Mật mã và độ phức tạp thuật toán (Complexity and Cryptography) Chủ đề 1: Thuật toán và máy Turing

PGS.TS. Nguyễn Đình Hân
(Mobile: 0915.046.320; Email: han.nguyendinh@hust.edu.vn)

Viện Toán ứng dụng và Tin học
Trường Đại học Bách khoa Hà Nội

Thuật toán và máy Turing

Hiện tượng thường gặp

- Trong thực tiễn có những bài toán dễ và cả những bài toán khó.
- Một bài toán có thể giải được bằng nhiều thuật toán tốt xấu khác nhau.
- Cùng một thuật toán, có thể có trường hợp nó cho kết quả nhanh, trường hợp cho kết quả chậm.
- Giữa giải được về mặt lý thuyết và giải được một cách thực tế đối với nhiều bài toán cũng có những sự khác biệt đáng kể.

Nguyên nhân từ đâu?

Lý thuyết độ phức tạp tính toán cho ta cách nhìn thống nhất về độ phức tạp của các thuật toán. Tuy nhiên, sẽ cần có thêm nhiều nghiên cứu để lý giải thỏa đáng!

Bài toán và cách tiếp cận

Phát biểu bài toán: gồm 2 phần tách biệt là *dữ kiện* và *yêu cầu*. Yêu cầu có thể là một câu hỏi hoặc yêu cầu tìm kiếm nghiệm.

⋆ Có hai dạng bài toán phổ biến:

(1) Bài toán quyết định (decision problem): đối với mỗi dữ kiện bài toán chỉ cần trả lời là "đúng" hoặc "sai".

(2) Bài toán tìm kiếm (search problem, các bài toán tối ưu - optimization problems, là trường hợp riêng): tìm kiếm nghiệm đối với dữ kiện bất kỳ cho trước.

Ví dụ một bài toán tìm kiếm

Bài toán xếp ba lô (phát biểu bằng ngôn ngữ tự nhiên): *Cho một lô hàng hóa gồm các gói hàng, mỗi gói có khối lượng cùng với giá trị cụ thể, và cho một chiếc ba lô. Hãy chọn từ lô hàng một số gói hàng nào đó và xếp đầy vào ba lô, nhưng không được quá, sao cho thu được một giá trị lớn nhất có thể.*

Bài toán trên được phát biểu lại bằng *ngôn ngữ toán học* như sau:

MAX-KNAPSACK

Dữ kiện: Cho hai dãy số nguyên dương

$$s_1, s_2, \ldots, s_n, S \quad \text{và} \quad v_1, v_2, \ldots, v_n.$$

Yêu cầu: Tìm một tập con $I \subseteq \{1, 2, \ldots, n\}$ sao cho

$$\sum_{i \in I} s_i \leq S \quad \text{và} \quad \sum_{i \in I} v_i \to \max.$$

Bài toán và cách tiếp cận

⋆ **Lưu ý:** Bài toán quyết định tương ứng với bài toán tìm kiếm có thể được xác định đơn giản bằng cách thay yêu cầu "*tìm nghiệm đối với dữ kiện bất kì cho trước*" bằng câu hỏi "*phải chăng tồn tại nghiệm đối với mỗi dữ kiện đã cho*?".

Chẳng hạn, bài toán quyết định tương ứng với bài toán trong ví dụ trên là:

MAX-KNAPSACK

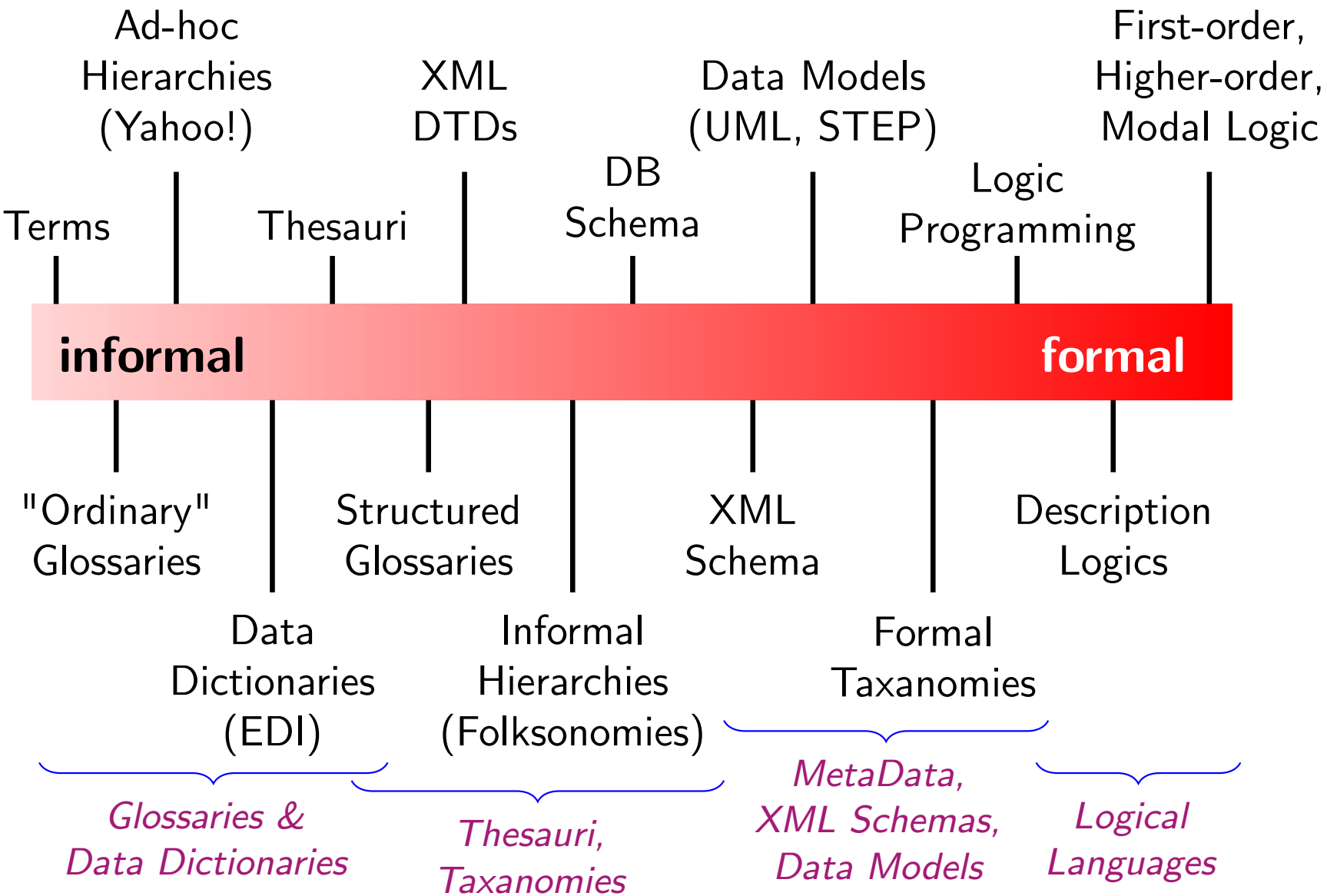
Dữ kiện: Cho hai dãy số nguyên dương

$$s_1, s_2, \ldots, s_n, S \quad \text{và} \quad v_1, v_2, \ldots, v_n, B.$$

Yêu cầu: Phải chăng có một tập con $I \subseteq \{1, 2, \ldots, n\}$ sao cho

$$\sum_{i \in I} s_i \leq S \quad \text{và} \quad \sum_{i \in I} v_i \geq B\,?$$

Phương tiện diễn tả bài toán



Ngôn ngữ hình thức

- Cho Σ là bảng hữu hạn các chữ cái.
- Một *từ* ω trên bảng chữ Σ là một dãy hữu hạn các phần tử của Σ

