Lecture 3. Grammars and formal languages

Hà Chí Trung, BM KHMT, KCNTT, HVKTQS

hct2009@yahoo.com

01685-582-102

Bài 3. Văn phạm và ngôn ngữ hình thức

1. Các khái niệm cơ bản
2. Các phép toán trên từ
3. Các phép toán trên ngôn ngữ
4. Văn phạm
5. Phân cấp Chomsky
6. Tính chất của văn phạm và ngôn ngữ
7. Все только начинается

1. Các khái niệm cơ bản

- **ĐN 3.1.** Tập Σ khác rỗng gồm hữu hạn hay vô hạn các ký hiệu được gọi là **bảng chữ cái**. Mỗi phần tử $a \in \Sigma$ được gọi là một **chữ cái** hay một **ký hiệu**.
  - ∑ = {a, b, c, … , x, y, z} bảng chữ cái tiếng Anh;
  - Δ = {α, β, γ, δ, ε, η, φ, κ, μ, χ, ν, π, θ, ρ, σ, τ, ω,ξ, ψ};
  - Γ = {0, 1} bảng chữ cái nhị phân;
- **ĐN 3.2.** Cho $\Sigma = (a_1, a_2, \dots, a_n)$, một dãy $a_{i1}a_{i2} \dots a_{im}, a_{ij} \in \Sigma$ được gọi là **một từ** hay một **xâu** (**chuỗi**) trên bảng Σ.
  - ε, 0, 01, 101, 1010, 110011 là các từ trên bảng chữ cái Γ = {0,1};
  - ε, beautiful, happy… là các từ trên Σ = {a, b, c, …, z}.

1. Các khái niệm cơ bản

- **ĐN 3.3. Độ dài chuỗi:** là số các ký hiệu tạo thành chuỗi. |abca| = 4
- **ĐN 3.4. Chuỗi rỗng:** ký hiệu **ε**, chuỗi không có ký hiệu nào. |ε| = 0
- **ĐN 3.5. Chuỗi con:** chuỗi v là chuỗi con của w nếu v được tạo bởi các ký hiệu liền kề nhau trong chuỗi w.
  - Chuỗi 10 là chuỗi con của chuỗi 0**10**001
- **ĐN 3.6. Chuỗi tiền tố:** là chuỗi con bất kỳ nằm ở đầu chuỗi
- **ĐN 3.7. Chuỗi hậu tố:** là chuỗi con bất kỳ nằm ở cuối chuỗi
  - Chuỗi abc có các tiền tố a, ab, abc
  - Chuỗi 0246 có các hậu tố 6, 46, 246, 0246

1. Các khái niệm cơ bản

- **ĐN 3.8:** Một ngôn ngữ (hình thức) L trên bộ chữ cái Σ là một **tập hợp các chuỗi** từ các ký hiệu của bộ chữ cái Σ.
  - Σ^* : tập hợp tất cả các chuỗi con, kể cả chuỗi rỗng ε, sinh ra từ bộ chữ cái Σ.
  - Σ^+ : tập hợp tất cả các chuỗi con, ngoại trừ chuỗi rỗng ε, sinh ra từ bộ chữ cái Σ.

$$\Sigma^* = \Sigma^+ + \{\varepsilon\} \qquad \Sigma+ = \Sigma^* - \{\varepsilon\}$$

- **Ví dụ 1:** Cho $\Sigma = \{0,1\}$:
  - $\Sigma^* = \{\boldsymbol{\varepsilon}, 0, 1, 00, 01, 10, 11, 000, \ldots\}$
  - $\Sigma^+ = \{0, 1, 00, 01, 10, 11, 000, \ldots\}$
  - Chuỗi 010**2**10 $\notin \Sigma^*$ vì có số 2 $\notin \Sigma$

2. Các phép toán trên từ

- **Phép nối kết** (concatenation), **phép đảo ngược** (reverse):
  - $\varepsilon w = w\varepsilon = w$ (với mọi w) → ε là đơn vị của phép nối kết
  - $w = abcd \rightarrow w^R = dcba$ $\varepsilon^R = \varepsilon$
- **Phép cắt trái** của từ α cho từ β - là phần còn lại của từ α sau khi cắt bỏ phần tiền tố β trong từ α.
- **Phép cắt phải** của từ α cho từ β - là phần còn lại của từ α sau khi cắt bỏ phần hậu tố β trong từ α.

3. Các phép toán trên ngôn ngữ

- Vì (có thể coi) mỗi ngôn ngữ là một tập hợp nên ta có các **phép toán đại số tập hợp:** phép hợp, phép giao, phép hiệu, phép lấy bù trên các ngôn ngữ.
- **Phép hợp:** $L_1 \cup L_2 = \{ \omega \in \Sigma^* \mid \omega \in L_1 \quad or \quad \omega \in L_2 \}$
- **Phép giao:** $L_1 \cap L_2 = \{ \omega \in \Sigma^* \mid \omega \in L_1 \quad and \quad \omega \in L_2 \}$
- **Phép phần bù (complement):** $\bar{L} = \Sigma^* - L$
- **Phép nhân ghép (concatenation):**

  $L_1L_2 = \{w_1w_2 \mid w_1 \in L_1$ và $w_2 \in L_2\}$ trên bộ chữ cái $\Sigma_1 \cup \Sigma_2$

  - LLL…LL = L^i (kết nối i lần trên cùng ngôn ngữ L)
  - $L^0 = \{\varepsilon\}$

3. Các phép toán trên ngôn ngữ

- **ĐN 3.10.** Ngôn ngữ **lặp** (bao đóng kleene, hoặc *-closure):

$$L^* = \bigcup_{i=0}^{\infty} L^i = \{\varepsilon\} \cup L \cup L^2 \cup \ldots \cup L^n \cup \ldots$$

- **ĐN 3.11.** Ngôn ngữ **lặp cắt** (bao đóng dương – positive closure):

$$L^+ = \bigcup_{i=1}^{\infty} L^i = L \cup L^2 \cup \ldots \cup L^n \cup \ldots$$

- **ĐN 3.12.** Ngôn ngữ **ngược:** $L^R = \{\omega \in \Sigma^* \mid \omega^R \in L\}$

- **ĐN 3.13.** Ngôn ngữ **cắt trái** của ngôn ngữ X cho ngôn ngữ Y:

$$Z = Y \setminus X = \{z \in \Sigma^* \mid x \in X, y \in Y \ mà \ x = yz\}$$

- **ĐN 3.14.** Ngôn ngữ **cắt phải** của ngôn ngữ X cho ngôn ngữ Y:

$$Z = X / Y = \{z \in \Sigma^* \mid x \in X, y \in Y \ mà \ x = zy\}$$

4. Văn phạm

- **Văn phạm** (được hiểu) là một tập các quy tắc về cấu tạo từ và các quy tắc về cách thức liên kết từ lại thành câu.

