

. . .

3 loại mô hình tính toán:

- Các văn phạm
- Các máy hữu hạn trạng thái
- Các máy Turing

1 Ngôn ngữ và văn phạm

Văn phạm cho biết 1 câu có đúng hay không. Chúng ta chỉ quan tâm **ngữ pháp** thay vì **ngữ nghĩa**. Cú pháp của một **ngôn ngữ tự nhiên** (ngôn ngữ nói) đều phức tạp.

Definition : Ngôn ngữ hình thức

- Làm tn xác định tổ hợp các từ có là câu đúng trong ngôn ngữ hình thức hay không
- Tạo ra 1 câu đúng

Thay thế đến khi không thay thế được nữa.

✔ Văn phạm cấu trúc trong câu

Văn phạm rộng nhất: VP cấu trúc câu.

- **Từ vựng V .** không rỗng, hữu hạn, chứa các kí hiệu.
- **Một ngôn ngữ trên V .** Một tập con của V^* .
- **Xâu rỗng λ .** Xâu không chứa kí hiệu, khác tập rỗng \emptyset .
- **Kí hiệu kết thúc T .** Không thể thay thế. (Khác với N)
- **Kí hiệu xuất phát S .** Phần tử bắt đầu.
- **Tập V^*** Mọi từ trên V .
- **Dẫn xuất.** $w_0 \rightarrow w_1$.

Definition : Văn phạm cấu trúc câu

Một VPCTC $G = (V, T, S, P)$ gồm một từ vựng V , một tập con T của V gồm các ký tự kết thúc, một ký hiệu xuất phát S và tập các dẫn xuất P .
 $V-T = N$ (không kết thúc). Mỗi dẫn xuất trong P cần chứa ít nhất 1 kí hiệu ko kết thúc ở vế trái.

📍 **VD1.** Văn phạm cấu trúc câu $G = VTSP$, $V = abABS$, $T = ab$, S là kí hiệu xuất phát và $P = \{(1)S \rightarrow ABa, (2)A \rightarrow BB, (3)B \rightarrow ab, (4)AB \rightarrow b\}$,

Definition : 3

$G = \{V, T, S, P\}$ là một văn phạm cấu trúc câu. Cho $w_0 = lz_0r$ và $w_1 = lz_1r$ là các xâu trên V . Nếu $z_0 \rightarrow z_1$ thì nói w_1 dẫn xuất trực tiếp từ w_0
■ Nếu $w_0 \rightarrow w_1 \rightarrow \dots w_n$ thì ta nói w_n dẫn xuất từ w_0 : $w_0 \rightarrow^* w_n$. Dãy các bước nhận được được gọi là **dẫn xuất**. Còn $S \rightarrow^* w_n$ là dẫn xuất đầy đủ của w_n .
■ Ngôn ngữ sinh ra bởi văn phạm G là tập các câu gồm toàn ký tự kết thúc xuất phát từ S :

$$L(G) = \{w \in T^*, S \rightarrow^* w\}$$

📍 **VD2.** Xâu Aaba được dẫn xuất trực tiếp từ ABa trong văn phạm VD1 vì $B \rightarrow ab$ là một dẫn xuất trong vavnp hạn đó. Hay abababa dc dẫn từ ABa. $Aba \rightarrow Aaba \rightarrow BBaba \rightarrow Bababa \rightarrow abababa$. Dùng lần lượt $B \rightarrow ab, A \rightarrow BB, B \rightarrow ab, B \rightarrow ab$.

📍 **VD3.** $V = S, A, a, b$, tập kết thúc $T = a, b$, ký hiệu xuất phát S và các dẫn xuất $P = \{(1)S \rightarrow aA, S \rightarrow B, 3)A \rightarrow aa\}$.

Ví dụ $S \rightarrow b$ là ký tự kết thúc rồi dừng. Còn từ aA sản xuất $A \rightarrow aa$ dẫn đến aaa . Ngoài ra ko dc từ nào. Do đó $L(G) = \{b, aaa\}$.

