiệu phủ định $\{\overline{a} | a \in \Sigma\}$
    - Đưa vào P_1 quy tắc $A \rightarrow \overline{X}$ trong đó $\overline{X}$ hình thành từ X bằng cách thay các kí hiệu cơ bản bởi kí hiệu phủ định tương ứng.
    - Bổ sung vào P_1 tập quy tắc $\{\overline{a} \rightarrow a | a \in \Sigma\}$.
  - Bổ sung vào P_1 các quy tắc trong P có dạng $A \rightarrow a$ và $A \rightarrow B_1 B_2 \ldots B_n$, $B_i \in V$.

Chứng minh

- Bước 2: Xây dựng văn phạm $G_2 = (\Sigma_2, V_2, \sigma_2, P_2)$ từ G_1 chỉ chứa quy tắc có dạng $A \to BC$ và $A \to a$ với $A, B, C \in V_2,\ a \in \Sigma_2$.
  - Trước hết ta đặt $\Sigma_2 = \Sigma_1,\ V_2 = V_1,\ \sigma_2 = \sigma_1$.
  - Nếu $A \to B_1 B_2 \ldots B_n \quad (n > 2)$ là quy tắc trong P_1 thì:
    - Bổ sung vào V_2 các kí hiệu phụ mới $K_1, K_2, \ldots, K_{n-2}$(chú ý rằng mỗi lần bổ sung các kí hiệu mới phải khác nhau).
    - Đưa vào P_2 các quy tắc:
      $A \to B_1 K_1, K_1 \to B_2 K_2, \ldots, K_{n-2} \to B_{n-1} B_n$
  - Bổ sung vào P_2 các quy tắc của P_1 có dạng
    $A \to a, A \to BC,\ A, B, C \in V_1,\ a \in \Sigma_1$.

Ví dụ

- Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ trong đó:
  $\Sigma = \{a, b, c, d\}, V = \{\sigma, A, B\}$
  $P = \{\sigma \to \sigma a, \sigma \to ABb, A \to ABBa, B \to b, A \to d, \sigma \to c\}$
  Quá trình biến đổi G thành văn phạm dạng chuẩn Chomsky như sau:
  - Bước 1: $G_1 = (\Sigma, V_1, \sigma, P_1)$ trong đó:
    $V_1 = \{\sigma, A, B, \overline{a}, \overline{b}\}$
    $P_1 = \{\sigma \to \sigma\overline{a}, \overline{a} \to a, \sigma \to AB\overline{b},\ A \to ABB\overline{a},$
    $\overline{b} \to b, B \to b, A \to d, \sigma \to c\}$
  - Bước 2: $G_2 = (\Sigma, V_2, \sigma, P_2)$ với:
    $V_2 = \{\sigma, A, B, \overline{a}, \overline{b}, K_1, S_1, S_2\}$
    $P_2 = \{\sigma \to \sigma\overline{a}, \overline{a} \to a, \sigma \to AK_1, K_1 \to B\overline{b}, \overline{b} \to b,\ A \to AS_1,$
    $S_1 \to BS_2, S_2 \to B\overline{a}, B \to b, A \to d, \sigma \to c\}$

  Đến đây ta được văn phạm dạng chuẩn Chomsky G_2 tương đương với G.

Quy tắc đệ quy trái

- **Định nghĩa.** Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ được gọi là đệ quy trái nếu trong P có suy dẫn dạng $A \vdash A\alpha$ với $A \in V, \alpha \in (\Sigma \cup V)^*$. Quy tắc thế có dạng $B \to B\beta$ với $B \in V, \beta \in (\Sigma \cup V)^*$ được gọi là quy tắc đệ quy trái.
- **Ví dụ 1.** Văn phạm phi ngữ cảnh $G = (\{a, b\}, \sigma, A, B, \sigma, P)$ với:
  $P = \{\sigma \to aBA,\ A \to Ab,\ B \to Bab,\ B \to \wedge, B \to B\sigma\}$
  có chứa 3 quy tắc đệ quy trái là $A \to Ab,\ B \to Bab,\ B \to B\sigma$ do đó văn phạm G là văn phạm đệ quy trái.
- **Ví dụ 2.** Cho văn phạm $G = (\{a, b, c\}, \{S, A, B\}, S, P)$ trong đó $P = \{S \to AB, A \to Ba, A \to cS, B \to Ab, B \to \wedge\}$ là văn phạm phi ngữ cảnh đệ quy trái vì tồn tại dẫn xuất $A \to Ba \to Aba$ mặc dù trong P không chứa quy tắc đệ quy trái.

Khử quy tắc đệ quy trái

- **Định lý 1**
  Mọi văn phạm phi ngữ cảnh đệ quy trái đều tồn tại văn phạm phi ngữ cảnh không chứa quy tắc đệ quy trái tương đương với nó.
- **Chứng minh**: Văn phạm $G_1 = (\Sigma_1, V_1, \sigma_1, P_1)$ được xây dựng như sau:
  - Gọi $A \to A\alpha_i, \quad i = 1, \dots n$ là tất cả các quy tắc đệ quy trái có vế trái là A , $\alpha_i \in (\Sigma \cup V)^*$.
    Gọi $A \to \beta_j, \quad j = 1, \dots, m$ là tất cả các quy tắc không đệ quy trái có vế trái là A , $\beta_j \in (\Sigma \cup V)^*$.
  - Khi đó các phép thế đệ quy trái trên được thay thế bởi các phép thế sau:
    - $A \to \beta_j A', \qquad j = \overline{1, m}$
    - $A' \to \alpha_i A', \qquad i = \overline{1, n}$
    - $A' \to \alpha_i, \qquad i = \overline{1, n}$
  - Quá trình thay thế này tiếp tục cho đến khi không còn quy tắc đệ quy trái trong P.

Dạng chuẩn Greiback

- **Định nghĩa**. Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ mà mỗi quy tắc của nó đều có dạng $A \to a\alpha, A \in V, a \in \Sigma, \alpha \in (\Sigma \cup V)^*$ được gọi là văn phạm phi ngữ cảnh dạng chuẩn Greiback hay dạng chuẩn Greiback.
- **Ví dụ.** Văn phạm $G = (\{a, b\}, \{S, A, B\}, S, P)$ với
  $P = \{S \to aA, A \to bBB, B \to aBA, A \to a, B \to b\}$
  là văn phạm phi ngữ cảnh dạng chuẩn Greiback.
- **Bổ đề 1.** Giả sử $G = (\Sigma, V, \sigma, P)$ là một văn phạm phi ngữ cảnh. Nếu $A \to \alpha_1 B \alpha_2$ là một quy tắc trong G và $B \to \beta_1, \beta_2, \ldots, \beta_n$ là tất cả các quy tắc trong văn phạm G mà vế trái là ký hiệu phụ B. Khi đó có thể thay thế quy tắc $A \to \alpha_1 B \alpha_2$ bằng tập các quy tắc:
  $$A \to \alpha_1\beta_1\alpha_2 | \alpha_1\beta_2\alpha_2 | \ldots | \alpha_1\beta_n\alpha_2$$

Thuật toán tìm dạng chuẩn Greiback

Văn phạm không chứa quy tắc rỗng G đưa về dạng chuẩn Greiback như sau:

- Bước 1: Đổi các ký hiệu phụ của G thành tập $V_1 = (A_1, A_2, \ldots, A_m)$, trong đó A_1 là tiên đề của văn phạm.
- Bước 2:
  - Nếu trong G còn quy tắc dạng $A_i \to A_j\alpha, j < i$ thì thay thế quy tắc này theo bổ đề 1 và quay lại bước 2.
  - Nếu xuất hiện quy tắc đệ quy trái $A_i \to A_i\alpha$ thì thêm vào ký hiệu mới A'_i rồi khử nó theo định lý 1.
  - Nếu mỗi quy tắc trong G đều có dạng $A \to a\alpha,\ A_i \to A_j\alpha$ với $a \in \Sigma_1,\ \alpha \in (\Sigma_1 \cup V_1)^*,\ j > i$ thì chuyển sang bước 3.
- Bước 3: Lần lượt thay thế hết các quy tắc dạng $A_i \to A_j\alpha,\ (j > i)$ theo bổ đề 1, bắt đầu từ những phép thế có dạng $A_{m-1} \to A_m\alpha$ đến những phép thế có dạng $A_1 \to A_k\alpha\ (k > 1)$, rồi chuyển sang bước 4.
- Bước 4: Thế hết các quy tắc dạng $A'_i \to A_j\alpha$ theo bổ đề 1.

Ví dụ

Cho văn phạm $G = (\{a, b\}, \{S, A, B\}, S, P)$ trong đó
$P = \{S \to AS, A \to BS, A \to a, B \to AA, B \to b, S \to a\}$
Hãy xây dựng văn phạm dạng chuẩn Greiback tương với G.

- Bước 1: Ký hiệu lại các chữ cái phụ của G
  $A_1 \equiv S, A_2 \equiv A, A_3 \equiv B$
  ta được văn phạm G_1 tương đương với G
  $G_1 = (\{a, b\}, \{A_1, A_2, A_3\}, A_1, P_1)$
  Trong đó
  $P_1 = \{A_1 \to A_2A_1, A_1 \to a, A_2 \to A_3A_1, A_2 \to a,$
  $A_3 \to A_2A_2, A_3 \to b\}$
- Bước 2: Thế quy tắc $A_3 \to A_2A_2$ bằng các quy tắc
  $A_3 \to A_3A_1A_2$ và $A_3 \to aA_2$.
  Vì $A_3 \to A_3A_1A_2$ là quy tắc đệ quy trái nên ta thế bởi:
  $A_3 \to bA_3', \; A_3 \to aA_2A_3'$
  $A_3' \to A_1A_2, \; A_3' \to A_1A_2A_3'$

Ví dụ

Đến đây ta được các quy tắc thế trong G_1 là:
$A_1 \longrightarrow A_2A_1,\ A_1 \longrightarrow a,\ A_2 \longrightarrow A_3A_1,\ A_2 \longrightarrow a,$
$A_3 \longrightarrow b,\ A_3 \longrightarrow aA_2,\ A_3 \longrightarrow bA_3',\ A_3 \longrightarrow aA_2A_3',$
$A_3' \longrightarrow A_1A_2,\ A_3' \longrightarrow A_1A_2A_3'.$

- Bước 3: Quy tắc $A_2 \longrightarrow A_3A_1$ thế bằng
  $A_2 \longrightarrow bA_1|aA_2A_1|bA_3'A_1|aA_2A_3'A_1$
  Quy tắc $A_1 \longrightarrow A_2A_1$ thế bằng
  $A_1 \longrightarrow bA_1A_1|aA_2A_1A_1|bA_3'A_1A_1|aA_2A_3'A_1A_1|aA_1$
- Bước 4: Thế quy tắc $A_3' \longrightarrow A_1A_2$ bởi
  $A_3' \longrightarrow bA_1A_1A_2|aA_2A_1A_1A_2|bA_3'A_1A_1A_2|aA_2A_3'A_1A_1A_2|aA_1A_2|aA_2$
  Cuối cùng thế quy tắc $A_3' \longrightarrow A_1A_2A_3'$ bởi:
  $A_3' \longrightarrow bA_1A_1A_2A_3'|aA_2A_1A_1A_2A_3'|bA_3'A_1A_1A_2A_3',$
  $A_3' \longrightarrow aA_2A_3'A_1A_1A_2A_3'|aA_1A_2|aA_2A_3'$

Điều kiện cần của ngôn ngữ phi ngữ cảnh

- **Định lý 1.** Nếu ngôn ngữ vô hạn L là ngôn ngữ phi ngữ cảnh thì tồn tại các số nguyên dương n_1, n_2 sao cho đối với từ tùy ý $z \in L$ mà $|z| > n_1$ đều có các từ u, v, x, y, w để $z = uxwyv$ thỏa mãn các điều kiện sau:
  - $|xwy| \leqslant n_2$
  - $xy \neq \wedge$
  - $\forall k \in \mathbb{N}$ ta có $z_k = ux^kwy^kv \in L$.
- **Hệ quả 1.** Ngôn ngữ $L = \{a^nb^nc^n | n = 1, 2, \ldots\}$ không phải là ngôn ngữ phi ngữ cảnh.
- **Chứng minh.** Thật vậy, giả sử L là ngôn ngữ phi ngữ cảnh, khi đó theo điều kiện cần tồn tại k đủ lớn để từ $z = a^kb^kc^k$ biểu diễn được thành tích ghép của 5 từ u, x, w, y, v trong đó $xy \neq \wedge$ tức là $a^kb^kc^k = uxwyv$. Ta xét tất cả các khả năng có thể xảy ra đối với x, y:

Chứng minh

1. x là từ con của a^k, y là từ con của a^k.
2. x là từ con của a^k, y chứa cả a và b.
3. x là từ con của a^k, y là từ con của b^k.
4. x là từ con của a^k, y chứa b và c.
5. x là từ con của a^k, y là từ con của c^k.
6. x chứa a và b, y là từ con của b^k.
7. x chứa a và b, y chứa b và c.
8. x chứa a và b, y là từ con của c^k.
9. x là từ con của b^k, y là từ con của b^k.
10. x là từ con của b^k, y chứa b và c.
11. x là từ con của b^k, y là từ con của c^k.
12. x chứa b và c, y là từ con của c^k.
13. x là từ con của c^k, y là từ con của c^k.

Chứng minh

Nhận xét: nếu $z \in L$ thì $l_a(z) = l_b(z) = l_c(z)$ và a, b, c không đứng xen kẽ nhau.

- **Trường hợp 1**: x, y là từ con của a^k
  Do $z_2 = ux^2wy^2v$ và $uxwyv = a^kb^kc^k$ nên suy ra $l_a(z_2) > k$.
  Mặt khác vì b, c không nằm trong x, y nên
  $l_b(z_2) = l_c(z_2) = k \Rightarrow z_2 \notin L \Rightarrow$ mâu thuẫn với điều kiện cần.
  Tương tự với các trường hợp 9 và 13.
- **Trường hợp 3**: x là từ con của a^k, y là từ con của b^k suy ra
  $z_2 = ux^2wy^2v$ có $l_a(z_2) > k$, mặt khác do c không nằm trong x, y nên
  $L_c(z_2) = k \Rightarrow z_2 \notin L$, mâu thuẫn với điều kiện cần.
  Tương tự đối với các trường hợp 5 và 11.
- **Trường hợp 6**: x chứa a và b, y là từ con của b^k, ta có:
  $x = a^mb^n \Rightarrow x^2 = a^mb^na^mb^n$ từ đó ta có a và b trong x^2 đứng xen kẽ nhau, suy ra $z_2 = ux^2wy^2v \notin L$, điều này mâu thuẫn với điều kiên cần.
  Tương tự cho các trường hợp 2,4,7,8,10,12.

Tính đóng của ngôn ngữ phi ngữ cảnh

- **Định lý 2.** Lớp ngôn ngữ phi ngữ cảnh đóng với các phép: hợp, tích ghép, lặp, lặp cắt, không đóng với phép giao và lấy phần bù.
- **Chứng minh.** Đối với tính đóng của các phép toán hợp, nhân, lặp, lặp cắt ta chứng minh tương tự như đối với định lý về tính đóng của ngôn ngữ sinh bởi văn phạm.
  - Tính không đóng với phép giao: Dễ thấy rằng ngôn ngữ $L = \{a^n b^n c^n | n \in Z^+\}$ là giao của hai ngôn ngữ phi ngữ cảnh $L_1 = \{a^m b^m c^k | m, k \in Z^+\}$ và $L_2 = \{a^s b^t c^t | s, t \in Z^+\}$.
    Trong đó văn phạm phi ngữ cảnh sinh ra L_1 là:
    $G_1 = (\{a, b, c\}, \{\sigma_1, A, B\}, \sigma_1, P_1)$
    $P_1 = \{\sigma_1 \to AB, A \to ab, A \to aAb, B \to cB, B \to c\}$
    Văn phạm phi ngữ cảnh sinh ra L_2 là:
    $G_2 = (\{a, b, c\}, \{\sigma_2, A, B\}, \sigma_2, P_2)$
    $P_2 = \{\sigma_2 \to AB, A \to aA, A \to a, B \to bBc, B \to bc\}$
    Theo định lý 2 thì $L = L_1 \cap L_2$ không phải là ngôn ngữ phi ngữ cảnh. Vậy ngôn ngữ phi ngữ cảnh không đóng với phép giao.

Tính đóng của ngôn ngữ phi ngữ cảnh

- **Chứng minh**
  - Tính không đóng với phép lấy phần bù:

    Giả sử lớp ngôn ngữ phi ngữ cảnh đóng với phép lấy phần bù . Khi đó CL_1 và CL_2 là những ngôn ngữ phi ngữ cảnh do đó:

    $L = L_1 \cap L_2 = C(CL_1 \cup CL_2)$ cũng là ngôn ngữ phi ngữ cảnh, điều này mâu thuẫn với định lý 2. Vậy ngôn ngữ phi ngữ cảnh không đóng với phép lấy phần bù.

Cây dẫn xuất đầy đủ trong văn phạm PNC

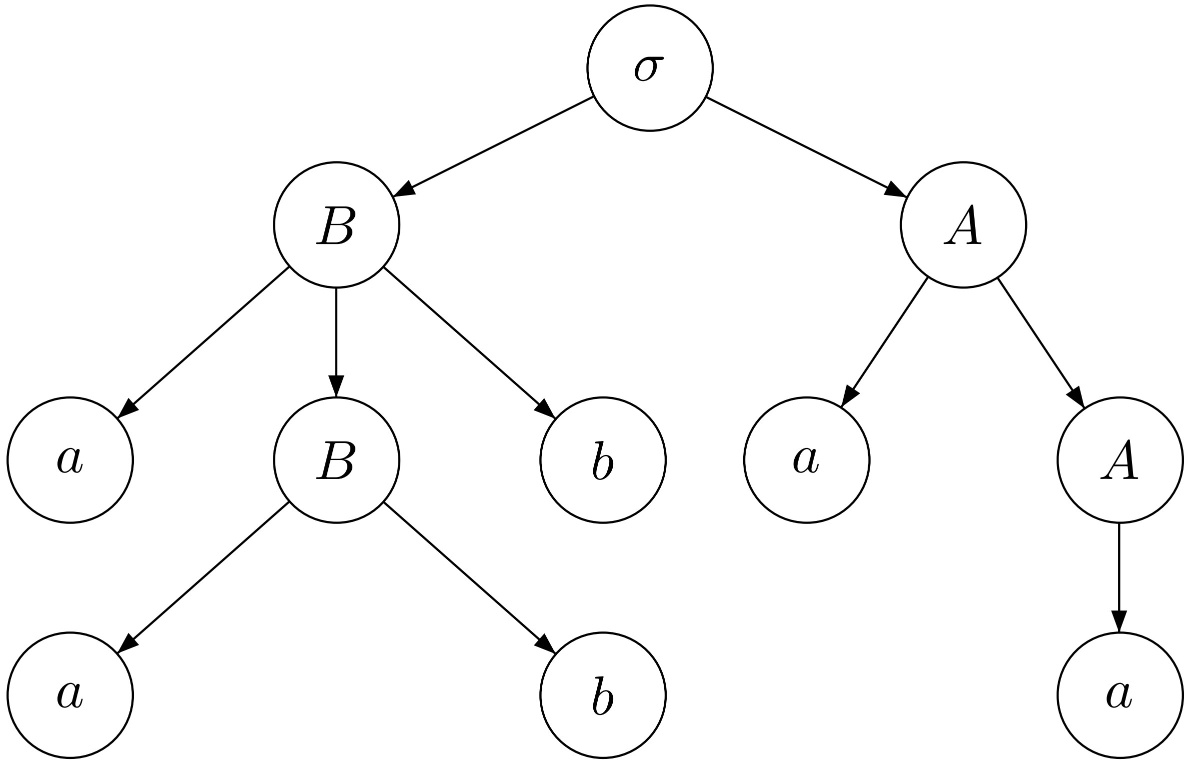
- **Định nghĩa.**
  Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm phi ngữ cảnh và $D = (\sigma = \omega_0, \omega_1, \ldots, \omega_m = x)$ là dẫn xuất đầy đủ của x trong G. Cây dẫn xuất (suy dẫn) của x được xây dựng từ D như sau:
  - Gốc của cây là σ.
  - Nếu phép thế $A \to X_1X_2\ldots X_k$ được thực hiện trong dẫn xuất trực tiếp $\omega_i \Rightarrow \omega_{i+1}$ thì ta vẽ thêm các đỉnh con của A lần lượt từ trái qua phải là $X_1, X_2, \ldots, X_k$.

Cây dẫn xuất đầy đủ trong văn phạm PNC

- **Tính chất**:
  - Ở mỗi đỉnh được gán một nhãn, gốc của cây được gán nhãn σ.
  - Mỗi đỉnh trong được gán nhãn là kí hiệu nào đó trong V.
  - Mỗi đỉnh ngoài (lá) được gán nhãn là một kí hiệu trong $\Sigma \cup \{\wedge\}$.
  - Nếu đỉnh m được gán nhãn là $A \in V$ còn các đỉnh $n_1, n_2 \ldots, n_k$ là các đỉnh con của đỉnh m theo thứ tự từ trái qua phải và được gán nhãn $X_1, X_2, \ldots, X_k$ tương ứng thì $A \to X_1 X_2 \ldots X_k$ là một quy tắc trong P được thực hiện trong dẫn xuất $\sigma \vdash x$.
  - Nếu đọc tất cả nhãn ở các lá theo thứ tự từ trái qua phải sẽ nhận được xâu x.

Cây dẫn xuất đầy đủ trong văn phạm PNC

- **Ví dụ.** Cho văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$
  $\Sigma = \{a, b\}, V = \{\sigma, A, B\}$
  $P = \{\sigma \to BA, B \to aBb, B \to ab, A \to aA, A \to a\}$
  Vẽ cây suy dẫn của $x = a^2b^2a^2$
- **Giải.** Dẫn xuất đánh dấu của x là:
  $D = ({}^*\sigma^*, B^*A^*, Ba^*A^*, {}^*B^*aa, a^*B^*ba^2, a^2b^2a^2)$



Tính nhập nhằng của ngôn ngữ PNC

- **Định nghĩa 1.** Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ được gọi là văn phạm nhập nhằng nếu tồn tại ít nhất một xâu $x \in L(G)$ mà được sinh ra bằng ít nhất hai cây suy dẫn khác nhau.
- **Định nghĩa 2.** Ngôn ngữ phi ngữ cảnh L được gọi là ngôn ngữ phi ngữ cảnh không nhập nhằng nếu tồn tại ít nhất một văn phạm không nhập nhằng sinh ra nó. Trường hợp ngược lại gọi là ngôn ngữ phi ngữ cảnh nhập nhằng.
- **Câu hỏi:** Nếu một ngôn ngữ sinh bởi một văn phạm phi ngữ cảnh nhâp nhằng thì nó có phải là ngôn ngữ nhập nhằng hay không?

Thuật toán phân tích từ trên xuống

Input: Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ không đệ quy trái và xâu $\omega = a_1 a_2 \dots a_n \in \Sigma^*$.

Output: Cây suy dẫn của ω nếu $\omega \in L(G)$ và thông báo lỗi nếu $\omega \notin L(G)$.

- Bước 1: Gốc của cây là σ, đỉnh hiện tại trên cây là σ và đưa con trỏ vào kí hiệu đầu tiên a_1 trên ω.
- Bước 2: Ký hiệu T là đỉnh hiện tại trên cây và v là kí hiệu hiện tại của con trỏ trên ω.
  - Nếu $T \in V$ thì liệt kê tất cả các quy tắc có vế trái là T và đánh số từ nhỏ đến lớn , chọn quy tắc có chỉ số từ bé đến lớn. Nếu không có phép thế nào hoặc tất cả các phép thế đã thực hiên rồi thì báo lỗi. Nếu có phép thế $T \to \theta$ thì:
    - Nếu $\theta = \wedge$ thì vẽ $\wedge$ là đỉnh con của T quay về bước 2 với đỉnh hiện tại trên cây chuyển sang đỉnh bên phải của T.
    - Nếu $\theta \neq \wedge$ và $\theta = X_1 X_2 \dots X_k$ thì vẽ $X_1, X_2, \dots, X_k$ lần lượt là các đỉnh con của T từ trái qua phải và quay lại bước 2 với đỉnh hiện tại trên cây là X_1.

Thuật toán phân tích từ trên xuống

- Bước 2
  - Nếu $T = a \in \Sigma$ thì so sánh T với v.
    - Nếu $a = v$ thì kiểm tra:
      -nếu a là kí hiệu cuối cùng trên ω và bên phải T trên cây không còn đỉnh nào thuộc V thì ta được cây suy dẫn của ω.
      -nếu a là kí hiệu cuối cùng trên ω nhưng bên phải T trên cây còn đỉnh thuộc V thì báo lỗi.
      -nếu a chưa phải kí hiệu cuối cùng thì dịch con trỏ sang phải một kí tự trên ω và quay lại bước 2 với đỉnh hiện tại trên cây chuyển sang phải của T.
    - Nếu $a \neq v$ thì quay lại bằng cách xóa đi cây con do cha của đỉnh T sinh ra và thực hiện phép thế tiếp theo với vế trái là cha của T (nếu không có thì quay lại tiếp).

Thuật toán phân tích từ trên xuống

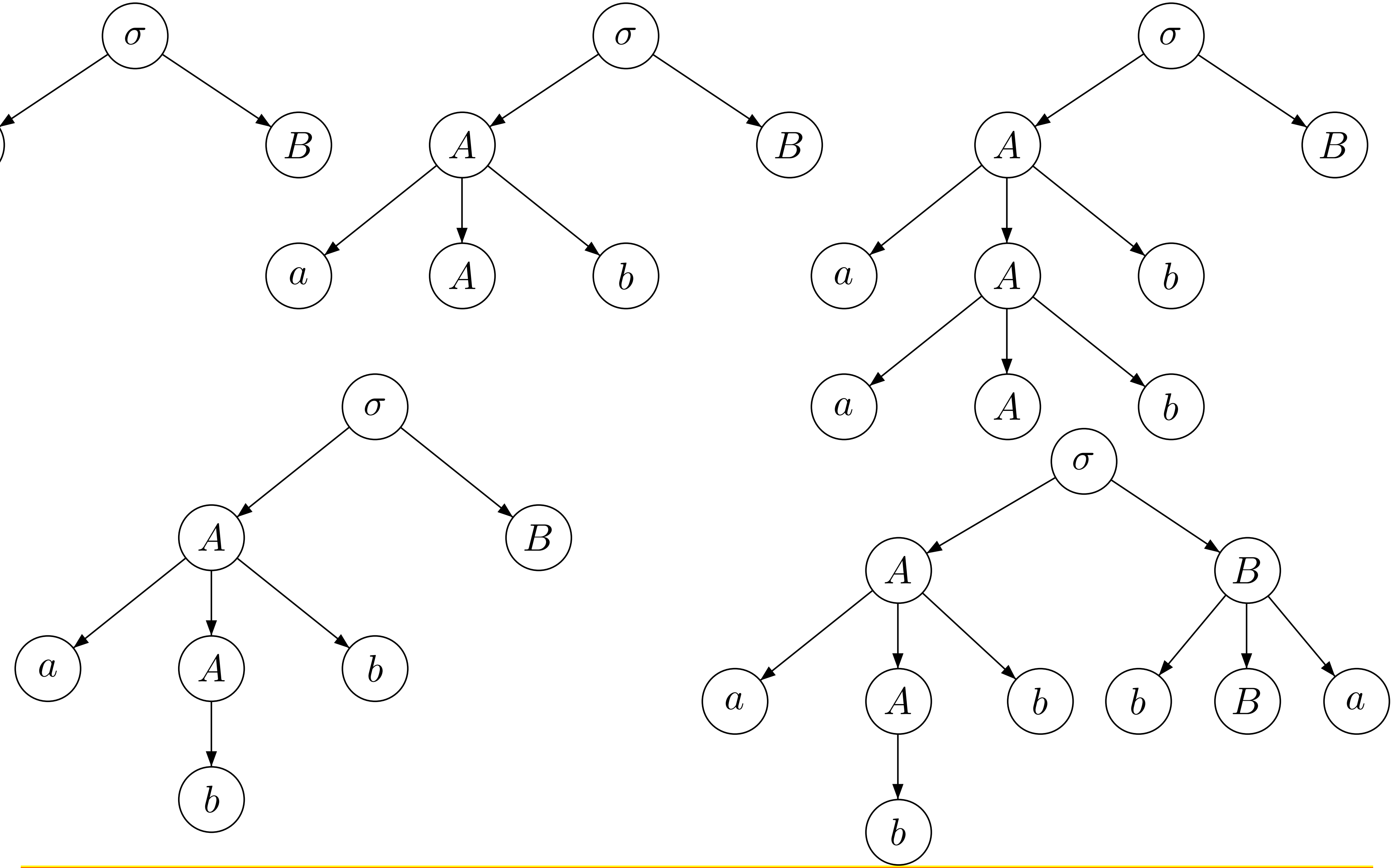
- **Ví dụ.** Cho $G = (\Sigma, V, \sigma, P)$, $\Sigma = \{a, b\}$, $V = \{\sigma, A, B\}$
  $P = \{\sigma \rightarrow AB\ (1), A \rightarrow aAb\ (2), A \rightarrow b\ (3), B \rightarrow bBa(4), B \rightarrow \wedge\}$
  Tìm cây phân tích từ trên xuống của $\omega = abbba$

- **Giải.** Gốc là σ, đỉnh hiện tại trên cây $T = \sigma$, con trỏ chỉ vào kí hiệu đầu tiên của ω là $*a*bbba$ với $v = a$.
  Do $\sigma \in V$ nên thực hiện phép thế (1): $\sigma \rightarrow AB$, lúc này $T = A$
  Do $A \in V$ nên thực hiện phép thế (2): $A \rightarrow aAb$, lúc này $T = a$
  $a = v$ nên con trỏ chuyển sang phải là $a*b*bba$ khi đó $v = b$, và $T = A$
  Do $A \in V$ nên thực hiên phép thế (2) $A \rightarrow aAb$, lúc này $T = a$.
  Do $a \neq v = b$ nên ta hủy phép thế (2) vừa thực hiện ở trên và chuyển sang thực hiện phép thế (3) là $A \rightarrow b$, lúc này $T = b$.
  Do $b = v$ nên con trỏ dịch sang phải với đỉnh hiện là $ab*b*ba$ với $v = b$ và $T = b$.
  Do $b = v$ nên con trỏ dịch sang phải là $abb*b*a$ khi đó $v = b$ và $T = B$.
  Do $B \in V$ nên ta thực hiện phép thế (4) là $B \rightarrow bBa$, lúc này $T = b$.
  Do $b = v$ nên con trỏ dịch sang là $abbb*a*$ khi đó $v = a$ và $T = B$.

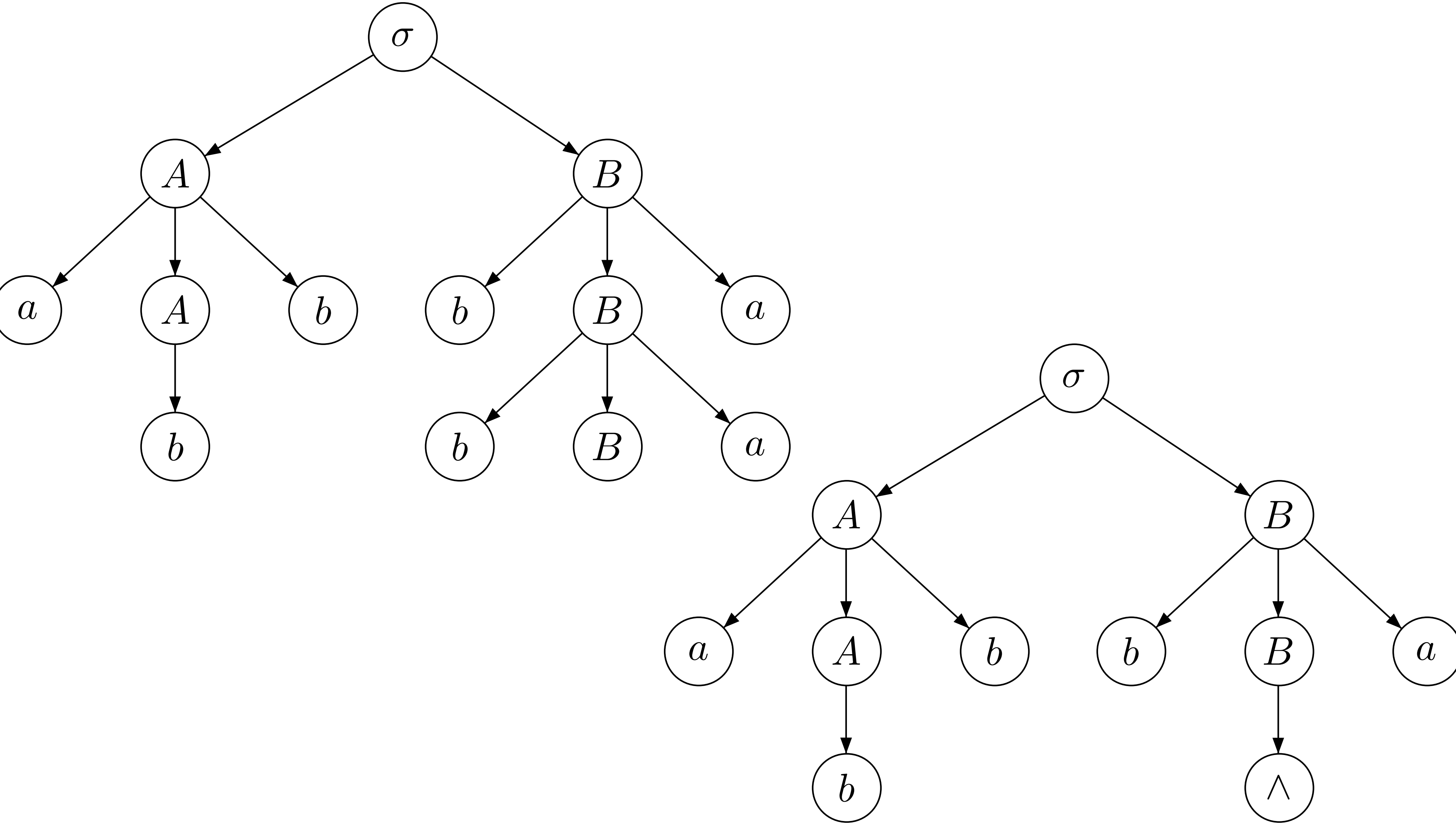
Thuật toán phân tích từ trên xuống

Do $B \in V$ nên ta thực hiện phép thế (4) là $B \to bBa$, lúc này $T = b$.
Do $b \neq v = a$ nên hủy bỏ phép thế (4) vừa thực hiện, thay vào đó phép thế (5) là $B \to \wedge$, lúc này $T = a$ (bỏ qua xâu rỗng).
Do ký hiệu hiện tại cuối cùng trên cây T trùng với ký hiệu cuối cùng của con trỏ v trên ω nên thuật toán kết thúc đoán nhận được xâu ω.

