Chương 9 Máy Turing

- PDA về một mặt nào đó mạnh hơn rất nhiều FSA.
- NNPNC-PDA vẫn còn giới hạn. Bên ngoài nó là gì?
- FSA và PDA khác nhau ở bản chất của bộ lưu trữ tạm thời.
- Nếu PDA dùng hai, ba stack, một hàng (queue), hay một thiết bị lưu trữ khác nào đó thì sức mạnh sẽ thế nào?
- Mỗi thiết bị lưu trữ định nghĩa một loại ôtômát mới và thông qua nó một họ ngôn ngữ mới?
- Ôtômát có thể được mở rộng đến chừng nào?
- Máy Turing ra đời và khái niệm về sự tính toán có tính máy móc hay giải thuật (mechanical or algorithmic computation).
- Máy Turing là khá thô sơ, nhưng đủ sức để bao trùm các quá trình rất phức tạp và **luận đề Turing** cho rằng bất kỳ quá trình tính toán nào thực hiện được bằng các máy tính ngày nay, đều có thể thực hiện được bằng máy Turing.

Chương 9 Máy Turing

9.1 Máy Turing chuẩn

9.2 Kết hợp các máy Turing cho các công việc phức tạp

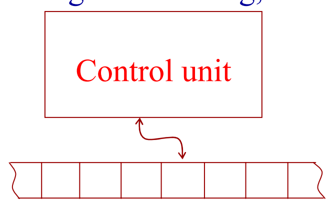
9.3 Luận đề Turing

Máy Turing chuẩn

- Định nghĩa 9.1
  - Một máy Turing M được định nghĩa bằng bộ bảy

$$M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F),$$

  - Q là tập hữu hạn các trạng thái nội,
  - Σ là tập hữu hạn các kí hiệu được gọi là bảng chữ cái ngõ nhập,
  - Γ là tập hữu hạn các kí hiệu được gọi là bảng chữ cái băng,
  - δ là hàm chuyển trạng thái,
  - $\square \in \Gamma$ là một kí hiệu đặc biệt, gọi là khoảng trắng (blank),
  - $q_0 \in Q$ là trạng thái khởi đầu,
  - $F \subseteq Q$ là tập các trạng thái kết thúc.



Input, Storage, Output

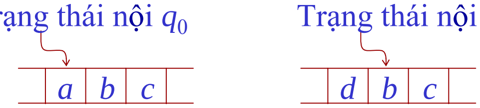
Máy Turing chuẩn (tt)

- Trong định nghĩa chúng ta giả thiết rằng $\Sigma \subseteq \Gamma - \{\square\}$.
- Hàm δ được định nghĩa như sau

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

- Nó được diễn dịch như sau: Các đối số của δ là trạng thái hiện hành của ôtômát và kí hiệu băng đang được đọc. Kết quả là một trạng thái mới của ôtômát, một kí hiệu băng mới thay thế cho kí hiệu đang được đọc trên băng và một sự di chuyển đầu đọc sang phải hoặc sang trái.
- Ví dụ $\delta(q_0, a) = \{q_1, d, R\}$

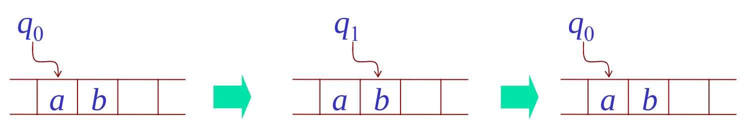
Trạng thái nội q_0 Trạng thái nội q_1



Ví dụ

- Xét một máy Turing được định nghĩa như sau
- $Q = \{q_0, q_1\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \square\}$, $F = \varnothing$, còn δ được định nghĩa

  $\delta(q_0, a) = (q_1, a, R)$ $\delta(q_1, a) = (q_0, a, L)$
  $\delta(q_0, b) = (q_1, b, R)$ $\delta(q_1, b) = (q_0, b, L)$
  $\delta(q_0, \square) = (q_1, \square, R)$ $\delta(q_1, \square) = (q_0, \square, L)$
- Xét hoạt động của M trong trường hợp sau