📍 **VD4.** $V = S, 0, 1$, tập kết thúc $T = 0, 1$, tập xuất phát S và các dẫn xuất $P = \{S \rightarrow 11S, S \rightarrow 0\}$. Xác định ngôn ngữ $L(G)$.

S có thể ra 0 hoặc 11S. Từ 11S dẫn xuất ra 110 hoặc 1111S, từ 1111S lại có thể ra 11110 hoặc 111111S.

$$L(G) = 0, 110, 11110, \dots$$

📍 **VD5.** Tìm văn phạm sinh ra $\{0^n 1^n \mid n = 0, 1, 2, \dots\}$

Có thể dùng 2 dẫn xuất. Xây dựng các dẫn xuất dẫn tới các xâu dài tuần tự trong ngôn ngữ bằng cách thêm 1 số 0 ở đầu và 1 số 1 ở cuối.

$T = \{0, 1\}$ Các sản xuất là: $\{S \rightarrow 0S1, S \rightarrow \lambda\}$.

📍 **VD6.** Số số 01 khác nhau được. $\{0^m 1^n, m, n > 0\}$.

2 văn phạm có thể sinh ra cùng ngôn ngữ.

$G_1 : V = S, 0, 1, T = 0, 1. \{S \rightarrow 0S, S \rightarrow S1, S \rightarrow \lambda\}$.

Bổ sung 1 ký tự không kết thúc A .

$$G_2 : V = S, A, 0, 1, T = 0, 1, \{S \rightarrow 0S, S \rightarrow 1A, S \rightarrow 1, A \rightarrow 1A, A \rightarrow 1, S \rightarrow \lambda\}$$

📍 **VD7.** $0^n 1^n 2^n \mid n = 0, 1, 2, \dots$

$G = \{V, T, S, P\}, V = \{0, 1, 2, S, A, B\}, T = \{0, 1, 2\}, S$.

Các dẫn xuất

$$\{S \rightarrow 0SAB, S \rightarrow \lambda, BA \rightarrow AB, 0A \rightarrow 01, 1A \rightarrow 11, 1B \rightarrow 12, 2B \rightarrow 22\}$$

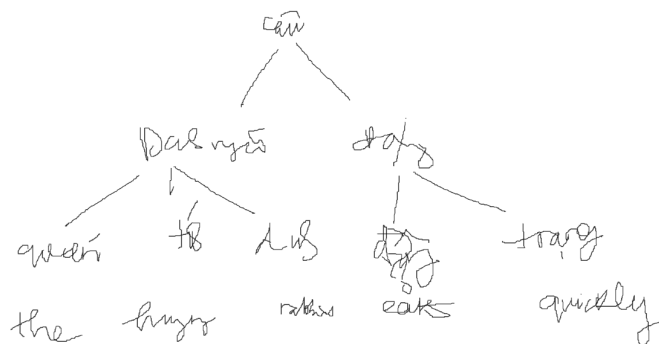
1.1 Các loại VPCTC

Được phân loại theo các sản xuất cho phép.

- **Loại 0 (Cấu trúc câu).** Không có hạn chế với các sản xuất của chúng.
- **Loại 1 (Cảm ngữ cảnh).** Ngày càng dài. Các dẫn xuất dạng $w_1 \rightarrow w_2$ trong đó w_2 dài hơn hoặc bằng w_1 hoặc có dạng $w_1 \rightarrow \lambda$.
 $l z r \rightarrow l z r$.
- **Loại 2 (Phi ngữ cảnh).** Chỉ có dạng $w_1 \rightarrow w_2$, trong đó w_1 phải là 1 ký hiệu đơn và không kết thúc.
- **Loại 3 (Chính quy).** $w_1 \rightarrow w_2 (w_1 = A, w_2 = aB)$ trong đó A, B không kết thúc hoặc $w_1 = S, w_2 = \lambda$.
Ngôn ngữ sinh ra gọi là **ngôn ngữ chính quy**.