Hình vẽ quá trình xây dựng cây suy dẫn



Hình vẽ quá trình xây dựng cây suy dẫn



Thuật toán phân tích từ dưới lên

- **Input**: Văn phạm phi ngữ cảnh $G = (\Sigma, V, \sigma, P)$ không đệ quy trái và xâu $\omega = a_1 a_2 \ldots a_n \in \Sigma^*$
- **Output**: Cây phân tích từ dưới lên của ω nếu $\omega \in L(G)$ và thông báo lỗi nếu $\omega \notin L(G)$
- Các bước của thuật toán:
  Khởi tạo: Dùng s dể chỉ trạng thái biến đổi của ω, ban đầu $s = \omega$
  Dùng hai con trỏ $First$ và $Last$ để trỏ lên xâu s . Xâu xác định bắt đầu từ $First$ đến $Last$ trong s được gọi là xâu con trỏ, ban đầu $First$ và $Last$ trỏ đến vị trí đầu tiên của xâu s.

Các bước của thuật toán

1. Bước 1:
   (a) Nếu $First = Last = \sigma$ thì $\omega \in L(G)$ và chuyển sang bước 3.
   (b) Nếu xâu con trỏ là vế phải của một quy tắc trong P thì thay thế xâu con trỏ trong s bằng vế trái của quy tắc đó ta được xâu s mới, cho $First$ và $Last$ cùng trỏ đến vị trí đầu tiên của s sau đó quay lại bước 1.(Nếu có nhiều quy tắc cùng có vế phải là xâu con trỏ thì đánh số các quy tắc đó để lựa chọn).
   (c) Nếu xâu con trỏ không là vế phải của một quy tắc nào trong P thì:
      - Nếu $Last$ không phải kí hiệu cuối cùng của s thì dịch $Last$ sang phải một kí tự rồi quay lại bước 1.
      - Nếu $Last$ là kí hiệu cuối cùng của s và $First$ không phải là kí hiệu cuối cùng của s thì dịch $First$ sang phải một kí tự và cho $Last = First$ rồi quay lại bước 1.
      - Nếu $Last$ và $First$ cùng là kí hiệu cuối cùng trong s thì chuyển sang bước 2.

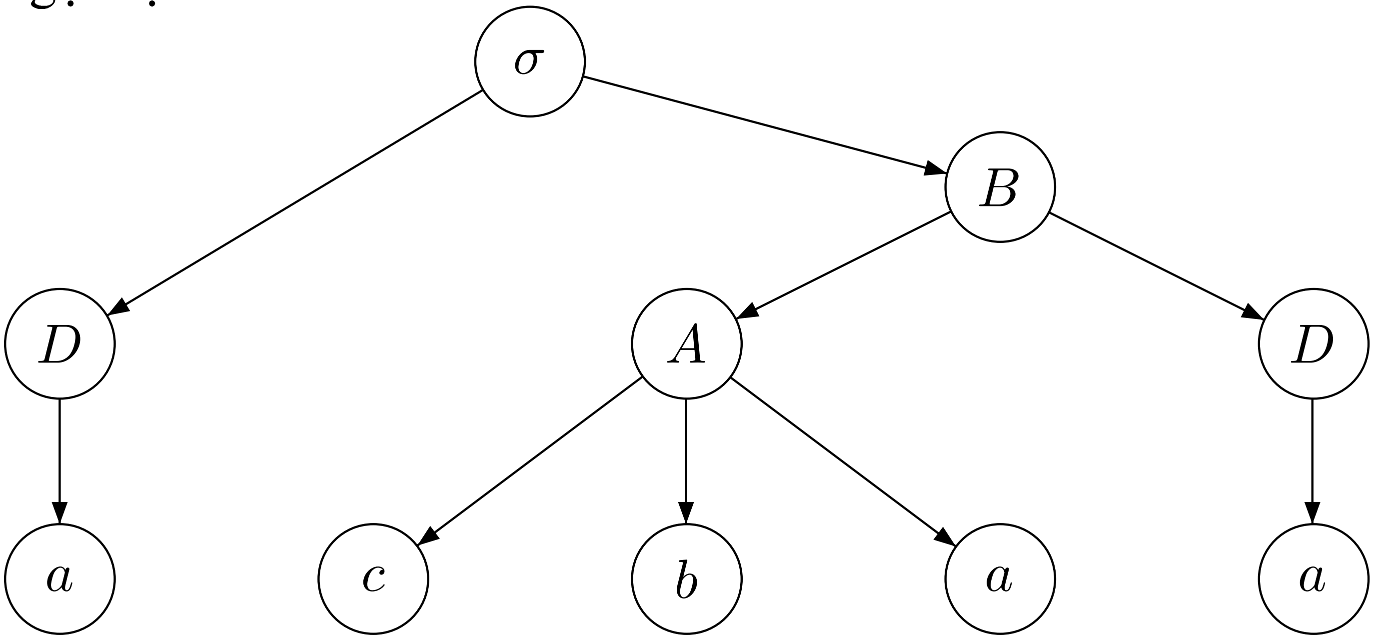
Các bước thuật toán

1. Bước 1
2. Bước 2: Hủy bỏ phép thế vừa thực hiện ở bước 1(b), khôi phục lại xâu s, $First$ và $Last$. Xét xem có còn quy tắc nào đánh số nhưng chưa thực hiện không:
   (a) Nếu còn thì quay lại bước 1 để thực hiện phép thế đó.
   (b) Nếu hết thì:
      - Quay lại bước 2 nếu $s \neq \omega$.
      - Báo lỗi nếu $s = \omega$.
3. Bước 3: Xây dựng cây suy dẫn nếu $\omega \in L(G)$.
   (a) Rút gọn lại quá trình diễn biến của s sao cho không trạng thái nào giống nhau.
   (b) Xét từ trái sang phải quá trình biến đổi của s, nếu phép thế $A \rightarrow X_1X_2 \ldots X_k$ được sử dụng trong quá trình đó thì vẽ thêm đỉnh A và các đỉnh X_i nếu $X_i \in \Sigma \cup \{\wedge\}$(nếu $X_i \in V$ thì nó ở trên cây và không có đỉnh cha), sau đó cho A trỏ đến $X_1, X_2, \ldots, X_k$.

Thuật toán từ dưới lên - Ví dụ

1. **Ví dụ.** Cho $G = (\Sigma, V, \sigma, P)$, $\Sigma = \{a, b, c\}$, $V = \{\sigma, A, B, D\}$
   $P = \{\sigma \to DB, B \to AD, A \to a, A \to cba, D \to a\}$
   $\omega = acbaa$ có thuộc $L(G)$ không?

2. **Giải.** Diễn biến của s như sau:
   $\omega = *a * cbaa \to A * cba * a \to AA * a* \to AAA \to AA * a* \to$
   $AAD \to AAa \to Acbaa \to *a * cbaa \to D * cba * a \to DA * a* \to$
   $DAA \to DA * a* \to DAD \to DB \to \sigma$
   Rút gọn lại ta có $\omega = acbaa \to Dcbaa \to DAa \to DAD \to DB \to \sigma$



Bài tập chương 2

- **1.** Giản lược ký hiệu thừa trong các văn phạm sau:

  a. $G = (\{a, b, c, d\}, \{S, A, B, T, K, V\}, S, P)$

  $P = \{S \to SA,\ A \to aA,\ A \to Tb,\ T \to cT,\ K \to bK,\ K \to d,$
  $S \to BS,\ B \to aB,\ B \to Sc,\ V \to d,\ S \to \wedge\}$

  b. $G = (\{a, b, c, d\}, \{\sigma, X, Y, M, N\}, \sigma, P)$

  $P = \{\sigma \to XY,\ X \to Xa,\ X \to bXM,\ M \to aM,\ M \to b,$
  $\sigma \to Ya,\ Y \to a\sigma,\ Y \to c, N \to d\}$

  c. $G = (\{a, b, c, d\}, \{S, A, B, C, K, F\}, S, P)$

  $P = \{S \to AB,\ A \to bB,\ A \to Fa,\ C \to cF,\ K \to bK,$
  $K \to dC,\ S \to BS,\ B \to a,\ F \to b\}$

- **2.** Loại bỏ quy tắc rỗng trong các văn phạm sau:

  a. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$

  $P = \{S \to Sb,\ S \to AcB,\ A \to \wedge,\ A \to BaBc, B \to \wedge\}$

  b. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$

  $P = \{S \to ASB,\ S \to Ac,\ A \to \wedge,\ A \to cBa, B \to \wedge\}$

  c. $G =< \{0, 1, 2\}, \{\sigma, A, B\}, \sigma, P >$

  $P = \{\sigma \to 01\sigma,\ \sigma \to B\sigma A,\ B \to AA1,\ B \to 2, B \to \wedge,\ A \to \wedge\}$

Bài tập chương 2

- **3.** Loại bỏ quy tắc đơn trong các văn phạm sau:
  a. $G =< \{a, b, c\}, \{S, A, B, C\}, S, P >$
  $P = \{S \to AbcB,\ S \to BAab,\ A \to B,\ A \to C, C \to a,$
  $A \to ca,\ B \to b\}$
  b. $G =< \{a, b, c\}, \{S, A, B, C\}, S, P >$
  $P = \{S \to AbC,\ A \to B,\ B \to C,\ C \to A,\ B \to Ab,\ A \to b,$
  $B \to cb,\ C \to ac\}$
  c. $G =< \{a, b, c\}, \{S, A, B, C\}, S, P >$
  $P = \{S \to ABc,\ A \to C,\ S \to C,\ C \to B,\ A \to a,\ B \to ac,$
  $S \to bA,\ C \to b\}$

- **4.** Đưa các văn phạm sau về dạng chuẩn Chomsky
  a. $G =< \{a, b, c, d\}, \{S, A, B\}, S, P >$
  $P = \{S \to bSa,\ S \to AcB,\ A \to Bac,\ A \to da,\ S \to a,\ B \to \wedge\}$
  b. $G =< \{a, b, c\}, \{S, K, T\}, S, P >$
  $P = \{S \to abST,\ T \to K,\ K \to aKb,\ T \to S,\ S \to ab,\ K \to c\}$
  c. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to ASB,\ S \to SaB,\ A \to Abc,\ B \to bc,\ S \to c,\ A \to \wedge\}$

Bài tập chương 2

- **5.** Khử quy tắc đệ quy trái trong các văn phạm sau:
  a. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to Sa,\ S \to b,\ A \to Ab,\ A \to ABc,\ A \to ca,$
  $A \to BaB,\ B \to Ba,\ B \to \wedge\}$
  b. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to baS,\ S \to AB,\ A \to AbA,\ A \to a,\ A \to b,$
  $A \to ASa,\ B \to Ba,\ B \to c\}$
  c. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to SAc,\ S \to Ba,\ A \to abB,\ A \to b,\ B \to c,$
  $A \to AbB,\ B \to Bc,\ B \to a\}$

- **6.** Đưa các văn phạm sau về dạng chuẩn Greiback
  a. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to BA,\ A \to Ba,\ B \to b,\ A \to b,\ B \to c,\ S \to a\}$
  b. $G =< \{a, b, c\}, \{S, A, B\}, S, P >$
  $P = \{S \to Sa,\ S \to a,\ S \to Bb,\ A \to Ba,\ A \to b,\ B \to a\}$

Chương 3 - Giới thiệu chung

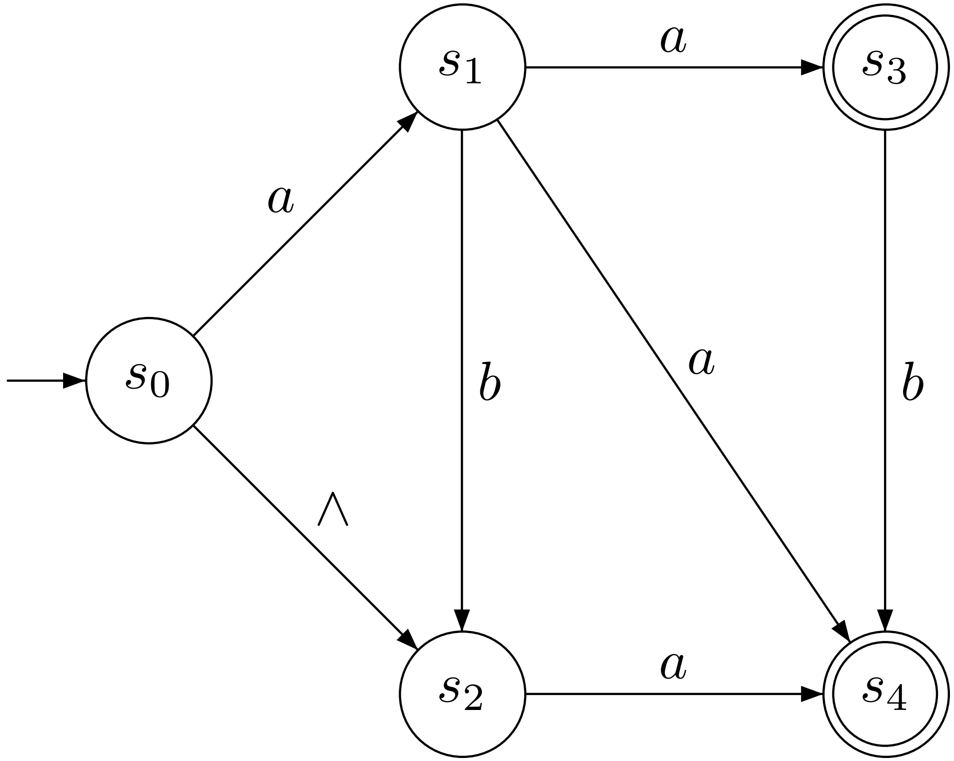
- Phần đầu của chương này trình bày về nguồn và các phép toán trên nguồn đó chính là dạng hình học của otomat hữu hạn.
- Phần hai cho chúng ta thấy mối quan hệ tương đương giữa nguồn với ôtomat hữu hạn, văn phạm chính quy và biểu thức chính quy. Mối quan hệ này không chỉ biểu hiện ở mặt sinh ra ngôn ngữ mà còn thể hiện ở các phép toán trên chúng. Phần này cũng cho ta một thuật toán cụ thể để tìm ngôn ngữ được sinh bởi một nguồn và do vậy nó cũng cho ta thuật toán xác định ngôn ngữ sinh ra văn phạm chính quy mà chương 1 ta chưa đề cập đến.
- Phần cuối cùng của chương 3 trình bày về điều kiện cần, điều kiện cần và đủ của ngôn ngữ chính quy nó cho phép ta có thể kiểm tra xem một ngôn ngữ có phải là ngôn ngữ chính quy hay không.

Nguồn và ngôn ngữ sinh bởi nguồn

- **Định nghĩa.** Nguồn I trên bảng chữ cái Σ là một đa đồ thị có hướng có khuyên trên đó có :
  - Một đỉnh vào $V(I)$ đánh dấu bằng mũi tên($\longrightarrow$) .
  - Một tập đỉnh ra $F(I)$ đánh dấu bằng cách đặt trong hình tròn kép, các đỉnh còn lại đặt trong hình tròn.
  - Trên mỗi cung ghi một ký hiệu thuộc $\Sigma \cup \{\wedge\}$.
- Cung trên đó ghi từ rỗng gọi là cung rỗng, còn lại là cung cốt yếu.
- Ký hiệu tập tất cả các đỉnh của nguồn I là $A(I)$.
- Nguồn I được gọi là đơn định nếu nó không chứa cung rỗng và hai cung bất kỳ xuất phát từ một đỉnh được ghi bằng hai ký hiệu khác nhau.
- Nguồn I được gọi là đầy đủ nếu $\forall a \in \Sigma, \ \ \forall s \in A(I)$ luôn tồn tại cung đi ra từ s và ghi a trên đó.
- Nguồn I được gọi là đơn định và đầy đủ nếu nó vừa đơn định vừa đầy đủ.

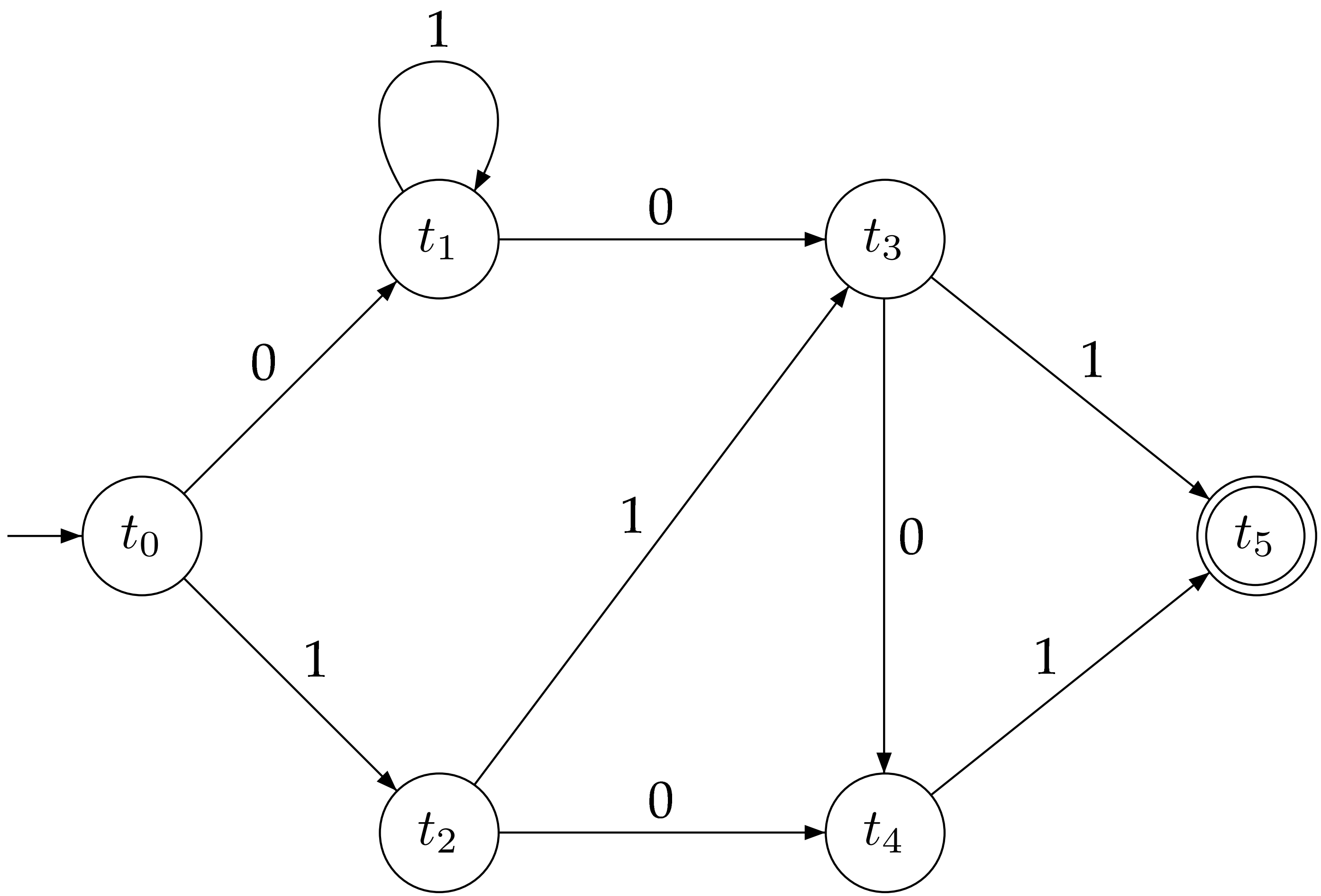
Các ví dụ về nguồn

- **Ví dụ 1**: Nguồn I_1



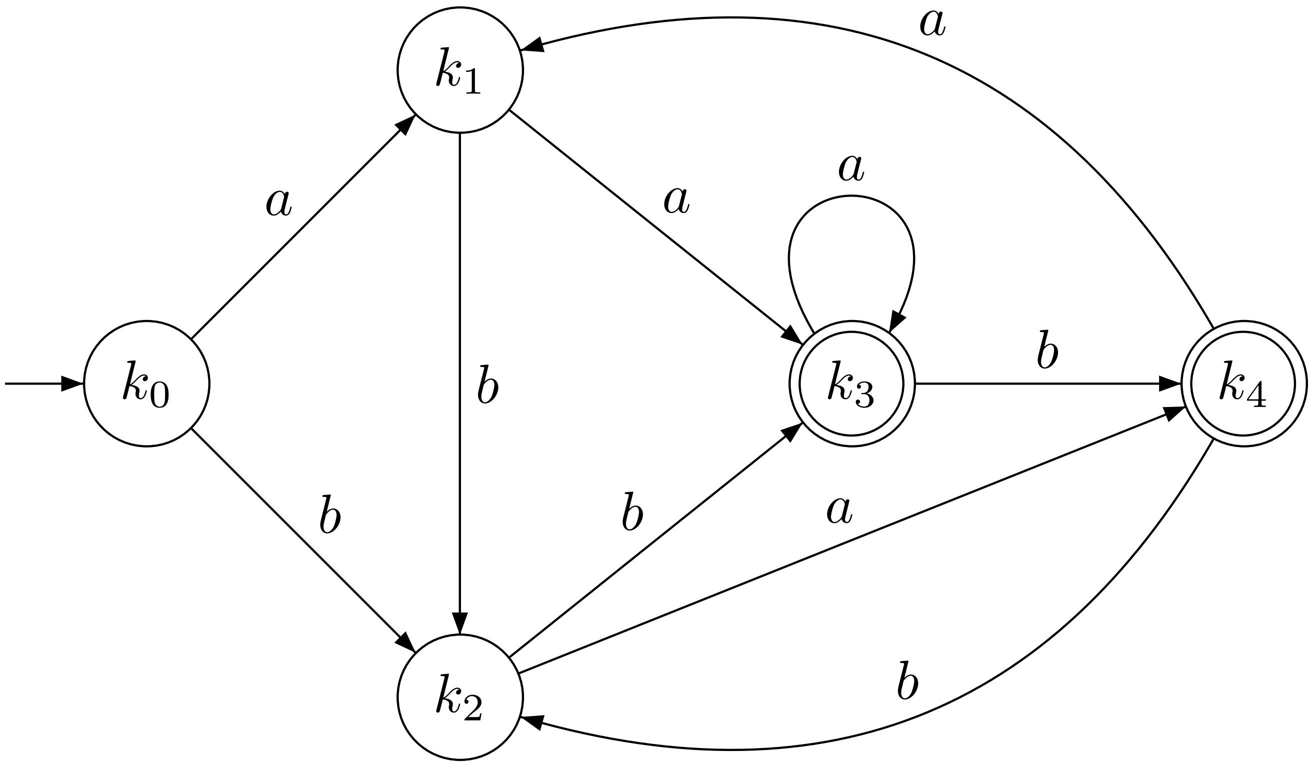
Các ví dụ về nguồn

- **Ví dụ 2**: Nguồn đơn định I_2



Các ví dụ về nguồn

- **Ví dụ 3**: Nguồn đơn định và đầy đủ I_3

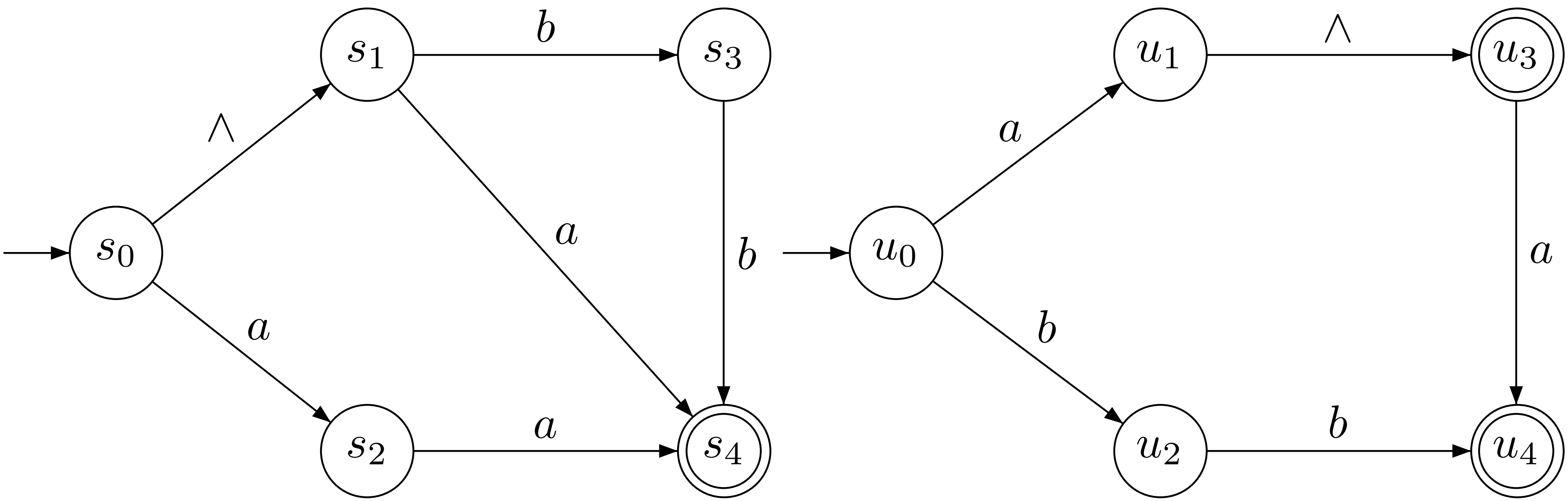


Ngôn ngữ sinh bởi nguồn

- **Định nghĩa.** Nếu $D = (s_0, s_1)(s_1, s_2)\ldots(s_{m-1}, s_m)$ là đường đi từ s_0 đến s_m trong I và $a_i \in \Sigma \cup \{\wedge\}$ là ký hiệu ghi trên cung (s_{i-1}, s_i) thì xâu $w = a_1a_2\ldots a_m$ được gọi là từ sinh bởi đường D.
- **Ký hiệu:**
  - $N_I(u, v)$ là tập tất cả các từ được sinh bởi tất cả các đường từ u đến v.
  - $T_I(s, a)$ là tập tất cả các đỉnh $u \in A(I)$ mà tồn tại đường đi từ s đến u sinh ra từ a, với $s \in A(I), a \in \Sigma \cup \{\wedge\}$.
    Như vậy $T_I(s, a) = \{u \in A(I) | a \in N_I(s, u)\}$.
  - $H_I(M, a)$ là tập tất cả các đỉnh $u \in A(I)$ mà tồn tại đường đi từ một đỉnh nào đó trong M đến u sinh ra từ a, tức là
    $H_I(M, a) = \bigcup_{s \in M} T_I(s, a)$, với $M \subset A(I), a \in \Sigma \cup \{\wedge\}$.
- Trong ví dụ 1 ở nguồn I_1 ta có:
  - $N_I(s_0, s_4) = \{a, aba, aa, aab\}$
  - $T_I(s_0, a) = \{s_1, s_4\}$
  - $H_I(\{s_1, s_3\}, b) = \{s_2, s_4\}$

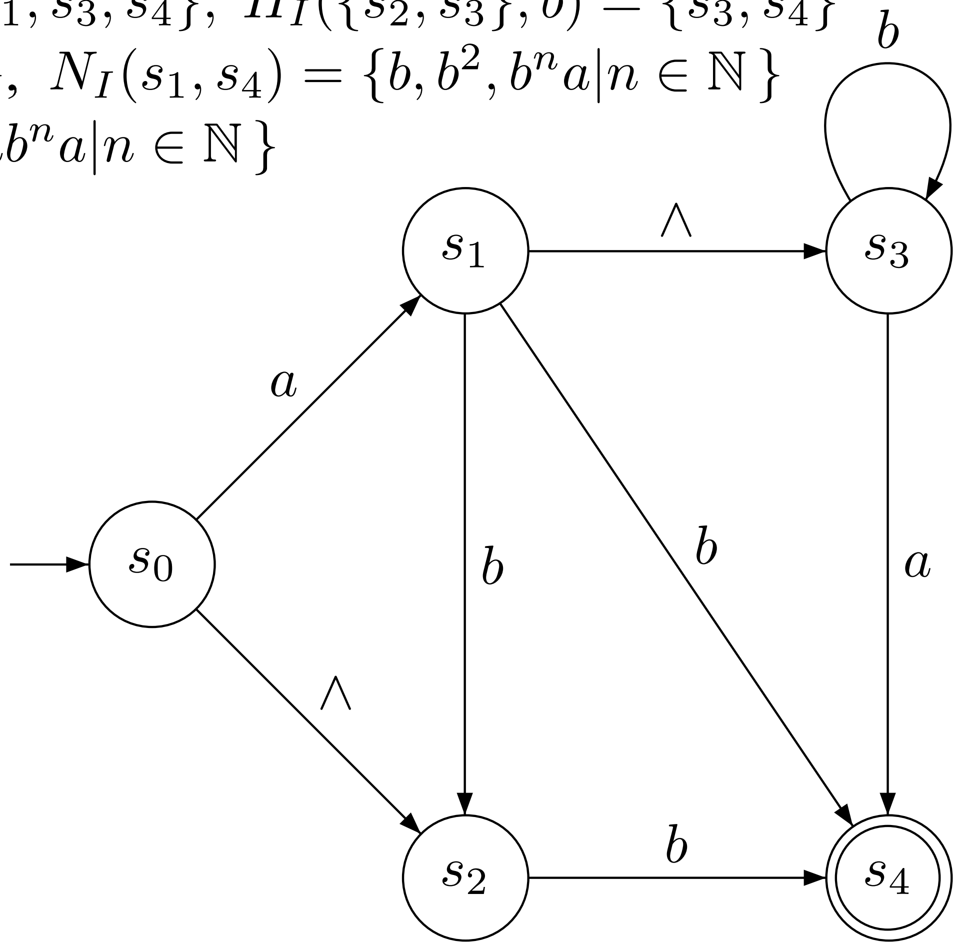
Ngôn ngữ sinh bởi nguồn

- **Định nghĩa.** Tập hợp tất cả các từ được sinh bởi tất cả các đường đi từ đỉnh vào $V(I)$ đến đỉnh kết thuộc $F(I)$ được gọi là ngôn ngữ sinh bởi nguồn I, ký hiệu là $N(I)$.
  Như vậy $N(I) = \{x \in \Sigma^* | x \in N_I(V(I), s), s \in F(I)\}$.
- **Định nghĩa.** Nguồn I và nguồn M được gọi là tương đương nếu chúng sinh ra cùng một ngôn ngữ, tức là $N(I) = N(M)$.
- Hai nguồn sau tương đương vì chúng cùng sinh ra ngôn ngữ $\{a, aa, bb\}$



Ngôn ngữ sinh bởi nguồn - Ví dụ

- Với nguồn I như sau ta có:
$T_I(s_0, a) = \{s_1, s_3\}$, $T_I(s_1, b) = \{s_3, s_4\}$, $T_I(s_0, \wedge) = \{s_2\}$
$H_I(\{s_0, s_3\}, a) = \{s_1, s_3, s_4\}$, $H_I(\{s_2, s_3\}, b) = \{s_3, s_4\}$
$N_I(s_0, s_2) = \{\wedge, ab\}$, $N_I(s_1, s_4) = \{b, b^2, b^n a | n \in \mathbb{N}\}$
$N(I) = \{b, ab^2, ab, ab^n a | n \in \mathbb{N}\}$

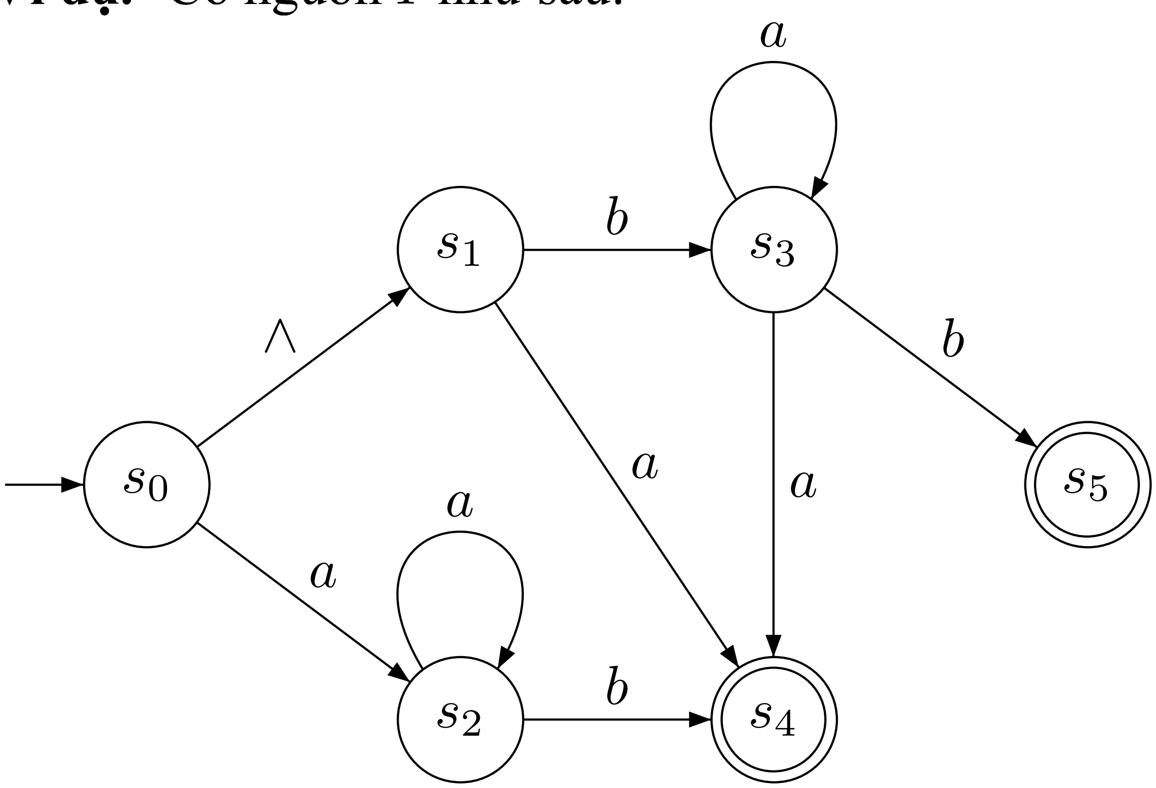


Các phép toán trên nguồn

- Phép đơn định hóa
- Phép lấy phần bù
- Phép hợp nguồn
- Phép giao nguồn
- Phép tích ghép
- Phép lặp
- Phép soi gương
- Phép chia trái
- Phép chia phải