$$\omega = (a_1, a_2, \ldots, a_n), a_i \in \Sigma.$$

- Tập tất cả các từ trên bảng chữ Σ được ký hiệu là Σ^* và được trang bị phép nhân (tích) ghép có tính chất kết hợp

$$(a_1, a_2, \ldots, a_n)(b_1, b_2, \ldots, b_m) = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m).$$

  Để tiện ta sẽ viết $\omega = a_1 a_2 \cdots a_n$ thay cho $\omega = (a_1, a_2, \ldots, a_n)$.
- Một phần tử $a \in \Sigma$ được gọi là một *chữ cái/kí tự*. *Từ rỗng* được ký hiệu là ε đóng vai trò là phần tử đơn vị trong phép nhân ghép. Do đó, tập Σ^* có cấu trúc vị nhóm và Σ^* được gọi là vị nhóm tự do trên Σ.

Ngôn ngữ hình thức

- Tập tất cả các từ khác rỗng trên Σ được ký hiệu là Σ^+. Ta có

$$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}.$$

- *Độ dài* $|\omega|$ của từ $\omega = a_1 a_2 \cdots a_n$ với $a_i \in \Sigma$ là n. Quy ước $|\varepsilon| = 0$.

- Mỗi tập con của Σ^* được gọi là *một ngôn ngữ* trên Σ.

- Một *X-phân tích* của một từ $\omega \in \Sigma^*$ theo X, với $X \subseteq \Sigma^*$, là một dãy $\omega = u_1 u_2 \cdots u_n$ với $u_1, u_2, \ldots, u_n \in X$, $n \geq 1$.

Ngôn ngữ đặc trưng của bài toán quyết định

- Giả sử Π là một bài toán quyết định với tập dữ kiện D_Π và câu hỏi Q_Π trên mỗi dữ kiện bài toán. Khi đó, đối với mỗi dữ kiện $d \in D_\Pi$, ta có $Q_\Pi(d) = \text{TRUE}$ khi câu hỏi Q_Π trên dữ kiện d được trả lời là "đúng". Trường hợp ngược lại, $Q_\Pi(d) = \text{FALSE}$.

- Giả sử e là một phép mã hóa thích hợp nào đó đối với bài toán Π, e ánh xạ các dữ kiện bài toán thành các xâu thuộc Σ^*. Ta kí hiệu $\langle d \rangle = e(d)$ với mỗi $d \in D_\Pi$. Khi đó *Ngôn ngữ đặc trưng* của Π, kí hiệu là $L(\Pi)$ (hoặc ngắn gọn bởi chữ nghiêng $\mathit{\Pi}$), được định nghĩa như sau:

$$L(D_\Pi) \stackrel{\text{def}}{=} \{\langle d \rangle \,|\, d \in D_\Pi\},$$

$$L(\Pi) \stackrel{\text{def}}{=} \{\langle d \rangle \,|\, d \in D_\Pi \,\&\, Q_\Pi = \text{TRUE}\}.$$

Bài toán và cách tiếp cận

Quy trình giải bài toán:

(1) Lập mô hình toán học cho bài toán (khi cần).

(2) Dựa trên mô hình đã có, xây dựng phương pháp giải hay *thuật toán* giải.

⋆ Lưu ý:

- Các công đoạn của quy trình phải dựa trên những trang thiết bị tính toán hiện có.
- Trong trường hợp bài toán chưa được phát biểu bằng ngôn ngữ toán học, thì việc đầu tiên phải làm là *dịch/chuyển đổi/mã hóa* bài toán đó sang một ngữ cảnh toán học thích hợp (Ví dụ, Bài toán xếp ba lô đã trình bày ở phần trước).

Bài toán thứ mười của Hilbert

- Năm 1900, David Hilbert đã đặt ra hai mươi ba bài toán dành cho thế kỉ 21. Trong số đó, bài toán thứ mười - được coi là bài toán quyết định cụ thể nổi tiếng nhất mọi thời đại, đề cập đến khái niệm *thuật toán*.
- Nội dung của bài toán là: *hãy cho một thuật toán để kiểm định một phương trình đa thức với hệ số nguyên cho trước có nghiệm nguyên hay không*? Khi đó, Hilbert không sử dụng thuật ngữ "algorithm" mà thay vào đó là "một quá trình xử lý để xác định nghiệm sau một số hữu hạn bước".
- Đến nay ta đã biết là không có một thuật toán như vậy. Nghĩa là, bài toán này không quyết định được. Tuy nhiên, việc chứng minh thuật toán không tồn tại **đòi hỏi phải có một định nghĩa chính xác về thuật toán**. Vì vậy, quá trình giải bài toán thứ mười đã phải chờ đến khi có định nghĩa đó.

Khái niệm trực giác về thuật toán

Thuật toán (algorithm) là một quy tắc để, với những dữ liệu ban đầu đã cho, tìm được lời giải của bài toán được xét sau một khoảng thời gian hữu hạn.

Yêu cầu:

- **Tính đúng đắn, hữu hạn:** thuật toán cần phải kết thúc sau một số hữu hạn bước và ta thu được câu trả lời cho vấn đề đặt ra.
- **Tính xác định:** ở mỗi bước, thuật toán cần phải xác định, nghĩa là chỉ rõ việc cần làm.
- **Tính hiệu quả:** thuật toán có thể sử dụng thật sự trên máy tính.

⋆ Lưu ý: khái niệm trực giác về thuật toán không thể là cơ sở để chứng minh điều khẳng định "không tồn tại một thuật toán nào giải bài toán đã cho".

Ví dụ về thuật toán

Bài toán tìm cực đại:

Cho n số $X[1], X[2], \ldots, X[n]$, tìm m và số j lớn nhất sao cho

$$m = X[j] = \max_{1 \le k \le n} X[k].$$

Tức là, tìm cực đại của các số đã cho và chỉ số lớn nhất trong các số cực đại.

Thuật toán:

B1. Khởi tạo, đặt $j = n, k = n - 1, m = X[n]$.

B2. Nếu $k = 0$ thì **Kết thúc**.

B3. Nếu $X[k] \le m$ thì **chuyển sang B5**.

B4. Đặt $j = k, m = X[k]$.