<câu>→<chủngữ><vịngữ>

<chủngữ>→tôi | anh | chị

<vịngữ>→<độngtừ><danhtừ>

<độngtừ>→ăn | uống

<danhtừ>→cơm | phở | sữa|…

4. Văn phạm

- **Backus - Naur Form – BNF**

<expression> ::= <expression> + <expression>

<expression> ::= <expression> * <expression>

<expression> ::= (<expression>)

<expression> ::= <identifier>

4. Văn phạm

- **Backus - Naur Form – BNF**

<sentence> ::= <noun phrase><verb phrase>

<noun phrase> ::= <article><adjective> <noun>|<article><noun>

<verb phrase> ::= <verb><adverb>|<verb>

<article> ::= *a*|*the*

<adjective> ::= *large*|*hungry*

<noun> ::= *rabbit*|*mathematician*

<verb> ::= *eats*|*hops*

<adverb> ::= *quickly*|*wildly*

4. Văn phạm

- **ĐN 3.15:** Văn phạm G là một bộ sắp thứ tự gồm 4 thành phần G = < Σ, Δ, S, P >, trong đó:
  - Σ - bảng chữ cái, gọi là bảng chữ cái cơ bản (bảng chữ cái kết thúc – **terminal symbol**);
  - Δ , Δ ∩ Σ =Ø, gọi là bảng ký hiệu phụ (báng chữ cái không kết thúc – **nonterminal symbol**);
  - S ∈ Δ - ký hiệu xuất phát hay tiên đề (**start variable**);
  - P - tập các luật sinh (**production rules**) dạng α→β, α, β ∈ (Σ ∪ Δ)*, trong α chứa ít nhất một ký hiệu không kết thúc (đôi khi, ta gọi chúng là các qui tắc hoặc luật viết lại).

4. Văn phạm

<Câu> => <Chủ ngữ><Vị ngữ>
<Chủ ngữ> => <Danh ngữ>
<Chủ ngữ> => <danh từ trừu tượng>
<Chủ ngữ> => <danh từ đơn thể>
<Chủ ngữ> => <đại từ số lượng>
<Chủ ngữ> => <đại từ không gian, thời gian>
....
<Vị ngữ> => <Động ngữ>
<Vị ngữ> => <Tính ngữ>
<Danh từ> => <danh từ trừu tượng>
<Danh từ> => <danh từ tổng thể>
...

4. Văn phạm

- **Qui ước:**
  - Chữ cái in hoa *A, B, C,…* để biểu thị các biến, trong đó *S* là ký hiệu xuất phát;
  - *X, Y, Z,…* để biểu diễn các ký tự chưa biết hoặc các biến;
  - *a, b, c, d, e,…* để biểu diễn chữ cái;
  - *u, v, w, x, y, z,…* để biểu diễn chuỗi chữ cái;
  - *α, β, ω,…* biểu thị chuỗi các biến hoặc các ký hiệu kết thúc.
- **Ví dụ 2:** G = ({a, b}, {S, A, B}, S, P), trong đó:

  P: S → aAS|bBS|ε,

  A → aaA|b,

  B → aaB|a

4. Văn phạm

- **ĐN 3.16. Dẫn xuất trực tiếp:** nếu $\alpha \rightarrow \beta$ là một luật sinh thì

$$\gamma\, \boldsymbol{\alpha}\, \delta \rightarrow\ \gamma\, \boldsymbol{\beta} \delta$$

- **ĐN 3.17. Dẫn xuất gián tiếp:** nếu các chuỗi $\alpha_1, \alpha_2, ..., \alpha_m \in \Sigma^*$ và $\alpha_1 \rightarrow \alpha_2, \alpha_2 \rightarrow \alpha_3, ..., \alpha_{m-1} \rightarrow \alpha_m$ thì α_m có thể được dẫn xuất gián tiếp từ α_1

$$\alpha_1 \rightarrow^* \alpha_m$$

- **ĐN 3.18. Ngôn ngữ L sinh bởi văn phạm G:**

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

- **ĐN 3.19. Văn phạm tương đương:** là 2 văn phạm sinh ra cùng một ngôn ngữ (G_1 tương đương $G_2 \Leftrightarrow L(G_1)=L(G_2)$)

4. Văn phạm

- **Cây dẫn xuất:** (hay cây phân tích cú pháp) của một văn phạm G = < Σ, Δ, S, P >:

  (1) Mỗi nút có một nhãn, là một ký hiệu ∈ (Σ ∪ Δ ∪ {ε})

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

  (3) Nếu nút trung gian có nhãn A thì A ∈ Δ

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

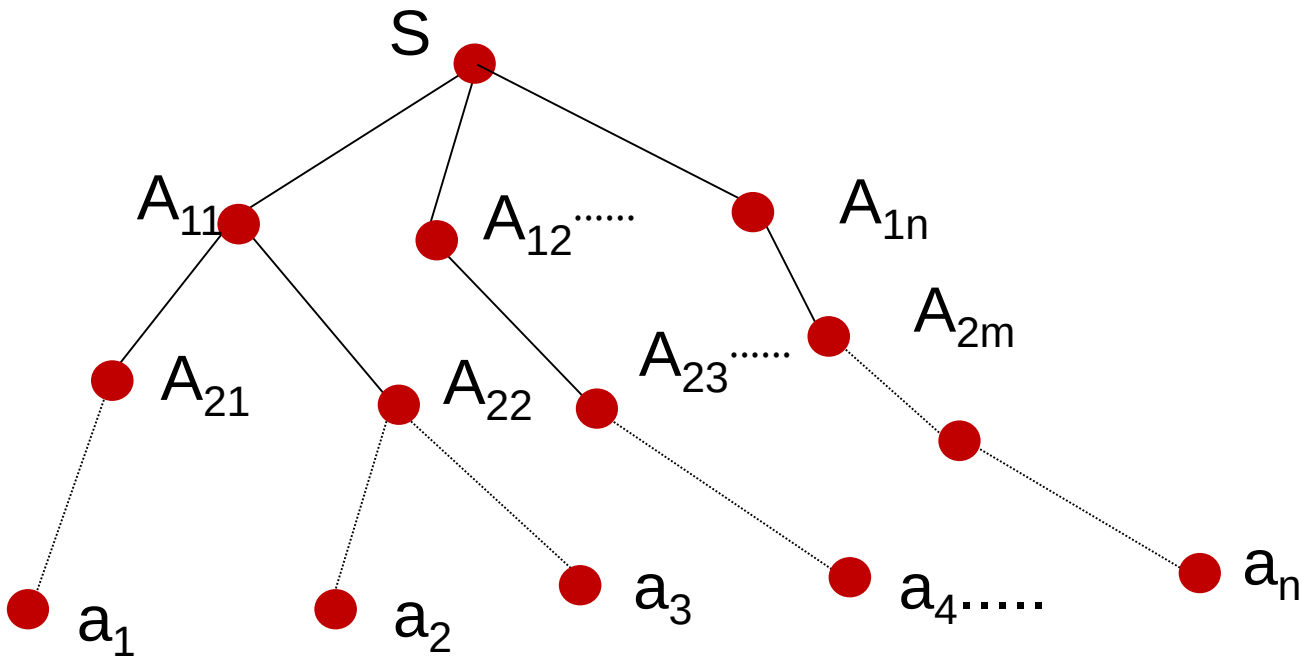
  (5) Nếu nút n có nhãn là ε thì n phải là nút lá và là nút con duy nhất của nút cha của nó