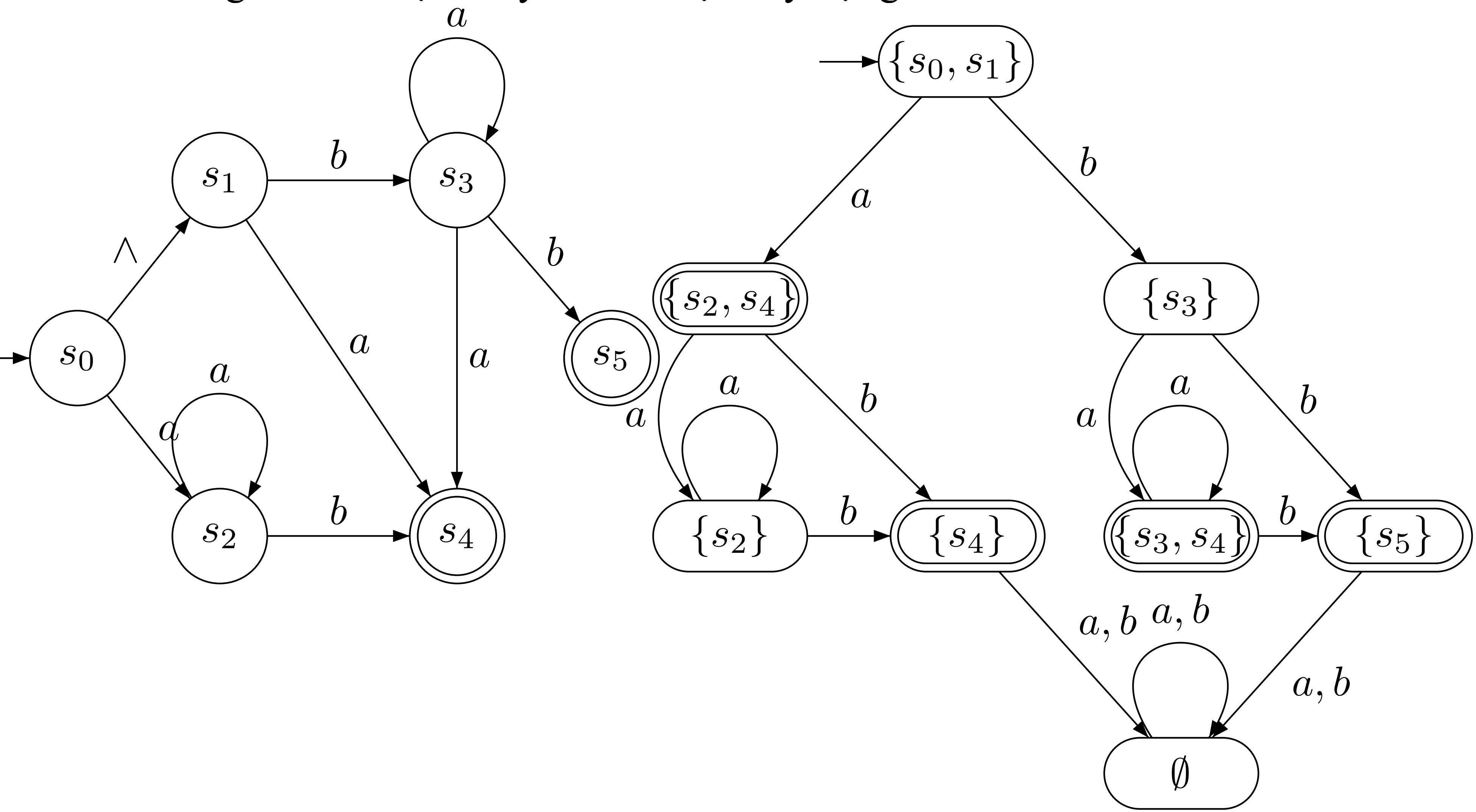
Phép đơn định hóa

- **Định nghĩa.** Nguồn K đơn định, đầy đủ tương đương với I được gọi là nguồn đơn định hóa của I.
- **Bài toán.** Cho trước nguồn I, cần xây dựng nguồn đơn định hóa của I.
- **Ví dụ.** Có nguồn I như sau:



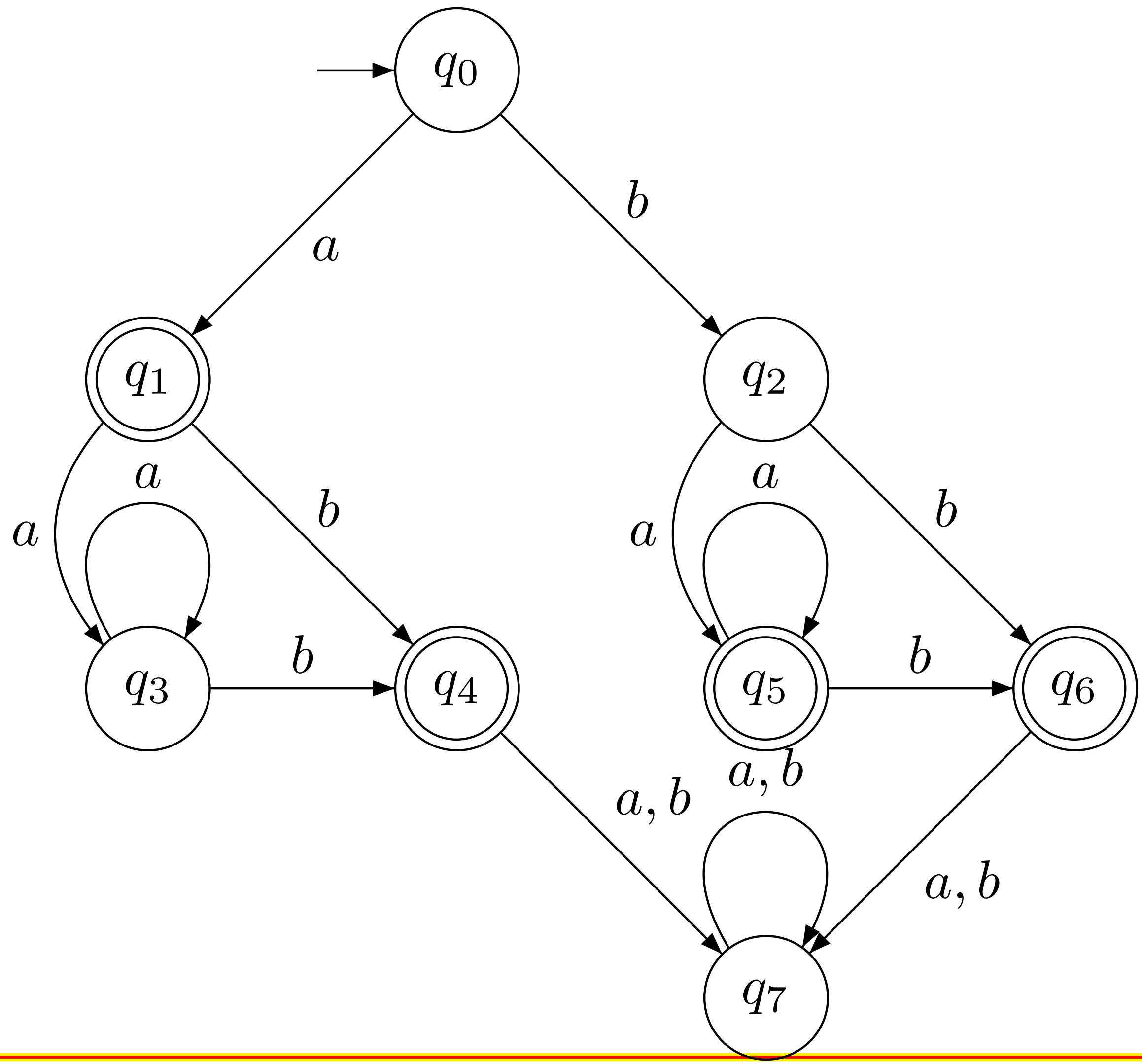
Phép đơn định hóa

- Nguồn đơn định đầy đủ K được xây dựng như sau:



Phép đơn định hóa

Bằng cách đặt tên đỉnh mới ta có nguồn thu gọn sau:



Thuật toán đơn định hóa

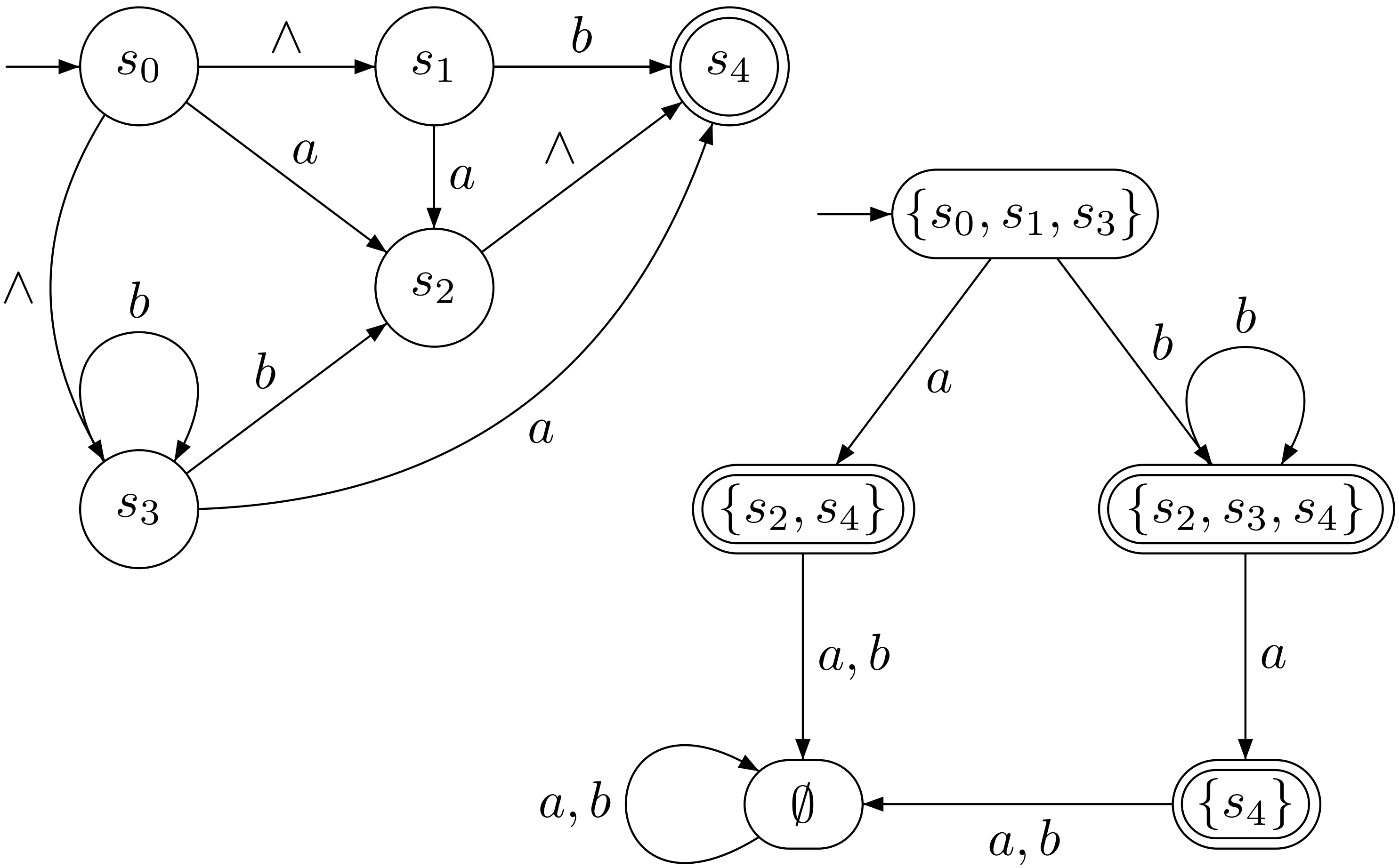
- Khởi tạo:
  - $V(K) = \{V(I)\} \cup T_I(V(I), \wedge)$
  - $A(K) = \{V(K)\}, T_I(s,a) = \{u \in A(I) | a \in N_I(s,u)\}$
  - $\forall M \subset A(I), H_I(M,a) = \bigcup_{s \in M} T_I(s,a)$
- Các bước của thuật toán:
  - Bước 1:
    - Nếu $\exists M \in A(K), \exists a \in \Sigma$ mà chưa có cung đi ra từ M ghi a trên đó thì xác định tập $H_I(M,a)$:

      Nếu $H_I(M,a) \notin A(K)$ thì xác định đỉnh mới là $H_I(M,a)$, nối cung từ M đến $H_I(M,a)$ và ghi a trên cung đó rồi quay lại bước 1.

      Nếu $H_I(M,a) \in A(K)$ thì nối cung từ M đến $H_I(M,a)$ và ghi a trên cung đó rồi quay lại bước 1.
    - Nếu $\forall M \in A(K), \forall a \in \Sigma$ đều có cung đi ra từ M với ký hiệu a trên đó thì chuyển sang bước 2.
  - Bước 2: $F(K) = \{M \in A(K) | M \cap F(I) \neq \emptyset\}$

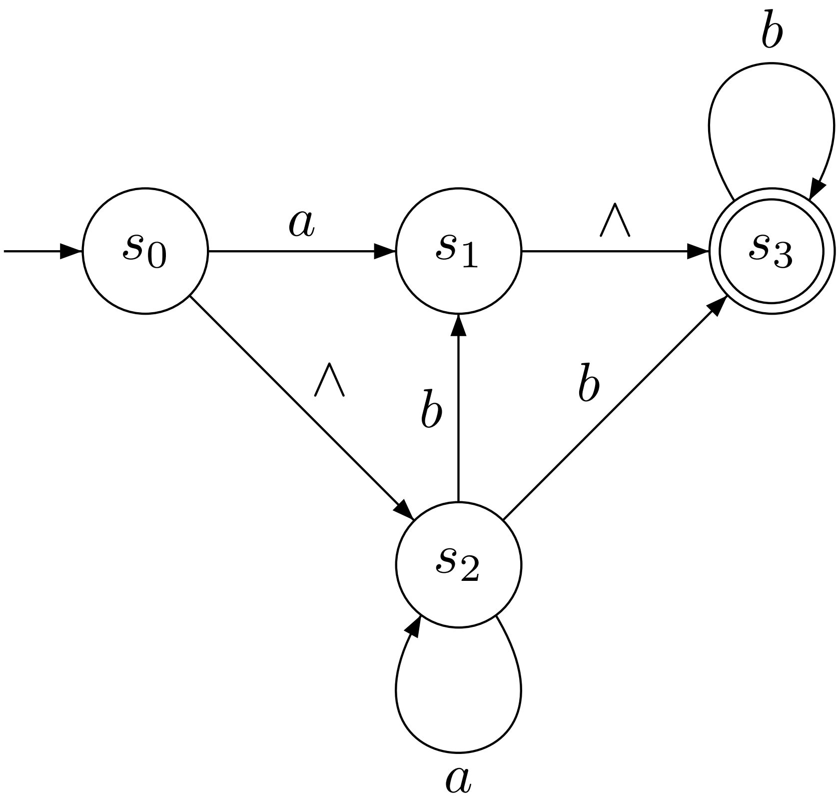
Ví dụ về thuật toán đơn định hóa

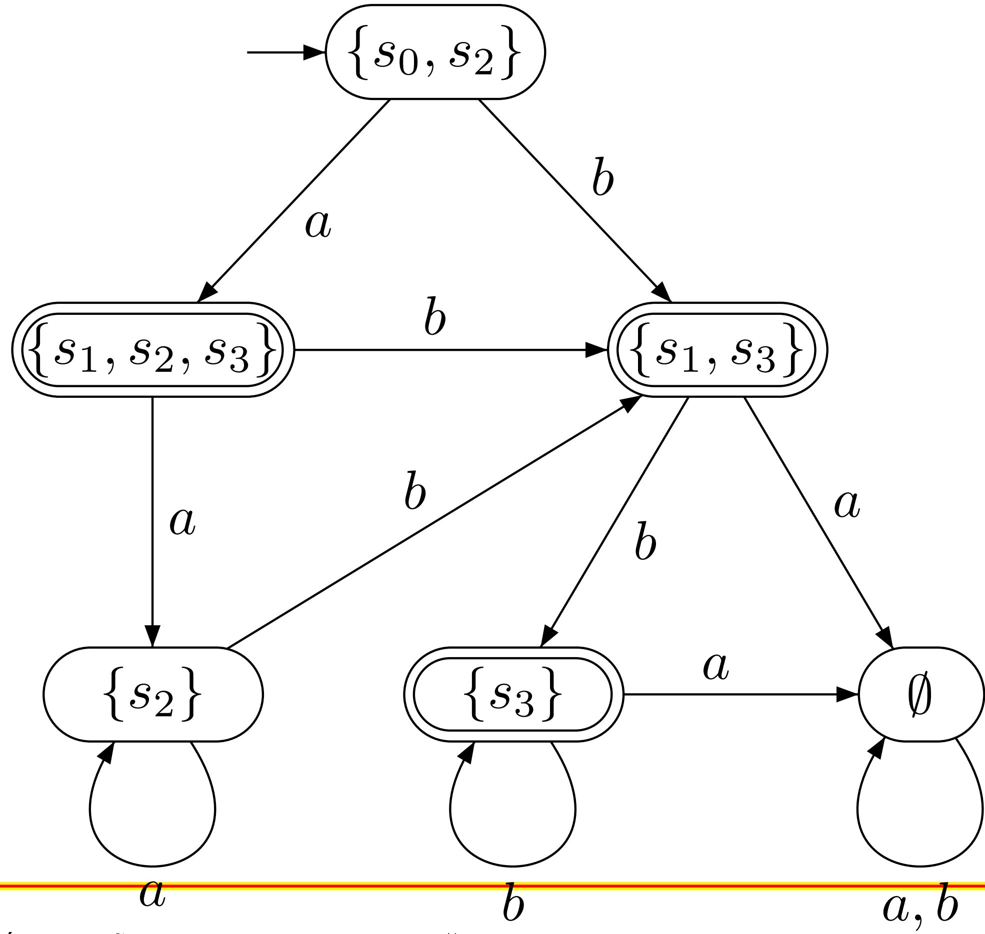
- Cho nguồn I
  Đặt $q_0 = \{s_0, s_1, s_3\}, q_1 = \{s_2, s_4\}, q_2 = \{s_2, s_3, s_4\}, q_3 = \emptyset, q_4 = \{s_4\}$
  ta có nguồn K đơn định, đầy đủ tương đương với I



Phép lấy phần bù

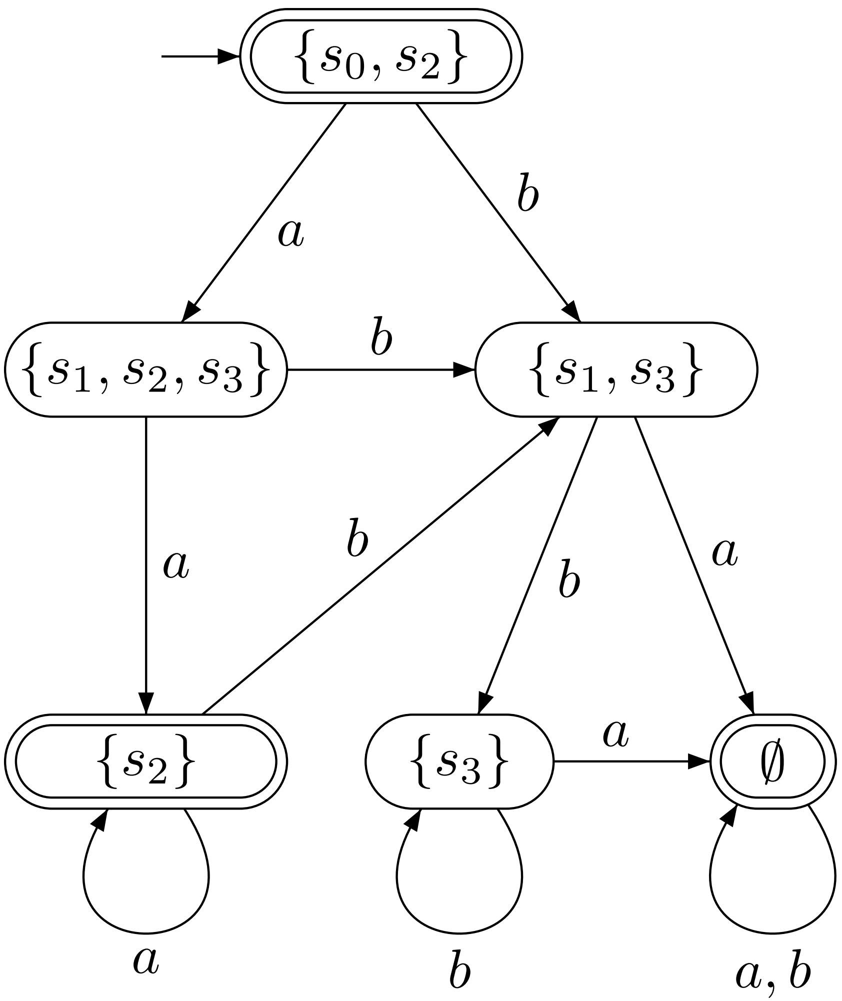
- **Định nghĩa.** Nguồn K được gọi là nguồn bù của I nếu
  $$N(K) = C_{\Sigma^*}N(I).$$
- **Bài toán.** Cho trước nguồn I, cần xây dựng nguồn bù K của I.
- **Ví dụ.** Có nguồn I và nguồn đơn định đầy đủ tương đương với I :

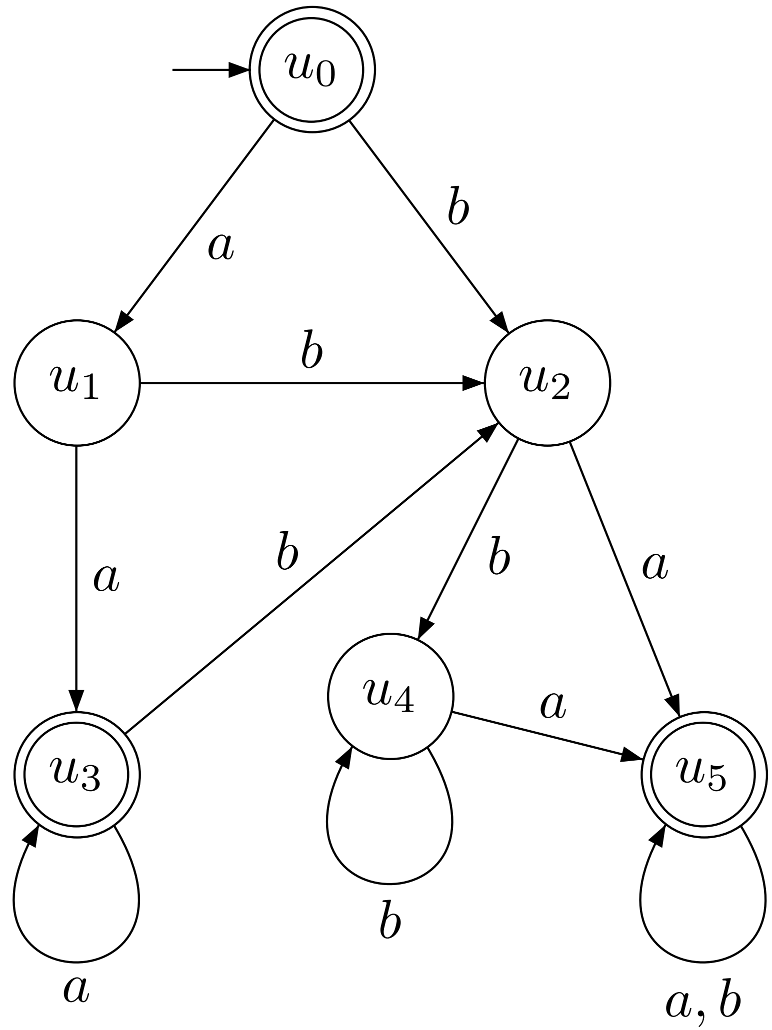




Phép lấy phần bù - Ví dụ

- Hình vẽ thu gọn nguồn bù K của I:



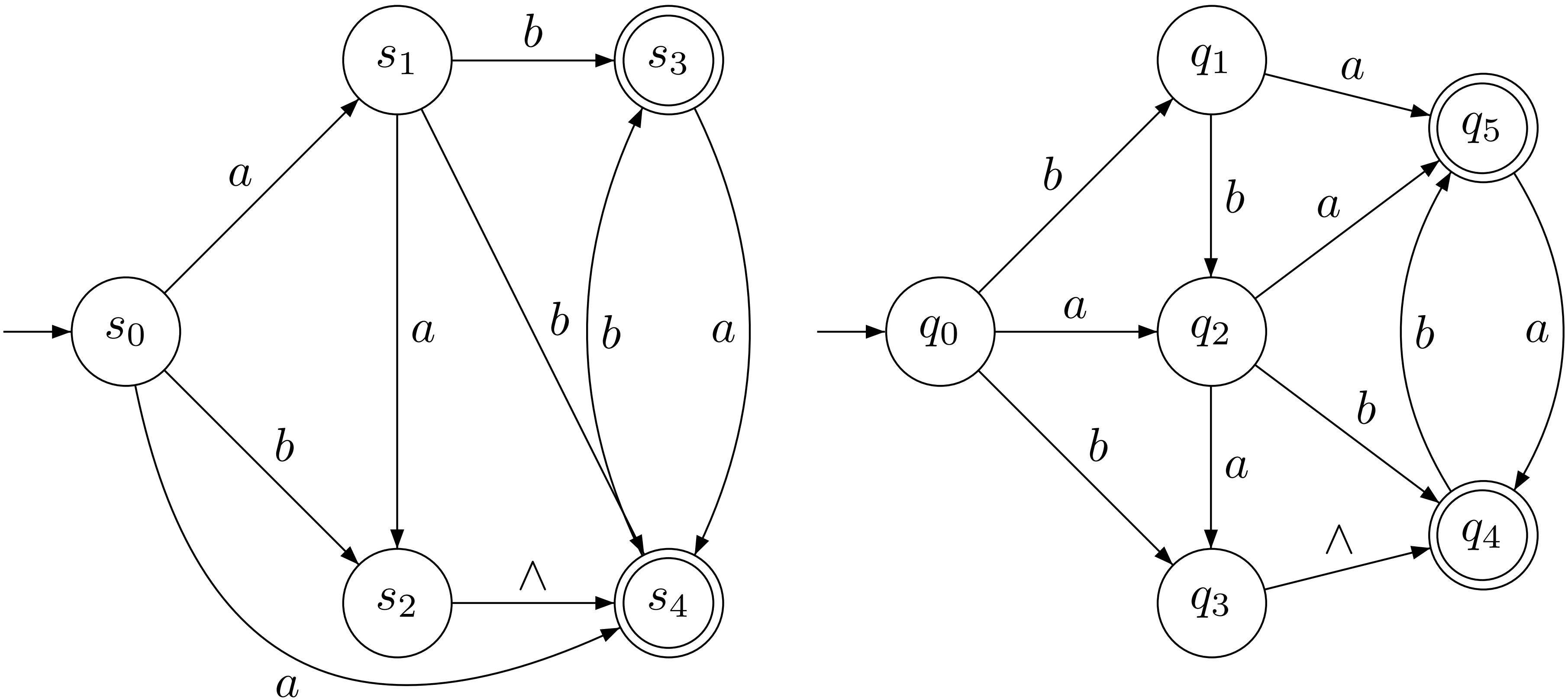


Thuật toán xây dựng nguồn bù

- Nếu I chưa đơn định, đầy đủ thì xây dựng nguồn K đơn định đầy đủ tương với I, tức là $N(K) = N(I)$.
- Trên nguồn K đổi các đỉnh kết thành không kết và ngược lại ta có nguồn bù M của I.

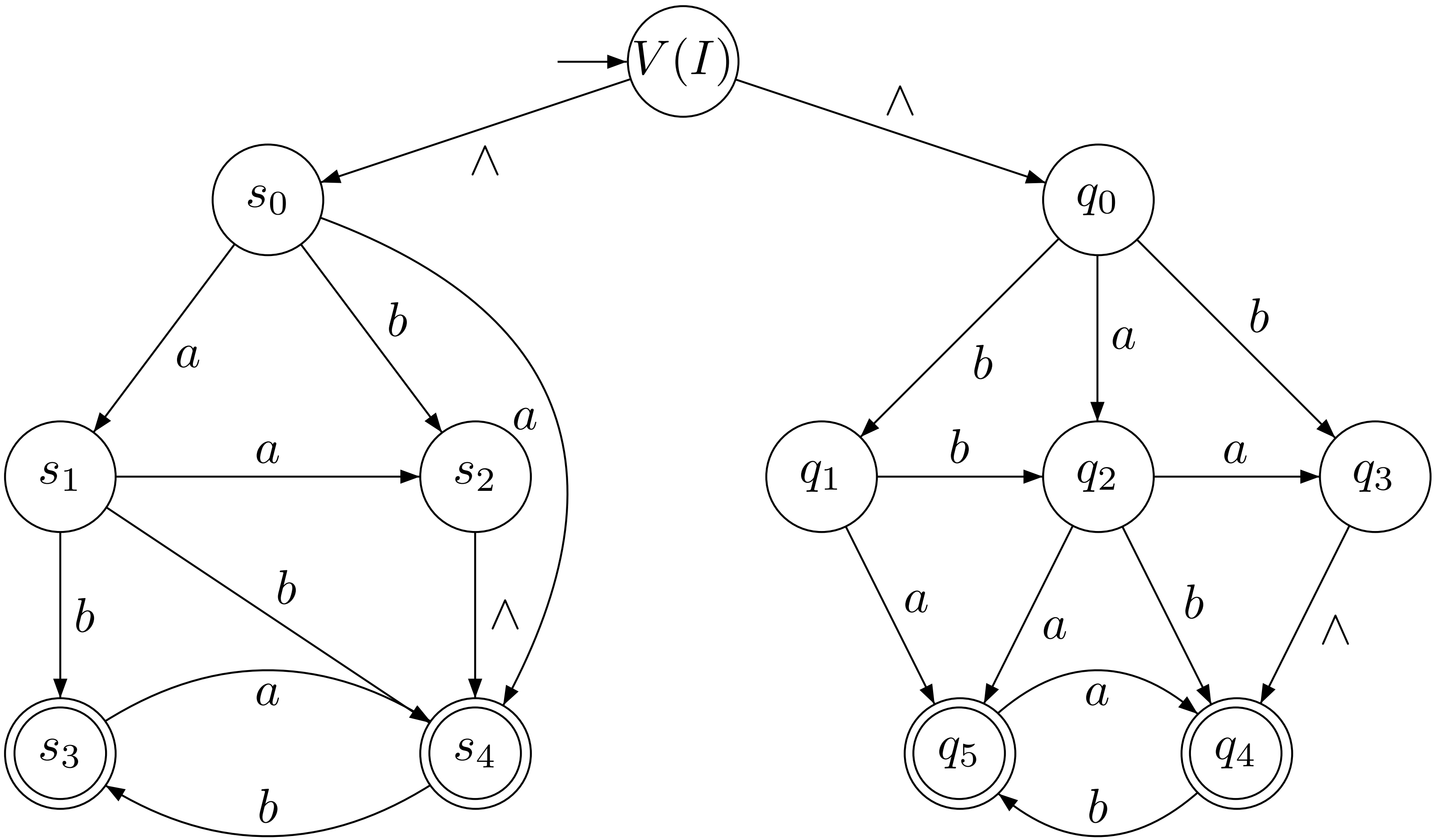
Phép hợp nguồn

- **Định nghĩa.** Nguồn I được gọi là hợp của các nguồn I_1 và I_2 nếu
$$N(I) = N(I_1) \cup N(I_2).$$
- **Bài toán.** Cho hai nguồn I_1, I_2, cần xây dựng nguồn hợp của hai nguồn này.
- **Ví dụ.** Cho các nguồn I_1 và I_2 như sau:



Phép hợp nguồn - Ví dụ

- Nguồn hợp I của I_1, I_2 là:

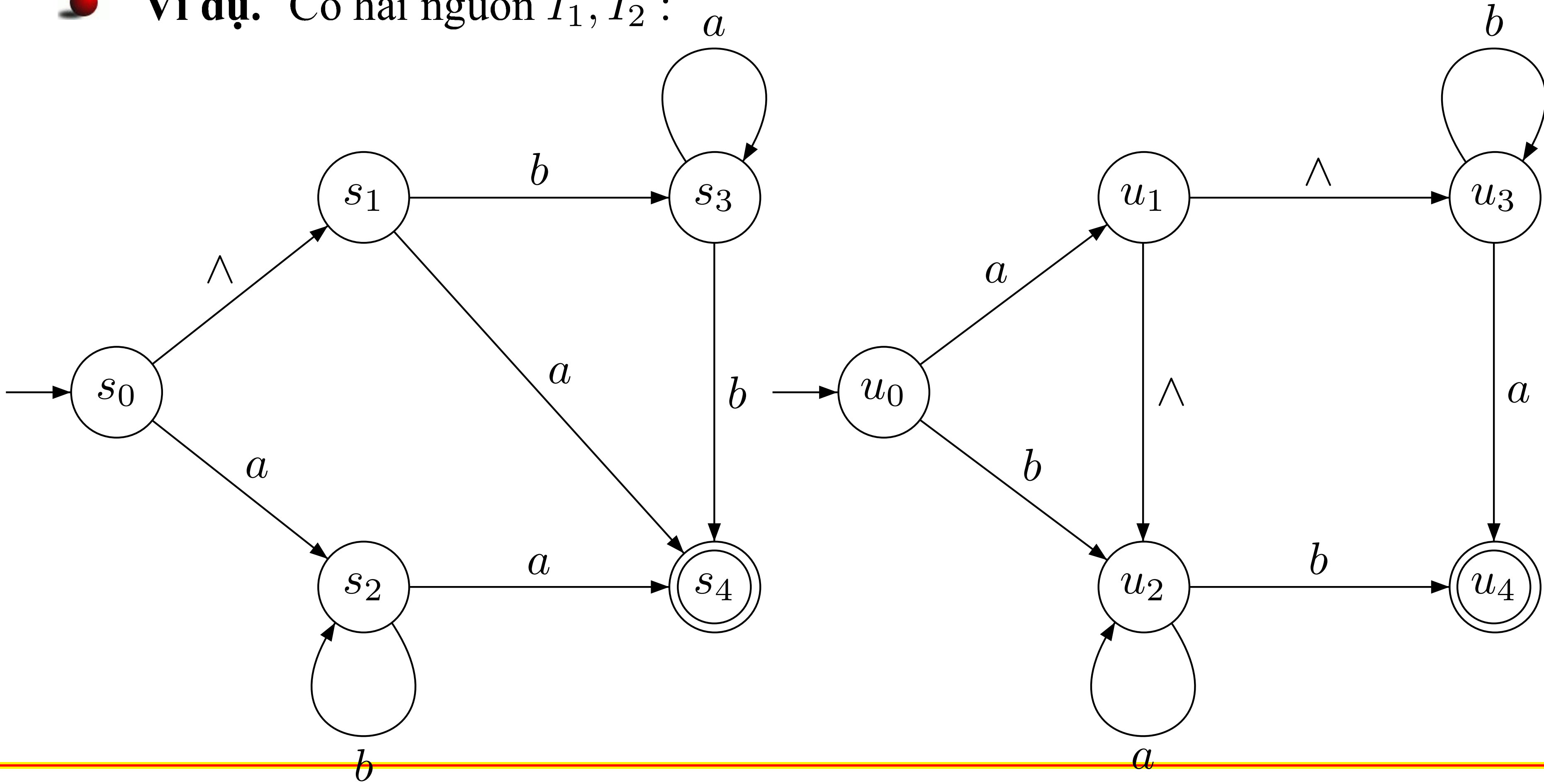


Thuật toán xây dựng nguồn hợp

- Giữ nguyên cấu trúc của I_1 và I_2.
- Thêm đỉnh mới $V(I)$ thừa nhận là đỉnh vào của I.
- Kẻ hai cung rỗng từ $V(I)$ đến $V(I_1)$ và $V(I_2)$.
- Tập đỉnh kết $F(I) = F(I_1) \cup F(I_2)$.