- Trường hợp này máy không dừng lại và rơi vào một **vòng lặp vô tận (infinite loop)**

Các đặc điểm của máy Turing chuẩn

- Có nhiều mô hình khác nhau của máy Turing.
- Sau đây là một số đặc điểm của máy Turing ***chuẩn***.
- Máy Turing có một băng không giới hạn cả hai đầu, cho phép di chuyển một số bước tùy ý về bên trái và phải.
- Máy Turing là đơn định trong ngữ cảnh là δ định nghĩa tối đa một chuyển trạng thái cho một cấu hình.
- Không có một băng nhập (input file) riêng biệt. Chúng ta giả thiết là vào thời điểm khởi đầu băng chứa một nội dung cụ thể. Một vài trong số này có thể được xem là chuỗi nhập (input). Tương tự không có một băng xuất (output file) riêng biệt. Bất kỳ khi nào máy dừng, một vài hay tất cả nội dung của băng có thể được xem là kết quả xuất (output).

Hình trạng tức thời

- Định nghĩa 9.2
  - Cho $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ là một máy Turing, thì một chuỗi
    $$a_1a_2 \ldots a_{k-1}q_1a_ka_{k+1} \ldots a_n$$
    bất kỳ với $a_i \in \Sigma$ và $q_1 \in Q$, là một hình trạng tức thời của M (gọi tắt là hình trạng).
  - Một di chuyển
    $$a_1a_2 \ldots a_{k-1}q_1a_ka_{k+1} \ldots a_n \vdash a_1a_2 \ldots a_{k-1}bq_2a_{k+1} \ldots a_n$$
    là có thể nếu và chỉ nếu
    $$\delta(q_1, a_k) = (q_2, b, R).$$
  - Một di chuyển
    $$a_1a_2 \ldots a_{k-1}q_1a_ka_{k+1} \ldots a_n \vdash a_1a_2 \ldots q_2a_{k-1}ba_{k+1} \ldots a_n$$
    là có thể nếu và chỉ nếu
    $$\delta(q_1, a_k) = (q_2, b, L).$$

Hình trạng tức thời (tt)

- M được gọi là dừng sau khi bắt đầu từ một cấu hình khởi đầu nào đó $x_1q_ix_2$ nếu
  $$x_1q_ix_2 \vdash^* y_1q_jay_2$$
  với bất kỳ q_j và a, mà đối với nó $\delta(q_j, a)$ không được định nghĩa.
- Dãy cấu hình dẫn tới một trạng thái dừng sẽ được gọi là một sự tính toán (computation).
- Ví dụ trong slide 290 trình bày khả năng rằng một máy Turing có thể không bao giờ dừng, thi hành trong một vòng lặp vô tận và từ đó nó không thể thoát.
- Trường hợp này đóng một vai trò cơ bản trong thảo luận về máy Turing, và được kí hiệu là
  $$x_1qx_2 \vdash^* \infty$$
  để chỉ ra rằng, bắt đầu từ cấu hình khởi đầu x_1qx_2, máy không bao giờ dừng.

Máy Turing như một bộ chấp nhận ngôn ngữ

- Định nghĩa 9.3
  - Cho $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ là một máy Turing, thì ngôn ngữ được chấp nhận bởi M là

    $L(M) = \{w \in \Sigma^+ : q_0 w \vdash^* x_1 q_f x_2$ và dừng, đối với một q_f nào đó $\in F,\ x_1, x_2 \in \Gamma^*\}$.
  - Định nghĩa này chỉ ra rằng chuỗi nhập w được viết trên băng với các khoảng trắng chặn ở hai đầu. Đây cũng là lý do các khoảng trắng bị loại ra khỏi bảng chữ cái ngõ nhập Σ.
  - Điều này đảm bảo chuỗi nhập được giới hạn trong một vùng rõ ràng của băng được bao bọc hai đầu bởi các kí hiệu trắng.
  - Không có qui ước này, máy không thể giới hạn vùng trong đó nó tìm kiếm chuỗi nhập.
  - Định nghĩa trên không nói rõ khi nào thì $w \notin L(M)$. Điều này đúng khi một trong hai trường hợp sau xảy ra

Ví dụ

(1) Máy dừng lại ở một trạng thái không kết thúc.