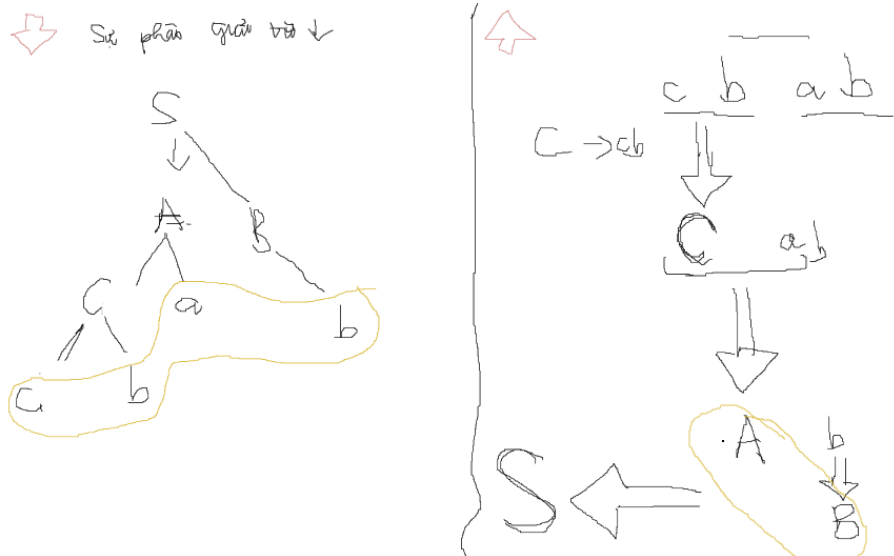
1.2 Cây dẫn xuất

1 dx trong nn sinh bởi phi ngữ cảnh là **cây dẫn xuất** hoặc **cây phân tích cú pháp**.



Tiếp cận trên xuống hoặc từ dưới lên.

VD12. XD $cbab$ có thuộc $V = \{a, b, c, A, B, C, S\}, T = \{a, b, c\}, S$ và dẫn xuất $S \rightarrow AB, A \rightarrow Ca, B \rightarrow Ba, B \rightarrow Cb, B \rightarrow B, C \rightarrow cb, C \rightarrow b$



1.3 Chuẩn BACKUS - NAUR

Các dẫn xuất trong VP2 có 1 ký hiệu đơn ko KT ở VT. Thay cho ký hiệu kéo theo thì $:=$. Đưa ký hiệu ko kết thúc vào $\langle \rangle$. Ví dụ $A \rightarrow a, A \rightarrow aA, A \rightarrow AB$ gộp thành

$$\langle A \rangle ::= a \langle A \rangle \mid \langle A \rangle \langle B \rangle$$

Trong ALGOL 60 một định danh bao gồm xâu các ký tự và chữ số. Phải bắt đầu bằng 1 chữ cái.

Ví dụ 1 định danh hợp lệ là **x99a**.

$$V = \{S, A, B, a, b\}, T = a, b, G = \{V, T, S, P\}$$

P gồm các dẫn xuất gồm:

- a) $S \rightarrow AB, A \rightarrow ab, B \rightarrow bb.$
- b) $S \rightarrow AB, S \rightarrow aA, A \rightarrow a, B \rightarrow ba$
- c) $S \rightarrow AB, S \rightarrow AA, A \rightarrow aB, A \rightarrow ab, B \rightarrow b$
- d) $S \rightarrow AA, S \rightarrow B, A \rightarrow aaA, A \rightarrow aa, B \rightarrow bB, B \rightarrow b$
- e) $S \rightarrow AB, A \rightarrow aAb, B \rightarrow bBa, A \rightarrow \lambda, B \rightarrow \lambda$

Giải.

$$a) L(G) = \{abbb\} \quad b) L(G) = \{\}$$

11. a)

$$V = \{S, 0, 1, A\}, T = \{0, 1\}, \{S \rightarrow A, A \rightarrow 00A, A \rightarrow \lambda\}$$

b)

c)

$$V = \{S, 0, 1, A, B\}, T = \{0, 1\}, \{S \rightarrow AB, A \rightarrow 00A, A \rightarrow \lambda, B \rightarrow B11, B \rightarrow \lambda\}$$

d)

2 Finite State Machine

Máy hữu hạn trạng thái dùng để **mô hình hóa** các loại máy và linh kiện trong máy tính.