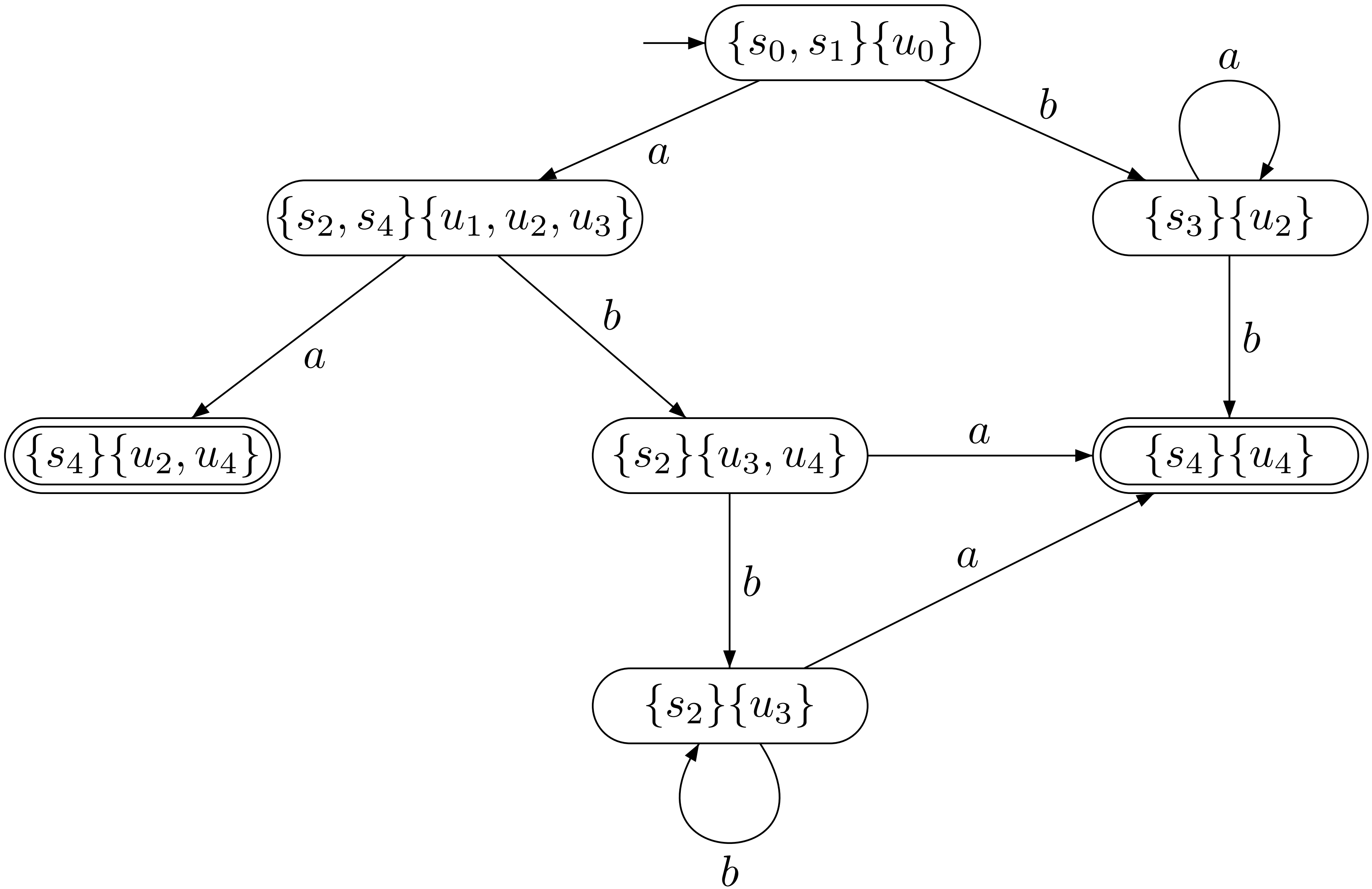
Phép giao nguồn

- **Định nghĩa.** Nguồn I được gọi là nguồn là giao của các nguồn I_1 và I_2 nếu $N(I) = N(I_1) \cap N(I_2)$.
- **Bài toán.** Cho hai nguồn I_1, I_2, cần xây dựng nguồn giao của chúng.
- **Ví dụ.** Có hai nguồn I_1, I_2 :



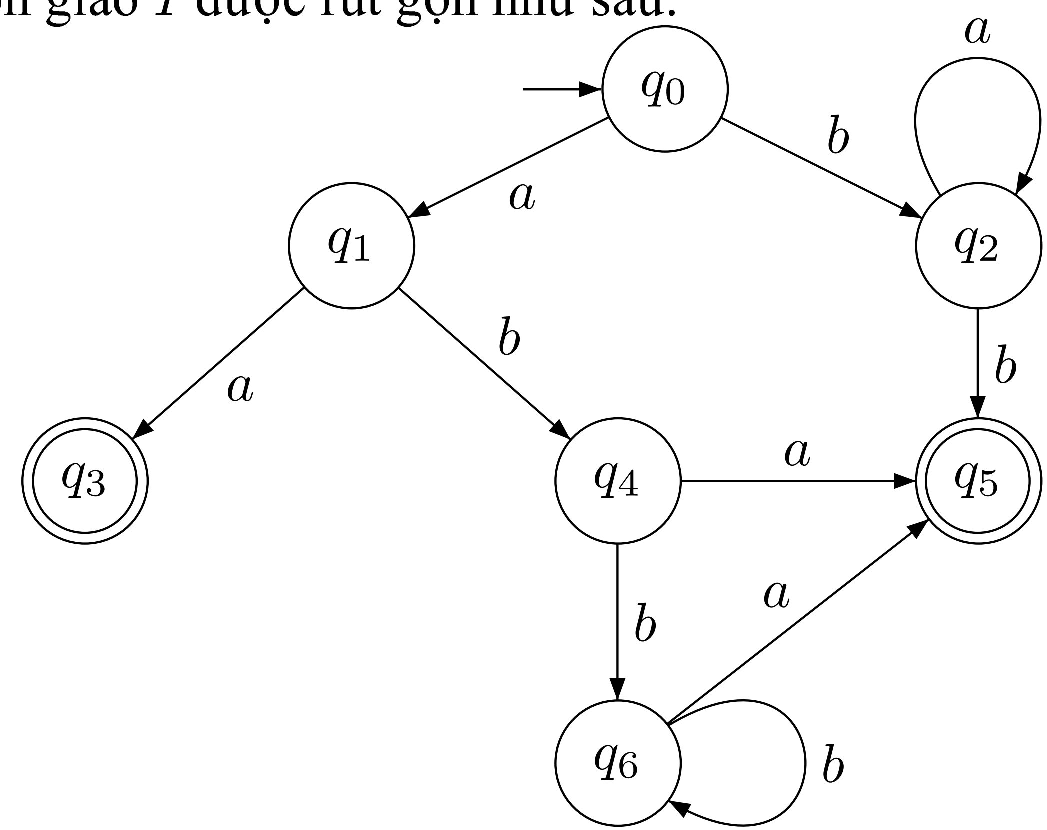
Phép giao nguồn

- Nguồn giao I được xây dựng như sau:



Phép giao nguồn

- Bằng cách đặt:
  $q_0 = (\{s_0, s_1\}, \{u_0\}),\ q_1 = (\{s_2, s_4\}, \{u_1, u_2, u_3\}),\ q_2 = (\{s_3\}, \{u_2\})$
  $q_3 = (\{s_4\}, \{u_2, u_4\}),\quad q_4 = (\{s_2\}, \{u_3, u_4\}),\quad q_5 = (\{s_4\}, \{u_4\})$
  $q_6 = (\{s_2\}, \{u_3\})$
- Nguồn giao I được rút gọn như sau:



Thuật toán xây dựng nguồn giao

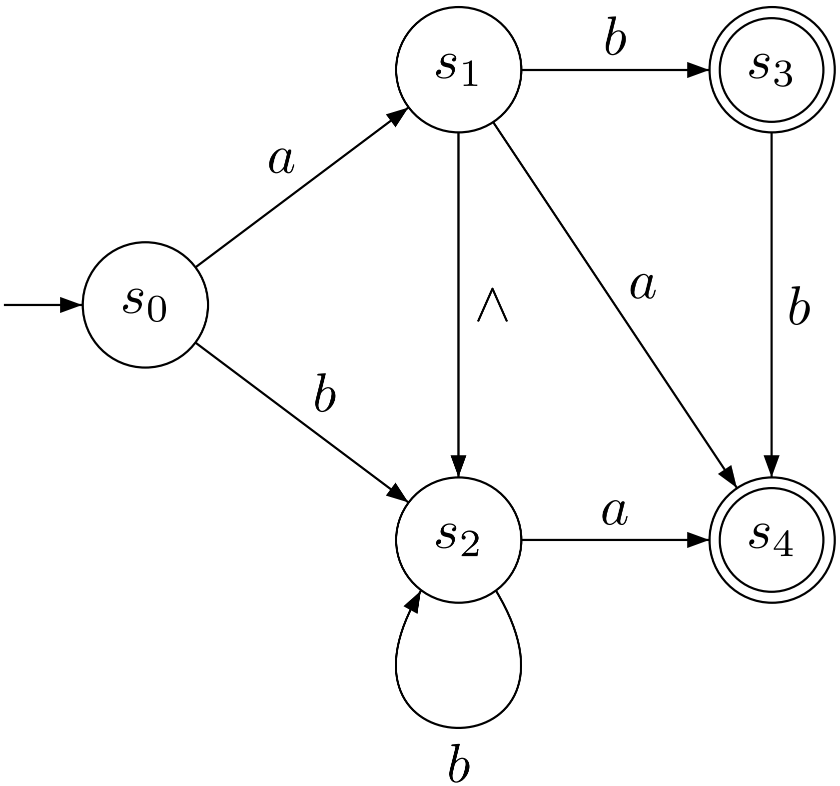
Giả sử I_1, I_2 là hai nguồn cho trước, nguồn giao I của I_1, I_2 được xây dựng như sau: Khởi tạo $P_0 = \{u \in A(I_1) | \wedge \in N(V(I_1), u)\}$,
$Q_0 = \{v \in A(I_2) | \wedge \in N(V(I_2), v)\}$, $V(I) = (P_0, Q_0)$, $A(I) = \{V(I)\}$

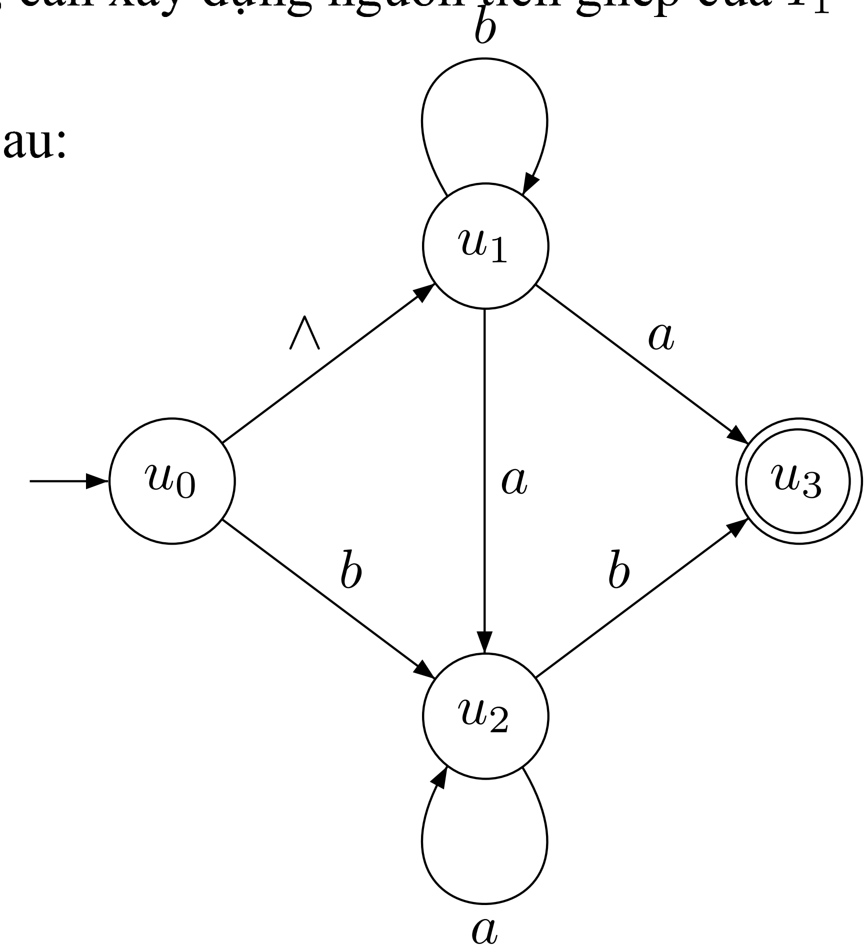
- Bước 1:
  - Nếu $\exists B = (P, Q) \in A(I)$ và $\exists a \in \Sigma$ mà chưa có cung đi từ B với nhãn a thì xác định đỉnh $C = (H_{I_1}(P, a), H_{I_2}(Q, a))$.
    - Nếu $C \in A(K)$ thì nối cung từ B đến C với nhãn a rồi quay lại bước 1.
    - Nếu $C \notin A(K)$ và $H_{I_1}(P, a), H_{I_2}(Q, a) \neq \emptyset$ thì đưa C vào $A(K)$, nối cung từ B đến C với nhãn a rồi quay lại bước 1.
  - Nếu $\forall B \in A(I), \forall a \in \Sigma$ đều có cung đi từ B với nhãn a thì chuyển sang bước 2.
- Bước 2: Xác định tập đỉnh kết:
  $F(I) = \{B = (P, Q) \in A(I) \, | P \cap F(I_1) \neq \emptyset, Q \cap F(I_2) \neq \emptyset\}$

Nhận xét: Do $A \cap B = \overline{\overline{A} \cup \overline{B}}$ nên ta có cách khác để xây dựng nguồn giao.

Phép tích ghép

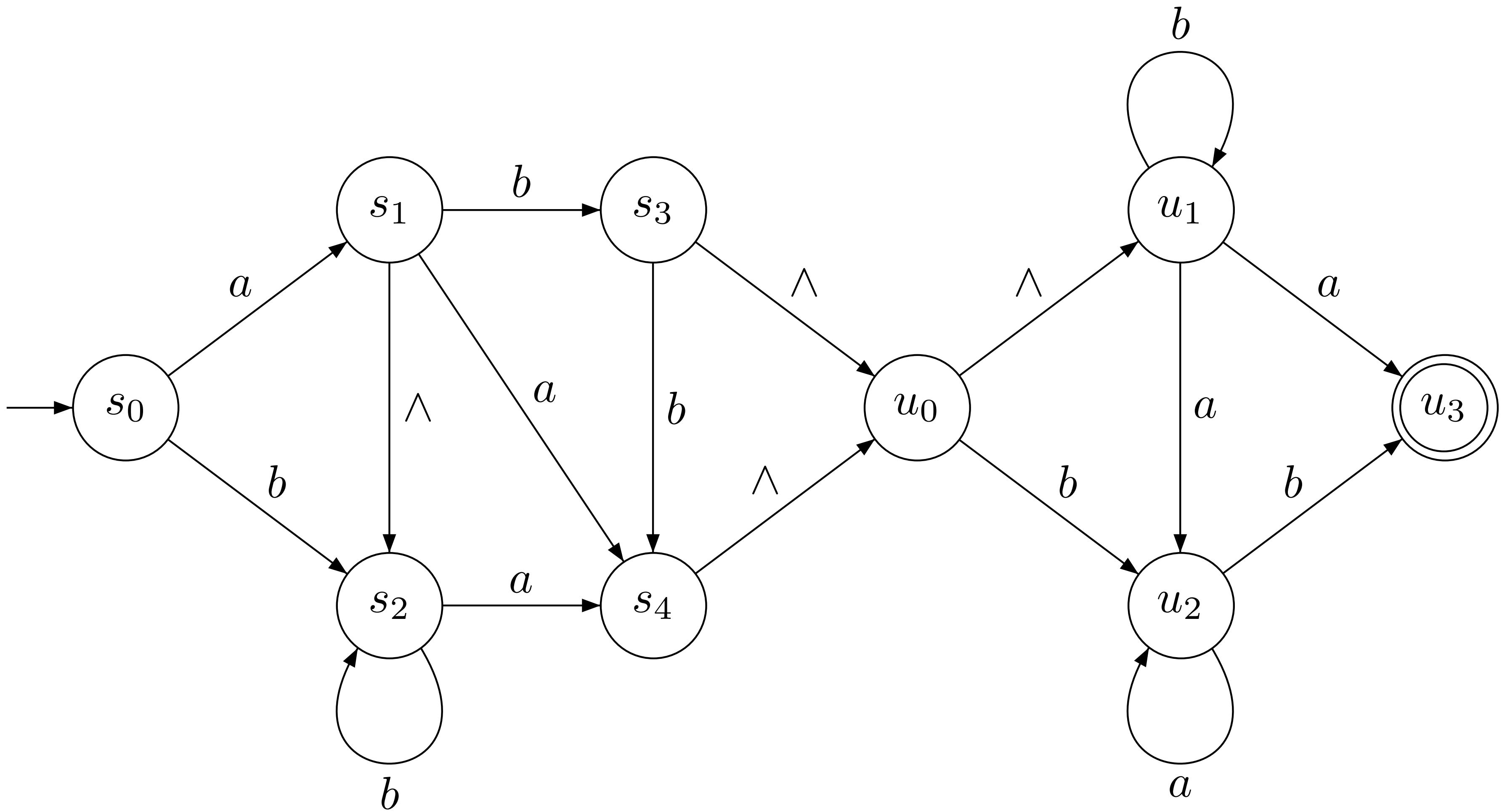
- **Định nghĩa.** Nguồn I được gọi là nguồn tích ghép của các nguồn I_1 vói I_2 nếu $N(I) = N(I_1).N(I_2)$.
- **Bài toán.** Cho hai nguồn I_1 và I_2, cần xây dựng nguồn tích ghép của I_1 với I_2.
- **Ví dụ.** Cho hai nguồn I_1, I_2 như sau:





Phép tích ghép

- Nguồn tích ghép I của hai nguồn I_1, I_2 được xây dựng như sau:

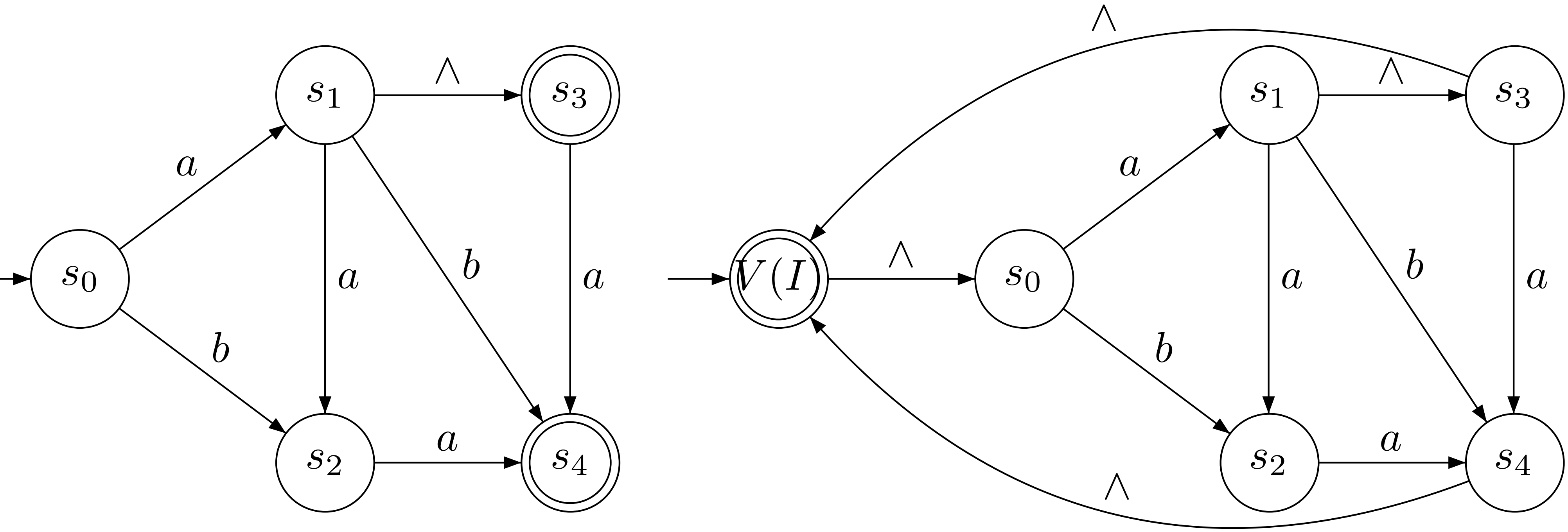


Phép tích ghép

- Thuật toán xây dựng nguồn tích ghép của hai nguồn I_1, I_2:
  - Giữ nguyên cấu trúc của I_1 và I_2.
  - Từ mỗi đỉnh kết của I_1 kẻ một cung rỗng đến $V(I_2)$.
  - Đặt đỉnh vào của I là đỉnh vào của I_1: $V(I) = V(I_1)$.
  - Đặt tập đỉnh kết của I là tập đỉnh kết của I_2: $F(I) = F(I_2)$.

Phép lặp

- **Định nghĩa.** Nguồn I được gọi là nguồn lặp của nguồn I_1 nếu $N(I) = (N(I_1))^*$.
- **Bài toán.** Cho nguồn I_1, cần xây dựng I là nguồn lặp của nguồn I_1.
- **Ví dụ.** Cho nguồn I_1 và nguồn lặp I của nó được xây dựng như sau:

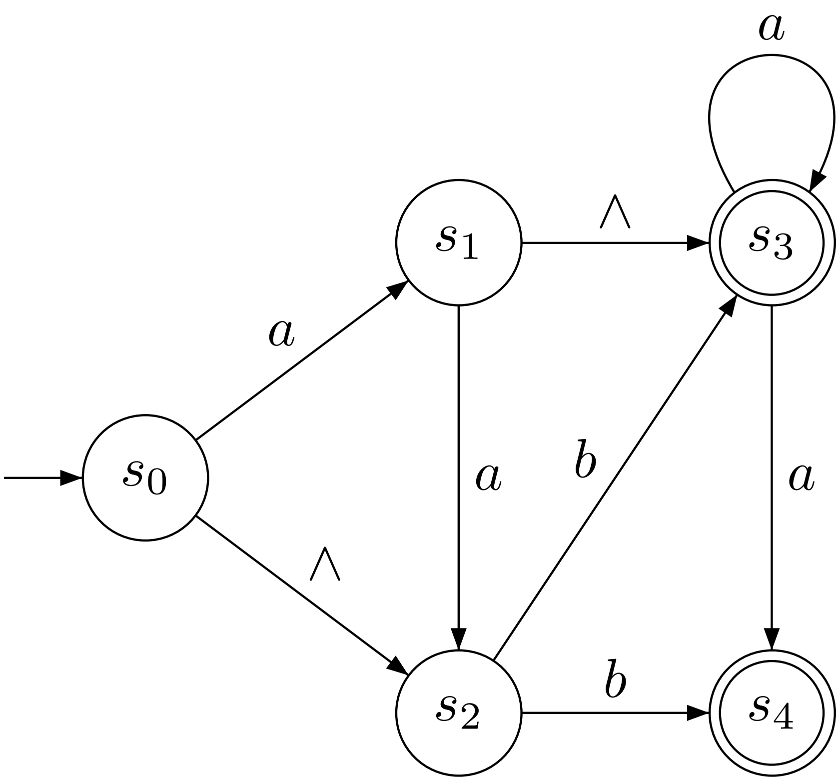


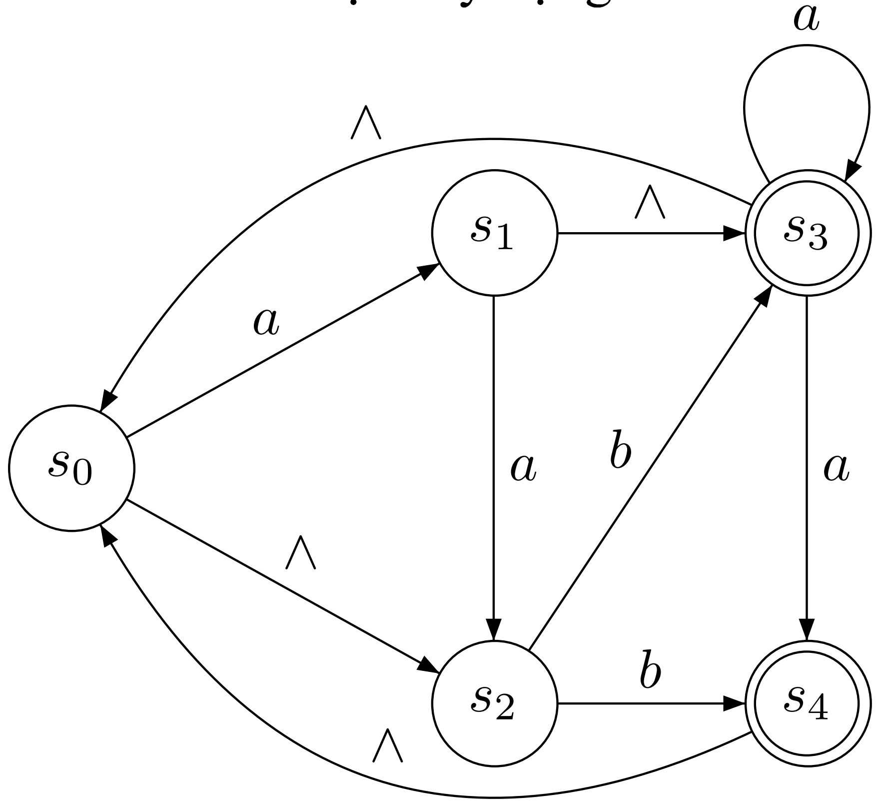
Thuật toán xây dựng nguồn lặp

- Giữ nguyên cấu trúc của I_1.
- Thêm vào đỉnh mới $V(I)$ là đỉnh vào của I.
- Kẻ một cung rỗng từ $V(I)$ đến $V(I_1)$.
- Từ mỗi đỉnh kết của I_1 kẻ một cung rỗng đến $V(I)$.
- $V(I)$ cũng là đỉnh kết duy nhất của I.

Phép lặp cắt

- **Định nghĩa.** Nguồn I được gọi là nguồn lặp cắt của nguồn I_1 nếu
$$N(I) = (N(I_1))^+.$$
- **Bài toán.** Cho nguồn I_1, cần xây dựng I là nguồn lặp cắt của nguồn I_1.
- **Ví dụ.** Cho nguồn I_1 và nguồn lặp cắt I của nó được xây dựng như sau:



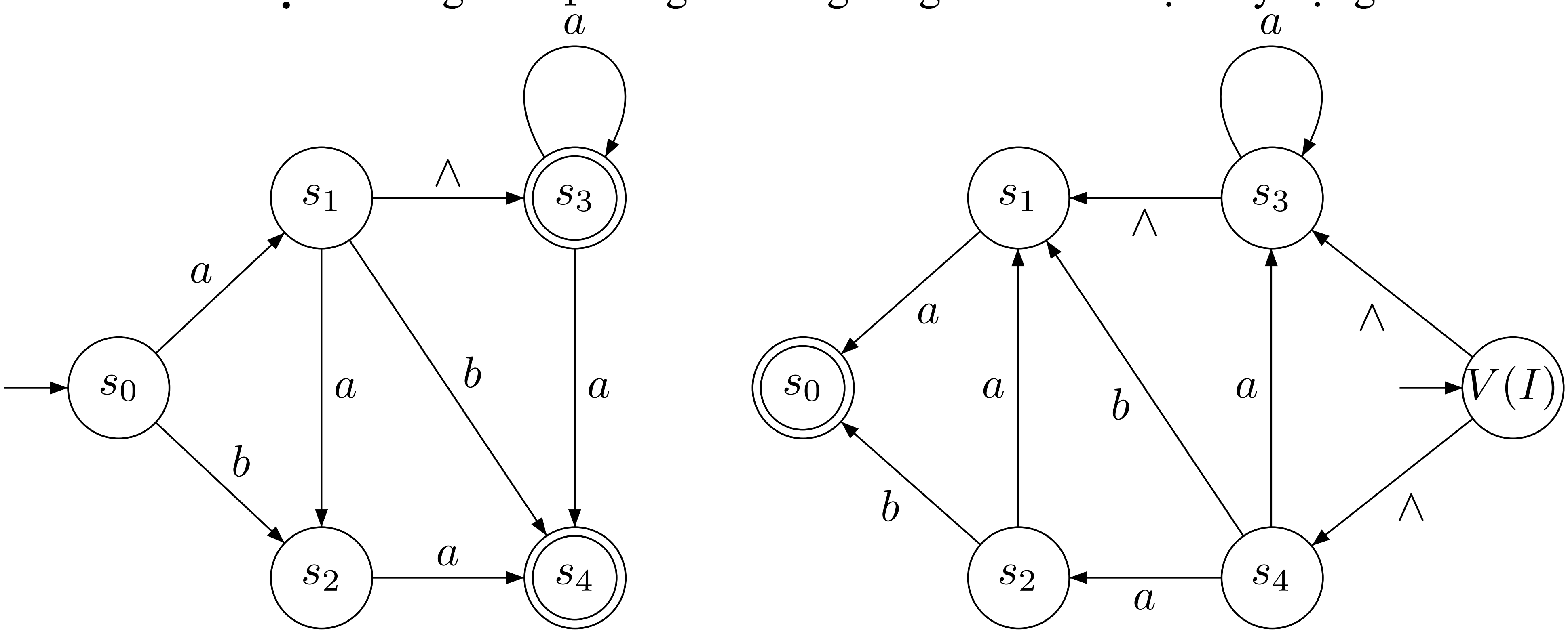


Thuật toán xây dựng nguồn lặp cắt

- Giữ nguyên cấu trúc của I_1.
- Từ mỗi đỉnh kết của I_1 kẻ một cung rỗng đến $V(I_1)$.
- Đỉnh vào và đỉnh kết của I giống như I_1.

Phép soi gương

- **Định nghĩa.** Nguồn I được gọi là nguồn soi gương của nguồn I_1 nếu
$$N(I) = \widetilde{N(I_1)}.$$
- **Bài toán.** Cho nguồn I_1, cần xây dựng nguồn soi gương I của I_1.
- **Ví dụ.** Cho nguồn I_1 và nguồn soi gương I của nó được xây dựng như sau:

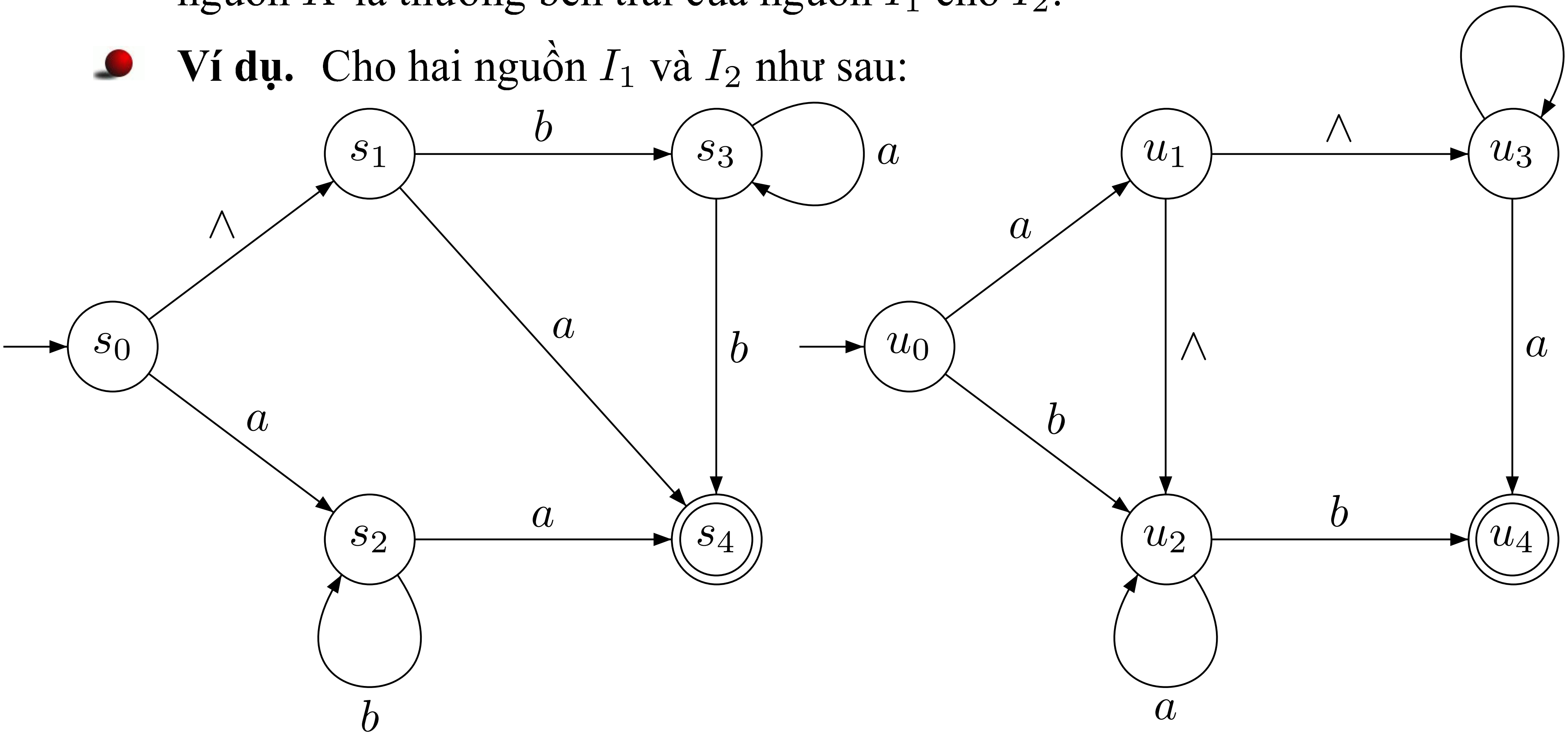


Thuật toán xây dựng nguồn soi gương

- Thêm vào I_1 một đỉnh mới $V(I)$ là đỉnh vào của nguồn I.
- Kẻ thêm một cung rỗng từ $V(I)$ đến các đỉnh kết của nguồn I_1.
- Đối với mỗi cung của I_1, giữ nguyên ký hiệu trên đó nhưng đổi chiều của cung.
- Đỉnh vào của I_1 là đỉnh kết duy nhất của nguồn I.