(2) Máy đi vào một vòng lặp vô tận và không bao giờ dừng.

- Ví dụ
  - Cho $\Sigma = \{a, b\}$, thiết kế máy Turing chấp nhận $L = \{a^n b^n : n \geq 1\}$.
  - Ý tưởng thiết kế là đọc một a thay bằng một x, đi kiếm một b thay bằng một y. Cứ như vậy cho đến khi không còn đồng thời a và b để thay thì dừng và chấp nhận chuỗi, các trường hợp khác thì không chấp nhận. Máy Turing kết quả như sau.

$Q = \{q_0, q_1, q_2, q_3, q_f\}$, $F = \{q_f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, x, y, \square\}$

$\delta(q_0, a) = \{q_1, x, R\}$ $\quad \delta(q_2, y) = \{q_2, y, L\}$ $\quad \delta(q_0, y) = \{q_3, y, R\}$

$\delta(q_1, a) = \{q_1, a, R\}$ $\quad \delta(q_2, a) = \{q_2, a, L\}$ $\quad \delta(q_3, y) = \{q_3, y, R\}$

$\delta(q_1, y) = \{q_1, y, R\}$ $\quad \delta(q_2, x) = \{q_0, x, R\}$ $\quad \delta(q_3, \square) = \{q_f, \square, R\}$

$\delta(q_1, b) = \{q_2, y, L\}$

Ví dụ

$q_0aaabbb \vdash xq_1aabbb \vdash xaq_1abbb \vdash xaaq_1bbb \vdash xaq_2aybb$
$\vdash xq_2aaybb \vdash q_2xaaybb \vdash xq_0aaybb \vdash xxq_1aybb$
$\vdash xxaq_1ybb \vdash xxayq_1bb \vdash xxaq_2yyb \vdash xxq_2ayyb$
$\vdash xq_2xayyb \vdash xxq_0ayyb \vdash xxxq_1yyb \vdash xxxyq_1yb$
$\vdash xxxyyq_1b \vdash xxxyyq_1b \vdash xxxyq_2yy \vdash xxxq_2yyy$
$\vdash xxq_2xyyy \vdash xxxq_0yyy \vdash xxxyq_3yy \vdash xxxyyq_3y$
$\vdash xxxyyyq_3\square \vdash xxxyyy\square q_f\square$ (chấp nhận)

$q_0aaabb \vdash xq_1aabb \vdash xaq_1abb \vdash xaaq_1bb \vdash xaq_2ayb$
$\vdash xq_2aaybq_2 \vdash xaayb \vdash xq_0aayb \vdash xxq_1ayb$
$\vdash xxaq_1yb \vdash xxayq_1b \vdash xxaq_2yy \vdash xxq_2ayy$
$\vdash xq_2xayy \vdash xxq_0ayy \vdash xxxq_1yy \vdash xxxyq_1y$
$\vdash xxxyyq_1\square$ (dừng)

Máy Turing như là transducer

- Máy Turing không chỉ được quan tâm như là một bộ chấp nhận ngôn ngữ mà trong tổng quát còn cung cấp một mô hình trừu tượng đơn giản của một máy tính số.
- Vì mục đích chính của một máy tính là biến đổi ***input*** thành ***output***, nó hoạt động như một transducer.
- Hãy thử mô hình hóa máy tính bằng cách dùng máy Turing.
- ***Input*** của một sự tính toán là tất cả các kí hiệu không trắng trên băng tại thời điểm khởi đầu. Tại kết thúc của sự tính toán, ***output*** sẽ là bất kì cái gì có trên băng.
- Vậy có thể xem một máy Turing M như là một sự hiện thực của một hàm f được định nghĩa bởi

$$\hat{w} = f(w)$$

trong đó $q_0w \vdash^*_M q_f\hat{w}$ với q_f là một trạng thái kết thúc nào đó.