B5. Đặt $k = k - 1$ và **chuyển sang B2**.

Định nghĩa thuật toán

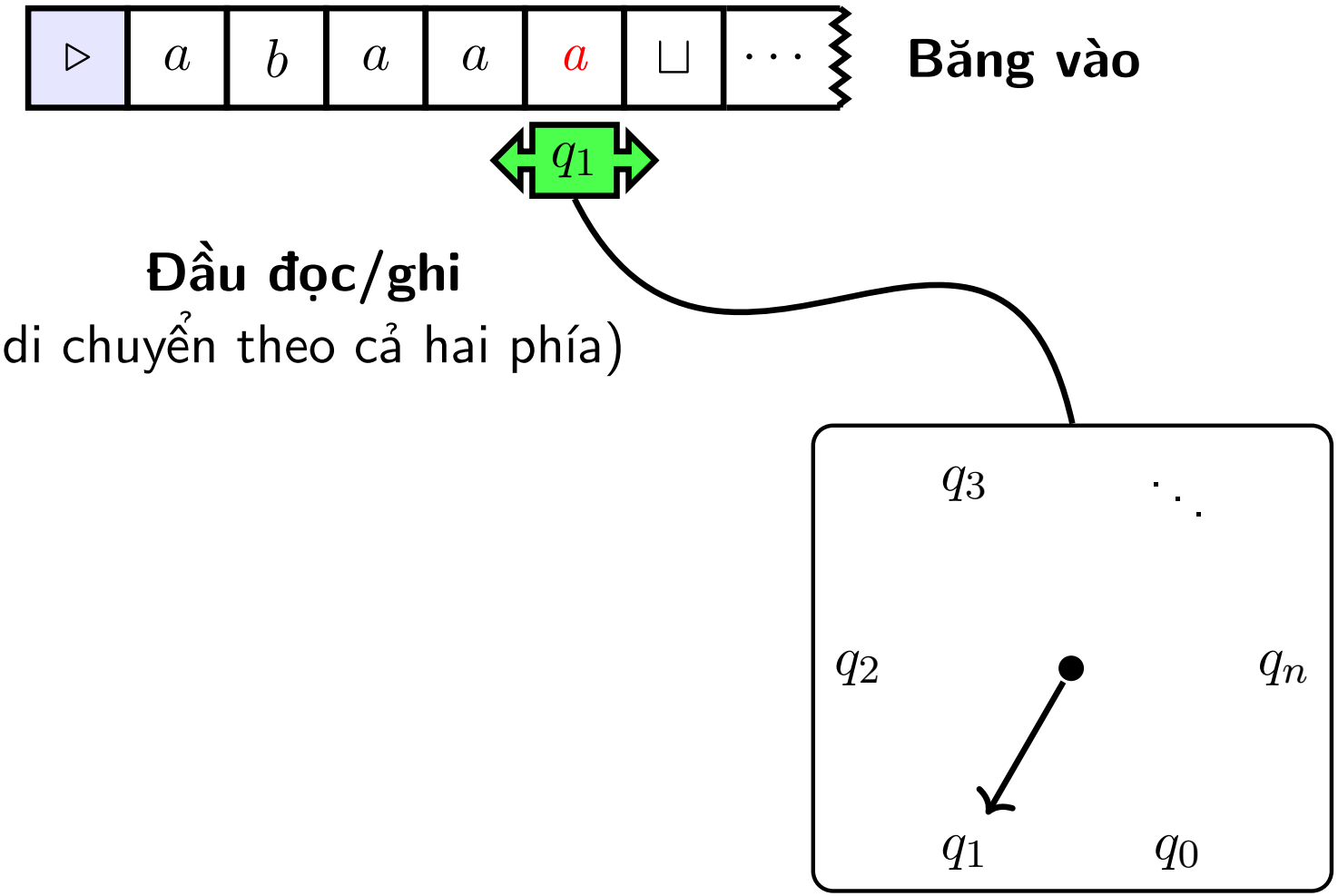
Năm 1936, Alonzo Church và Alan Turing đưa ra định nghĩa thuật toán và đồng nhất nó với thuật toán theo nghĩa trực giác. Từ đó, cung cấp định nghĩa thuật toán cần thiết để giải bài toán thứ mười của Hilbert. Đến năm 1970, Yuri Matiyasevich chứng minh thuật toán cho bài toán này không tồn tại.

Định nghĩa 1.1 **Thuật toán** là máy Turing dừng.

Thuật toán được định nghĩa bởi mô hình máy Turing còn được gọi là **thuật toán máy Turing**.

Luận đề Church-Turing (Church-Turing thesis) Thuật toán theo nghĩa trực giác đồng nhất với thuật toán máy Turing.

Máy Turing


▷
a
b
a
a
a
⊔
· · ·
Băng vào
q_1
Đầu đọc/ghi
(di chuyển theo cả hai phía)
q_3
q_2
q_n
q_1
q_0
Bộ phận điều khiển

Định nghĩa 1.2 Một máy Turing là một bộ năm $(\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$, với:

- $\mathcal{K}$ là một tập hữu hạn các trạng thái;
- Σ là một bảng chữ cái, chứa đựng **kí tự khoảng trống/dấu cách** $\sqcup$ và **kí tự trái nhất** $\triangleright$, nhưng không chứa các ký tự $\leftarrow$ và $\rightarrow$;
- $s \in \mathcal{K}$ là trạng thái khởi đầu;
- $\mathcal{H} \subseteq \mathcal{K}$ là tập trạng thái kết thúc;
- δ, **hàm chuyển**,
  là một hàm từ $(\mathcal{K} - \mathcal{H}) \times \Sigma$ sang $\mathcal{K} \times (\Sigma \cup \{\leftarrow, \rightarrow\})$ sao cho:

(a) với mọi $q \in \mathcal{K} - \mathcal{H}$, nếu $\delta(q, \triangleright) = (p, b)$ thì $b = \rightarrow$.

(b) với mọi $q \in \mathcal{K} - \mathcal{H}$ và $a \in \Sigma$, nếu $\delta(q, a) = (p, b)$ thì $b \neq \triangleright$.