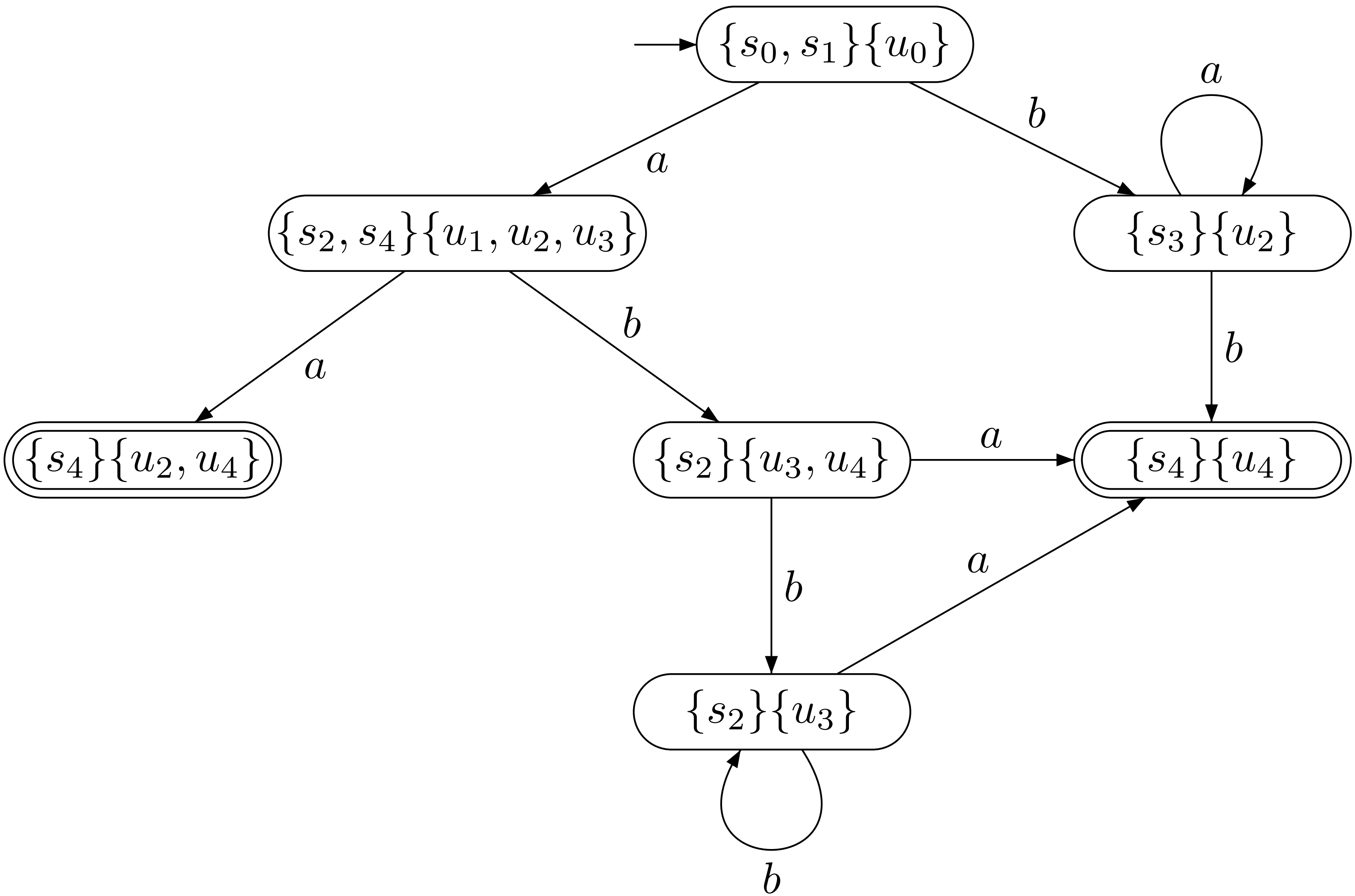
Phép chia trái

- **Định nghĩa.** Nguồn K được gọi là nguồn thương bên trái của nguồn I_1 cho I_2 nếu $N(K) = N(I_2) \setminus N(I_1)$.
- **Bài toán.** Giả sử I_1, I_2 là hai nguồn trên bảng chữ cái Σ, cần xây dựng nguồn K là thương bên trái của nguồn I_1 cho I_2.
- **Ví dụ.** Cho hai nguồn I_1 và I_2 như sau:



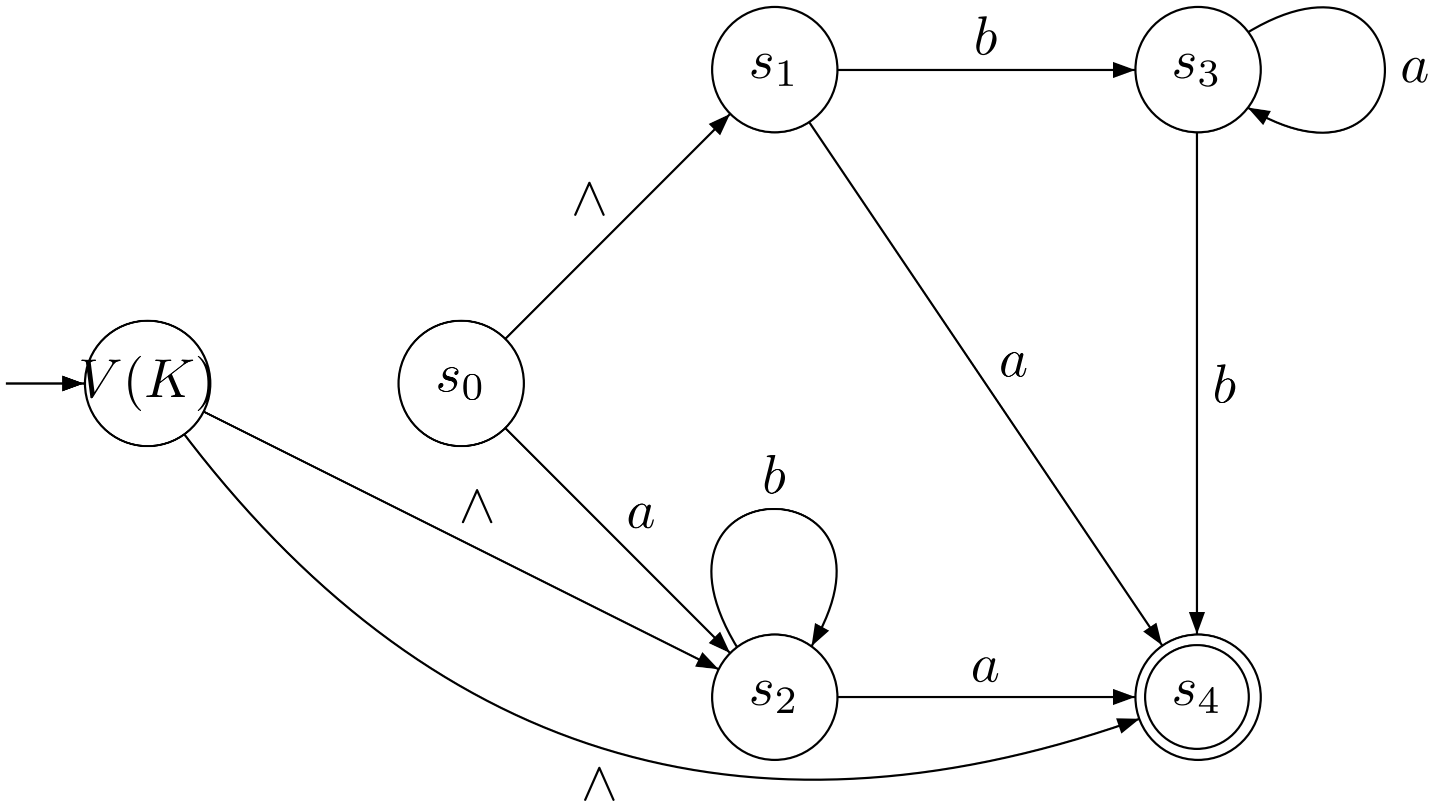
Phép chia trái

- Từ nguồn giao I của I_1 và I_2 ta có các đỉnh xuất phát trên I_1 là $\{s_2, s_4\}$



Phép chia trái

- Nguồn thương bên trái của I_1 cho I_2 được xây dựng như sau:



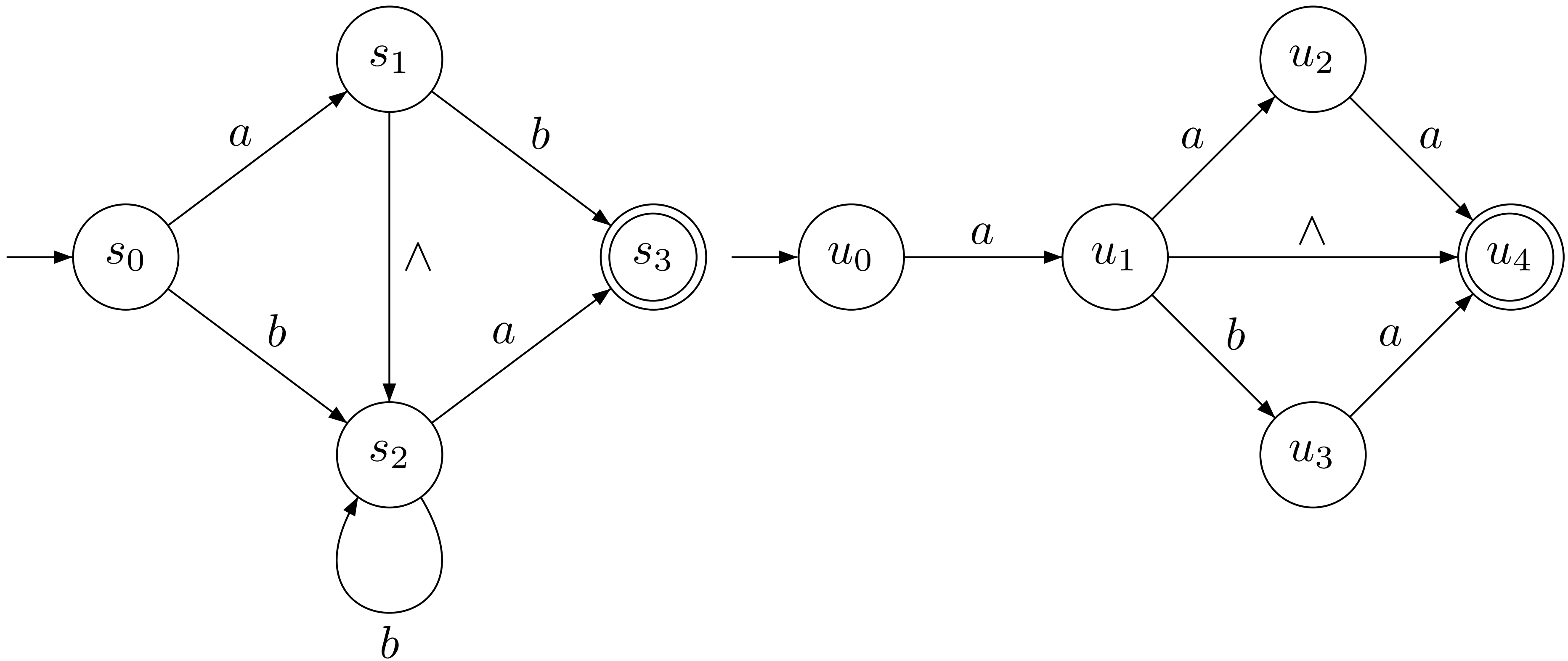
Thuật toán xây dựng phép chia trái

Để nhận được nguồn thương bên trái K của các nguồn I_1, I_2 ta thực hiện các bước sau:

- Xây dựng nguồn giao I của I_1, I_2.
- Thêm đỉnh mới $V(K)$ vào I_1 là đỉnh vào của nguồn K.
- Giả sử $Q = (S, T)$ là một đỉnh tùy ý của nguồn giao I. Khi đó từ đỉnh vào $V(K)$ của nguồn K có cung rỗng đến tất cả các đỉnh $v \in S$ khi và chỉ khi $T \cap F(I_2) \neq \emptyset$.
- Tập đỉnh kết: $F(K) = F(I_1)$.

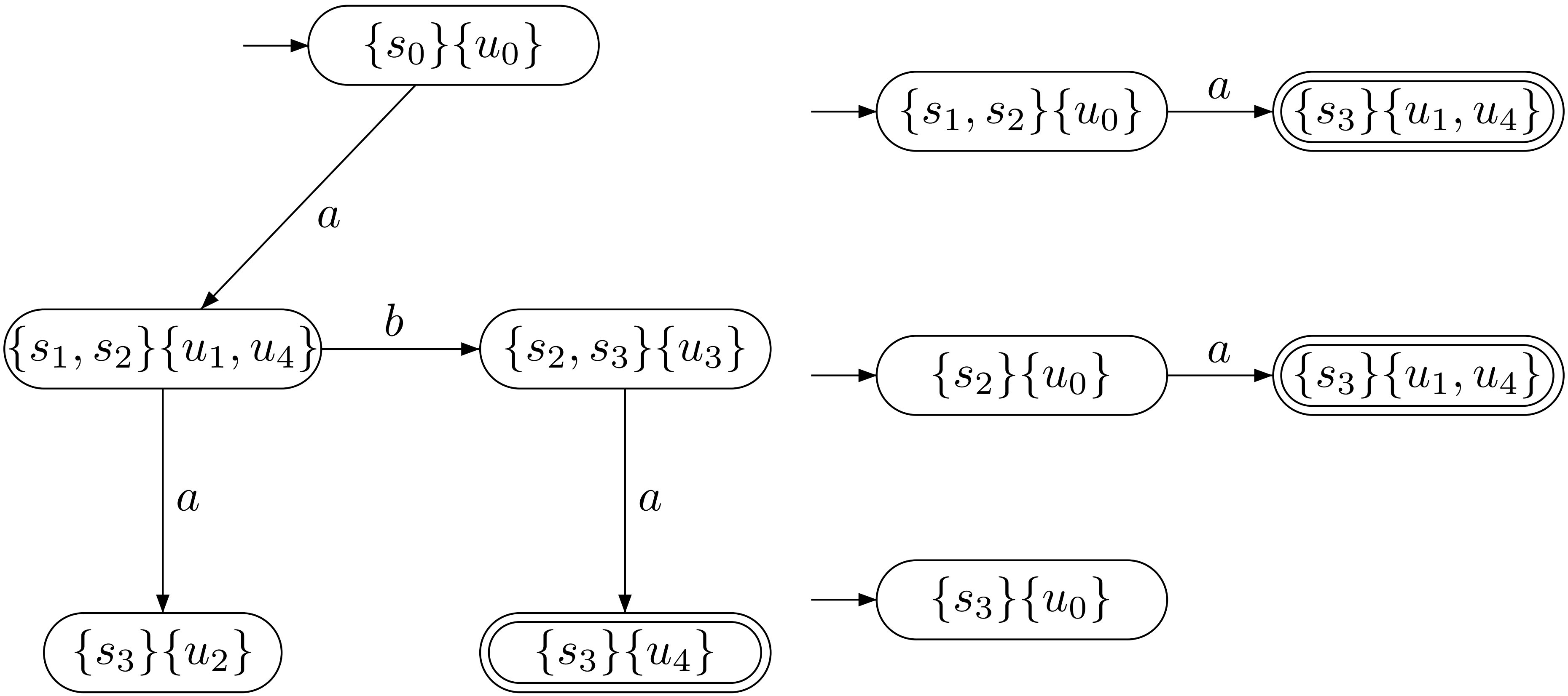
Phép chia phải

- **Định nghĩa.** Nguồn K được gọi là nguồn thương bên phải của nguồn I_1 cho I_2 nếu $N(K) = N(I_1)/N(I_2)$.
- **Bài toán.** Giả sử I_1, I_2 là hai nguồn trên bảng chữ cái Σ, cần xây dựng nguồn K là thương bên phải của nguồn I_1 cho I_2.
- **Ví dụ.** cho hai nguồn I_1 và I_2 như sau:



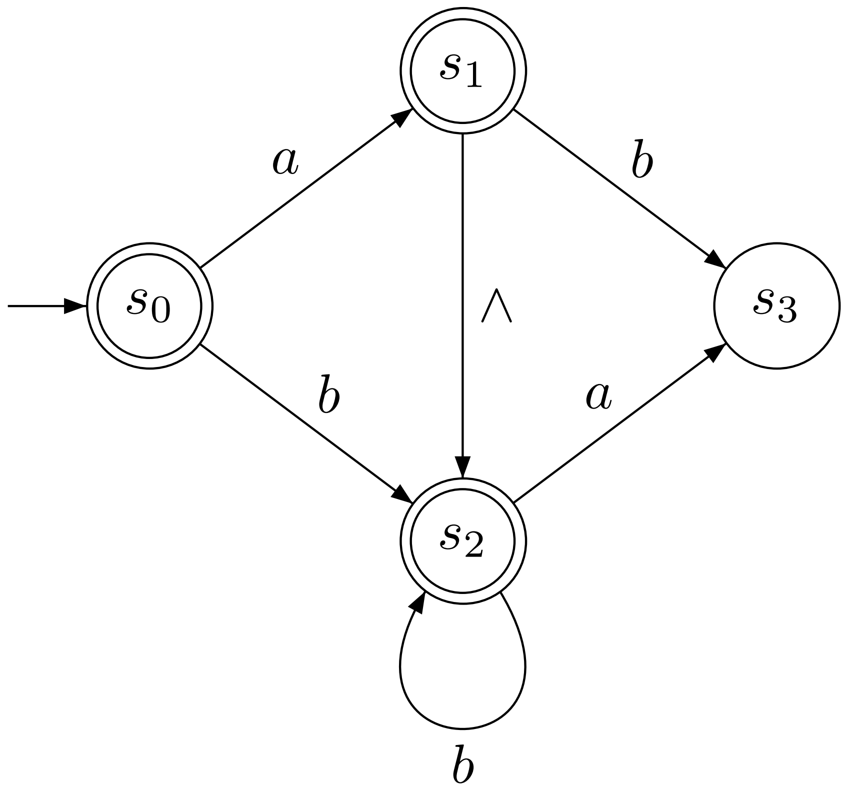
Phép chia phải

- Gọi I_{s_m} là nguồn được thành lập từ I_1 bằng cách lấy s_m là đỉnh xuất phát ($s_m \in A(I_1)$).
- Các nguồn giao $K_m = I_{s_m} \cap I_2$ được xây dựng như sau:



Phép chia phải

- Do $N(K_0), N(K_1), N(K_2) \neq \emptyset$ và $N(K_3) = \emptyset$ nên s_0, s_1, s_2 là các đỉnh kết của nguồn I.



Thuật toán xây dựng phép chia phải

Thuật toán xây dựng nguồn thương bên phải được thực hiện theo 2 bước sau:

- Bước 1: Với mỗi $s_i \in A(I_1)$ ta gọi I_{s_i} là nguồn trên I_1 nhưng đỉnh vào là s_i, xây dựng nguồn giao K_i của I_{s_i} với I_2.
- Bước 2: Nguồn K xây dựng trên cơ sở của nguồn I_1 như sau:
  - Thừa nhận đỉnh vào của I_1 là đỉnh vào của K.
  - Đỉnh $s_i \in A(I_1)$ là đỉnh kết của nguồn K khi và chỉ khi $N(K_i) \neq \emptyset$.

Ôtômat hữu hạn

- Ôtômat đơn định và không đơn định
  - **Định nghĩa.** Ôtômat hữu hạn (FA) trên bảng chữ cái Σ là bộ năm đối tượng:
    $$A = (S, \Sigma, s_0, \delta, F) \text{ trong đó:}$$
    S: là tập hữu hạn các trạng thái của ôtômat.
    Σ: là bảng chữ cái vào của ôtômat.
    $s_0 \in S$: trạng thái mở đầu.
    $F \subseteq S$: tập trạng thái kết thúc của ôtômat.
    - Nếu hàm chuyển trạng thái δ: $U \rightarrow S$ trong đó $U \subset S \times \Sigma$ thì A được gọi là ôtômat đơn định (viết tắt là DFA). Nếu miền xác định δ là $U = S \times \Sigma$ thì A được gọi là ôtômat đơn định đầy đủ.
    - Nếu hàm chuyển trạng thái δ: $U \rightarrow 2^S$ trong đó $U \subset S \times \Sigma$ thì A được gọi là ôtômat không đơn định (viết tắt là $NDFA$).
  - **Chú ý.** Nếu δ không xác định tại (s, a) thì trong tính toán có thể coi $\delta(s, a) = \emptyset$.

Ví dụ

- **Ví dụ 1.** Cho ôtômat đơn định
  - $A_1 = (\{s_0, s_1, s_2, s_3\}, \{a, b, c\}, s_0, \delta_1, \{s_2, s_3\})$
    Với hàm chuyển δ_1 được xác định bằng bảng chuyển.

δ_1	s_0	s_1	s_2	s_3
a	s_1	s_2	s_1	
b	s_2	s_3		s_3
c	s_3		s_3	s_2

- **Ví dụ 2.** Cho ôtômat không đơn định
  - $A_2 = (\{t_0, t_1, t_2\}, \{0, 1\}, t_0, \delta_2, \{t_2\})$
    Với hàm chuyển δ_2 được xác định bằng bảng chuyển.

δ_2	t_0	t_1	t_2
0	$\{t_1, t_2\}$	$\{t_0, t_1\}$	$\{t_1\}$
1	$\{t_2\}$	$\{t_0\}$	

Ngôn ngữ được đoán nhận bởi DFA

- Giả sử δ là hàm chuyển trong ôtômat A, khi đó hàm chuyển mở rộng $\delta^*\colon S \times \Sigma^* \to S$ được xác định bởi các quy tắc sau:
  - $\forall s \in S, \delta^*(s, \wedge) = s$
  - $\forall s \in S, \forall a \in \Sigma, \delta^*(s, a) = \delta(s, a)$
  - $\forall s \in S, \forall a \in \Sigma, \forall x \in \Sigma^*, \delta^*(s, ax) = \delta^*(\delta^*(s, a), x)$
- **Định nghĩa**. Ta gọi tập tất cả các xâu trên bảng chữ cái Σ, mà mỗi xâu này đẩy ôtômat A từ trạng thái khởi đầu (s_0) đến một trong những trạng thái kết là ngôn ngữ được đoán nhận bởi (hay được sinh bởi) ôtômat đơn định A, ký hiệu là $T(A)$.
  Tóm lại, $T(A) = \{x \in \Sigma^* \mid \delta^*(s_0, x) \in F\}$

Mô tả quá trình đoán nhận xâu ω

- Xâu vào $\omega = x_1 x_2 \dots x_n$

$$
\begin{array}{ccccccccccc}
x_1 & \longrightarrow & x_2 & \longrightarrow & x_3 & \longrightarrow & \dots & \longrightarrow & x_n & & \\
\big\uparrow & & \big\uparrow & & \big\uparrow & & \big\uparrow & & & & \\
q_0 & \longrightarrow & q_1 & \longrightarrow & q_2 & \longrightarrow & \dots & \longrightarrow & q_{n-1} & \longrightarrow & q_n
\end{array}
$$

- Ban đầu ôtômat ở trạng thái q_0, đầu đọc nhìn vào ô có ký hiệu x_1, dưới tác động của hàm chuyển δ, ôtômat chuyển sang trạng thái $q_1 = \delta(q_0, x_1)$ và đầu đọc chuyển sang phải 1 ô nhìn vào ký hiệu x_2. Sau đó ôtômat tiếp tục chuyển sang trạng thái $q_2 = \delta(q_1, x_2)$. Quá trình đó tiếp cho tới khi đầu đọc nhìn vào ký hiệu x_n, trạng thái của ôtômat là q_{n-1}. Hàm chuyển δ đưa ôtômat từ trạng thái q_{n-1} sang trạng thái $q_n = \delta(q_{n-1}, x_n)$. Nếu $q_n \in F$ thì ôtômat đã đoán nhận xâu ω hay $\omega \in T(A)$, ngược lại ta nói rằng ôtômat không đoán nhận được xâu ω hay $\omega \notin T(A)$.

Ví dụ

- Cho $A = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, q_0, \delta, \{q_0\})$. Hàm chuyển xác định như sau:

δ	q_0	q_1	q_2	q_3
0	q_2	q_3	q_0	q_1
1	q_1	q_0	q_3	q_2

- Cho xâu $\omega = 110101$. Quá trình hoạt động của A diễn ra như sau:

$$
\begin{array}{ccccccccccccc}
1 & \longrightarrow & 1 & \longrightarrow & 0 & \longrightarrow & 1 & \longrightarrow & 0 & \longrightarrow & 1 & & \\
\Big\uparrow & & \Big\uparrow & & \Big\uparrow & & \Big\uparrow & & \Big\uparrow & & \Big\uparrow & & \\
q_0 & \longrightarrow & q_1 & \longrightarrow & q_0 & \longrightarrow & q_2 & \longrightarrow & q_3 & \longrightarrow & q_1 & \longrightarrow & q_0
\end{array}
$$

$\delta^*(q_0, 110101) = \delta^*(\delta^*(q_0, 1), 10101) = \delta^*(q_1, 10101) =$
$\delta^*(\delta^*(q_1, 1), 0101) = \delta^*(q_0, 0101) = \delta^*(\delta^*(q_0, 0), 101) = \delta^*(q_2, 101) =$
$\delta^*(\delta^*(q_2, 1), 01) = \delta^*(q_3, 01) = \delta^*(\delta^*(q_3, 0), 1) = \delta^*(q_1, 1) = q_0 \in F$

Ngôn ngữ được đoán nhận bởi NDFA

- Giả sử δ là hàm chuyển trong ôtômat không đơn định A, khi đó hàm chuyển mở rộng
  δ^*: $2^S \times \Sigma^* \to 2^S$ được xác định bởi các quy tắc sau:
  - $\forall Q \subset S,\ \delta^*(Q, \wedge) = Q.$
  - $\forall Q \subset S, \forall a \in \Sigma, \delta^*(Q, a) = \bigcup_{s \in Q} \delta(s, a)$
  - $\forall Q \subset S, \forall a \in \Sigma, \forall x \in \Sigma^*, \delta^*(Q, ax) = \delta^*(\delta^*(Q, a), x)$
- **Định nghĩa.** Ta gọi tập tất cả các xâu trên bảng chữ cái vào Σ mà mỗi xâu này đẩy ôtômat không đơn định A từ trạng thái khởi đầu đến một tập trạng thái trong đó có ít nhất một trạng thái kết là ngôn ngữ được đoán nhận bởi Ôtômat không đơn định A.
  Tóm lại: $T(A) = \{x \in \Sigma^* \mid \delta^*(s_0, x) \cap F \neq \emptyset\}$ là ngôn ngữ được đoán nhận bởi ôtômat không đơn định A.
- **Định nghĩa.** Hai ôtomat A và A' được gọi là tương đương nếu chúng đoán nhận cùng một ngôn ngữ, tức là $T(A) = T(A')$.

Ví dụ

Cho ôtômat không đơn định $A = (\{s_0, s_1, s_2, s_3\}, \{a, b, c\}, s_0, \delta, \{s_2, s_3\})$
Hàm chuyển δ cho bởi bảng:

δ	s_0	s_1	s_2	s_3
a	$\{s_1\}$	$\{s_1, s_2\}$	$\{s_1, s_3\}$	$\{s_2\}$
b	$\{s_1\}$	$\{s_1\}$	$\{s_1, s_2\}$	$\{s_2\}$
c		$\{s_2\}$	$\{s_1\}$	

- Cho xâu $\omega_1 = baac$ vào A, quá trình đoán nhận ω_1 như sau:
  $\delta^*(\{s_0\}, baac) = \delta^*(\delta^*(\{s_0\}, b), aac) = \delta^*(\{s_1\}, aac) =$
  $\delta^*(\delta^*(\{s_1\}, a), ac) = \delta^*(\{s_1, s_2\}, ac) = \delta^*(\delta^*(\{s_1, s_2\}, a), c) =$
  $\delta^*(\{s_1, s_2, s_3\}, c) = \{s_1, s_2\}$
  Vậy $\delta^*(s_0, baac) \cap F = \{s_1, s_2\} \cap \{s_2, s_3\} = \{s_2\} \neq \emptyset$ nên ôtômat đoán nhận được xâu $\omega_1 = baac$ hay $\omega_1 \in T(A)$.
- Cho xâu $\omega_2 = bccb$ vào A, quá trình đoán nhận ω_2 như sau:
  $\delta^*(\{s_0\}, bccb) = \delta^*(\delta^*(\{s_0\}, b), ccb) = \delta^*(\delta^*(\{s_1\}, c), cb) =$
  $\delta^*(\delta^*(\{s_2\}, c), b) = \delta^*(\{s_1\}, b) = \{s_1\}$
  Vậy $\delta^*(s_0, bccb) \cap F = \{s_1\} \cap \{s_2, s_3\} = \emptyset$ nên $\omega_2 = bccb \notin T(A)$.

Thuật toán đơn định hóa

- **Định lý.** Mọi ôtômat không đơn định đều tồn tại ôtômat đơn định, đầy đủ tương đương với nó.
- **Chứng minh.** Giả sử $A = (S, \Sigma, s_0, \delta, F)$ là ôtômat không đơn định . Ta cần xây dựng ôtômat $A' = (S', \Sigma', s'_0, \delta', F')$ đơn định, đầy đủ tương đương với A.
  Khởi tạo $\Sigma' = \Sigma$, $S' = \{\{s_0\}\}$ (S' là tập hợp chứa các tập con của S)
  - Bước 1: Xây dựng tập trạng thái S' và hàm chuyển δ':
    - Nếu $\exists s \in S', \exists a \in \Sigma'$ mà chưa tính $\delta'(s, a)$ thì tính:
      $\delta'(s, a) = \bigcup_{u \in s} \delta(u, a)$
      - Nếu $\delta'(s, a) \in S'$ thì quay lại bước 1.
      - Nếu $\delta'(s, a) \notin S'$ thì bổ xung $\delta'(s, a)$ vào S' rồi quay lại bước 1.
    - Nếu $\forall s \in S', \forall a \in \Sigma', \delta'(s, a) \in S'$ thì chuyển sang bước 2.
  - Bước 2: Xây dựng tập trạng thái kết:
    - $F' = \{s \in S' | s \cap F \neq \emptyset\}$.