Máy Turing như là transducer (tt)

- Định nghĩa 9.4
  - Một hàm f với miền xác định D được gọi là khả tính toán-Turing hay đơn giản là khả tính toán nếu tồn tại một máy Turing nào đó $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ sao cho

$$q_0 w \vdash^*_M q_f f(w),\ q_f \in F,\ \forall\ w \in D.$$

- Ví dụ
  - Cho x, y nguyên dương, thiết kế máy Turing tính $x + y$.
  - Chúng ta đầu tiên chọn qui ước để biểu diễn số nguyên dương.
  - Ta đã biết cách biểu diễn số nguyên dương bằng chuỗi nhị phân và cách cộng hai số nhị phân, tuy nhiên để ứng dụng điều đó vào trong trường hợp này thì hơi phức tạp một chút.
  - Vậy để đơn giản hơn ta biểu diễn số nguyên dương x bằng chuỗi $w(x)$ các số 1 có chiều dài bằng x.

Ví dụ

- Chúng ta cũng phải quyết định các số x và y vào lúc ban đầu được đặt như thế nào trên băng và tổng của chúng xuất hiện như thế nào lúc kết thúc sự tính toán.
- Chúng ta giả thiết rằng $w(x)$ và $w(y)$ được phân cách bằng một kí hiệu 0, với đầu đọc ở trên kí tự trái cùng của $w(x)$. Sau khi tính toán, $w(x + y)$ sẽ ở trên băng và được theo sau bởi một kí tự 0, và đầu đọc sẽ được đặt trên kí tự trái cùng của kết quả.
- Chúng ta vì vậy muốn thiết kế một máy Turing để thực hiện sự tính toán (trong đó q_f là một trạng thái kết thúc)

$$q_0 w(x)0w(y) \vdash^* q_f\, w(x + y)0,$$

$$Q = \{q_0, q_1, q_2, q_3, q_f\},\ F = \{q_f\}$$

$\delta(q_0, 1) = (q_0, 1, R)$ $\delta(q_0, 0) = (q_1, 1, R)$ $\delta(q_1, 1) = (q_1, 1, R)$

$\delta(q_1, \square) = (q_2, \square, L)$ $\delta(q_2, 1) = (q_3, 0, L)$ $\delta(q_3, 1) = (q_3, 1, L)$

$\delta(q_3, \square) = (q_f, \square, R)$

Kết hợp các máy Turing cho các công việc phức tạp

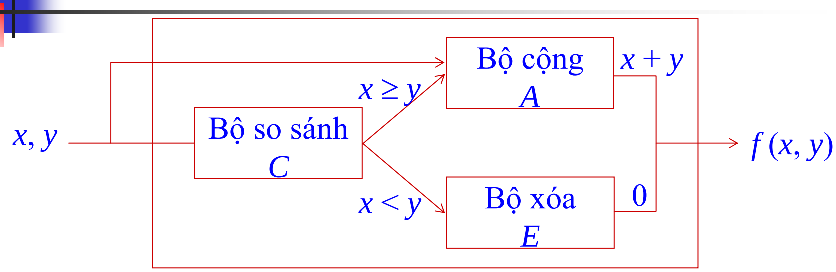
- Chúng ta đã thấy máy Turing có thể thực hiện được các phép toán cơ bản và quan trọng những cái mà có trong tất cả các máy tính.
- Vì trong các máy tính số, các phép toán cơ bản như vậy là các thành phần cơ bản cho các lệnh phức tạp hơn, vì vậy chúng ta ở đây cũng sẽ trình bày máy Turing có khả năng kết hợp các phép toán này lại với nhau.

- Ví dụ
  - Thiết kế một máy Turing tính toán hàm sau

$$f(x, y) = x + y \text{ nếu } x \geq y$$
$$= 0 \quad \text{ nếu } x < y$$

  - Ta xây dựng mô hình tính toán cho nó như sau

Kết hợp các máy Turing cho các công việc phức tạp (tt)



- Chúng ta sẽ xây dựng bộ so sánh C mà sau khi thực hiện xong có kết quả như sau:

$$q_{C,0}w(x)0w(y) \vdash^* q_{A,0}w(x)0w(y), \text{ nếu } x \geq y$$
$$q_{C,0}w(x)0w(y) \vdash^* q_{E,0}w(x)0w(y), \text{ nếu } x < y$$

trong đó $q_{C,0}$, $q_{A,0}$ và $q_{E,0}$ lần lượt là trạng thái khởi đầu của bộ so sánh, bộ cộng và bộ xóa.