4. Văn phạm

- **Cây dẫn xuất:** Giả sử dẫn xuất của từ w= $a_1a_2...a_n$ là dãy qui tắc dạng:

$$S \rightarrow A_{11}A_{12}...A_{1n} \rightarrow A_{21}A_{22}...A_{2m}... \rightarrow a_1a_2...a_n$$

- Khi đó, ta có thể biểu diễn nó dưới dạng cây như sau:



4. Văn phạm

- **Định lý:** G = < Σ, Δ, S, P > là một CFG thì **S** → * α nếu và chỉ nếu có cây dẫn xuất trong văn phạm sinh ra α.
- **Dẫn xuất trái nhất (phải nhất):** nếu tại mỗi bước dẫn xuất, luật sinh được áp dụng vào biến bên trái nhất (phải nhất)
- **Ví dụ 3:** Cho G: S → AB; A → aA | a; B → bB | b;
  (a) S → AB → aAB → aaAB → aaaB → aaabB → aaabb
  (b) S → AB → AbB → Abb → aAbb → aaAbb → aaabb
  (c) S → AB → aAB → aAbB → aAbb → aaAbb → aaabb
  (d) S → AB → aAB → aaAB → aaAbB → aaabB → aaabb
- ***Lưu ý:*** *(a) là dẫn xuất trái nhất, (b) -phải nhất. Các dãy ở trên có cùng một cây dẫn xuất.*

4. Văn phạm

- **Ví dụ 4:** Xét văn phạm G ={{a, b}, {S, A}, S, P}, trong đó
  P = { S → aS, S → aA, A → bA, A → b }
- Ngôn ngữ sinh bởi văn phạm G?

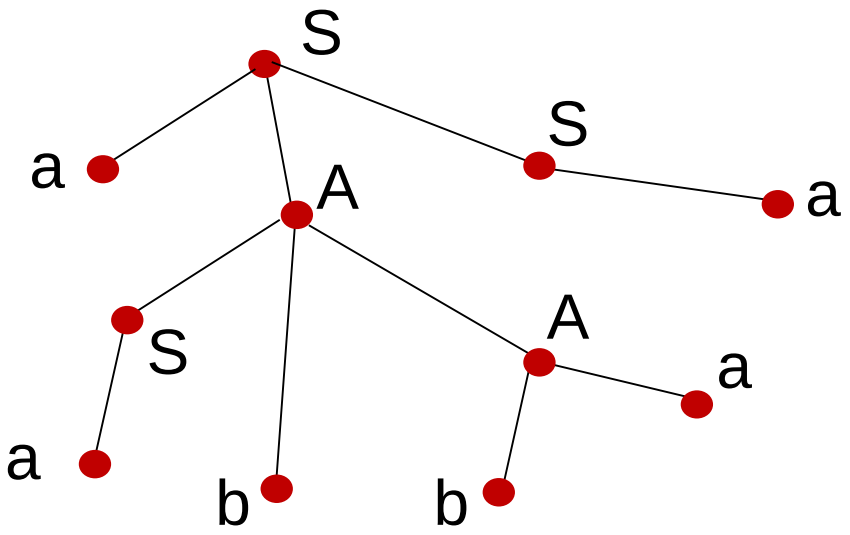
4. Văn phạm

- **Ví dụ 5:** Cho văn phạm:

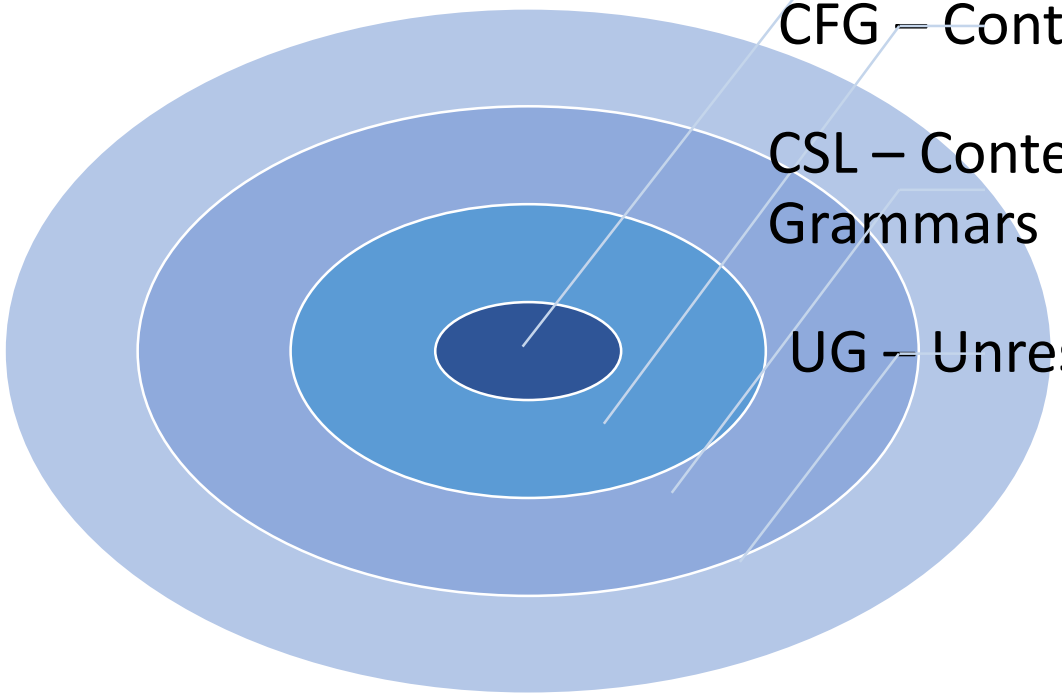
$$S \rightarrow aAS;\ A \rightarrow SbA\ ;\ A \rightarrow SS;\ S \rightarrow a;\ A \rightarrow ba$$

- Xét chuỗi: aabbaa

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



5. Chomsky hierarchy (1956)



RG – Regular Grammars

CFG – Context-Free Grammars

CSL – Context Sensitive Grammars

UG – Unrestricted Grammars

5. Chomsky hierarchy (1956)

- **ĐN 3.20. Văn phạm loại 0** – Văn phạm không hạn chế **(UG – Unrestricted Grammar):** không cần thỏa điều kiện ràng buộc nào trên tập các luật sinh;
- **ĐN 3.21. Văn phạm loại 1** – Văn phạm cảm ngữ cảnh **(CSG – Context Sensitive Grammar):** nếu văn phạm G có các luật sinh dạng α→β và:

$$\alpha = \alpha' A \alpha'',\ A \in \Delta, \alpha', \alpha'', \beta \in \left(\Sigma \cup \Delta\right)^*, |\beta| \geq |\alpha|;$$

- **ĐN 3.22. Văn phạm loại 2** – Văn phạm phi ngữ cảnh **(CFG – Context-Free Grammar):** có luật sinh dạng A→α với A là một biến đơn và α là chuỗi các ký hiệu thuộc (Σ ∪ Δ)*;

5. Chomsky hierarchy (1956)

- **ĐN 3.23. Văn phạm loại 3** – Văn phạm chính quy **(RG – Regular Grammar):** có mọi luật sinh dạng tuyến tính phải hoặc tuyến tính trái.
  - Tuyến tính phải: A → aB hoặc A → a;
  - Tuyến tính trái: A → Ba hoặc A → a;
  - Với A, B là các biến đơn, a là ký hiệu kết thúc (có thể là rỗng).