Ví dụ

- Cho ôtômat không định $A = (S, \Sigma, p_0, \delta, F)$ trong đó
  $S = \{p_0, p_1, p_2\},\ \Sigma = \{a, b, c\},\ F = \{p_1, p_2\}$
  Hàm chuyển δ cho bởi bảng:

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

- Ta xây dựng ôtômat $A' = (S', \Sigma', s_0, \delta', F')$ đơn định tương đương với A.
  Khởi tạo $\Sigma' = \Sigma$, $s_0 = \{p_0\},\ S' = \{s_0\}$
  S' và δ' được xây dựng đồng thời như sau:
  $\delta'(s_0, a) = \{p_1\} = s_1,\ \delta'(s_0, b) = \{p_1, p_2\} = s_2,\ \delta'(s_0, c) = \{p_2\} = s_3$
  $\delta'(s_1, a) = \{p_2\} = s_3,\ \delta'(s_1, b) = \emptyset = s_4,\ \delta'(s_1, c) = \{p_0, p_2\} = s_5$
  $\delta'(s_2, a) = \delta(p_1, a) \cup \delta(p_2, a) = \{p_1, p_2\} = s_2$
  $\delta'(s_2, b) = \delta(p_1, b) \cup \delta(p_2, b) = \{p_1\} = s_1$
  $\delta'(s_2, c) = \delta(p_1, c) \cup \delta(p_2, c) = \{p_0, p_2\} = s_5$

Ví dụ

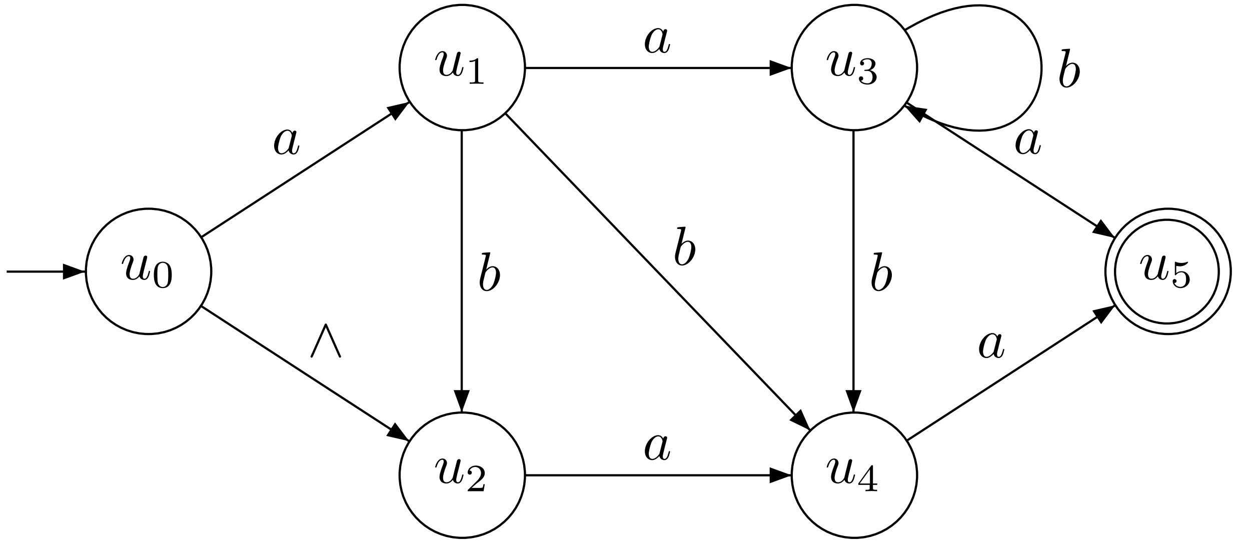
$\delta'(s_3, a) = \delta(p_2, a) = \{p_1\} = s_1$
$\delta'(s_3, b) = \delta(p_2, b) = \{p_1\} = s_1$
$\delta'(s_3, c) = \delta(p_2, c) = \{p_2\} = s_3$
$\delta'(s_4, a) = \delta'(s_4, b) = \delta'(s_4, c) = \emptyset = s_4$
$\delta'(s_5, a) = \delta(p_0, a) \cup \delta(p_2, a) = \{p_1\} = s_1$
$\delta'(s_5, b) = \delta(p_0, b) \cup \delta(p_2, b) = \{p_1, p_2\} = s_2$
$\delta'(s_5, c) = \delta(p_0, c) \cup \delta(p_2, c) = \{p_2\} = s_3$
Vậy $S' = \{s_0, s_1, s_2, s_3, s_4, s_5\}$, hàm chuyển δ' biểu diễn bởi bảng:

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

Tập trạng thái kết là $F' = \{s_1, s_2, s_3, s_5\}$

Sự tương đương của nguồn và FA

- **Định nghĩa.** Nguồn I và Ôtômat hữu hạn A được gọi là tương đương nếu chúng sinh ra cùng một ngôn ngữ, tức là $N(I) = T(A)$.
- Cho nguồn I và ôtômat A tương đương với I xây dựng như sau:



- $A = (\{u_0, u_1, u_2, u_3, u_4, u_5\}, \{a, b\}, u_0, \{u_5\})$, hàm chuyển δ như sau

δ	u_0	u_1	u_2	u_3	u_4	u_5
a	$\{u_1, u_4\}$	$\{u_3\}$	$\{u_4\}$	$\{u_5\}$	$\{u_5\}$	
b		$\{u_2, u_4\}$		$\{u_3, u_4\}$		

Xây dựng FA tương đương với nguồn

Thuật toán xây dựng ôtômat $A = (S, \Sigma, s_0, \delta, F)$ tương đương với nguồn I như sau:

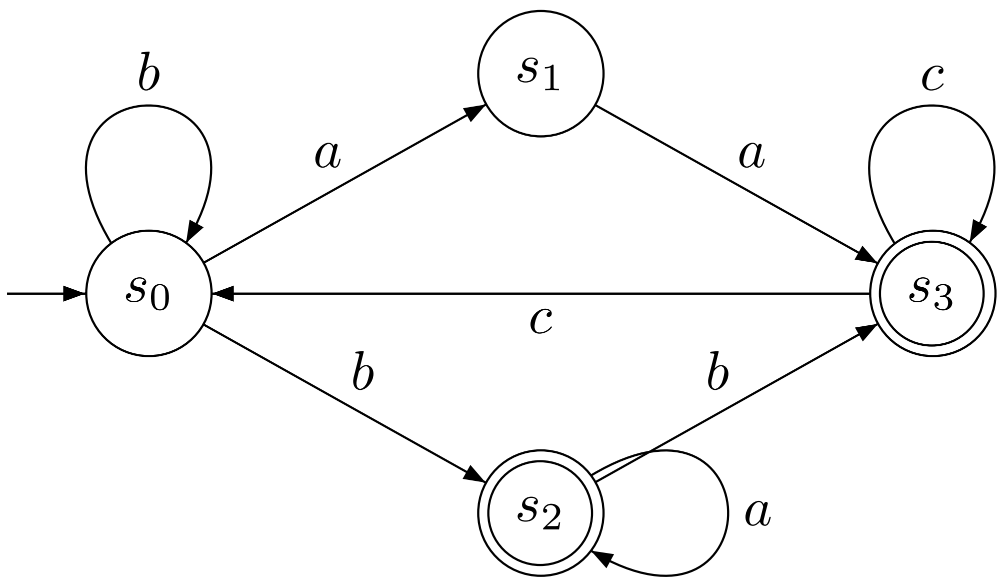
- Bảng chữ cái: $\Sigma = \Sigma_I$
- Tập trạng thái: $S = A(I)$
- Tập trạng thái kết: $F = F(I) \cup \{s \in S | \exists u \in F(I), \wedge \in N_I(s, u)\}$
- Trạng thái ban đầu: $s_0 = V(I)$
- Hàm chuyển: $\delta(s, a) = \{s' | a \in N_I(s, s')\}$.

Xây dựng nguồn tương đương với FA

- **Ví dụ**. Cho ôtômat $A = (S, \Sigma, s_0, \delta, F)$. Trong đó $S = \{s_0, s_1, s_2, s_3\}$, $\Sigma = \{a, b, c\}$, $F = \{s_2, s_3\}$
Hàm chuyển δ cho bởi bảng:

δ	s_0	s_1	s_2	s_3
a	s_1	s_3	s_2	
b	$\{s_0, s_2\}$		s_3	
c				$\{s_0, s_3\}$

- Nguồn I tương đương với A được xây dựng như sau:

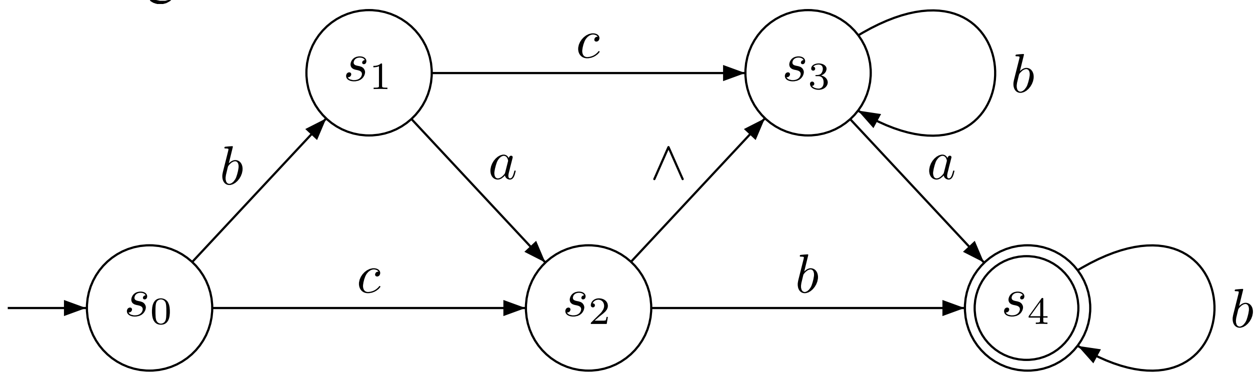


Xây dựng nguồn tương đương với FA

- Giả sử có ôtômat $A = (S, \Sigma, s_0, \delta, F)$, thuật toán xây dựng nguồn I tương đương với A được thực hiện theo 2 bước như sau:
  - Bước 1:
    - Xác định đỉnh: Lấy các điểm trên mặt phẳng hoặc không gian tương ứng với các trạng thái của ôtômat, dùng ngay các ký hiệu trạng thái để đặt tên cho các đỉnh tương ứng.
    - Đỉnh tương ứng với trạng thái khởi đầu thừa nhận là đỉnh vào ký hiệu $V(I)$.
    - Đỉnh s được thừa nhận là đỉnh kết thúc khi và chỉ khi $s \in F$.
  - Bước 2: Xác định các cung:
    - Nếu A là ôtômat đơn định thì nối cung từ đỉnh s đến s', trên cung đó ghi ký hiệu a khi và chỉ khi $s' = \delta(s, a)$.
    - Nếu A là ôtômat không đơn định thì nối cung từ đỉnh s đến s', trên cung đó ghi ký hiệu a khi và chỉ khi $s' \in \delta(s, a)$.

Sự tương đương của nguồn và VPCQ

- **Định nghĩa.** Nguồn I và văn phạm chính quy G được gọi là tương đương nếu chúng sinh ra cùng một ngôn ngữ, nghĩa là $N(I) = L(G)$.
- Xây dựng văn phạm chính quy tương đương với nguồn.
  - **Ví dụ.** Cho nguồn I như sau:



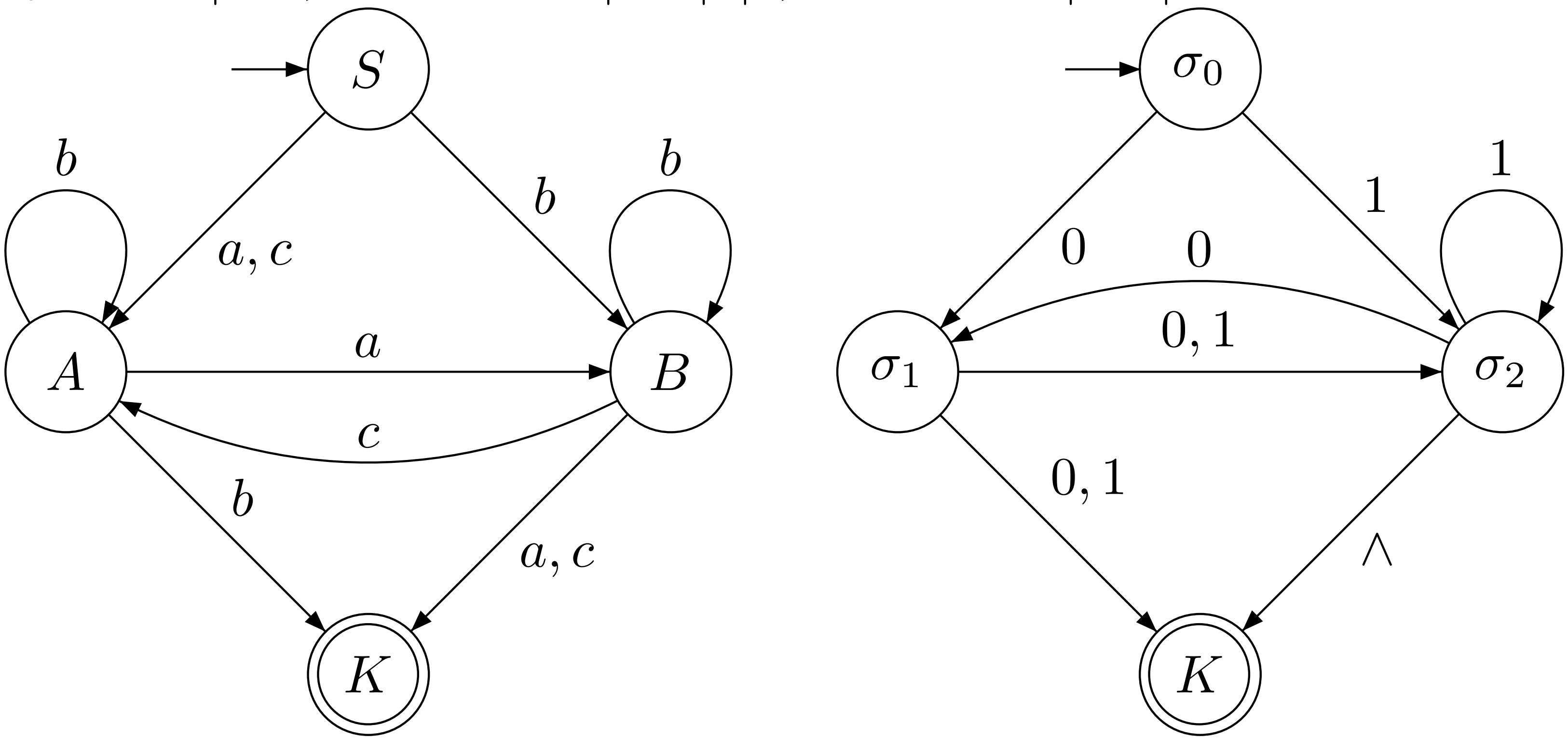
  - Văn phạm chính quy G được xây dựng như sau:
    $G = (\{a, b, c\}, \{s_0, s_1, s_2, s_3, s_4\}, s_0, P)$, trong đó quy tắc P gồm:
    $s_0 \longrightarrow bs_1|cs_2, \quad s_3 \longrightarrow bs_3|as_4$
    $s_1 \longrightarrow as_2|cs_3, \quad s_4 \longrightarrow bs_4$
    $s_2 \longrightarrow bs_4|bs_3|as_4, \quad s_3 \longrightarrow a, \quad s_2 \longrightarrow b, \quad s_2 \longrightarrow a, \quad s_4 \longrightarrow b$
  - Nếu thay 4 quy tắc kết bởi quy tắc $s_4 \longrightarrow \wedge$ ta được VPCQ suy rộng.

Xây dựng VPCQ tương đương với nguồn

- Giả sử có nguồn I trên bảng chữ cái Σ và $\wedge \notin N(I)$. Văn phạm chính quy (chính quy suy rộng) $G = (\Sigma, V, \sigma, P)$ tương đương với I được xác định như sau:
  - Ký tự ban đầu σ là đầu vào của I $(\sigma = V(I))$.
  - Tập ký tự phụ V là tất cả ký tự ghi trên các đỉnh của nguồn I.
  - Tập ký tự cơ bản là các ký tự khác $\wedge$ ghi trên các cung của I.
  - Tập P các quy tắc của văn phạm chính quy:
    - Quy tắc không kết: có quy tắc $s \to as'$ khi và chỉ khi có đường đi từ s đến s' sinh ra ký hiệu a tức là $a \in N_I(s, s')$.
    - Quy tắc kết: có quy tắc $u \to a$ khi và chỉ khi ký hiệu $a \in N_I(u, u')$ trong đó u' là đỉnh kết.
  - Tập P các quy tắc của văn phạm chính quy suy rộng: Thay các quy tắc kết của văn phạm chính quy bởi các quy tắc có dạng $s \to \wedge$ trong đó s là đỉnh kết của I.

Xây dựng nguồn tương đương với VPCQ

- **Ví dụ.** Xây dựng các nguồn I_1, I_2 tương đương với các văn phạm chính quy, chính quy suy rộng G_1, G_2 sau:
  $G_1 = (\{a, b, c\}, \{S, A, B\}, S, P_1)$, trong đó P_1 gồm:
  $S \to aA|cA|bB, \quad A \to bA|aB|b, \quad B \to cA|bB|a|c$
  $G_2 = (\{0, 1\}, \{\sigma_0, \sigma_1, \sigma_2\}, \sigma_0, P_2)$, trong đó P_2 gồm:
  $\sigma_0 \to 0\sigma_1|1\sigma_2, \quad \sigma_1 \to 0\sigma_2|1\sigma_2|0|1, \quad \sigma_2 \to 0\sigma_1|1\sigma_2|\wedge$



Xây dựng nguồn tương đương với VPCQ

- Giả sử có văn phạm chính quy (Chính quy suy rộng):
$$G = (\Sigma, V, \sigma, P)$$
Nguồn I tương đương với G được xây dựng như sau:
- Đỉnh:
  - Lấy các điểm trên mặt phẳng hoặc không gian với các phần tử thuộc V, đồng thời dùng ngay ký hiệu phụ đó để ghi tên các đỉnh tương ứng.
  - Thêm vào một đỉnh mới là đỉnh kết của I, ký hiệu là K.
  - Đỉnh tương ứng với ký hiệu xuất phát σ của văn phạm là đỉnh vào của nguồn.
- Cung:
  - $\forall a \in \Sigma \cup \{\wedge\}, \forall A, B \in V$. Trong I có cung đi từ đỉnh A sang đỉnh B trên đó ghi a khi và chỉ khi trong G có quy tắc $A \to aB$.
  - $\forall b \in \Sigma \cup \{\wedge\}, \forall C \in V$. Trong I có cung đi từ đỉnh C sang K trên đó ghi b khi và chỉ khi trong G có quy tắc $C \to b$.

Biểu thức chính quy

- **Định nghĩa.** Trên bảng chữ cái Σ, biểu thức chính quy được định nghĩa theo các bước quy nạp sau đây:
  - $\emptyset$ là biểu thức chính quy biểu diễn ngôn ngữ rỗng.
  - a là biểu thức chính quy biểu diễn ngôn ngữ $\{a\}$,($a \in \Sigma \cup \{\wedge\}$).
  - Nếu r và s là hai biểu thức chính quy biểu diễn hai ngôn ngữ R và S tương ứng thì:
    - $(r) \vee (s)$ là biểu thức chính quy biểu diễn ngôn ngữ $R \cup S$.
    - $(r)(s)$ là biểu thức chính quy biểu diễn ngôn ngữ $R.S$.
    - $(r)^*$ là biểu thức chính quy biểu diễn ngôn ngữ $(R)^*$.
- **Ví dụ.**
  - $A = (ab)^3 \vee (c)^*((b)^* \vee c^2)$ là biểu thức chính quy biểu diễn ngôn ngữ $\{ababab, c^n b^m, c^{n+2} | m, n \in \mathbb{N}\}$.
  - $B = (a \vee b)^*(bab)(a \vee b)^*$ là biểu thức chính quy biểu diễn ngôn ngữ là tất cả các xâu trên a, b mà có chứa từ con bab.
- **Ký hiệu**: $M(B)$ là ngôn ngữ mà biểu thức chính quy B biểu diễn.

Biểu thức chính quy suy rộng

Để việc biểu diễn ngôn ngữ chính quy được thuận tiện và gọn gàng, người ta bổ xung một số cách biểu diễn ngôn ngữ chính quy trên bảng chữ cái Σ sau:

- $^\wedge\omega$ là biểu thức chính quy biểu diễn ngôn ngữ là tập tất cả các xâu bắt đầu bằng xâu ω. Dễ thấy $^\wedge\omega = \omega.\Sigma^*$.
- $\omega\$$ là biểu thức chính quy biểu diễn ngôn ngữ là tập tất cả các xâu kết thúc bằng xâu ω. Dễ thấy $\omega\$ = \Sigma^*.\omega$.
- $\omega\{n,\}$ là biểu thức chính quy biểu diễn ngôn ngữ $\{\omega^k \mid k \geq n\}$.
- $\omega\{n,m\}$ là biểu thức chính quy biểu diễn ngôn ngữ $\{\omega^k \mid n \leq k \leq m\}$.
- $[^\wedge a_1 a_2 \ldots a_n]$ là biểu thức chính quy biểu diễn các ký tự thuộc Σ không phải là $a_1, a_2, \ldots, a_n$.
- $[a - z], [A - Z]$ là biểu thức chính quy của tập ký tự $a \rightarrow z$ và $A \rightarrow Z$.
- $[0 - 9]$ là biểu thức chính quy biểu diễn tập các ký tự từ 0 đến 9.
- $[^\wedge a - z]$ là biểu thức chính quy biểu diễn ngôn ngữ là tập các ký tự thuộc Σ không phải là $a, b, c, \ldots, x, y, z$. Tức là $[^\wedge a - z] = \Sigma \setminus [a - z]$.

Ví dụ

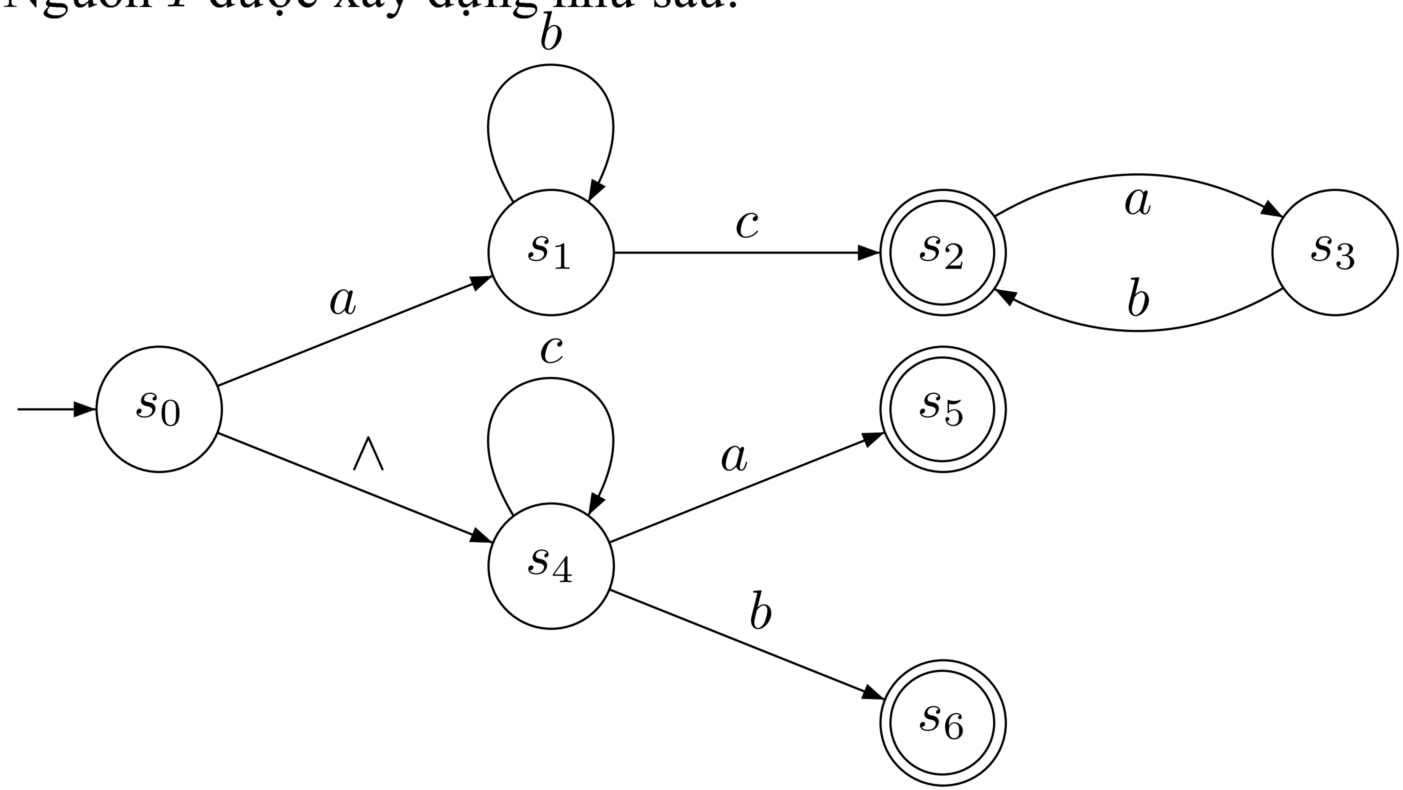
- Ngôn ngữ gồm tất cả các xâu bắt đầu bằng từ "anh" hoặc kết thúc bằng từ "em" được biểu diễn bởi biểu thức chính quy $^\wedge$anh$\vee$em\$
- Ngôn ngữ $\{a^2b, a^3b, a^4b, a^5b, c(bc)^4, c(bc)^4, c(bc)^5, c(bc)^6, c(bc)^7\}$ được biểu diễn bởi biểu thức chính quy $(a\{2,5\})b \vee c(bc\{3,7\})$
- Ngôn ngữ $\{abc^6, abc^7, abc^8, \ldots\}$ được biểu diễn bởi biểu thức chính quy $ab(c\{6,\})$
- Tập các số điện thoai có thể có của mạng mobifone được biểu diễn bởi biểu thức chính quy $09(0 \vee 3)[0-9]^7$.
- Tập tất cả các biển số xe máy có thể có đăng ký tại Hà Nội được biểu diễn bởi biểu thức chính quy $29-[A-Z][0-9]-[0-9]^4$
- Tập các mã lớp của trường Đại học Thăng Long được biểu diễn bởi biểu thức chính quy $([2-7] \vee CN)(S \vee C \vee 1 \vee 3 \vee 6 \vee 8)[1-4][0-9]$

Sự tương đương của nguồn và BTCQ

- **Định nghĩa.** Nguồn I và biểu thức chính quy B được gọi là tương đương nếu chúng xác định cùng một ngôn ngữ, nghĩa là $N(I) = M(B)$
- **Ví dụ.** Xây dựng nguồn I tương đương với biểu thức chính quy sau:

$$B = a(b)^*c(ab)^* \vee c^*(a \vee b)$$

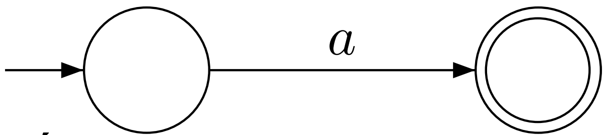
Nguồn I được xây dựng như sau:



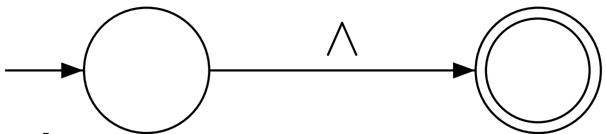
Xây dựng nguồn tương đương với BTCQ

- Giả sử có biểu thức chính quy B trên bảng chữ cái Σ. Khi đó nguồn I tương đương với B được xây dựng nhờ các bước quy nạp sau:
- Cơ sở quy nạp:

Nếu $N(I_1) = \{a\} = M(a)$ thì I_1 là:

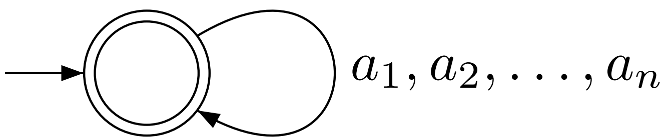


Nếu $N(I_2) = \{\wedge\} = M(\wedge)$ thì I_2 là:



Nếu $N(I_3) = \emptyset = M(\emptyset)$ thì I_3 là:

Nếu $N(I_4) = \Sigma^* = M((a_1 \vee a_2 \vee \ldots \vee a_n)^*)$ thì I_4 là:

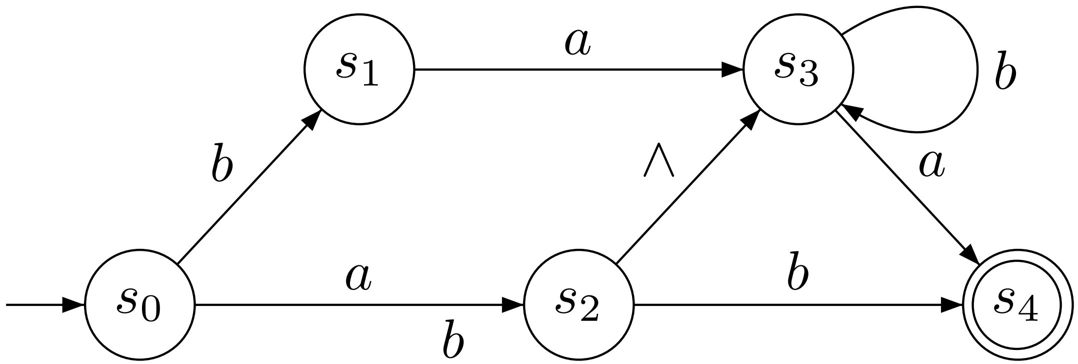


Xây dựng nguồn tương với BTCQ

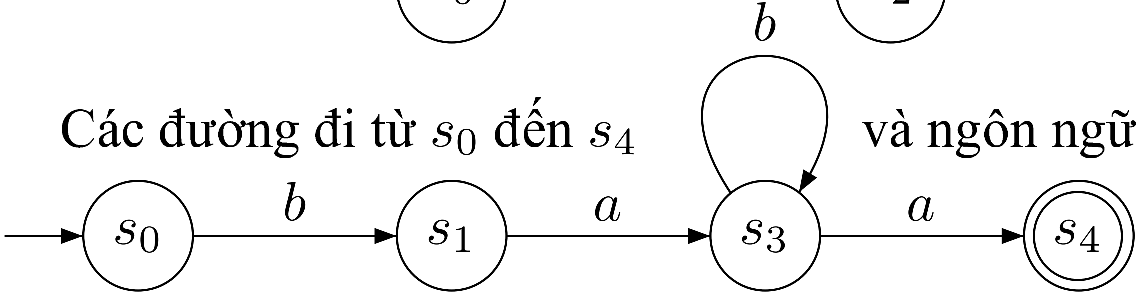
- Các bước quy nạp chính:
  - Giả sử $B = D_1 \vee D_2$ với D_1, D_2 là các biểu thức chính quy và đã có $N(I_1) = M(D_1), N(I_2) = M(D_2)$. Khi đó nguồn I tương đương với nguồn B là hợp của các nguồn I_1 và I_2.
    $$N(I) = N(I_1) \cup N(I_2) = M(D_1) \cup M(D_2) = M(B)$$
  - Giả sử $B = D_1 D_2$ với D_1, D_2 là các biểu thức chính quy và đã có $N(I_1) = M(D_1), N(I_2) = M(D_2)$. Khi đó nguồn I tương đương với nguồn B là tích ghép của các nguồn I_1 và I_2.
    $$N(I) = N(I_1) N(I_2) = M(D_1) M(D_2) = M(B)$$
  - Giả sử $B = (D_1)^*$ với D_1 là biểu thức chính quy và đã có $N(I_1) = M(D_1)$. Khi đó nguồn I tương đương với B là nguồn lặp của nguồn I_1
    $$N(I) = (N(I_1))^* = (M(D_1))^* = M(B)$$
- **Chú ý**: Trước khi xây dựng nguồn I tương đương với B để tránh nhầm lẫn ta có thể phân tích $B = \bigvee_{s=1}^{m} F_s$ trong đó F_s không chứa phép $\vee$.

Xây dựng BTCQ tương đương với nguồn

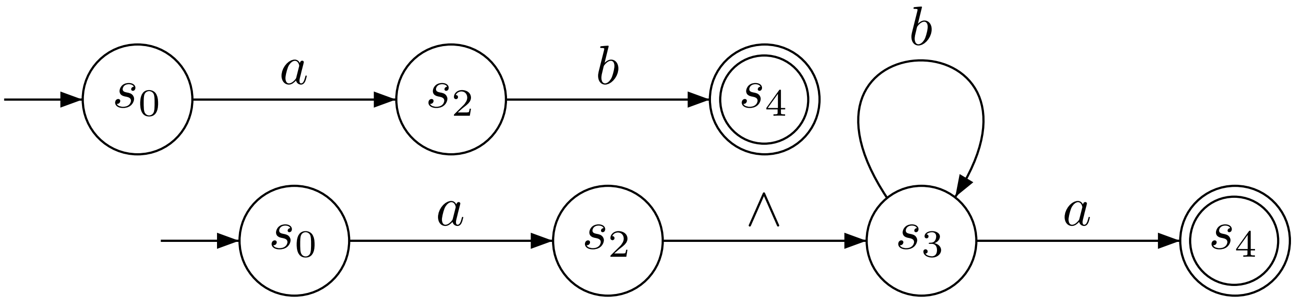
- **Ví dụ.** Có nguồn I trên $\{a, b\}$ như sau:



Các đường đi từ s_0 đến s_4 và ngôn ngữ sinh ra tương ứng là:



$$R(\pi_1) = ba(b)^*a$$



$$R(\pi_2) = ab$$

$$R(\pi_3) = a(b)^*a$$

Biểu thức chính quy $Q(I)$ tương đương với nguồn I là:

$Q(I) = R(\pi_1) \vee R(\pi_2) \vee R(\pi_3) = ba(b)^*a \vee ab \vee a(b)^*a$

BTCQ tương đương với nguồn

Thuật toán xây dựng biểu thức chính quy tương đương với nguồn I có duy nhất đỉnh kết trùng với đỉnh vào được thực hiện theo các bước quy nạp sau:

- Cơ sở quy nạp:
  - Nếu I có một đỉnh duy nhất và không có cung, khi đó $B(I) = \wedge$.
  - Nếu I có một đỉnh v duy nhất là đỉnh vào và đỉnh kết, có m cung với các nhãn là: $a_1, a_2, \ldots, a_m$. Khi đó:
    $$B(I) = (a_1 \vee a_2 \vee \ldots \vee a_m)^*$$
- Quy nạp chính:
  Giả sử mọi nguồn I_1 có đỉnh vào trùng với đỉnh kết với số đỉnh $\leqslant n$ đã được xây dựng. Xét nguồn I có $(n+1)$ đỉnh, với đỉnh vào và đỉnh ra là v_0. Giả sử có m vòng sơ cấp qua v_0 là $\pi_1, \pi_2, \ldots, \pi_m$

$$\pi_i = v_0 \xrightarrow{a_{i_1}} v_1 \xrightarrow{a_{i_2}} \ldots \xrightarrow{a_{i_k}} v_k \xrightarrow{a_{i_{k+1}}} v_0$$

BTCQ tương đương với nguồn

- Quy nạp chính
  - Nếu $k = 0$ thì $R(\pi_i) = a_{i_1}$
  - Nếu $k > 0$ thì xét các nguồn:

$$I_1 = I^{v_1}_{v_0},\ I_2 = I^{v_2}_{v_0,v_1}, \dots, I_k = I^{v_k}_{v_0,\dots,v_{k-1}}$$

trong đó $I^{v_j}_{v_0,\dots,v_{j-1}}$ là nguồn còn lại từ I bằng cách xóa đi các đỉnh $v_0, \dots, v_{j-1}$ với đỉnh kết và vào là v_j.
Dễ thấy $|I_j| \leqslant n$, theo giả thiết quy nạp ta gọi biểu thức chính quy tương đương với I_j là $R_j = B(I_j)$. Do vậy biểu thức chính quy tương ứng với vòng π_i là:

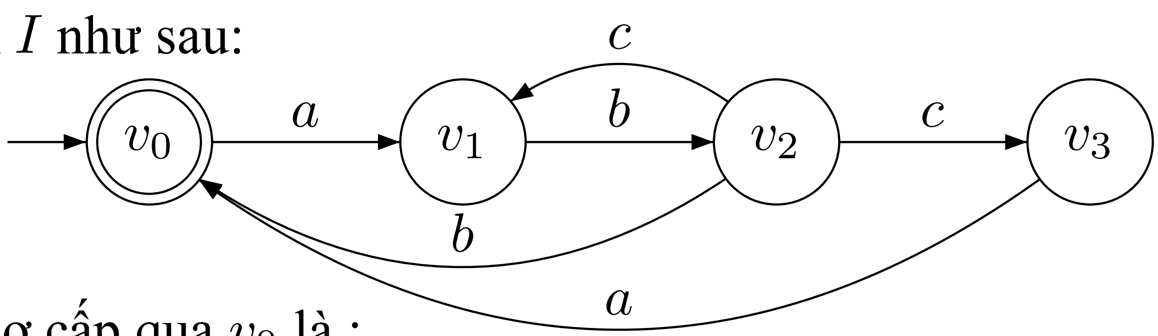
$$R(\pi_i) = a_{i_1} R_1 a_{i_2} R_2 \dots a_{i_k} R_k a_{i_{k+1}}$$

Khi đó biểu thức chính quy tương đương với I là:

$$B(I) = (\bigvee_{i=1}^{m} R(\pi_i))^*$$

Ví dụ

- Cho nguồn I như sau:



- Các vòng sơ cấp qua v_0 là :
  $\pi_1 = (v_0, v_1, v_2, v_3, v_0)$
  $\pi_2 = (v_0, v_1, v_2, v_0)$
  - Biểu thức chính quy ứng với vòng π_1 là:
    $R(\pi_1) = aR(I_{v_0}^{v_1})bR(I_{v_0,v_1}^{v_2})cR(I_{v_0,v_1,v_2}^{v_3})a$
    Mặt khác ta có: $R(I_{v_0}^{v_1}) = (bc)^*, \quad R(I_{v_0,v_1}^{v_2}) = R(I_{v_0,v_1,v_2}^{v_3}) = \wedge$
    Do vậy: $R(\pi_1) = a(bc)^*bca$
  - Biểu thức chính quy ứng với vòng π_2 là:
    $$R(\pi_2) = aR(I_{v_0}^{v_1})bR(I_{v_0,v_1}^{v_2})b = a(bc)^*bb$$
- Biểu thức chính quy tương đương với nguồn I là:
  $$B(I) = (a(bc)^*bca \vee a(bc)^*bb)^*$$

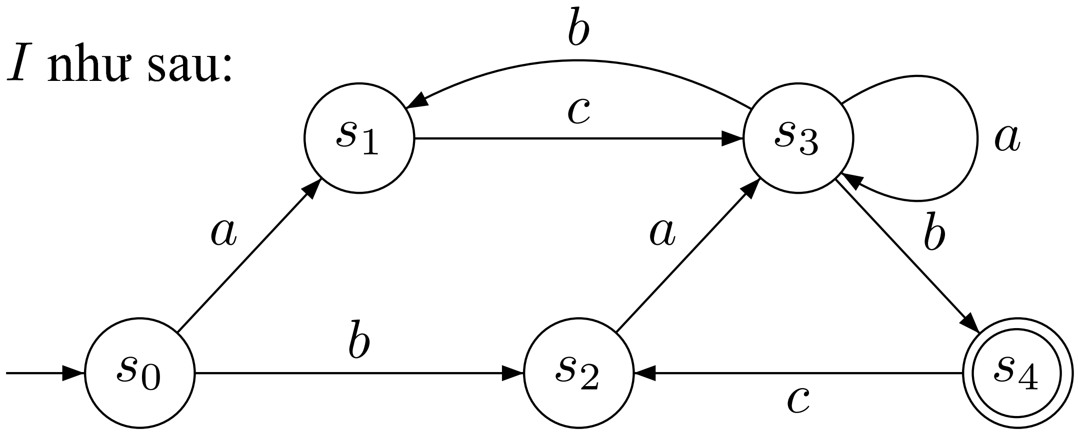
BTCQ tương đương với nguồn

Thuật toán xây dựng biểu thức chính quy tương đương với nguồn bất kì

- Bước 1: Liệt kê tất cả các đường đơn đi từ v_0 đến một trong những đỉnh kết của nguồn là $\pi_1, \ldots, \pi_n$
- Bước 2: Đối với mỗi đường π_i là $v_0 \xrightarrow{a_{i_1}} v_1 \xrightarrow{a_{i_2}} \ldots \xrightarrow{a_{i_k}} v_k \in F$
  $I_0 = I^{v_0},\ I_1 = I^{v_1}, \ldots, I_k = I^{v_k}$ là nguồn trên I với đỉnh vào và đỉnh ra lần lượt là $v_0, v_1, \ldots, v_k$.
  Theo thuật toán 1 ta có biểu thức chính quy tương đương với I_m là R_m. Do vậy
  $$R(\pi_i) = R_0 a_{i_1} R_1 a_{i_2} \ldots a_{i_k} R_k$$
  là biểu thức chính quy ứng với nguồn π_i. Khi đó:
  $$B(I) = \begin{cases} \bigvee_{i=1}^{n} R(\pi_i) & \text{nếu } v_0 \notin F \\ \bigvee_{i=1}^{n} R(\pi_i) \vee R_0 & \text{nếu } v_0 \in F \end{cases}$$
  là biểu thức chính quy tương đương với nguồn I.