Bài tập

- Nếu chúng ta xây dựng được các bộ so sánh, bộ cộng và bộ xóa thì với mô hình kết hợp như trên chúng ta có thể xây dựng được hàm tính toán được yêu cầu.
- Xây dựng máy Turing thực hiện các phép toán sau
  - Hàm $f(x, y)$ trong slide trên
  - Phép AND, OR, XOR
  - Phép cộng hai số nhị phân

Luận đề Turing

- Máy Turing có thể được xây dựng từ các phần đơn giản hơn, tuy nhiên khá cồng kềnh cho dù phải thực hiện các phép toán đơn giản. Điều này là vì "tập lệnh" của một máy Turing là quá đơn giản và hạn chế.
- Vậy máy Turing có sức mạnh đến đâu và như thế nào trong sự so sánh với sức mạnh của máy tính ngày nay?
- Mặc dầu với cơ chế đơn giản nhưng máy Turing có thể giải quyết được các bài toán phức tạp mà máy tính ngày nay giải quyết được.
- Để chứng minh điều này người ta đã chọn ra một máy tính điển hình, sau đó xây dựng một máy Turing thực hiện được tất cả các lệnh trong tập lệnh của máy tính (tập lệnh của CPU).
- Tuy làm được điều này nhưng đó cũng chưa phải là một chứng minh chặt chẽ để chứng tỏ máy Turing có sức mạnh ngang bằng với các máy tính ngày nay.

Luận đề Turing (tt)

- Tuy nhiên cũng không ai đưa ra được phản chứng chứng minh rằng máy Turing không mạnh bằng với máy tính ngày nay.
- Cuối cùng, với khá nhiều bằng chứng mạnh mẽ tuy chưa đủ là một chứng minh chặt chẽ, chúng ta chấp nhận luận đề Turing sau như là một định nghĩa của một "***sự tính toán cơ học***"

- Luận đề Turing
  - Bất kỳ cái gì có thể được thực hiện trên bất kỳ máy tính số đang tồn tại nào đều có thể được thực hiện bởi một máy Turing.
  - Không ai có thể đưa ra một bài toán, có thể giải quyết được bằng những gì mà một cách trực quan chúng ta xem là một giải thuật, mà đối với nó không tồn tại máy Turing nào giải quyết được.
  - Các mô hình thay thế khác có thể được đưa ra cho sự tính toán cơ học nhưng không có cái nào trong số chúng là mạnh hơn mô hình máy Turing.

Giải thuật

- Luận đề trên đóng một vai trò quan trọng trong khoa học máy tính cũng giống như vai trò của các định luật cơ bản trong vật lý và hóa học.
- Bằng việc chấp nhận luận đề Turing, chúng ta sẵn sàng để định nghĩa chính xác khái niệm giải thuật, cái mà là khá cơ bản trong khoa học máy tính.

- Định nghĩa 9.5
  - Một giải thuật cho một hàm f: $D \to R$ là một máy Turing M sao cho cho một chuỗi nhập $d \in D$ trên băng nhập, cuối cùng M dừng với kết quả $f(d) \in R$ trên băng. Một cách cụ thể là:

$$q_0 d \vdash_M^* q_f f(d),\ q_f \in F,\ \forall\ d \in D.$$

Chương 10 Phụ lục

Máy Turing không đơn định

- Định nghĩa 10.6
  - Là máy Turing mà trong đó hàm δ được định nghĩa như sau:

$$\delta: Q \times \Sigma \rightarrow 2^{Q \times \Sigma \times \{L, R\}}$$

- Định lý 10.5
  - Lớp máy Turing không đơn định tương đương với lớp máy Turing chuẩn.
- Định lý 10.6
  - Tập tất cả các máy Turing là vô hạn đếm được.