Ví dụ 1.1 Cho máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \{h\})$, với:

$\mathcal{K} = \{q_0, q_1, h\}$,

$\Sigma = \{a, \sqcup, \triangleright\}$,

$s = q_0$,

và δ được cho trong bảng sau:

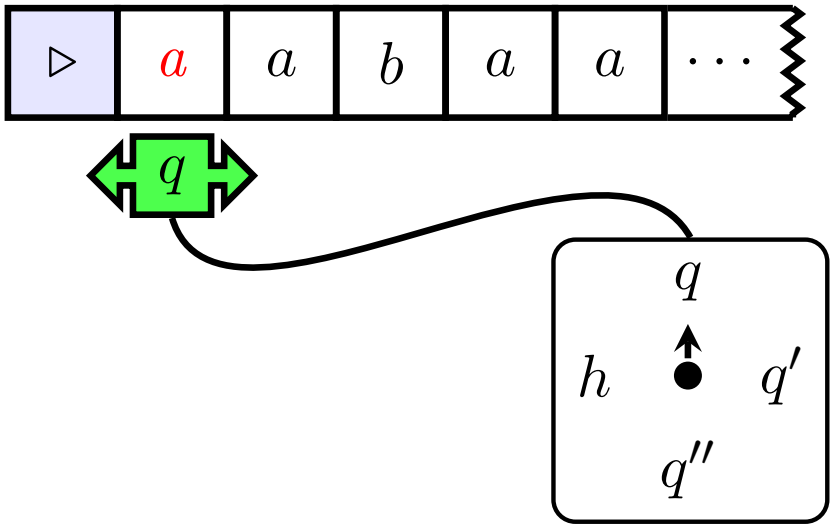
$q,$	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	(q_0, a)
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

Định nghĩa 1.3 **Một cấu hình (configuration)** của máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ là một phần tử của $\mathcal{K} \times \triangleright\Sigma^* \times (\Sigma^*(\Sigma - \{\sqcup\}) \cup \{\varepsilon\})$.

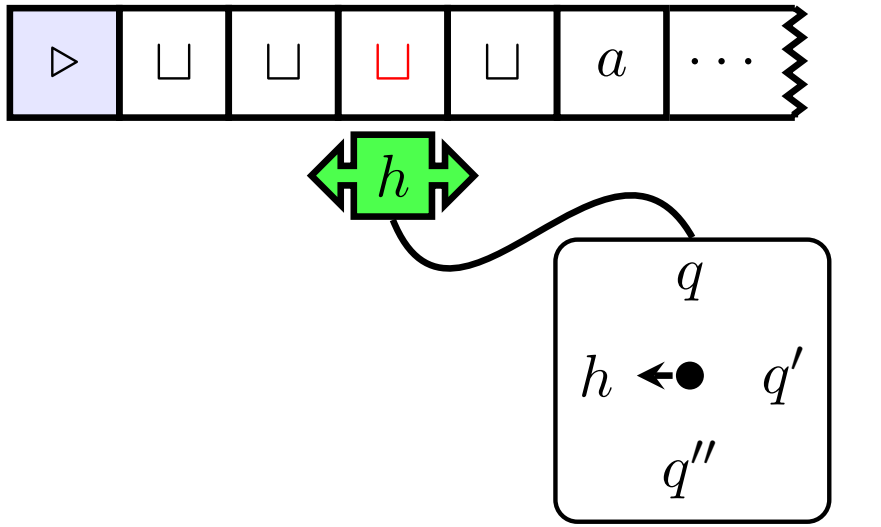
Chú ý:

- Mọi cấu hình đều được giả thiết là bắt đầu bởi ký hiệu trái nhất và không bao giờ kết thúc với một dấu cách trừ khi dấu cách đang là ký tự hiện tại được đọc. Vì vậy, $(q, \triangleright a, aba), (h, \triangleright \sqcup \sqcup\sqcup, \sqcup a)$ và $(q, \triangleright \sqcup a \sqcup \sqcup, \varepsilon)$ là các cấu hình, nhưng $(q, \triangleright baa, a, bc\sqcup)$ và $(q, \sqcup aa, ba)$ thì không phải.
- Một cấu hình mà thành phần của nó có chứa trạng thái trong $\mathcal{H}$ được gọi là **một cấu hình dừng (halted configuration)**.
- Ta cũng qui ước dùng một kí hiệu đơn giản để mô tả nội dung băng vào của cấu hình. Cụ thể, ta sẽ viết $\omega\underline{a}u$ để diễn tả nội dung băng vào của cấu hình $(q, \omega a, u)$. Kí hiệu được gạch dưới chỉ ra vị trí của đầu đọc. Từ đây, ta có thể viết gọn, chẳng hạn $(q, \omega\underline{a}u)$ thay cho $(q, \omega a, u)$.

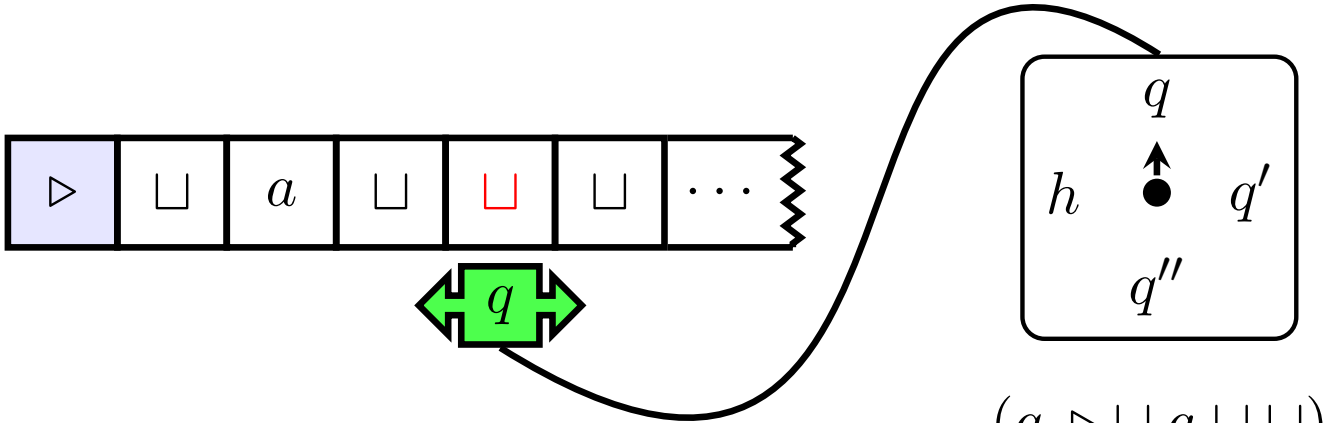
Máy Turing



$(q, \triangleright \underline{a}abaa)$



$(h, \triangleright \sqcup \sqcup \underline{\sqcup} \sqcup a)$



$(q, \triangleright \sqcup a \sqcup \underline{\sqcup})$

Định nghĩa 1.4 Cho máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ và hai cấu hình của M, $(q_1, \omega_1 \underline{a_1}, u_1)$ và $(q_2, \omega_2 \underline{a_2}, u_2)$, với $a_1, a_2 \in \Sigma$. Khi đó, ta có

$$(q_1, \omega_1 \underline{a_1}, u_1) \vdash_M (q_2, \omega_2 \underline{a_2}, u_2)$$

khi và chỉ khi, với mỗi $b \in \Sigma \cup \{\leftarrow, \rightarrow\}, \delta(q_1, a_1) = (q_2, b)$ tương ứng với một trong các trường hợp sau đây:

1. $b \in \Sigma, \omega_1 = \omega_2, u_1 = u_2$ và $a_2 = b$.
2. $b = \leftarrow, \omega_1 = \omega_2 a_2$ và
   (a) $u_2 = a_1 u_1$ (nếu $a_1 \neq \sqcup$ hoặc $u_1 \neq \varepsilon$) hoặc
   (b) $u_2 = \varepsilon$ (nếu $a_1 = \sqcup$ và $u_1 = \varepsilon$).
3. $b = \rightarrow, \omega_2 = \omega_1 a_1$ và
   (a) $u_1 = a_2 u_2$ hoặc
   (b) $u_1 = u_2 = \varepsilon$ và $a_2 = \sqcup$.