- Nếu ký hiệu L_0, L_1, L_2, L_3 là lớp các ngôn ngữ được sinh ra bởi văn phạm loại 0, 1, 2, 3 tương ứng, ta có:

$$L_3 \subset L_2 \subset L_1 \subset L_0.$$

5. Chomsky hierarchy (1956)

- **Ví dụ 6:** Xét văn phạm sau:

  G = ({a, b}, {S, A}, S, P), trong đó:

$$P = \begin{cases} S \rightarrow aS \\ S \rightarrow aA \\ A \rightarrow bA \\ A \rightarrow b \end{cases}$$

- Đây là văn phạm loại 3 (dạng tuyến tính phải)
- Một dẫn xuất từ S có dạng:

$$S \rightarrow aS \rightarrow aaS \rightarrow aaaA \rightarrow aaabA \rightarrow aaabbA \rightarrow aaabbbA \rightarrow aaabbbb = a^3b^4$$

$$L(G) = a^+b^+ = \{a^nb^m \mid n,m \geq 1\}$$

5. Chomsky hierarchy (1956)

- **Ví dụ 7:** Xét văn phạm sau:

$$G = (\{a, b\}, \{S\}, S , P) , \qquad P = \begin{cases} S \rightarrow aSb \\ S \rightarrow ab \end{cases}$$

- Đây là văn phạm loại 2 (dạng $A \rightarrow \alpha$)
- Một dẫn xuất từ S có dạng:

$$S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb \rightarrow aaaabbbb = a^4b^4$$

$$L(G) = \{a^n b^n \mid n \geq 1\}$$

5. Chomsky hierarchy (1956)

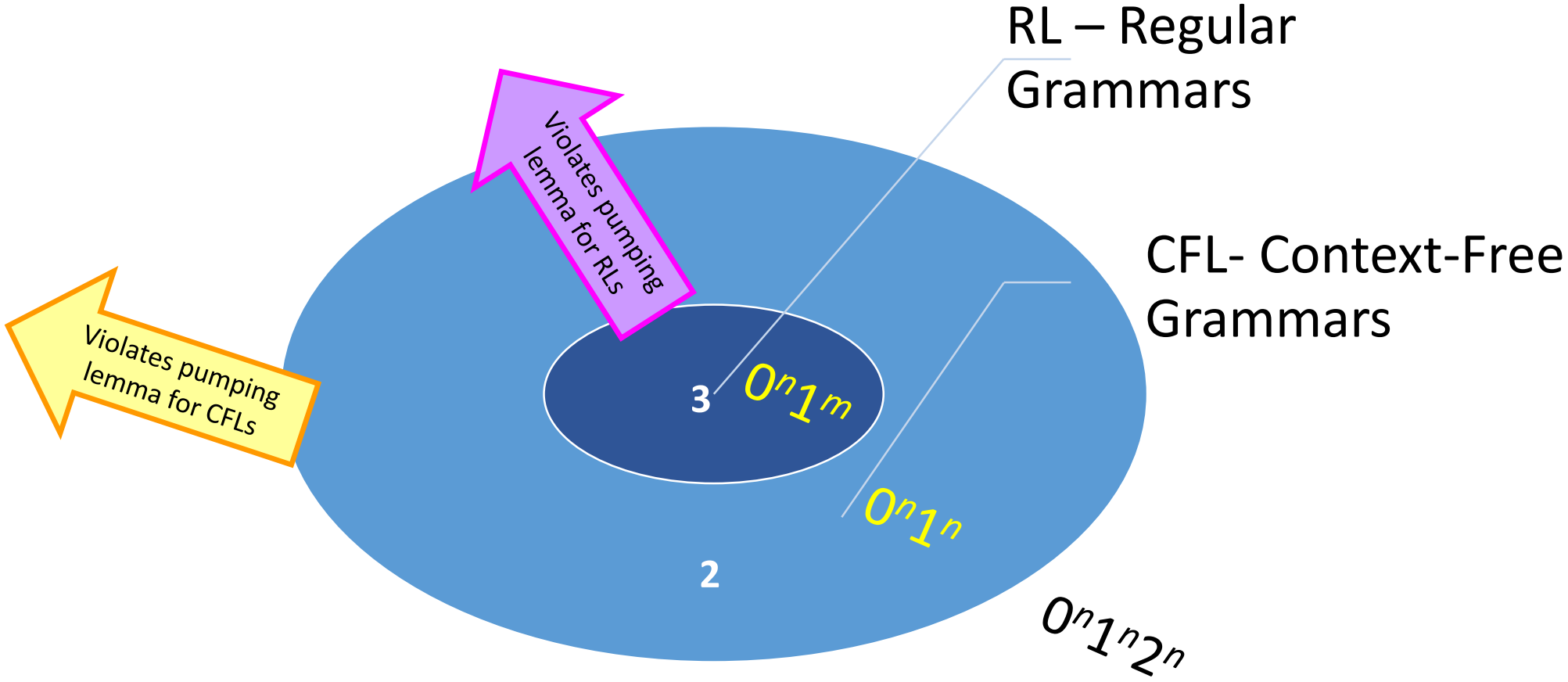
- **Ví dụ 8:** Xét văn phạm sau:

$$G(\{a, b, c\}, \{S, B, C\}, P, S), \quad P = \begin{cases} S \to aSBC \\ S \to aBC \\ CB \to BC \\ aB \to ab \\ bB \to bb \\ bC \to bc \\ cC \to cc \end{cases}$$

- Đây là văn phạm loại 1
- Một dẫn xuất từ S: $S \to aSBC \to aaBCBC \to aabCBC \to aabBCC \to aabbCC \to aabbcC \to aabbcc=a^2b^2c^2$