Ví dụ

- Cho nguồn I như sau:



- Trước hết ta tính: $R_0 = R(I^{s_0}) = \wedge$, $R_1 = R(I^{s_1}) = R(s_1 s_3 s_1) = (cR(I^{s_3}_{s_1})b)^* = (cR(s_3 \vee s_3 s_4 s_2 s_3)b)^* = (c(a \vee bca)^* b)^*$
$R_2 = R(I^{s_2}) = R(s_2 s_3 s_4 s_2) = (aR(I^{s_3}_{s_2})bR(I^{s_4}_{s_2,s_3})c)^* = (a(a \vee bc)^* bc)^*$
$R_3 = R(I^{s_3}) = R(s_3 \vee s_3 s_1 s_3 \vee s_3 s_4 s_2 s_3) = (a \vee bc \vee bca)^*$
$R_4 = R(I^{s_4}) = R(s_4 s_2 s_3 s_4) = (cR(I^{s_2}_{s_4})aR(I^{s_3}_{s_4,s_2})b)^* = (ca(a \vee bc)^* b)^*$
- Các đường đi từ s_0 đến s_4 là: $\pi_1 = s_0 s_1 s_3 s_4$, $\pi_2 = s_0 s_2 s_3 s_4$. Ta có:
$R(\pi_1) = R_0 a R_1 c R_3 b R_4 = a(c(a \vee bca)^* b)^* c(a \vee bc \vee bca)^* b(ca(a \vee bc)^* b)^*$
$R(\pi_2) = R_0 b R_2 a R_3 b R_4 = b(a(a \vee bc)^* bc)^* a(a \vee bc \vee bca)^* b(ca(a \vee bc)^* b)^*$
Biểu thức chính quy tương đương với nguồn I là:

$$B(I) = R(\pi_1) \vee R(\pi_2)$$

Điều kiện cần của ngôn ngữ chính quy

- **Định lý 1.** Nếu L là ngôn ngữ chính quy thì tồn tại số nguyên dương k sao cho với mọi $z \in L$ mà $|z| > k$ đều biểu diễn được thành tích ghép của từ $u, x, v : z = uxv \;\; (x \neq \wedge)$ đồng thời $z_i = ux^iv \in L \; \forall i = 0, 1, 2 \ldots$
- **Ứng dụng**: Chứng minh rằng ngôn ngữ $L = \{a^p | p$ là số nguyên tố$\}$ không phải là ngôn ngữ chính quy.
  Thật vậy giả sử L là ngôn ngữ chính quy, khi đó với các số nguyên tố p đủ lớn từ $z = a^p$ được biểu diễn thành uxv $(x \neq \wedge)$.
  Do vậy tồn tại ba số nguyên không âm $n_1, n_2, n_3 (n_2 > 0)$ để
  $p = n_1 + n_2 + n_3, u = a^{n_1}, x = a^{n_2}, v = a^{n_3}$.
  Khi đó: $z_i = ux^iv = a^{n_1}(a^{n_2})^i a^{n_3} = a^{n_1}a^{n_2}a^{(i-1)n_2}a^{n_3} = a^{p+(i-1)n_2}$
  Lấy $i = p + 1$ ta được $z_{p+1} = a^{p+pn_2} = a^{(1+n_2)p} \notin L$ vì $(1 + n_2)p$ không phải là số nguyên tố. Mâu thuẫn này chứng tỏ L không phải là ngôn ngữ chính quy.
- **Bài tập.** Chứng minh rằng ngôn ngữ $L = \{a^n b^n | n \in Z^+\}$ không phải là ngôn ngữ chính quy.

Điều kiện cần và đủ của NNCQ

Quan hệ tương đương bất biến phải chỉ số hữu hạn.

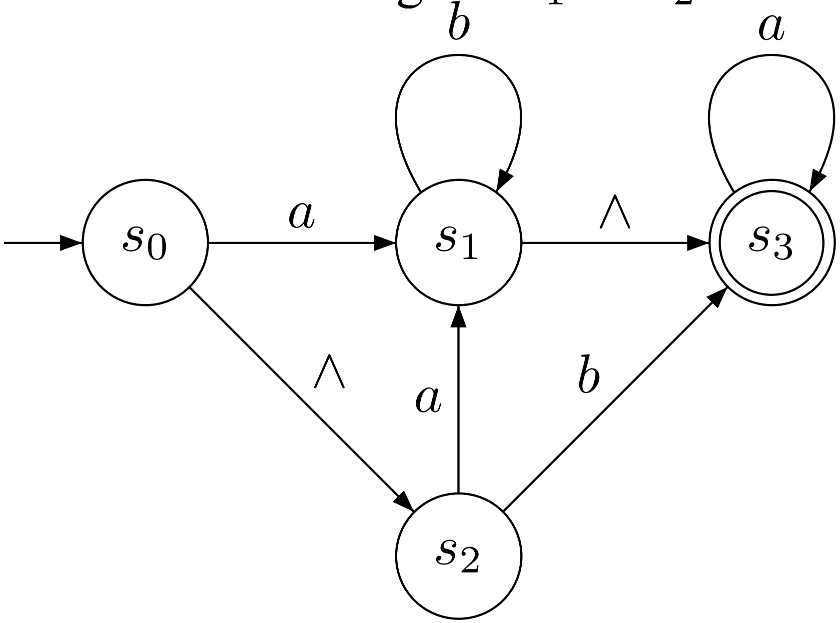
- **Định nghĩa 0.** Quan hệ hai ngôi T trên tập X là một tập con của tập $X \times X$.
  Nếu phần tử x quan hệ với phần tử y thì ta ký hiệu là xTy.
- **Định nghĩa 1.** Quan hệ hai ngôi T trên tập X được gọi là quan hệ tương đương nếu nó có ba tính chất:
  1. Phản xạ: $\forall x \in X$ có xTx.
  2. Đối xứng: $\forall x, y \in X$ nếu xTy thì yTx.
  3. Bắc cầu: $\forall x, y, z \in X$ nếu có xTy và yTz thì có xTz.
- **Định nghĩa 2.** Quan hệ hai ngôi trên ngôn ngữ X được gọi là quan hệ bất biến phải nếu nó thỏa mãn điều kiện:
  $$\forall x, y, z \in X \text{ nếu có } xTy \text{ thì có } (xz)T(yz)$$
- **Định nghĩa 3.** Quan hệ hai ngôi T trên ngôn ngữ X được gọi là quan hệ tương đương bất biến phải nếu nó vừa là quan hệ tương đương vừa là quan hệ bất biến phải.

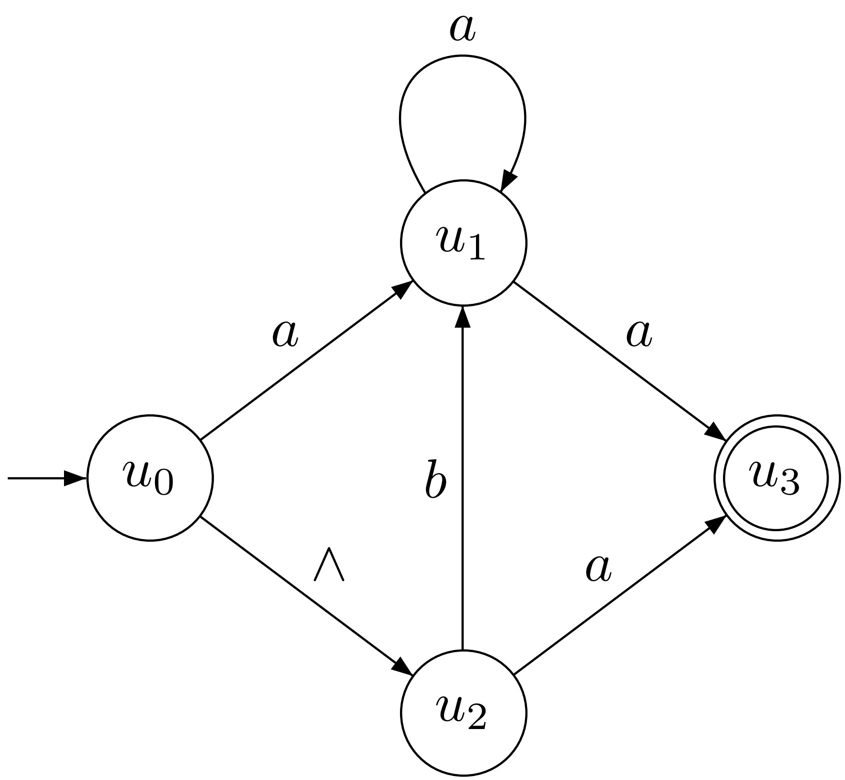
Điều kiện cần và đủ của NNCQ

- **Định nghĩa 4.** Dãy tập con $X_1, X_2, \ldots X_m$ của tập X được gọi là một phân hoạch của tập X nếu nó thỏa mãn đồng thời ba điều kiện:
  1. $X_i \neq \emptyset, \forall i = \overline{1,m}$.
  2. $X_i \cap X_j = \emptyset, \forall i \neq j,\ i,j = \overline{1,m}$.
  3. $X = \bigcup_{i=1}^{m} X_i$
- **Định lý.** Dãy gồm tất cả các lớp tương đương do quan hệ tương đương T trên tập X lập thành một phân hoạch của X.
- **Định nghĩa 5.** Số lớp tương đương trong phân hoạch do quan hệ tương đương T xác định trên tập X được gọi là chỉ số của quan hệ tương đương T trên tập X.
  Nếu chỉ số của quan hệ tương đương trên tập X là hữu hạn thì quan hệ T trên X gọi là quan hệ tương đương với chỉ số hữu hạn.
- **Định lý Myhill-Norode.** Ngôn ngữ L là ngôn ngữ chính quy khi và chỉ khi nó là hợp của một số lớp tương đương theo quan hệ tương đương bất biến phải với chỉ số hữu hạn.

Bài tập chương 3

- **1.** Cho 2 nguồn I_1 và I_2 như sau:





a. Tìm nguồn bù của I_1 và I_2.

b. Tìm nguồn giao của I_1 và I_2.

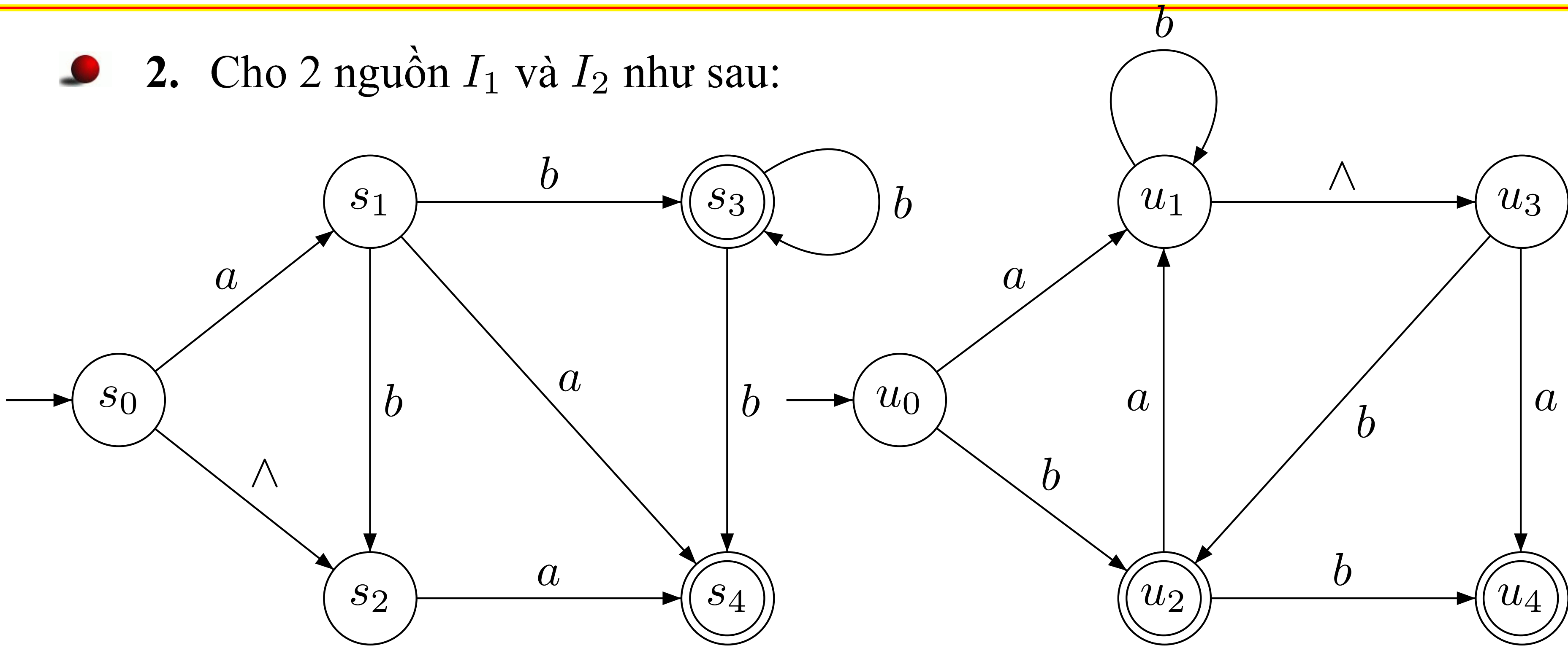
c. Tìm nguồn tích ghép của I_1 và I_2.

d. Tìm nguồn soi gương của I_1 và I_2.

e. Tìm nguồn lặp của I_1 và lặp cắt của I_2.

Bài tập chương 3

- **2.** Cho 2 nguồn I_1 và I_2 như sau:



a. Tìm nguồn bù của I_1 và I_2.

b. Tìm nguồn giao của I_1 và I_2.

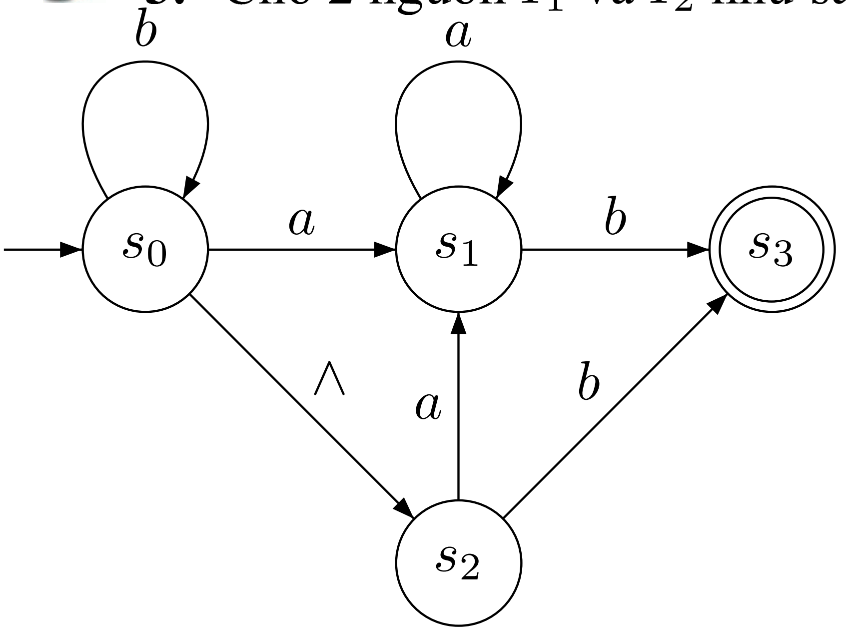
c. Tìm nguồn tích ghép của I_1 và I_2.

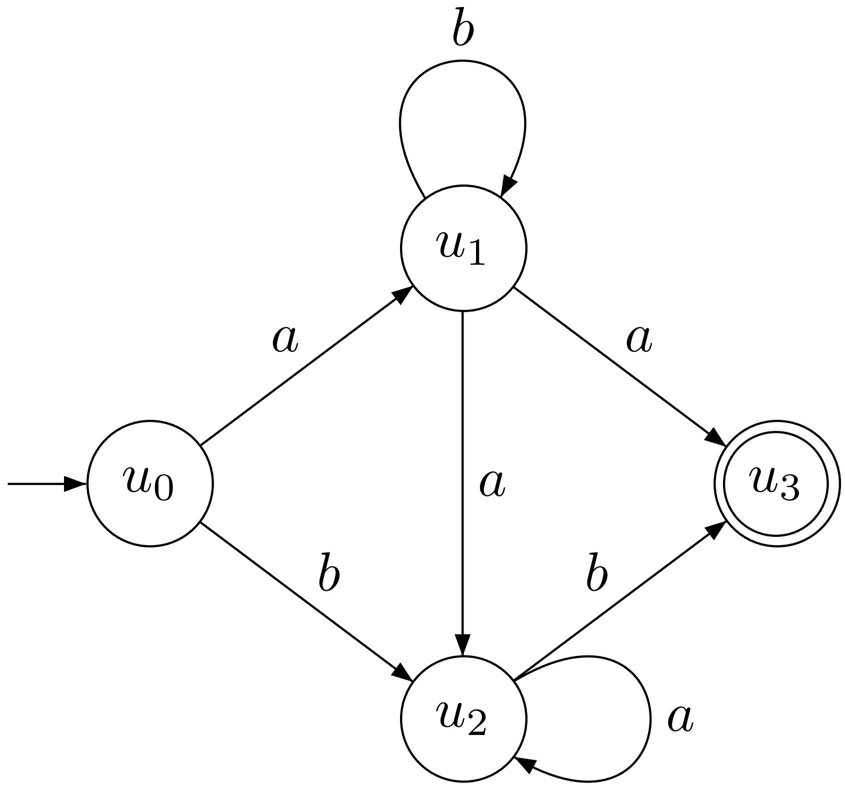
d. Tìm nguồn soi gương của I_1 và I_2.

e. Tìm nguồn lặp của I_1 và lặp cắt của I_2.

Bài tập chương 3

3. Cho 2 nguồn I_1 và I_2 như sau:





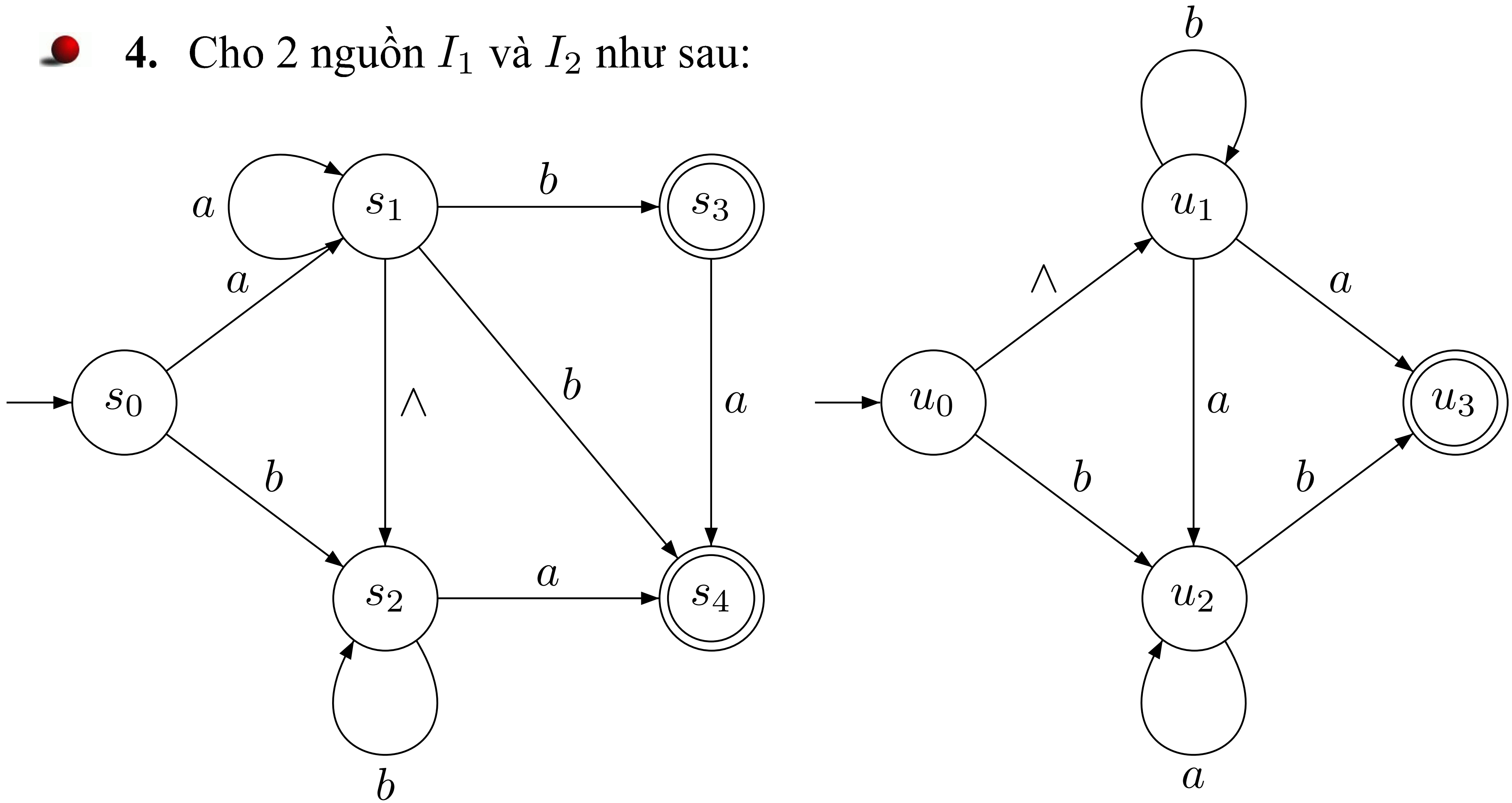
a. Tìm nguồn giao của I_1 và I_2.

b. Tìm nguồn tích ghép của I_1 và I_2.

c. Đơn định hóa nguồn I_2, tìm nguồn bù của I_2.

Bài tập chương 3

- **4.** Cho 2 nguồn I_1 và I_2 như sau:



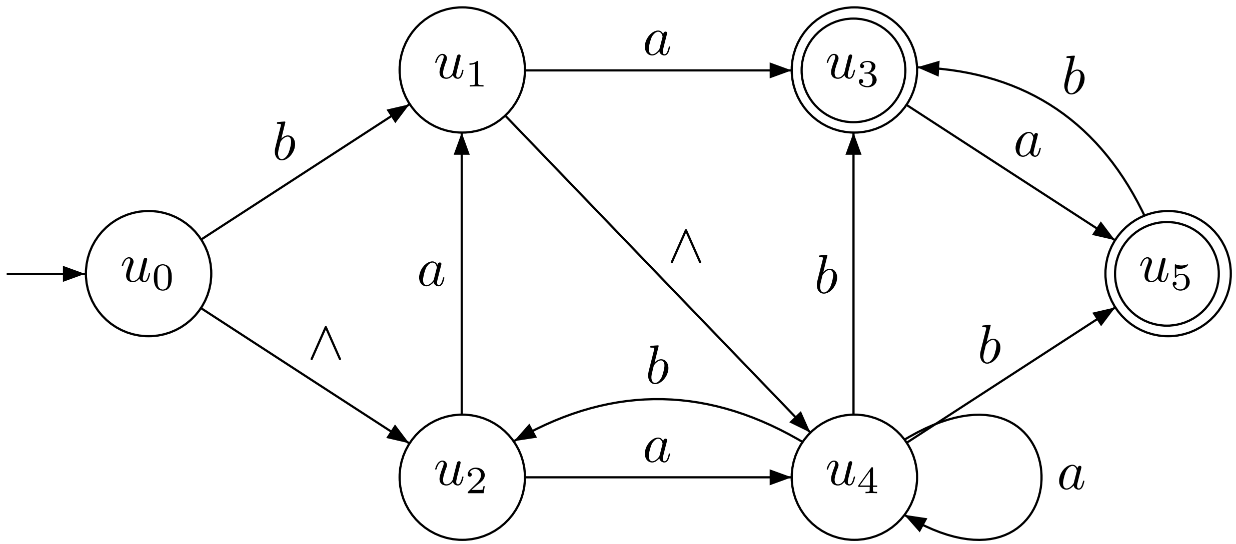
a. Tìm nguồn giao của I_1 và I_2.

b. Tìm nguồn tích ghép của I_1 và I_2.

c. Đơn định hóa nguồn I_2, tìm nguồn bù của I_2.

Bài tập chương 3

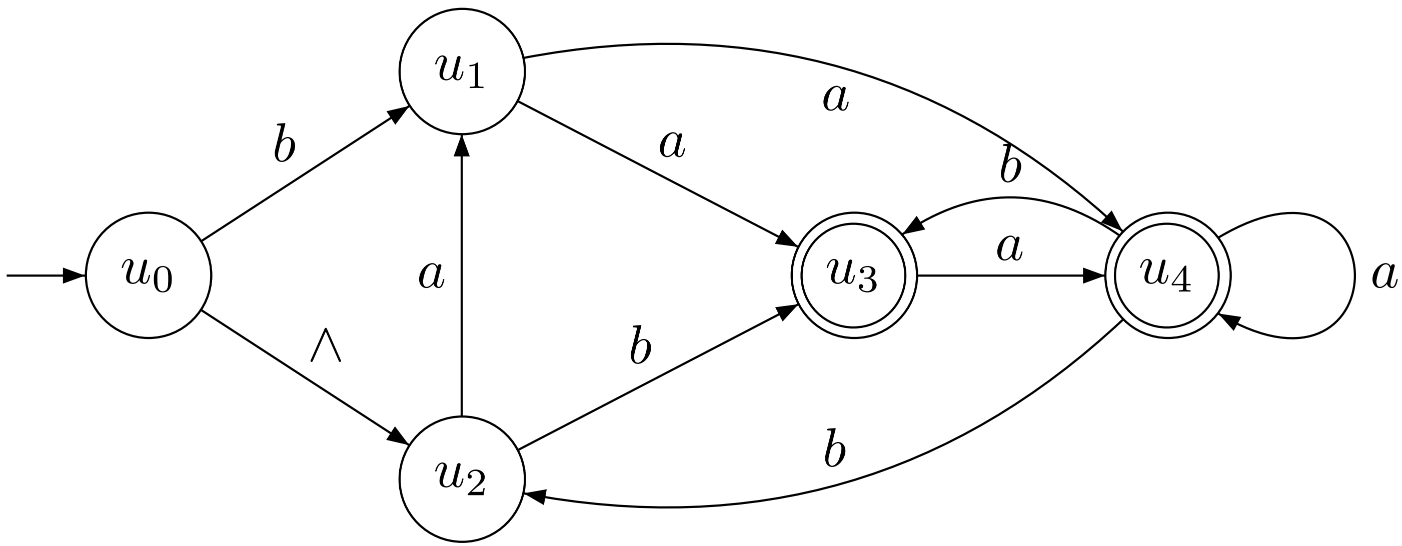
- **5.** Cho nguồn I như sau:



a. Xây dựng FA tương đương với nguồn I.
b. Xây dựng văn phạm chính quy tương đương với nguồn I.
c. Xây dựng biểu thức chính quy tương đương với nguồn I.

Bài tập chương 3

- **6.** Cho nguồn I như sau:



a. Xây dựng FA tương đương với nguồn I.

b. Xây dựng văn phạm chính quy tương đương với nguồn I.

c. Xây dựng biểu thức chính quy tương đương với nguồn I.

Bài tập chương 3

- **7.** Cho ôtômat $A = (S, \Sigma, s_0, \delta, F)$. Trong đó $S = \{s_0, s_1, s_2, s_3\}$, $\Sigma = \{a, b, c\}$, $F = \{s_2, s_3\}$
Hàm chuyển δ cho bởi bảng:

δ	s_0	s_1	s_2	s_3
a	s_2	$\{s_2, s_3\}$	s_0	s_2
b	$\{s_0, s_1\}$	$\emptyset$	s_3	$\emptyset$
c	$\emptyset$	s_2	$\emptyset$	$\{s_1\}$

a. Xây dựng nguồn tương đương với ôtômat A.
b. Xây dựng văn phạm chính quy tương đương với ôtômat A.
c. Xây dựng biểu thức chính quy tương đương với ôtômat A.

Bài tập chương 3

- **8.** Cho ôtômat $A = (S, \Sigma, s_0, \delta, F)$. Trong đó
  $S = \{s_0, s_1, s_2, s_3, s_4\}$, $\Sigma = \{a, b, c\}$, $F = \{s_3, s_4\}$
  Hàm chuyển δ cho bởi bảng:

δ	s_0	s_1	s_2	s_3	s_4
a	s_1	$\{s_1, s_2\}$	$\emptyset$	s_2	s_2
b	$\{s_1, s_3\}$	$\emptyset$	s_4	$\emptyset$	s_0
c	$\emptyset$	s_4	$\emptyset$	$\{s_1\}$	$\emptyset$

a. Xây dựng nguồn tương đương với ôtômat A.
b. Xây dựng văn phạm chính quy tương đương với ôtômat A.
c. Xây dựng biểu thức chính quy tương đương với ôtômat A.

Bài tập chương 3

- **9.** Cho văn phạm chính quy G như sau:
  $G = (\{a, b, c\}, \{S, A, B\}, S, P)$, trong đó P gồm:
  $S \to a|bA|cB|aA, \quad A \to bB|a, \quad B \to cA|cB|a|c$
  a. Xây dựng nguồn tương đương với văn phạm chính quy G.
  b. Xây dựng ôtômat tương đương với văn phạm chính quy G.
  c. Xây dựng biểu thức chính quy tương đương với văn phạm chính quy G.

- **10.** Xây dựng nguồn, ôtômat, văn phạm chính quy tương đương với các biểu thức chính quy sau:
  a. $B_1 = (b \vee ac)^* \vee ab(c \vee bab \vee (a \vee ca)^*)$
  b. $B_2 = (a)^*ba(b \vee a \vee aba)^*ba(cba \vee cba(a \vee b)^*)^*.$
  c. $B_3 = (c \vee ac)^*bc(a \vee cb \vee a(c \vee a)^*)^*ab.$

- **11.** Chứng minh rằng ngôn ngữ $L = \{a^{n+1}b^{n+2} | n \in Z^+\}$ không phải là ngôn ngữ chính quy.