Ví dụ 1.1 (tiếp) Giả sử máy Turing M đã cho khởi đầu với cấu hình $(q_1, \triangleright\underline{\sqcup}aaaa)$. Các bước tính toán của M như sau:

$$
\begin{aligned}
(q_1, \triangleright\underline{\sqcup}aaaa) &\vdash_M (q_0, \triangleright \sqcup \underline{a}aaa)\\
&\vdash_M (q_1, \triangleright \sqcup \underline{\sqcup}aaa)\\
&\vdash_M (q_0, \triangleright \sqcup \sqcup\underline{a}aa)\\
&\vdash_M (q_1, \triangleright \sqcup \sqcup\underline{\sqcup}aa)\\
&\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \underline{a}a)\\
&\vdash_M (q_1, \triangleright \sqcup \sqcup \sqcup \underline{\sqcup}a)\\
&\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \sqcup\underline{a})\\
&\vdash_M (q_1, \triangleright \sqcup \sqcup \sqcup \sqcup\underline{\sqcup})\\
&\vdash_M (q_0, \triangleright \sqcup \sqcup \sqcup \sqcup \sqcup \underline{\sqcup})\\
&\vdash_M (h, \triangleright \sqcup \sqcup \sqcup \sqcup \sqcup \underline{\sqcup})
\end{aligned}
$$

Định nghĩa 1.5 Với máy Turing M bất kỳ, gọi $\vdash_M^*$ là bao đóng (phản xạ, bắc cầu) của $\vdash_M$. Khi đó ta nói cấu hình C dẫn xuất (gián tiếp) cấu hình C' nếu $C \vdash_M^* C'$. Một phép tính toán (computation) thực hiện bởi M là một dãy các cấu hình $C_0, C_1, \ldots, C_n$, với $n \geq 0$ sao cho

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \cdots \vdash_M C_n.$$

Ta cũng nói rằng phép tính toán có độ dài n hoặc nó có n bước và ta ký hiệu là $C_0 \vdash_M^n C_n$.

Ví dụ 1.2 Cho máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \{h\})$, với: $\mathcal{K} = \{q_0, q_1, h\}$, $\Sigma = \{a, b, \sqcup, \triangleright\}$, $s = q_0$, và δ được cho trong bảng sau:

$q,$	σ	$\delta(q, \sigma)$
q_0	a	(q_1, b)
q_0	b	(q_1, a)
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	a	$(q_0, \rightarrow)$
q_1	b	$(q_0, \rightarrow)$
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

(a) Liệt kê các bước tính toán của M bắt đầu từ cấu hình $(q_0, \triangleright \underline{a} abbba)$?

(b) Mô tả M làm gì khi nó bắt đầu tại q_0 và đọc ô bất kỳ của băng vào?

Ví dụ 1.3 Cho máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \{h\})$, với:
$\mathcal{K} = \{q_0, q_1, q_2, h\}$, $\Sigma = \{a, b, \sqcup, \triangleright\}$, $s = q_0$, và δ được cho trong bảng sau:

$q,$	σ	$\delta(q, \sigma)$
q_0	a	$(q_1, \leftarrow)$
q_0	b	$(q_0, \rightarrow)$
q_0	$\sqcup$	$(q_0, \rightarrow)$
q_1	a	$(q_1, \leftarrow)$
q_1	b	$(q_2, \rightarrow)$
q_1	$\sqcup$	$(q_1, \leftarrow)$
q_2	a	$(q_2, \rightarrow)$
q_2	b	$(q_2, \rightarrow)$
q_2	$\sqcup$	$(h, \sqcup)$

- Lặp lại hai yêu cầu của Ví dụ 1.2 với cấu hình ban đầu là $(q_0, \triangleright a\underline{b}b \sqcup bb \sqcup \sqcup \sqcup aba)$?

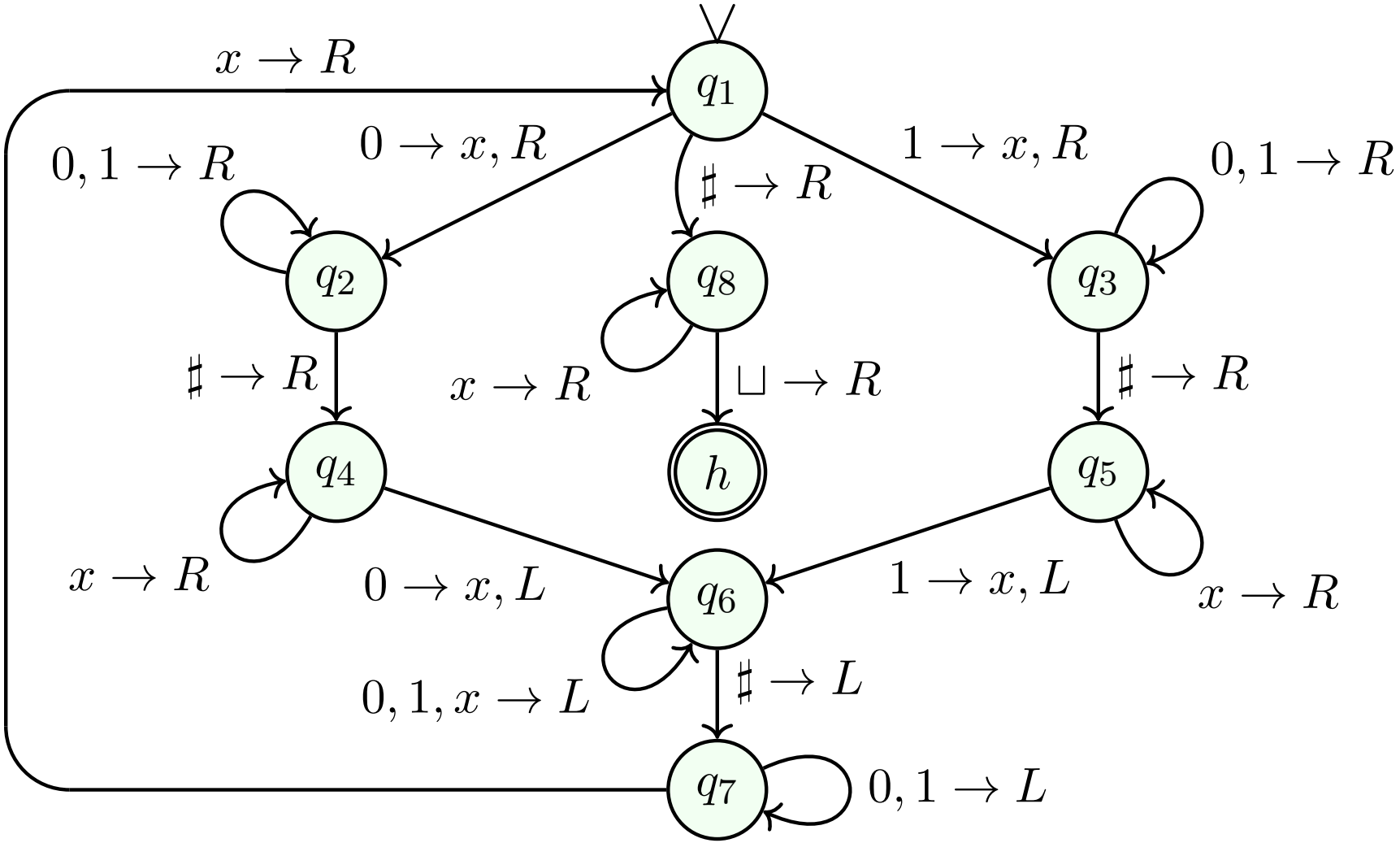
Bài tập 1.1 Thiết kế một máy Turing tìm dấu cách trong một xâu cho trước.