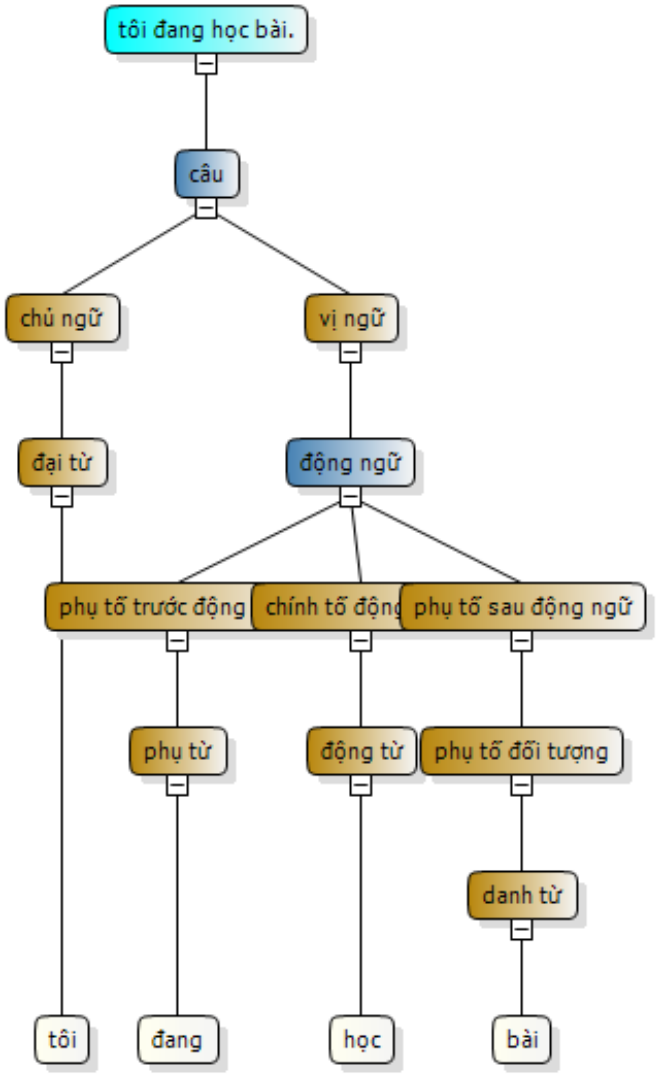
$$L(G) = \{a^n b^n c^n \mid n \geq 1\}$$

5. Chomsky hierarchy (1956)



6. Phân tích cú pháp

- tôi đang học bài.
  - câu
    - chủ ngữ
      - đại từ
        - tôi
    - vị ngữ
      - động ngữ
        - phụ tố trước động ngữ
          - phụ từ
            - đang
        - chính tố động ngữ
          - động từ
            - học
        - phụ tố sau động ngữ
          - phụ tố đối tượng
            - danh từ
              - bài



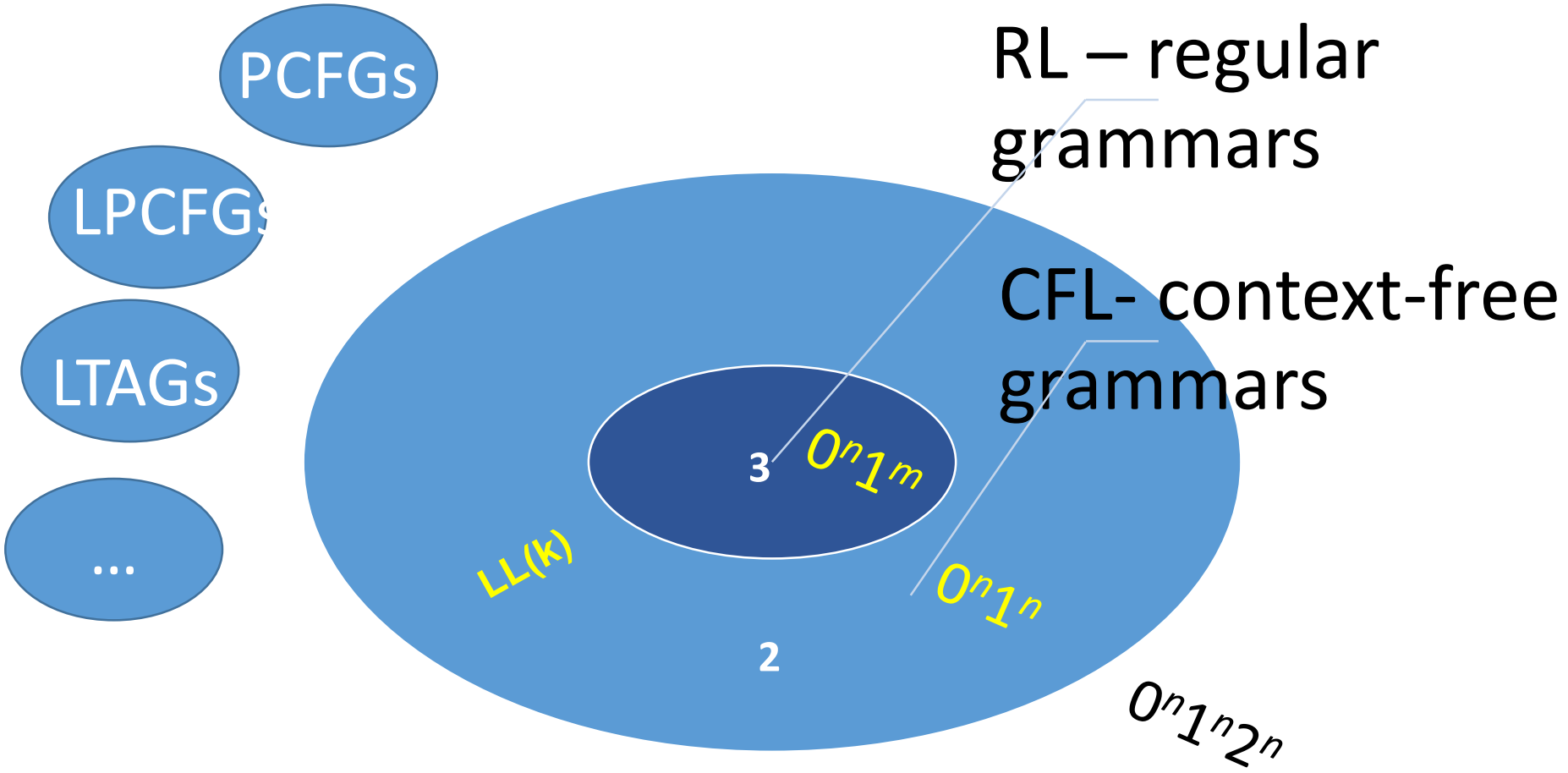
6. Phân tích cú pháp

- Top-down parsing.
- Bottom-up parsing.
  - Phân tích đệ quy ($O(c^n)$);
  - thuật toán phân tích **CYK** (**Coke-Younger-Kasami**) ($O(n^3)$);
  - (thuật toán phân tích **Earley)** ($O(n^3)$ hoặc $O(n^2)$ hoặc $O(n)$);
  - **LL(k)** for top-down parsing ($O(n)$);
  - **LR(k)** for với bottom-up parsing ($O(n)$).

6. Phân tích cú pháp

- LL(k) (Left-to-right parse, Leftmost-derivation, k-symbol lockahead);
- PCFGs
- LPCFGs
- LTAGs