Bài tập chương 3

- **12.** Cho văn phạm chính quy G như sau:
  $G = (\{a, b, c\}, \{S, A, B\}, S, P)$, trong đó P gồm:
  $S \to aS|cB|bA, \quad A \to aB|b|c, \quad B \to bS|cA|a|b$
  a. Xây dựng nguồn tương đương với văn phạm chính quy G.
  b. Xây dựng ôtômat tương đương với văn phạm chính quy G.
  c. Xây dựng biểu thức chính quy tương đương với văn phạm chính quy G.
- **13.** Xây dựng nguồn, ôtômat, văn phạm chính quy tương đương với các biểu thức chính quy sau:
  a. $B_1 = (c \vee ba)^* \vee bc(a \vee cab \vee (b \vee ca)^*)$
  b. $B_2 = (b)^*(c \vee a)(a \vee b \vee cb)^* a(acb \vee c(c \vee b)^*)^*.$
  c. $B_3 = (b \vee a)^* ba(b \vee ca \vee b(cb \vee a)^*)^* c \vee b.$
- **14.** Chứng minh rằng ngôn ngữ $L = \{a^{n+1}cb^{2n} | n \in Z^+\}$ không phải là ngôn ngữ chính quy.

Chương 4 - Giới thiệu chung

Như ta đã biết trong chương 3, văn phạm chính quy tương đương với ôtômat hữu hạn. Trong chương này ta sẽ nghiên cứu ôtômat đẩy xuống, sự khác biệt chủ yếu giữa ôtômat đẩy xuống với ôtômat hữu hạn là ôtômat đẩy xuống có thêm một ngăn xếp chứa các ký tự theo kiểu "Last in First out" tức là phần tử mới được đưa vào sẽ được xếp trên phần tử trước đó và do đó nó được đưa ra trước. Chính sự khác biệt đó đã giúp ôtômat đẩy xuống đoán nhận được ngôn ngữ phi ngữ cảnh mà ôtômat hữu hạn không làm được. Phần cuối chương này đề cập tới vấn đề kiểm tra xem một xâu có thuộc ngôn ngữ sinh bởi văn phạm đã cho hay không. Đây là một bài toán rất khó đối với một văn phạm bất kỳ nhưng đối với văn phạm LL_1 thì ta có thuật toán phân tích tất định để trả lời câu hỏi đó.

Ôtômat đẩy xuống

- **Định nghĩa.** Một ôtômat đẩy xuống PA là một bộ gồm 7 thành phần:
  $M = (\Sigma, Q, K, \delta, q_0, z_0, F)$ trong đó
  - Σ là tập hữu hạn, khác rỗng các ký tự đầu vào (bảng chữ cái).
  - Q là tập hữu hạn khác rỗng các trạng thái sao cho $\Sigma \cap Q = \emptyset$.
  - K tập hữu hạn, khác rỗng các ký tự của ngăn xếp.
  - $z_0 \in K$ là ký hiệu đầu tiên của danh sách đẩy xuống.
  - $F \subseteq Q$ là tập trạng thái kết thúc.
  - δ là hàm chuyển của M, có hai dạng:
    - Nếu δ là ánh xạ: $(\Sigma \cup \{\wedge\}) \times Q \times K \rightarrow Q \times K^*$ thì M được gọi là ôtômat đẩy xuống đơn định.
    - Nếu δ là ánh xạ: $(\Sigma \cup \{\wedge\}) \times Q \times K \rightarrow 2^{Q \times K^*}$ thì M được gọi là ôtômat không đơn định.
      (Trong chương này ta chỉ nghiên cứu ôtômat không đơn định và ký hiệu là PA)

Ví dụ

- **Ví dụ 1.** Cho ôtômat đẩy xuống
  $M_1 =< \Sigma, Q, K, \delta, q_0, z_0, F >$ trong đó
  $\Sigma = \{a, b\}, \quad Q = \{q_0, q_1\}, \quad K = \{a, b, z_0\}, \quad F = \{q_1\}$
  Hàm chuyển δ được xác định như sau:
  - $\delta(a, q_0, z_0) = (q_0, z_0 a)$
  - $\delta(a, q_0, a) = (q_0, a)$
  - $\delta(b, q_0, a) = (q_1, \wedge)$
- **Ví dụ 2.** Cho ôtômat đẩy xuống
  $M_2 =< \Sigma, Q, K, \delta, q_0, z_0, F >$ trong đó
  $\Sigma = \{0, 1, c\}, \qquad Q = \{q_0, q_1, q_2\}$
  $K = \{0, 1, c, z_0\}, \qquad F = \{q_2\}$
  Hàm chuyển δ được xác định như sau:
  - $\delta(a, q_0, k) = (q_0, ka) \quad \forall a \in \Sigma \backslash \{c\}, \forall k \in K$
  - $\delta(c, q_0, k) = (q_1, k) \quad \forall k \in K$
  - $\delta(a, q_1, a) = (q_1, \wedge) \quad \forall a \in \Sigma \backslash \{c\}$
  - $\delta(\wedge, q_1, z_0) = (q_2, \wedge)$

Bước chuyển của $\mathcal{PA}$

Giả sử PA đang ở trạng thái q, đầu đọc đang nhìn vào ký hiệu a trên bảng đầu vào và z là ký hiệu ở đỉnh ngăn xếp hiện tại và hàm chuyển có đẳng thức:

- $\delta(a, q, z) = \{(q_1, \gamma_1), (q_2, \gamma_2), \cdots (q_m, \gamma_m)\}$
  Khi đó PA có thể thực hiện một bước chuyển như sau:
  - PA chuyển từ trạng thái q sang trạng thái q_i.
  - Thay ký hiệu z ở đỉnh ngăn xếp bằng xâu γ_i (ký hiệu bên phải nhất của γ_i nằm ở đỉnh ngăn xếp, còn ký hiệu bên trái nhất nằm ở dưới thay vị trí của z).
  - Đầu đọc chuyển sang phải một ô.
- $\delta(\wedge, q, z) = \{(q_1, \gamma_1), (q_2, \gamma_2), \cdots (q_m, \gamma_m)\}$
  Khi đó PA có thể thực hiện một bước chuyển "nhắm mắt" như sau:
  - PA chuyển từ trạng thái q sang trạng thái q_i.
  - Thay ký hiệu z ở đỉnh ngăn xếp bằng xâu γ_i (ký hiệu bên phải nhất của γ_i nằm ở đỉnh ngăn xếp, còn ký hiệu bên trái nhất nằm ở dưới thay vị trí của z) nhưng đầu đọc không dịch chuyển.

Ngôn ngữ đoán nhận của PA

- **Hình trạng.**
  Ở một thời điểm khi PA đang đọc xâu $\omega = a_1a_2\ldots a_n$, hình trạng tức thời của PA xác nhận bởi ba yếu tố:
  - Xâu $\alpha \in K^*$ trong ngăn xếp (xác định từ đáy của ngăn xếp lên đỉnh của ngăn xếp).
  - Trạng thái $q \in Q$.
  - Phần chưa đọc đến của ω là $\omega' = a_ka_{k+1}\ldots a_n$ (nếu đầu đọc đang nhìn vào a_k).

  Khi đó $H = (\omega', q, \alpha)$ được gọi là hình trạng hiện tại của PA.
- **Suy dẫn.**
  Giả sử PA có hình trạng hiện tại là $H = (x\omega, q, v_0k)$ đồng thời $(q', \gamma) \in \delta(x, q, k)$, với $x \in \Sigma \cup \{\wedge\},\ \omega \in \Sigma^*,\ k \in K,\ v_0 \in K^*$.
  Sau khi thực hiện bước chuyển, PA chuyển sang trạng thái q' thì hình trạng hiện thời của PA là $H' = (\omega, q', v_0\gamma)$.
  Khi đó ta nói rằng hình trạng H suy dẫn trực tiếp sang H', ký hiệu là $H \Rightarrow H'$.

Ngôn ngữ đoán nhận của PA

- **Định nghĩa.** Hình trạng H được gọi là suy dẫn ra hình trạng H' nếu tồn tại dãy các hình trạng: $H = H_0, H_1, \ldots, H_m = H'$ sao cho $H_i \Rightarrow H_{i+1}\ \forall i = \overline{0, m-1}$ và ký hiệu là $H \vdash H'$.
- **Định nghĩa.**
  $T(M) = \{\omega \in \Sigma^* \mid (\omega, q_0, z_0) \vdash (\wedge, q, v), q \in F, v \in K^*\}$ được gọi là ngôn ngữ đoán nhận bởi M theo trạng thái kết thúc.
- Chứng minh rằng: trong **ví dụ 1** thì $T(M_1) = \{a^n b \mid n \in \mathbb{N}^*\}$
- Chứng minh rằng: trong **ví dụ 2** thì $T(M_2) = \{\omega\, c\, \widetilde{\omega} \mid \omega \in \{0, 1\}^*\}$

Ví dụ

- Cho ôtômat đẩy xuống $M =< \Sigma, Q, K, \delta, q_0, z_0, F >$ trong đó
  $\Sigma = \{a, b, c\}, \quad Q = \{q_0, q_1, q_2\}$
  $K = \{a, b, c, I, A, \xi\}, \quad z_0 = I, \quad F = \{q_2\}$
  Hàm chuyển δ được xác định như sau:
  1. $\delta(\wedge, q_0, I) = \{(q_1, \xi I)\}$
  2. $\delta(\wedge, q_1, I) = \{(q_1, b^2 Ia), (q_1, Ac)\}$
  3. $\delta(\wedge, q_1, A) = \{(q_1, Ac), (q_1, \wedge)\}$
  4. $\delta(a, q_1, a) = \{(q_1, \wedge)\} \quad \forall a \in \Sigma.$
  5. $\delta(\wedge, q_1, \xi) = \{(q_2, \xi)\}$
- Quá trình đoán nhận xâu $\omega = a^2cb^4$ thể hiện qua diễn biến các hình trạng sau:
  $(a^2cb^4, q_0, I) \xmapsto{1} (a^2cb^4, q_1, \xi I) \xmapsto{2.1} (a^2cb^4, q_1, \xi b^2 Ia) \xmapsto{4}$
  $(acb^4, q_1, \xi b^2 I) \xmapsto{2.1} (acb^4, q_1, \xi b^4 Ia) \xmapsto{4} (cb^4, q_1, \xi b^4 I) \xmapsto{2.2}$
  $(cb^4, q_1, \xi b^4 Ac) \xmapsto{4} (b^4, q_1, \xi b^4 A) \xmapsto{3.2} (b^4, q_1, \xi b^4) \xmapsto{4} (b^3, q_1, \xi b^3) \xmapsto{4}$
  $(b^2, q_1, \xi b^2) \xmapsto{4} (b, q_1, \xi b) \xmapsto{4} (\wedge, q_1, \xi) \xmapsto{5} (\wedge, q_2, \xi)$

- **Định lý 1.**
Với mọi văn phạm phi ngữ cảnh G luôn tồn tại một ôtômat đẩy xuống không đơn định M, đoán nhận $L(G)$ theo tập trạng thái kết thúc, hay $T(M) = L(G)$.

- **Ví dụ.** Cho văn phạm phi ngữ cảnh $G = (\{a,b\}, \{\sigma, A, B\}, \sigma, P)$ với:
$P = \{\sigma \to AB, A \to aA, A \to \wedge, B \to bB, B \to b\}$
Ôtômat đẩy xuống không đơn định đoán nhận $L(G)$ là:
$M = (\{a,b\}, \{q_0, q_1, q_2\}, \{a, b, \sigma, A, B, \xi\}, \delta, q_0, \sigma, \{q_2\})$ với δ được xác định như sau:

1. $\delta(\wedge, q_0, \sigma) = \{(q_1, \xi\sigma)\}$
2. $\delta(\wedge, q_1, \sigma) = \{(q_1, BA)\}$
   $\delta(\wedge, q_1, A) = \{(q_1, Aa), (q_1, \wedge)\}$
   $\delta(\wedge, q_1, B) = \{(q_1, Bb), (q_1, b)\}$
3. $\delta(a, q_1, a) = \{(q_1, \wedge)\}, \quad \delta(b, q_1, b) = \{(q_1, \wedge)\}$
4. $\delta(\wedge, q_1, \xi) = \{(q_2, \xi)\}$

Ví dụ đoán nhận trong $L(G)$ trong M

Dễ thấy $\omega = a^2b^3 \in L(G)$ và có dẫn xuất:
$D = (\sigma, AB, aAB, a^2AB, a^2B, a^2bB, a^2b^2B, a^2b^3)$
Quá trình biến đổi hình trạng của ω trong M như sau:
$(a^2b^3, q_0, \sigma) \xmapsto{1} (a^2b^3, q_1, \xi\sigma) \xmapsto{2} (a^2b^3, q_1, \xi BA) \xmapsto{2}$
$((a^2b^3, q_1, \xi BAa) \xmapsto{3} (ab^3, q_1, \xi BA) \xmapsto{2} (ab^3, q_1, \xi BAa) \xmapsto{3}$
$(b^3, q_1, \xi BA) \xmapsto{2} (b^3, q_1, \xi B) \xmapsto{2} (b^3, q_1, \xi Bb) \xmapsto{3} (b^2, q_1, \xi B) \xmapsto{3}$
$(b^2, q_1, \xi Bb) \xmapsto{3} (b, q_1, \xi B) \xmapsto{2} (b, q_1, \xi b) \xmapsto{3} (\wedge, q_1, \xi) \xmapsto{4} (\wedge, q_2, \xi)$
Ngược lại, từ suy dẫn trên ta ghi lại những phép thế loại 2 và viết ngược lại dẫn xuất đánh dấu của ω như sau:

$$D' = ({}^*\sigma^*, {}^*A^*B, a^*A^*B, a^{2*}A^*B, a^{2*}B^*, a^2b^*B^*, a^2b^{2*}B^*, a^2b^3)$$

Thuật toán 1

- Thuật toán xây dựng ôtômat đẩy xuống không đơn định đoán nhận ngôn ngữ phi ngữ cảnh $L(G)$ là:

$$M = (\Sigma, Q, K, \delta, q_0, z_0, F) \text{ trong đó}$$

  - Σ là bảng kí hiệu đầu vào chính là Σ của G.
  - $Q = \{q_0, q_1, q_2\}$ là tập các trạng thái.
  - $K = \Sigma \cup V \cup \{\xi\}$ là các kí hiệu của ngăn xếp.
  - $z_0 = \sigma$ là kí hiệu đầu tiên của ngăn xếp.
  - $q_0 \in Q$ là trạng thái ban đầu.
  - $F = \{q_2\}$ là tập trạng thái kết thúc.

Hàm chuyển $\delta : (\Sigma \cup \{\wedge\}) \times Q \times K \rightarrow 2^{Q \times K^*}$ xác định như sau:

1. $\delta(\wedge, q_0, \sigma) = \{(q_1, \xi\sigma)\}$
2. $\delta(\wedge, q_1, \varphi) = \{(q_1, \widetilde{\omega}) | (\varphi \rightarrow \omega) \in P, \varphi \in \Delta, \omega \in K^*\}$
3. $\delta(a, q_1, a) = \{(q_1, \wedge)\}, \forall a \in \Sigma$
4. $\delta(\wedge, q_1, \xi) = \{(q_2, \xi)\}$

Sự tương đương của VPPNC và PA

- **Định lý 2.**
  Mọi ôtômat đẩy xuống M đều tồn tại văn phạm phi ngữ cảnh G sao cho $L(G) = T(M)$.
  Kết hợp định lý 1 với định lý 2 ta có định lý sau:
- **Định lý 3.** Lớp ngôn ngữ đoán nhận bởi văn phạm phi ngữ cảnh bằng lớp ngôn ngữ đoán nhận bởi ôtômat đẩy xuống.

Định lý 4

- **Định nghĩa.**
  Hai ôtômat đẩy xuống M và M' được gọi là tương đương nếu chúng đoán nhận cùng một ngôn ngữ, tức là $T(M) = T(M')$.

- **Định lý 4.** Đối với mỗi ôtômat đẩy xuống M luôn tồn tại ôtômat đẩy xuống không đơn định M' tương đương với M, có một trạng thái kết và quá trình chuyển về trạng thái này ngăn xếp trở nên rỗng.

- **Ví dụ.** Cho ôtômat đẩy xuống $M = (\Sigma, Q, K, \delta, q_0, z_0, F)$ là:
  $M = (\{a, b, c, d\}, \{q_0, q_1, q_2, q_3\}, \{a, b, c, d, z_0\}, \delta, q_0, z_0, \{q_1, q_3\})$ với:
  1. $\delta(a, q_0, k) = \{(q_0, kc)\}, \quad \forall k \in \{a, b, c, z_0\}$
  2. $\delta(b, q_0, k) = \{(q_0, k)\}, \quad \forall k \in \{a, b, c, z_0\}$
  3. $\delta(c, q_0, c) = \{(q_0, \wedge)\}$
  4. $\delta(\wedge, q_0, z_0) = \{(q_1, z_0)\}$
  5. $\delta(d, q_0, k) = \{(q_2, k)\}, \quad \forall k \in \{a, b, c, z_0\}$
  6. $\delta(d, q_2, k) = \{(q_2, \wedge)\}, \quad \forall k \in \{a, b, c, z_0\}$
  7. $\delta(\wedge, q_2, z_0) = \{(\wedge, q_3, z_0)\}$

Ví dụ

Ôtômat đẩy xuống M' tương đương với M được xây dựng như sau:

$$M' = (\{a, b, c, d\}, \{q_0, q_1, q_2, q_3, q'_0, q'_1, q'_2\}, \{a, b, c, d, z_0, t_0\}, \delta', q'_0, t_0, \{q'_2\})$$

1. $\delta'(\wedge, q'_0, t_0) = \{(q_0, t_0 z_0)\}$
2. $\delta'(x, q, z) = \delta(x, q, z) \quad \forall x \in \Sigma,\ q \in \{q_0, q_1, q_2, q_3\}, z \in K$
3. $\delta'(\wedge, q, z) = \begin{cases} \delta(\wedge, q, z) & \text{nếu } q \in \{q_0, q_2\},\ z \in K \\ \delta(\wedge, q, z) \cup \{(q'_1, z)\} & \text{nếu } q \in \{q_1, q_3\},\ z \in K \end{cases}$
4. $\delta'(\wedge, q'_1, z) = \{(q_1, \wedge)\}, \quad z \in K$
5. $\delta'(\wedge, q'_1, t_0) = \{(q'_2, \wedge)\}$
6. $\delta'(\wedge, q, t_0) = \{(q'_2, \wedge)\}, \quad q \in \{q_1, q_3\}$

Đoán nhận trong M và M'

- $\omega = ab^3c$ được đoán nhận trong M như sau:
$(ab^3c, q_0, z_0) \xmapsto{1} (b^3c, q_0, z_0c) \xmapsto{2} (b^2c, q_0, z_0c) \xmapsto{2} (bc, q_0, z_0c) \xmapsto{2}$
$(c, q_0, z_0c) \xmapsto{3} (\wedge, q_0, z_0) \xmapsto{4} (\wedge, q_1, z_0)$
- $\omega = ab^3c$ được đoán nhận trong M' như sau:
$(ab^3c, q_0', t_0) \xmapsto{1} (ab^3c, q_0, t_0z_0) \xmapsto{2} (b^3c, q_0, t_0z_0c) \xmapsto{2}$
$(b^2c, q_0, t_0z_0c) \xmapsto{2} (bc, q_0, t_0z_0c) \xmapsto{2} (c, q_0, t_0z_0c) \xmapsto{2} (\wedge, q_0, t_0z_0) \xmapsto{2}$
$(\wedge, q_1, t_0z_0) \xmapsto{3} (\wedge, q_1', t_0z_0) \xmapsto{4} (\wedge, q_1', t_0) \xmapsto{5} (\wedge, q_2', \wedge)$

Thuật toán xây dựng M' từ M

Giả sử $M = (\Sigma, Q, K, \delta, q_0, z_0, F)$. Khi đó
$M' = (\Sigma', Q', K', \delta', q_0', \xi, F')$ xác định như sau:
$\Sigma' = \Sigma$, $Q' = Q \cup \{q_0', q_1', q_2'\}$
$K' = K \cup \{\xi\}, \quad F' = \{q_2'\}$

1. $\delta'(\wedge, q_0', \xi) = \{(q_0, \xi z_0)\}$
2. $\delta'(x, q, z) = \delta(x, q, z),\ x \in \Sigma',\ q \in Q,\ z \in K$
3. $\delta'(\wedge, q, z) = \begin{cases} \delta(\wedge, q, z) & \text{nếu } q \in Q \setminus F,\ z \in K \\ \delta(\wedge, q, z) \cup \{(q_1', z)\} & \text{nếu } q \in F,\ z \in K \end{cases}$
4. $\delta'(\wedge, q_1', z) = \{(q_1', \wedge)\}\ \forall z \in K$
5. $\delta'(\wedge, q_1', \xi) = \{(q_2', \wedge)\}$
6. $\delta'(\wedge, q, \xi) = \{(q_2', \wedge)\},\ \forall q \in F$

Hàm $First$

- **Định nghĩa.** Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm phi ngữ cảnh, không đệ quy trái và $\alpha \in (\Sigma \cup V)^*$, hàm $First(\alpha)$ xác định như sau:
  1. $\forall a \in \Sigma, First(a) = \{a\}, First(\wedge) = \{\wedge\}$.
  2. $\forall A \in V, First(A) = \bigcup_{i=1}^{k} First(\alpha_i)$ nếu $A \to \alpha_1|\alpha_2|\ldots|\alpha_k$ là tập hợp tất cả các phép thế có vế trái là A.
  3. Nếu $\wedge \in First(\beta_1), \ldots, First(\beta_i)$ và $\wedge \notin First(\beta_{i+1})$ $(i < n)$ thì $First(\beta_1\beta_2\ldots\beta_n) = (First(\beta_1) \cup \ldots \cup First(\beta_{i+1})) \setminus \{\wedge\}$
  4. Nếu $\wedge \in First(\beta_1), First(\beta_2), \ldots, First(\beta_n)$ thì: $First(\beta_1\beta_2\ldots\beta_n) = First(\beta_1) \cup \ldots \cup First(\beta_n)$
- **Nhận xét.** Nếu $\wedge \notin First(\alpha)$ thì $First(\alpha)$ là tất cả các ký hiệu cơ bản ở vị trí đầu tiên của tất cả các xâu β mà α có thể suy dẫn được.
- **Ví dụ.** $G = (\Sigma, V, \sigma, P)$ trong đó: $\Sigma = \{a, b, c, d\}$, $V = \{\sigma, A, B\}$ $P = \{\sigma \to AB, \sigma \to cBA, A \to aB, A \to Bd, B \to b, B \to \wedge\}$
  Khi đó: $First(B) = \{\wedge, b\}$, $First(A) = First(Bd) \cup First(aB) = ((First(B) \cup First(d)) \setminus \{\wedge\}) \cup \{a\} = \{a, b, d\}$.
  $First(\sigma) = First(AB) \cup First(cBA) = \{a, b, c, d\}$.

Hàm $Follow$

- **Định nghĩa.** Giả sử $G = (\Sigma, V, \sigma, P)$ là văn phạm phi ngữ cảnh, không đệ quy trái và $A \in V$. Hàm $Follow(A)$ được xác định như sau:
  Gọi $\alpha_i A \beta_i$ $(i = \overline{1, m})$, $\alpha_i, \beta_i \in (\Sigma \cup V)^*$ là tập hợp tất cả các vế phải chứa A của các phép thế trong P. Đặt $T(A) = \bigcup_{i=1}^{m} First(\beta_i)$
  - Nếu $\wedge \notin T(A)$ thì $Follow(A) = T(A)$
  - Nếu $\wedge \in T(A)$ thì $Follow(A) = T(A) \cup \{\$\}$
- **Ví dụ.** $G = (\Sigma, V, \sigma, P), \Sigma = \{a, b, c, d\}, \; V = \{\sigma, A, B, C\}$
  $P = \{\sigma \to AB, \sigma \to cBd, \sigma \to Aa, A \to BCA, A \to b$
  $B \to a, B \to \wedge, C \to cBa\}$
  Khi đó ta có:
  - $T(A) = First(B) \cup First(a) \cup First(\wedge) = \{a, \wedge\}$
    nên $Follow(A) = \{a, \wedge, \$\}$
  - $T(B) = First(\wedge) \cup First(d) \cup First(CA) \cup First(a) = \{\wedge, d, c, a\}$
    nên $Follow(B) = \{\wedge, d, c, a, \$\}$
  - $T(C) = First(A) = First(BCA) \cup First(b) = \{a, c, b\}$
    nên $Follow(C) = \{a, c, b\}$

Bảng phân tích của văn phạm

- **Định nghĩa.** Bảng phân tích M của văn phạm $G = (\Sigma, V, \sigma, P)$ là mảng hai chiều $M(A, b)$ ($A \in V, b \in \Sigma$) trong đó các quy tắc thế ghi trong ô có tọa độ (A, b) được xác định như sau:
  - Nếu $b \in First(A)$ thì liệt kê tất cả các dẫn xuất $(A, \omega_1, \ldots, b\alpha)$ sau đó đưa phép thế $A \to \omega_1$ vào ô (A, b).
  - Nếu $\wedge \in First(A)$ và $b \in Follow(A)$ thì đưa tất cả các phép thế có dạng $A \to \omega_1$ vào ô (A, b) trong đó $(A = \omega_0, \omega_1, \ldots, \omega_k = \wedge)$ một dẫn xuất từ A đến $\wedge$.
- **Ví dụ.** Cho văn phạm $G = (\{x, y, z\}, \{I, A, B\}, I, P)$ với $P = \{I \to xA,\ I \to yB,\ A \to x,\ A \to z,\ A \to By$ $B \to xA,\ B \to y,\ B \to \wedge\}$. Bảng phân tích M của văn phạm G là:

	x	y	z
I	$I \to xA$	$I \to yB$	
A	$A \to x, A \to By$	$A \to By$	$A \to z$
B	$B \to xA$	$B \to y, B \to \wedge$	

Văn phạm LL_1

- **Định nghĩa.** Văn phạm phi ngữ cảnh, không đệ quy trái G được gọi là văn phạm LL_1 nếu bảng phân tích của nó không có quá một phép thế ở mỗi ô.
- **Ví dụ.** Cho văn phạm $G = (\{a, b, c\}, \{I, A, B\}, I, P)$ với
  $P = \{I \to aA,\ I \to bB,\ A \to a,\ A \to Bc,\ B \to bA,\ B \to \wedge\}$
  Bảng phân tích M của văn phạm G là:

	a	b	c
I	$I \to aA$	$I \to bB$	
A	$A \to a$	$A \to Bc$	$A \to Bc$
B		$B \to bA$	$B \to \wedge$

  Theo bảng phân tích trên thì G là văn phạm LL_1.
- Nếu G là văn phạm LL_1 thì ta có thuật toán phân tích tất định để kiểm tra xem xâu ω có thuộc $L(G)$ không? Nếu có thì ta có thể in ra cây suy dẫn của nó.

Thuật toán phân tích tất định

Dùng một bảng để phân tích gồm 3 cột lần lượt là: $Stack$, $Input$, $Output$. Đặt $\omega\$$ vào dòng đầu tiên của $Input$, đặt $\$\sigma$ vào dòng đầu tiên của $Stack$. Gọi X là ký hiệu ở đỉnh của $Stack$, a là ký hiệu hiện thời của đầu đọc trên ω. Ban đầu, đầu đọc nhìn vào ký tự đầu tiên bên trái của ω, đỉnh của $Stack$ là σ.

- Nếu $X \in \Sigma$ thì:
  - Nếu $X = a = \$$ thì phân tích thành công, in ra cây theo lần lượt các phép thế ở $Output$.
  - Nếu $X = a \neq \$$ thì lấy X ra khỏi stack rồi viết xuống dòng dưới của cột $Stack$, dịch đầu đọc sang phải một ô rồi viết xâu còn lại trong $Input$ xuống dòng dưới của cột $Input$.
- Nếu $X \in V$, xét vị trí $M[X, a]$ trên bảng phân tích M.
  - Nếu có quy tắc $X \to \alpha\beta\gamma$ thì thay X bởi $\widetilde{\alpha\beta\gamma}$ trong $Stack$ (α nằm trên cùng của $Stack$) rồi viết xuống dòng dưới của cột $Stack$ và viết quy tắc $X \to \alpha\beta\gamma$ ra dòng tương ứng của $Output$.
  - Nếu không có quy tắc thì báo lỗi $error$ ra cột $Ouput$.

Ví dụ

- **Ví dụ.** Cho văn phạm $G = (\{a, b, c\}, \{I, A, B\}, I, P)$ với M là:

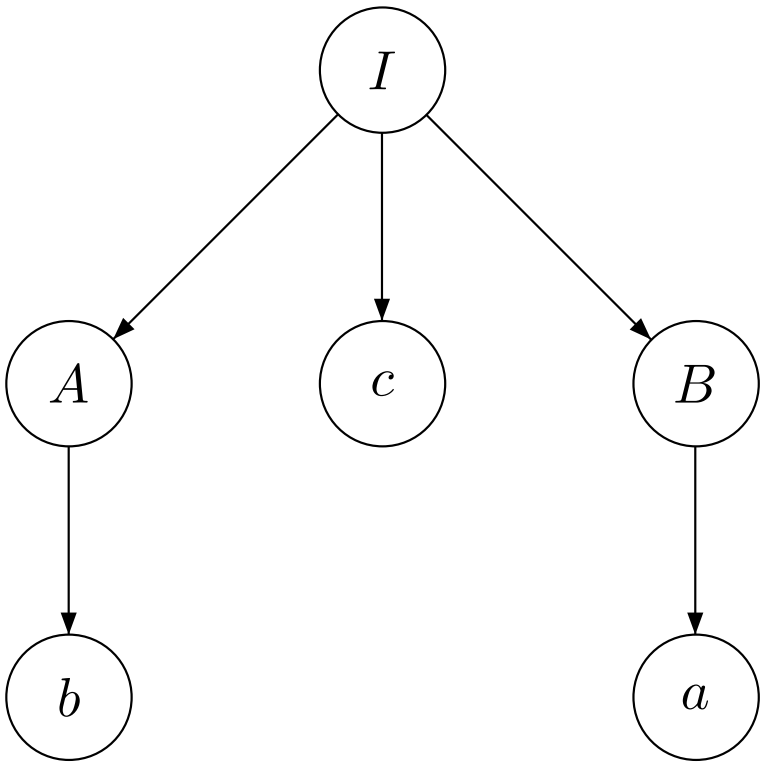
	a	b	c
I	$I \to aA$	$I \to AcB$	$I \to AcB$
A		$A \to b$	$A \to \wedge$
B	$B \to a$	$B \to bA$	

- Bảng phân tích xâu $\omega_1 = bca$ và $\omega_2 = cba$ là:

$Stack$	$Input$	$Output$	$Stack$	$Input$	$Output$
$\$I$	$bca\$$		$\$I$	$cba\$$	
$\$BcA$	$bca\$$	$I \to AcB$	$\$BcA$	$cba\$$	$I \to AcB$
$\$Bcb$	$bca\$$	$A \to b$	$\$Bc$	$cba\$$	$A \to \wedge$
$\$Bc$	$ca\$$		$\$B$	$ba\$$	
$\$B$	$a\$$		$\$Ab$	$ba\$$	$B \to bA$
$\$a$	$a\$$	$B \to a$	$\$A$	$a\$$	$error$
$\$$	$\$$				

Ví dụ

Cây phân tích của xâu $\omega_1 = bca$ là:



Bài tập chương 4

- **1.** Xây dựng ôtômat đẩy xuống tương đương với các văn phạm phi ngữ cảnh sau:

  a. $G_1 = (\{a, b, c\}, \{S, A, B, C\}, S, P_1)$

  $P_1 = \{S \to aS,\ S \to BA,\ A \to abB,\ B \to bAC,\ C \to c,\ A \to \wedge\}$

  b. $G_2 = (\{a, b, c\}, \{S, A, B, C\}, S, P_2)$

  $P_2 = \{S \to bS,\ S \to CbB,\ A \to aB,\ B \to bC,\ C \to b,\ A \to b\}$

- **2.** Tìm $First$ và $Follow$ của các ký hiệu bổ xung trong các văn phạm phi ngữ cảnh sau:

  a. $G_1 = (\{a, b, c\}, \{S, A, B, C\}, S, P_1)$
  $P_1 = \{S \to BC,\ S \to Ac,\ A \to abC,\ B \to aAC,\ C \to c,$
  $A \to \wedge,\ B \to \wedge\}$

  b. $G_2 = (\{a, b, c\}, \{S, A, B, C\}, S, P_2)$
  $P_2 = \{S \to aS,\ S \to CaB,\ A \to bA,\ B \to AbC,\ C \to a,\ C \to \wedge,$
  $A \to b,\ A \to \wedge\}$

Bài tập chương 4

- **3.** Cho các văn phạm phi ngữ cảnh sau:

$$G_1 = (\{a, b, c\}, \{S, A, B\}, S, P_1)$$
$$P_1 = \{S \to cA,\ S \to BA,\ A \to baB,\ B \to aA,\ B \to \wedge\}$$
$$G_2 = (\{a, b, c\}, \{S, A, B\}, S, P_2)$$
$$P_2 = \{S \to cS,\ S \to AB,\ A \to aB,\ B \to bA,\ A \to \wedge,\ B \to \wedge\}$$
$$G_3 = (\{a, b, c\}, \{S, A, B\}, S, P_3)$$
$$P_3 = \{S \to bcS,\ S \to aBAc,\ A \to BS,\ B \to cA,\ A \to \wedge,\ B \to \wedge\}$$

a. Chứng minh rằng $G_1,\ G_2,\ G_3$ là các văn phạm LL_1.

b. Lập bảng phân tích tất định xem xâu $\omega_1 = cbaaba$ và $\omega_2 = bcaa$ có thuộc $L(G_1)$ không? Nếu thuộc hãy vẽ cây suy dẫn.

c. Lập bảng phân tích tất định xem xâu $\omega_3 = ccba$ và $\omega_4 = cbab$ có thuộc $L(G_2)$ không? Nếu thuộc hãy vẽ cây suy dẫn.

d. Lập bảng phân tích tất định xem xâu $\omega_5 = bcacacc$ và $\omega_6 = accbcb$ có thuộc $L(G_3)$ không? Nếu thuộc hãy vẽ cây suy dẫn.

Tài liệu tham khảo

[1] Đặng Huy Ruận, *Lý thuyết ngôn ngữ hình thức và ôtômat,* NXB Đại học Quốc Gia Hà Nội, 2002.

[2] Đỗ Đức Giáo, *Toán rời rạc,* NXB Đại học Quốc Gia Hà Nội, 2000.

[3] Đỗ Đức Giáo, Đặng Huy Ruận, *Ngôn ngữ hình thức,* NXB Khoa học và Kỹ thuật, 1991.

[4] Phan Đình Diệu, *Lý thuyết ôtômat và thuật toán,* NXB Đại học và Trung học chuyên nghiệp, 1971.