Ôtômát ràng buộc tuyến tính

- Định nghĩa 10.7
  - Một ôtômát ràng buộc tuyến tính (Linear Bounded Automata - LBA) là một máy Turing không đơn định $M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$, như trong Định nghĩa 10.6, ngoại trừ bị giới hạn rằng Σ phải chứa hai kí tự đặc biệt [và], sao cho $\delta(q_i, [)$ có thể chứa chỉ một phần tử dạng $(q_j, [, R)$ và $\delta(q_i,])$ có thể chứa chỉ một phần tử dạng $(q_j,], L)$.
  - Bằng lời, khi đầu đọc chạm đến dấu móc vuông ở một trong hai đầu nó phải giữ lại và đồng thời không thể vượt ra vùng nằm giữa hai dấu móc vuông.
  - Trong trường hợp này chúng ta nói đầu đọc bị giới hạn giữa hai dấu móc vuông hai đầu.

Ôtômát ràng buộc tuyến tính (tt)

- Định nghĩa 10.7
  - Một chuỗi được chấp nhận bởi một ôtômát ràng buộc tuyến tính nếu có một dãy chuyển hình trạng có thể

$$q_0[w] \vdash^* [x_1 q_f x_2]$$

với một q_f nào đó $\in F$, $x_1, x_2 \in \Sigma^*$. Ngôn ngữ được chấp nhận bởi ***lba*** là tập tất cả các chuỗi được chấp nhận bởi ***lba***.
- Ví dụ
  - Ngôn ngữ $L = \{a^n b^n c^n : n \geq 0\}$ là một ngôn ngữ ràng buộc tuyến tính vì chúng ta có thể xây dựng được một ***lba*** chấp nhận đúng nó.

Ngôn ngữ khả liệt kê đệ qui, đệ qui

- Định nghĩa 10.8
  - Một ngôn ngữ *L* được gọi là khả liệt kê đệ qui nếu tồn tại một máy Turing *M* chấp nhận nó.
  - Từ định nghĩa này cũng dễ dàng suy ra được mọi ngôn ngữ mà đối với nó tồn tại một thủ tục liệt kê (các phần tử của nó) thì khả liệt kê đệ qui.
- Định nghĩa 10.9
  - Một ngôn ngữ *L* trên Σ được gọi là đệ qui nếu tồn tại một máy Turing *M* chấp nhận nó và dừng đối với $w \in \Sigma^+$. Hay nói cách khác một ngôn ngữ là đệ qui nếu và chỉ nếu tồn tại một giải thuật thành viên cho nó.

Văn phạm

- Định nghĩa 10
  - Một văn phạm mà mọi luật sinh không cần thõa bất kỳ ràng buộc nào tức là có dạng

    $$\alpha \to \beta$$

    trong đó $\alpha \in (V \cup T)^*V(V \cup T)^*$, $\beta \in (V \cup T)^*$ thì được gọi là ***văn phạm loại 0*** hay là ***văn phạm không hạn chế***.
  - Một văn phạm mà mọi luật sinh có dạng chiều dài vế trái nhỏ hơn hoặc bằng chiều dài vế phải tức là có dạng

    $$\alpha \to \beta$$

    trong đó $\alpha \in (V \cup T)^*V(V \cup T)^*$, $\beta \in (V \cup T)^*$ và $|\alpha| \le |\beta|$ thì được gọi là ***văn phạm loại 1*** hay ***văn phạm cảm ngữ cảnh***.
  - Văn phạm phi ngữ cảnh còn được gọi là ***văn phạm loại 2***.
  - Văn phạm chính qui còn được gọi là ***văn phạm loại 3***.