6. Phân tích cú pháp



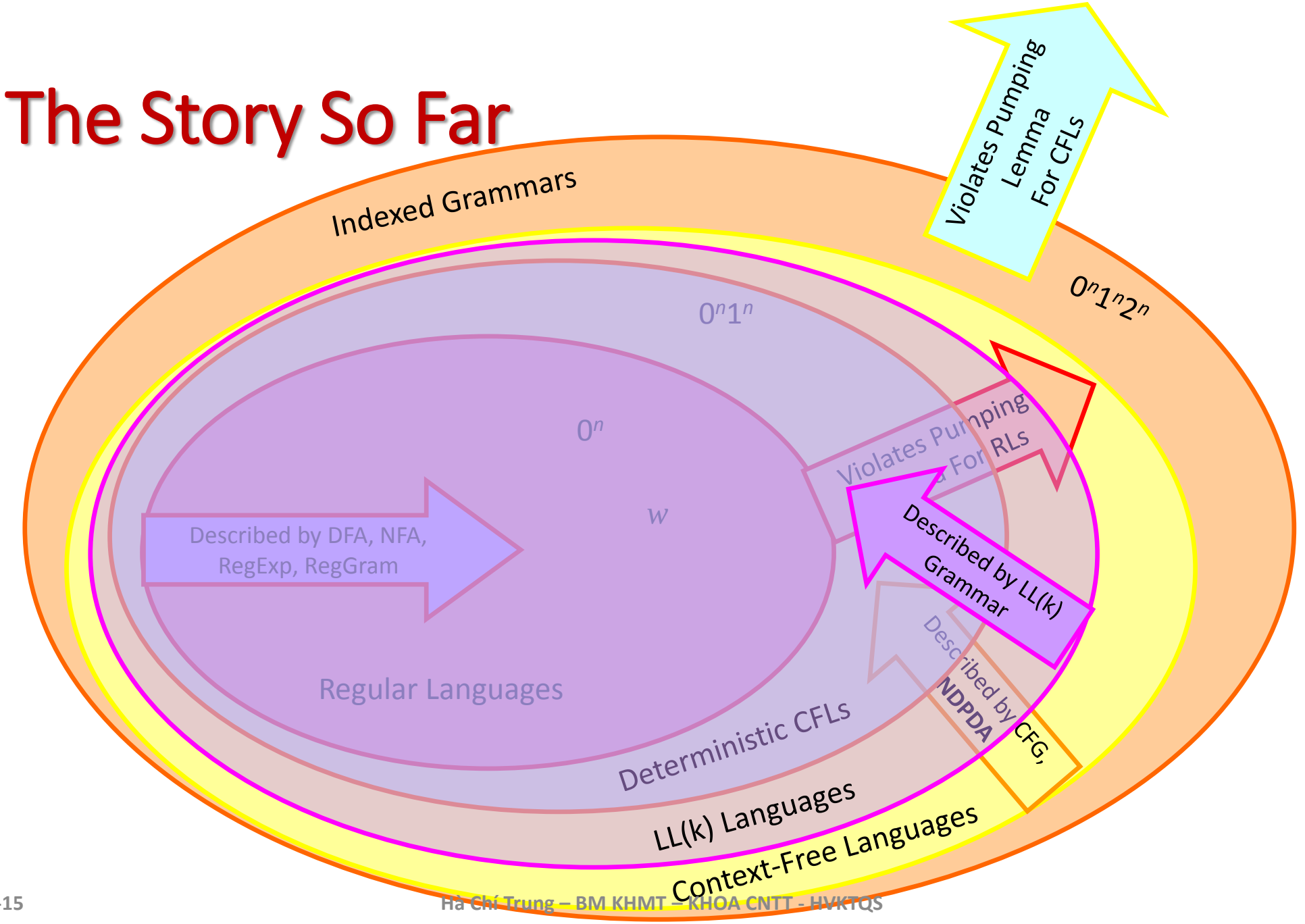
6. Tính chất của văn phạm và ngôn ngữ

- Cho văn phạm G=(Σ, Δ, S, P) tùy ý:
  - **Văn phạm tương đương**: luôn tồn tại một văn phạm G'=(Σ', Δ', S', P') tương đương với G, tức là L(G')=L(G).
  - Nếu tồn tại trong P quy tắc chứa ký hiệu xuất phát S ở vế phải thì tồn tại G'⇔G mà trong P' không chứa S ở vế phải.
  - Có thể xây dựng văn phạm G' tương đương với G mà các quy tắc của nó không chứa ký hiệu kết thúc ở vế trái.
- Với hai **dẫn xuất** $D = \omega_0\omega_1...\omega_k$ và $D' = \omega'_0\omega'_1...\omega'_m$ trong G:
  - hai dẫn xuất là **đồng lực** nếu $\omega_0 = \omega'_0$ và $\omega_k = \omega_m$.
  - dẫn xuất D là **không lặp** nếu với mọi $i \neq j$ thì $\omega_i \neq \omega_j$.
  - Với mọi dẫn xuất trong văn phạm G, luôn luôn tồn tại một dẫn xuất không lặp và đồng lực với nó.