2.1 Máy hữu hạn trạng thái có đầu ra

Definition : 1

1 máy hữu hạn trạng thái $M = (S, I, O, f, g, s_0)$ gồm:

- S là tập các trạng thái
- I tập chữ cái đầu vào
- O tập chữ đầu ra
- f hàm chuyển trạng thái ($f : S \times I \rightarrow S$)
- g hàm đầu ra ($g : S \times I \rightarrow O$)
- s_0 trạng thái ban đầu

3 FSM no output: Automat hữu hạn

Không cho đầu ra, có một tập trạng thái kết thúc và đoán nhận 1 xâu nếu và chỉ nếu xâu đó đưa đến 1 trạng thái kết thúc.

Definition : 1

A, V là 2 tập con của V^* , V là một từ vựng. Ghép AB là tập xâu dạng xy ($x \in A, y \in B$).

Ghép A^n .
$$\begin{cases} A^0 = \{\lambda\} \\ A^{n+1} = A^0 A \end{cases}$$

VD: $A = \{0, 11\}, B = \{1, 10, 110\}$. Then ...

Definition : Bao đóng Kleene

A là tập con của V^* . Bao đóng A^* gồm các phép ghép 1 số tùy ý các xâu thuộc A.

3.1 Automat hữu hạn - no output

Definition : 3

Một automat hữu hạn $M = (S, I, f, s_0, F)$ gồm:

- S là tập các state
- I là chữ cái đầu vào
- f cho trạng thái tiếp theo của (trạng thái, đầu vào) ($f : S \times I \rightarrow S$). f mở rộng: $S \times I^* \rightarrow S$.
- trạng thái xuất phát s_0
- tập con F gồm các trạng thái kết thúc

Ngôn ngữ đón nhận bởi M là $L(M)$, đưa từ đầu thành **trạng thái kết thúc**. Cùng đoán nhận thì 2 automat tương đương.

ATM tất định

ATM không tất định

Mỗi (s_i, i) đều có f là một s_f nào đó.

Mỗi (s_i, i) có thể có nhiều đầu ra.

Definition : ATM HH ko tất định

$M = (S, I, f, s_0, F)$, có $f : S \times I \rightarrow P(S)$ (tập bao gồm các tập con của S).

- **Tính chất.** Một ngôn ngữ đoán nhận bởi một automat hữu hạn không tất định cũng sẽ được đoán nhận bởi một automat hữu hạn tất định.

B4. Sự đoán nhận ngôn ngữ.

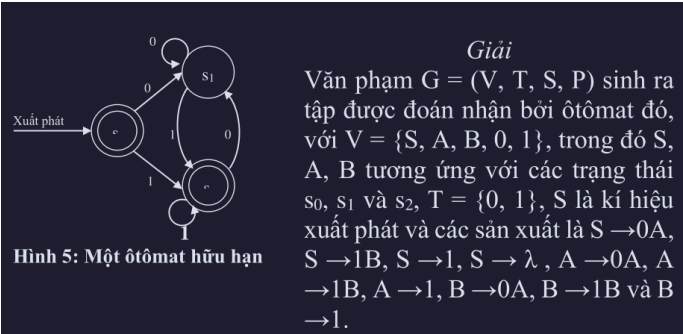
3.2 Tập chính quy

Các tập chính quy tạo từ ghép, hợp vào bao đóng Kleene (các biểu thức chính quy), đón nhận bởi **automat hữu hạn**.

- \emptyset : tập rỗng
- λ : xâu rỗng
- (AB) sự ghép

- $A \cup B$ sự hợp
- A^* bao đóng Kleene

Xây dựng văn phạm.



$1 * \cup (01)$

$(1 * \cup 0)1$

4 Máy Turing