Tổng kết các lớp đối tượng

Các lớp ngôn ngữ		Kí hiệu
Chính qui	**Reg**ular	L_{REG}
Tuyến tính	**Lin**ear	L_{LIN}
Phi ngữ cảnh đơn định	**D**eterministic **C**ontext-**F**ree	L_{DCF}
Phi ngữ cảnh	**C**ontext-**F**ree	L_{CF}
Cảm ngữ cảnh	**C**ontext-**S**ensitive	L_{CS}
Đệ qui	**Rec**usive	L_{REC}
Khả liệt kê đệ qui	**R**ecusively **E**numerable	L_{RE}

Tổng kết các lớp đối tượng (tt)

Các lớp văn phạm		Kí hiệu
Chính qui ≡ Tuyến tính-phải và tuyến tính-trái ≡ Loại 3	**Reg**ular ≡ **R**ight-**Lin**ear và **L**eft-**Lin**ear	$G_{REG} \equiv G_{R\text{-}LIN}$ và $G_{L\text{-}LIN}$
Tuyến tính	**Lin**ear	G_{LIN}
Phi ngữ cảnh đơn định: điển hình là LL(k) và LR(k)	LL(k) và LR(k)	G_{LL} và G_{LR}
Phi ngữ cảnh ≡ Loại 2	**C**ontext-**F**ree	G_{CF}
Cảm ngữ cảnh ≡ Loại 1	**C**ontext-**S**ensitive	G_{CS}
Không hạn chế ≡ Loại 0	**U**n**R**estricted	G_{UR}

Tổng kết các lớp đối tượng (tt)

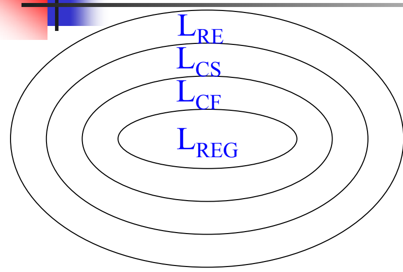
Các lớp ôtômát		Kí hiệu
Hữu hạn	**F**inite **S**tate	FSA (nfa, dfa)
Đẩy xuống đơn định	**D**eterministic **P**ush **D**own	DPDA
Đẩy xuống không đơn định	**N**ondeterministic **P**ush **D**own	NPDA
Ràng buộc tuyến tính	**L**inear **B**ounded	LBA
Máy Turing	**T**uring **M**achine	TM

Mối quan hệ giữa các lớp đối tượng

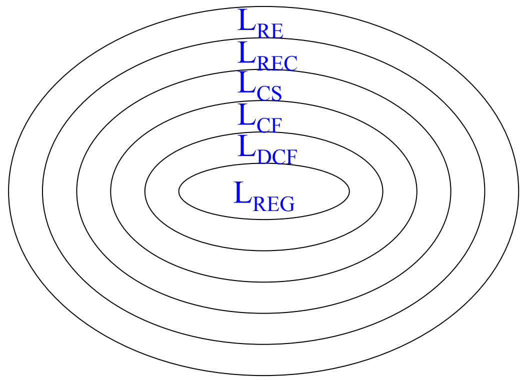
Ngôn ngữ	Văn phạm	Ôtômát
L_{REG}	$G_{REC} \equiv G_{L\text{-}LIN}$ và $G_{R\text{-}LIN}$	FSA $\equiv$ DFA $=$ NFA
L_{LIN}	G_{LIN}	$\subset$ NPDA
L_{DCF}	$\supset$ LL(k) và LR(k)	DPDA
L_{CF}	G_{CF}	NPDA
L_{CS}	G_{CS}	LBA
L_{REC}	$\subset G_{UR}$	$\subset$ TM
L_{RE}	G_{UR}	TM

- Dấu $\equiv$ có nghĩa là theo định nghĩa, còn dấu $=$ có nghĩa là tương đương, dấu $\supset$ có nghĩa là tập cha (không bằng), dấu $\subset$ có nghĩa là tập con (không bằng).

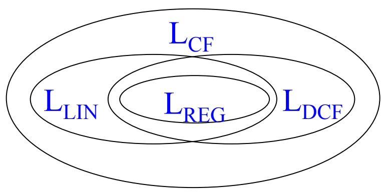
Phân cấp ngôn ngữ theo Chomsky



Sơ đồ phân cấp đơn giản



Sơ đồ phân cấp chi tiết



Sơ đồ phân cấp trong lớp PNC