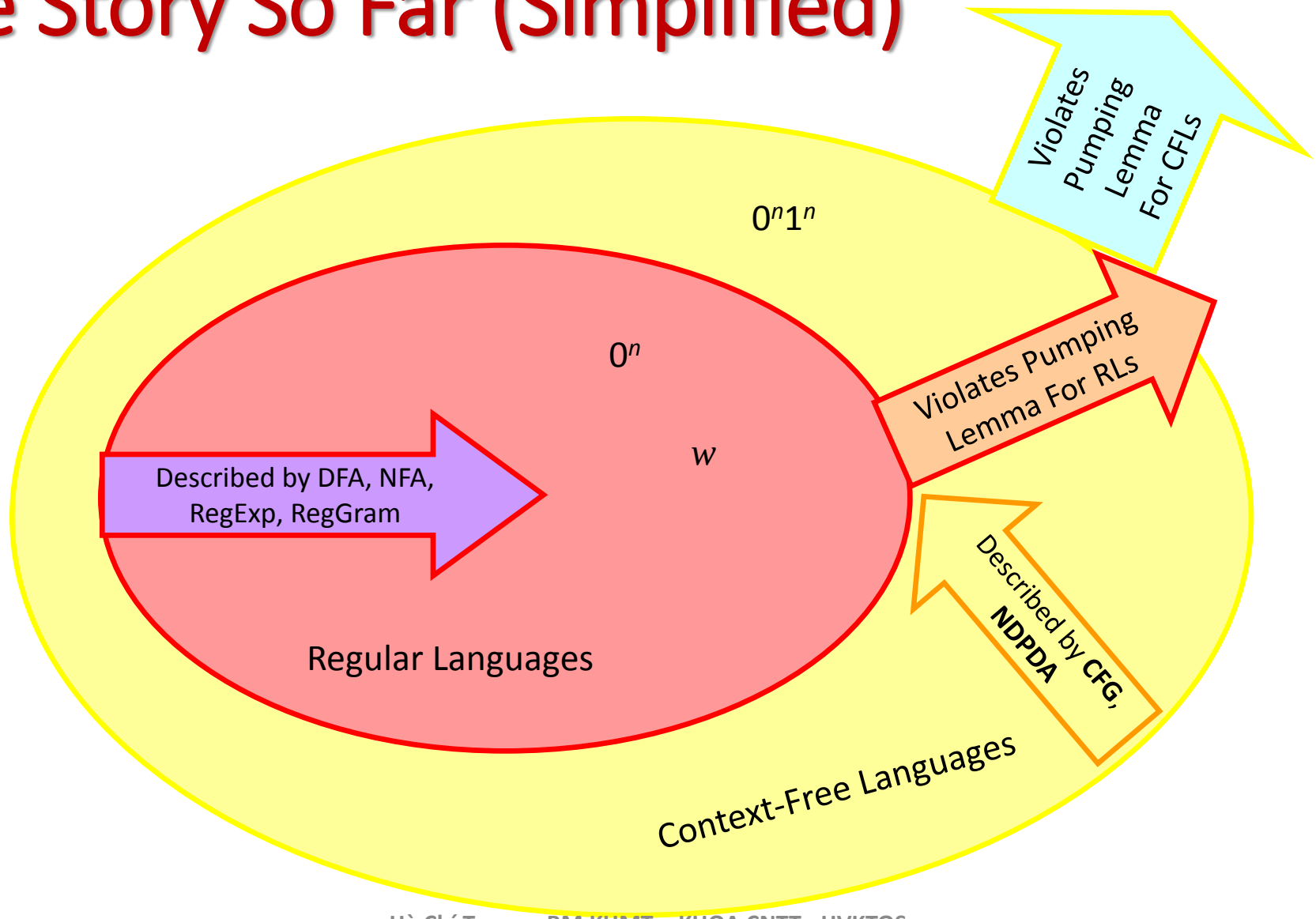
6. Tính chất của văn phạm và ngôn ngữ

- **Tính đóng: Lớp ngôn ngữ sinh bởi văn phạm là đóng đối với hầu hết các phép toán trên ngôn ngữ.**
  - **Định lý:** Lớp ngôn ngữ sinh bởi văn phạm là đóng đối với phép nhân ghép ngôn ngữ (.), phép lặp, lặp cắt, phép chia trái và chia phải, và phép hiệu của 2 ngôn ngữ.
  - **Định lý:** phép hợp (∪), phép giao (∩), phép lấy phần bù đóng với ngôn ngữ loại 3.
- **Tính đệ quy:** Chúng ta nói rằng văn phạm G là đệ qui nếu tồn tại thuật toán xác định một từ w cho trước có thuộc L(G) hay không.

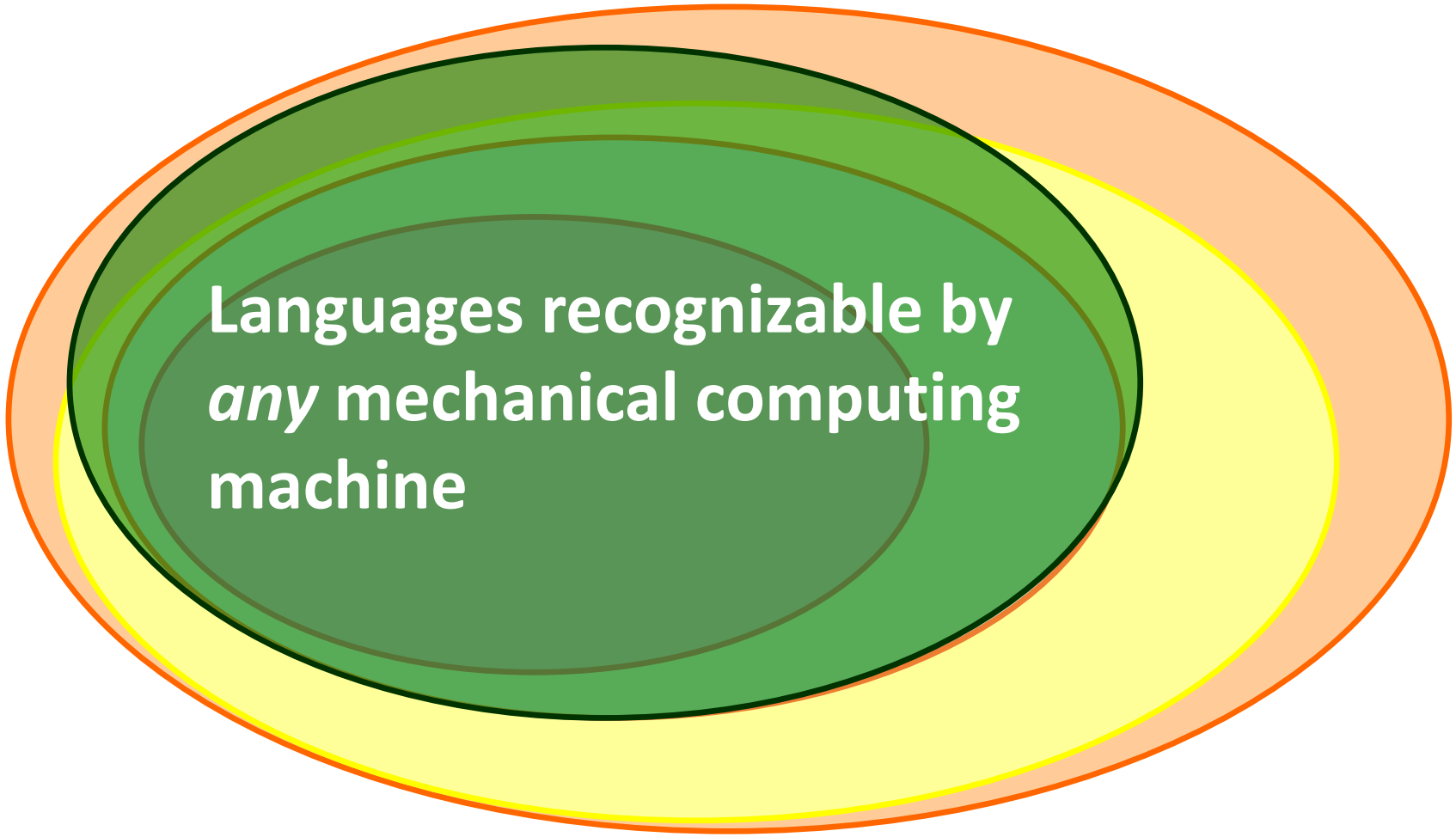
7. The Story So Far



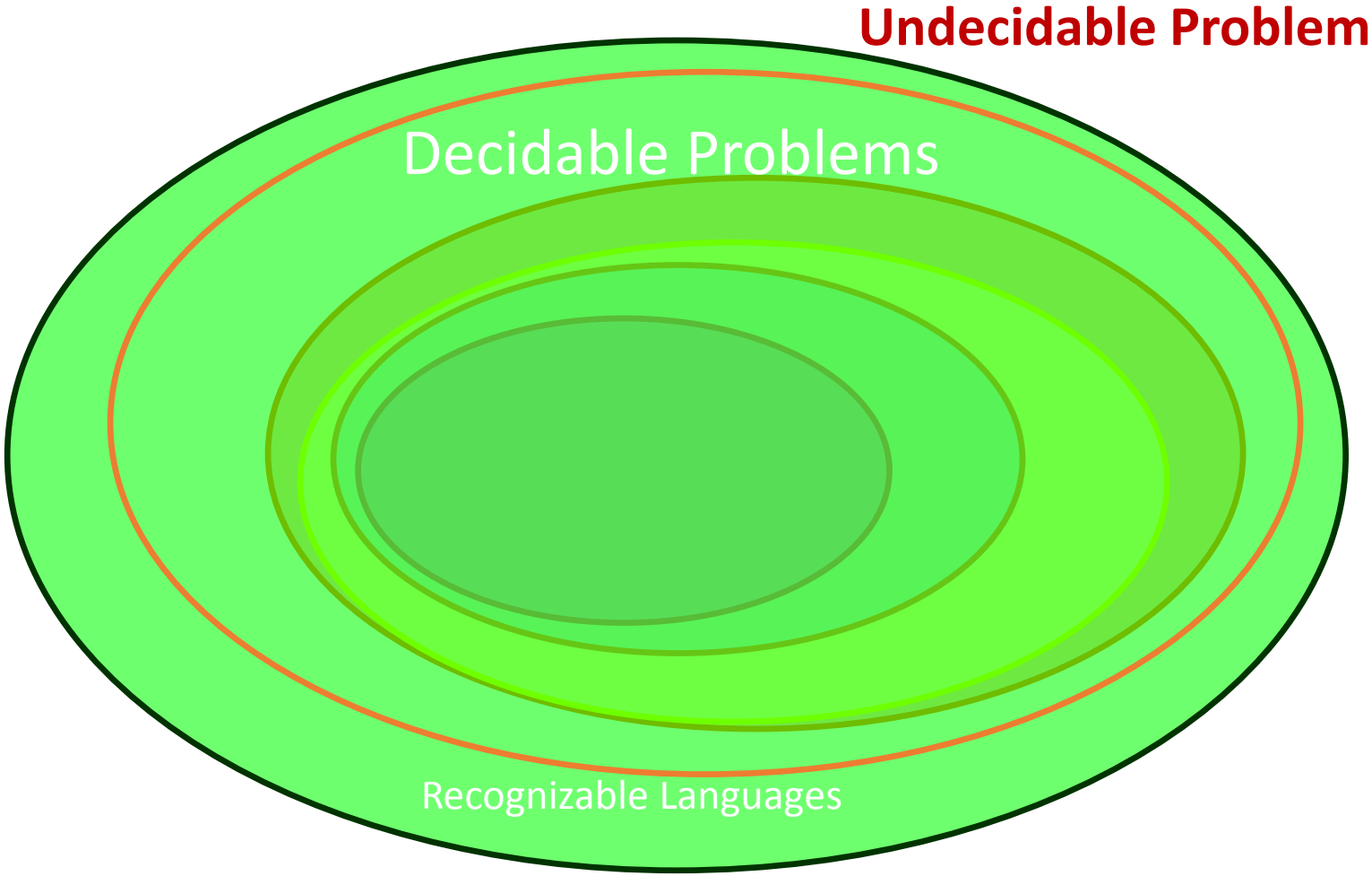
7. The Story So Far (Simplified)



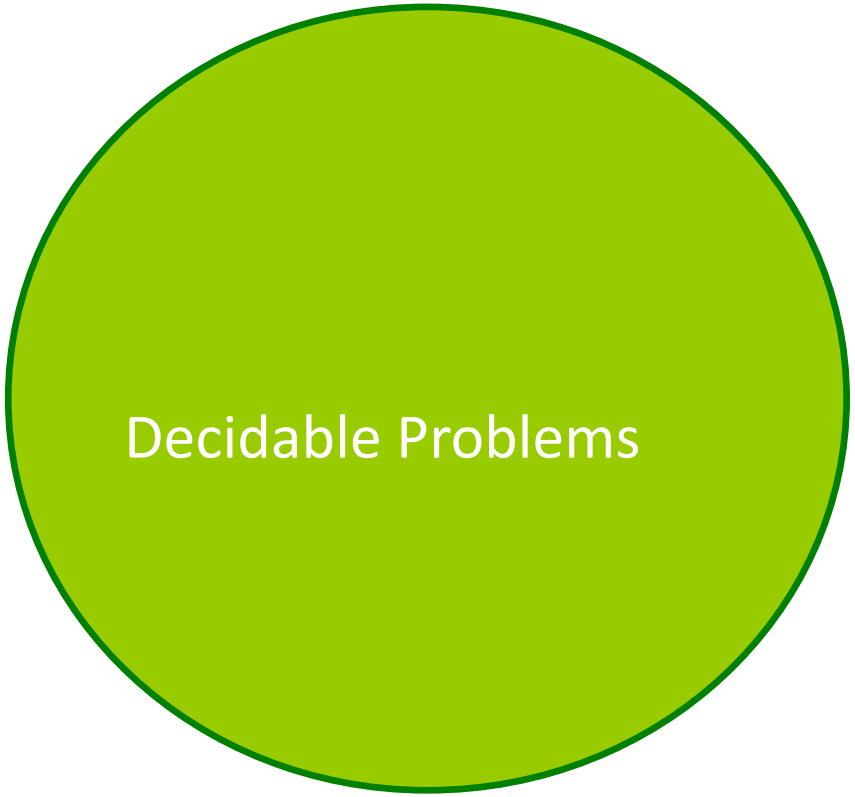
7. The Story So Far (Theory of Computation)



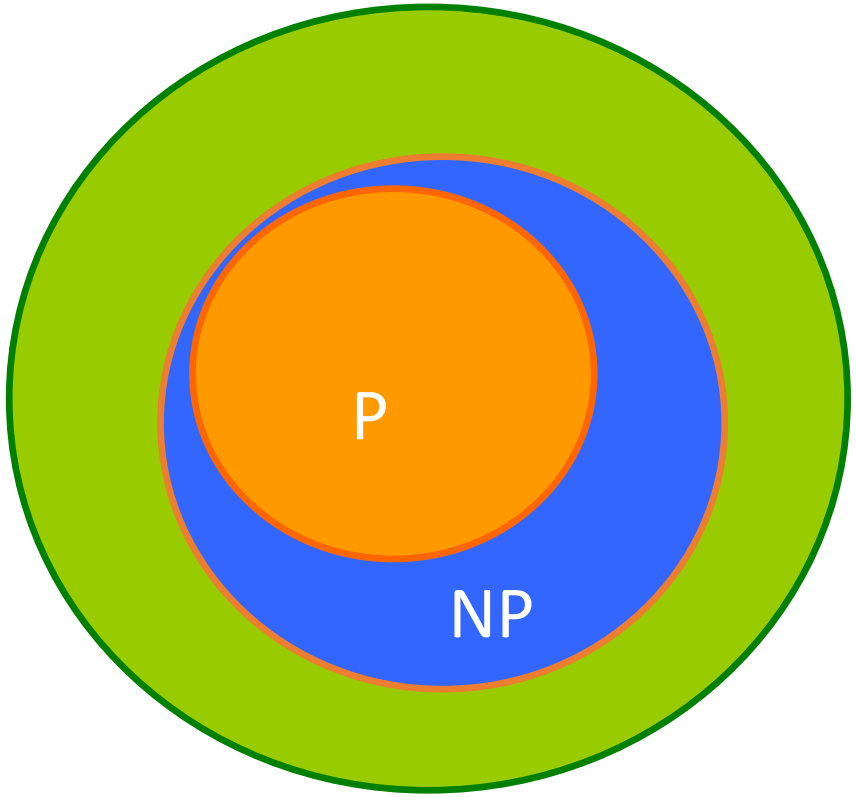
7. The Story So Far (Theory of Computation)



7. The Story So Far (Theory of Computation)



Problems that can be solved by a computer (eventually).



Note: not known if P $\subset$ NP or P = NP

Problems that can be solved by a computer in a ***reasonable*** time.