Bài tập 1.2 Thiết kế một máy Turing kiểm tra tính cân xứng của các từ trên bảng chữ Σ cho trước.

Bài tập 1.3 Xây dựng một máy Turing kiểm tra xem hai số nhị phân cho trước có bằng nhau không.

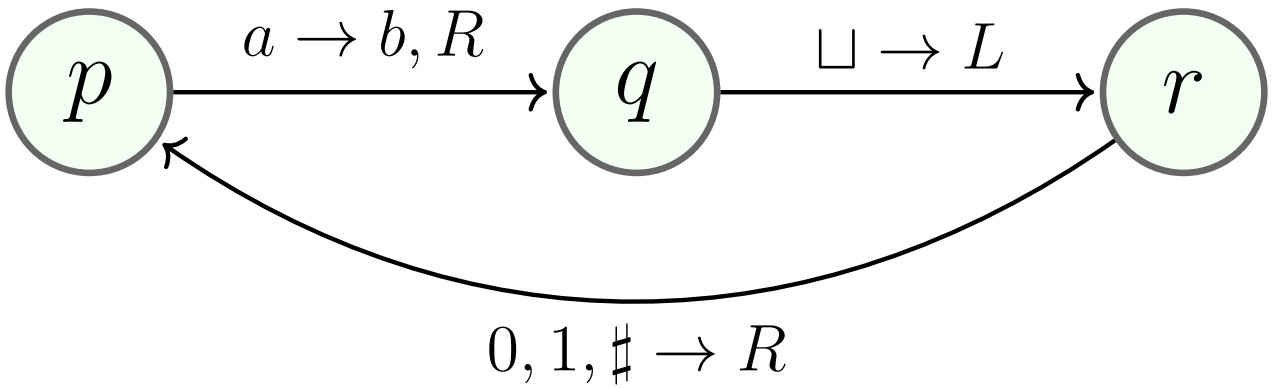
Máy Turing

Câu đố 1.1 Máy Turing M sau đây tương ứng với bài tập nào?



Máy Turing

Lưu ý: Trong câu đố là máy Turing M với hàm chuyển $\delta : (\mathcal{K} - \mathcal{H}) \times \Sigma \longrightarrow \mathcal{K} \times \Sigma \times \{L, R\}$. Hàm chuyển có thể được cho dưới dạng đồ thị chuyển. Chẳng hạn, đồ thị chuyển sau đây biểu diễn $\delta(p, a) = (q, b, R)$, $\delta(q, \sqcup) = (r, \sqcup, L)$, $\delta(r, 0) = (p, 0, R)$, $\delta(r, 1) = (p, 1, R)$ và $\delta(r, \sharp) = (p, \sharp, R)$.



(⋆) Trong [5], hàm chuyển được định nghĩa như sau

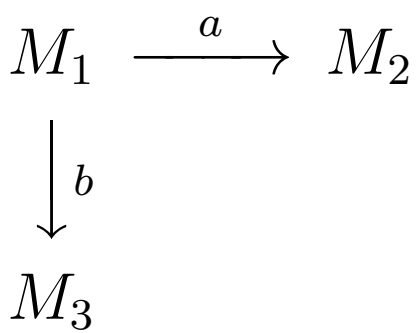
$$\delta : (\mathcal{K} - \mathcal{H}) \times \Sigma \longrightarrow \mathcal{K} \times \Sigma \times \{L, R, S\}.$$

Các máy Turing cơ bản

- *Máy Turing ghi kí tự* và *máy Turing di chuyển đầu đọc*: Cho Σ là bảng chữ cái cố định. Với mỗi $a \in \Sigma \cup \{\rightarrow, \leftarrow\} - \{\triangleright\}$ ta định nghĩa một máy Turing $M_a = (\{s, h\}, \Sigma, \delta, s, \{h\})$. Trong đó, với mỗi $b \in \Sigma - \{\triangleright\}$, $\delta(s, b) = (h, a)$. Lưu ý, $\delta(s, \triangleright) = (s, \rightarrow)$.
- Thực chất, các máy Turing này chỉ làm một việc duy nhất là ghi ký tự a vào ô hiện tại trên băng vào nếu $a \in \Sigma$ hoặc di chuyển đầu đọc nếu $a \in \{\rightarrow, \leftarrow\}$. Sau công việc đó, nó dừng ngay lập tức (ngoại trừ trường hợp nó đọc phải ký hiệu trái nhất $\triangleright$ thì đầu đọc di chuyển sang phải).
- Ta sẽ đơn giản ký hiệu bằng cách viết a thay vì M_a. Nghĩa là, nếu $a \in \Sigma$ thì máy Turing ghi kí tự$-a$ được ký hiệu là a. Các máy Turing di chuyển đầu đọc $M_{\leftarrow}$ và $M_{\rightarrow}$ sẽ được ký hiệu là L và R.

Qui tắc kết hợp các máy Turing cơ bản

- Mỗi máy Turing cơ bản giống như một trạng thái của một otomat hữu hạn. Cách kết hợp các máy Turing cơ bản tương tự cách kết nối các trạng thái của một otomat hữu hạn với nhau. Tuy nhiên cần lưu ý là việc kết nối từ máy này tới máy khác chỉ có hiệu lực khi máy thứ nhất dừng và máy thứ hai bắt đầu từ trạng thái khởi đầu của nó với băng vào và vị trí đầu đọc như ở thời điểm máy thứ nhất dừng.
- Ví dụ, cho M_1, M_2 và M_3 là ba máy Turing cơ bản. Ta có một máy Turing kết hợp làm việc như sau:



Qui tắc kết hợp các máy Turing cơ bản

Giả sử $M_1 = (\mathcal{K}_1, \Sigma, \delta_1, s_1, \mathcal{H}_1)$, $M_2 = (\mathcal{K}_2, \Sigma, \delta_2, s_2, \mathcal{H}_2)$ và $M_3 = (\mathcal{K}_3, \Sigma, \delta_3, s_3, \mathcal{H}_3)$. Giả sử $\bigcap_{i=1}^{3} \mathcal{K}_i = \emptyset$. Khi đó ta có $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ với:

$$\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3, s = s_1, \mathcal{H} = \mathcal{H}_2 \cup \mathcal{H}_3.$$

Với mỗi $\sigma \in \Sigma$ và $q \in \mathcal{K} - \mathcal{H}$, $\delta(q, \sigma)$ được định nghĩa như sau:

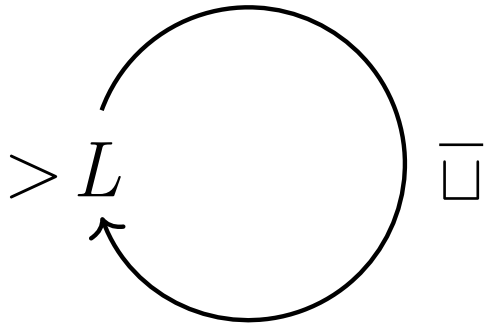
(a) Nếu $q \in \mathcal{K}_1 - \mathcal{H}_1$ thì $\delta(q, \sigma) = \delta_1(q, \sigma)$.

(b) Nếu $q \in \mathcal{K}_2 - \mathcal{H}_2$ thì $\delta(q, \sigma) = \delta_2(q, \sigma)$.

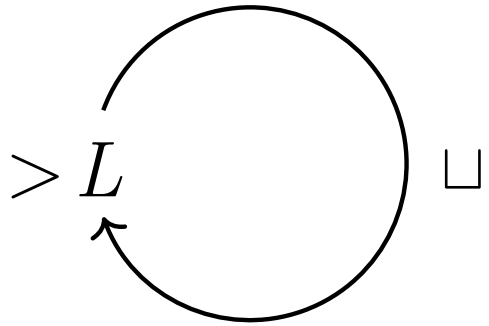
(c) Nếu $q \in \mathcal{K}_3 - \mathcal{H}_3$ thì $\delta(q, \sigma) = \delta_3(q, \sigma)$.

(d) Cuối cùng, nếu $q \in \mathcal{H}_1$ - trường hợp duy nhất - thì $\delta(q, \sigma) = s_2$ nếu $\sigma = a$, $\delta(q, \sigma) = s_3$ nếu $\sigma = b$, và ngược lại $\delta(q, \sigma) \in \mathcal{H}$.

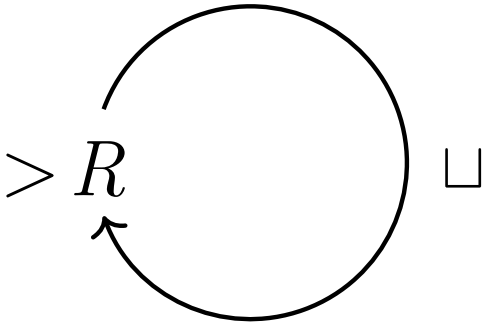
Ví dụ 1.4 Máy Turing tìm kí tự:



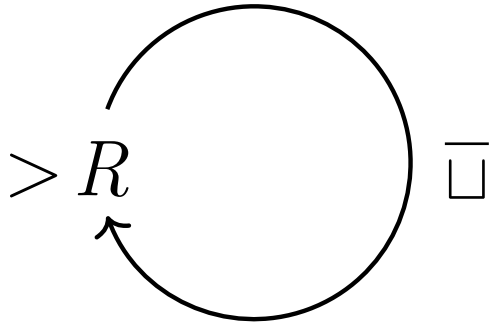
(a) $L_{\sqcup}$



(b) $L_{\overline{\sqcup}}$

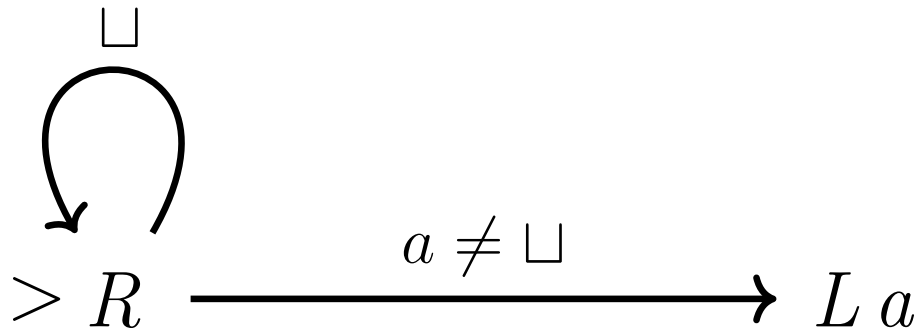


(c) $R_{\overline{\sqcup}}$

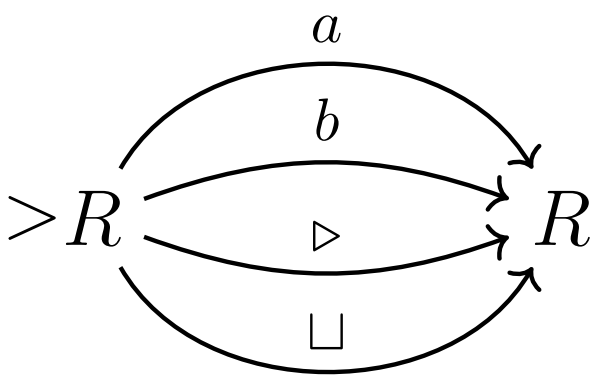


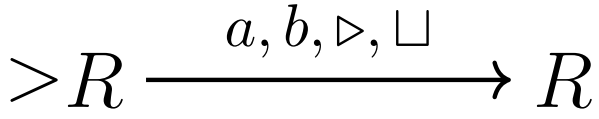
(d) $R_{\sqcup}$

Ví dụ 1.5 Máy Turing sao chép kí tự:

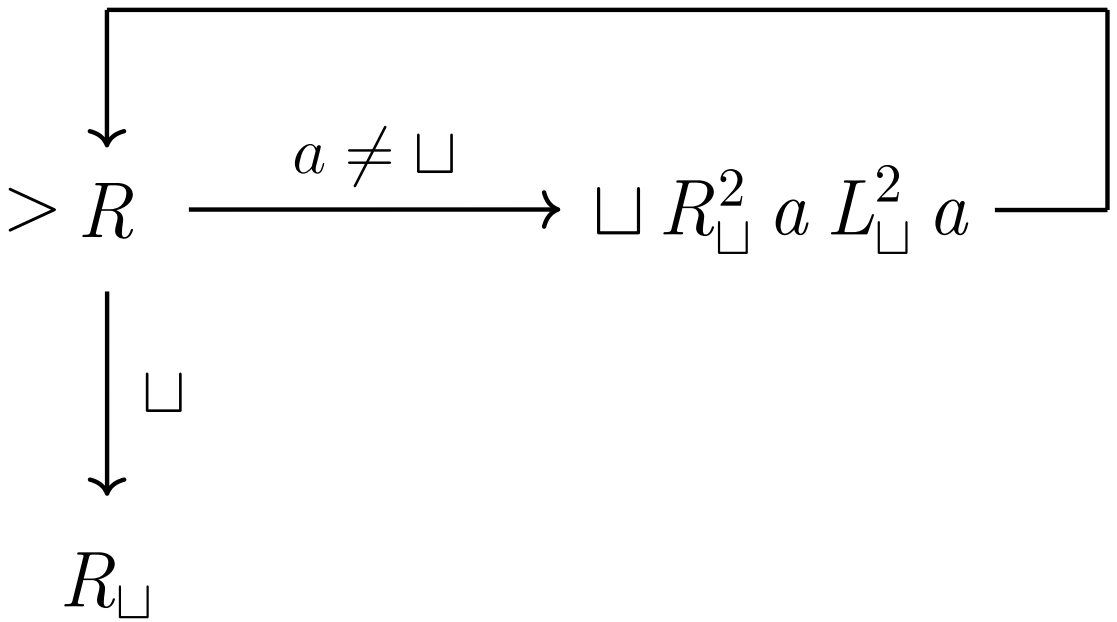


⋆ Lưu ý: Giả sử $\Sigma = \{a, b, \triangleright, \sqcup\}$, máy Turing $R \to R$ có các biểu diễn tương ứng sau đây:





Bài tập 1.4 Mô tả hoạt động của máy Turing sau đây:



Định nghĩa 1.6 Cho máy Turing $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ với $\mathcal{H} = \{y, n\}$ gồm hai trạng thái dừng phân biệt (y và n tương ứng với "yes" và "no"). Bất kỳ cấu hình nào có chứa thành phần y được gọi là **cấu hình chấp nhận**, còn cấu hình chứa thành phần n được gọi là **cấu hình bác bỏ**. Ta nói rằng máy Turing M **chấp nhận đầu vào** (accepts input) ω nếu $q_0\omega \vdash_M^* uyv$, nghĩa là khi tính toán trên từ vào ω, máy M chuyển từ cấu hình ban đầu đến cấu hình chấp nhận.

Tập các từ vào mà máy M chấp nhận tạo thành *ngôn ngữ chấp nhận được* của M, được gọi là **ngôn ngữ của máy Turing** (language of a Turing machine) M và được ký hiệu bởi L_M.

Máy Turing đoán nhận ngôn ngữ

Định nghĩa 1.7 Ngôn ngữ L được gọi là **đoán nhận được theo Turing** (Turing-recognizable) hay đơn giản là **đoán nhận được** (recognizable) nếu nó là ngôn ngữ chấp nhận được của một máy Turing nào đó, nghĩa là nếu tồn tại một máy Turing M sao cho $L = L_M$. Khi đó ta nói rằng "máy Turing M đoán nhận ngôn ngữ L", hay "ngôn ngữ L được đoán nhận bởi máy Turing M".

Ví dụ, ngôn ngữ (được đoán nhận bởi máy Turing) bao gồm tất cả các điệp từ trong bảng chữ Σ là:

$$L = \{\omega \,|\, \omega \in \Sigma^*, \omega = \omega^{\mathcal{R}}\},$$

trong đó $\omega^{\mathcal{R}}$ là từ ngược của từ ω.

Máy Turing đoán nhận ngôn ngữ

$\star$ Lưu ý: Máy Turing dừng được gọi là **máy quyết định** (decider) bởi vì nó luôn luôn có khả năng quyết định hay bác bỏ đối với mỗi từ vào của nó.

Định nghĩa 1.8 Ngôn ngữ L được gọi là **khẳng định được** (decidable), hay cụ thể hơn là **khẳng định được theo Turing** (Turing-decidable) nếu nó được đoán nhận bởi máy quyết định M nào đó. Trong trường hợp ấy, ta nói rằng "máy Turing dừng M khẳng định ngôn ngữ L"; ngược lại, ngôn ngữ L được gọi là **không khẳng định được** (undecidable).

Bài tập 1.5 Cho ngôn ngữ

$$L = \{0^i 1^i \,|\, i = 1, 2, \ldots\}.$$

Chứng minh rằng L là ngôn ngữ khẳng định được.

Máy Turing đoán nhận ngôn ngữ

Trong Bài tập 1.5, ta cần chứng tỏ rằng tồn tại máy Turing dừng M đoán nhận ngôn ngữ L.

Ý tưởng ban đầu: Máy sẽ thực hiện tính toán theo một qui trình lặp mà nội dung mỗi phép lặp bao gồm:

- ⋆ Di chuyển zigzac đầu đọc-ghi từ kí tự khác $\sqcup$ đầu tiên đến kí tự khác $\sqcup$ cuối cùng trên băng và xóa các kí tự ấy nếu chúng tương ứng là 0 và 1; ngược lại, *bác bỏ*.
- ⋆ Khi thực hiện bước lặp đầu tiên, máy đồng thời loại bỏ từ rỗng cũng như những từ đầu vào mà bắt đầu bởi kí tự 1 và những từ vào mà sau kí tự 1 còn xuất hiện kí tự 0.
- ⋆ Khi bắt đầu một phép lặp tiếp theo (máy ở trạng thái q_0'), không kể phép lặp đầu tiên, nếu bắt gặp $\sqcup$, tức băng còn toàn ô trống, thì *chấp nhận*.

Qui ước cấu hình khởi đầu của các máy Turing:

Giả sử $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ là một máy Turing và $\omega \in (\Sigma - \{\sqcup, \triangleright\})^*$ thì **cấu hình khởi đầu** của M đối với xâu vào ω là:

$$(s, \triangleright \underline{\sqcup} \omega)$$

Bài tập 1.6 Cho ngôn ngữ

$$L = \{a^n b^n c^n : n \geq 0\}.$$

Chứng minh rằng L là ngôn ngữ khẳng định được.

Máy Turing tính các hàm

Cho hàm $f : \Sigma^* \to \Sigma^*$. Ta có thể hình dung máy Turing tính hàm f bằng cách xử lý mỗi từ vào $\omega \in \Sigma^*$ sao cho, nếu tại ω hàm f xác định thì máy dừng ở trạng thái chấp nhận và nội dung trên băng là từ $f(\omega) \in \Sigma^*$; ngược lại, máy dừng ở trạng thái bác bỏ.

Giả sử $M = (\mathcal{K}, \Sigma, \delta, s, \mathcal{H})$ là một máy Turing. *Hàm tương ứng với máy Turing* M là hàm từ Σ^* vào Σ^*, được ký hiệu là F_M và được định nghĩa như sau:

$$F_M(\omega) = \begin{cases} uv, & \text{nếu } q_0\omega \vdash_M^* uyv, \\ \textit{không xác định} & \text{nếu ngược lại.} \end{cases}$$

Định nghĩa 1.9 Hàm $f : \Sigma^* \to \Sigma^*$ được gọi là **tính được** (computable), hay cụ thể hơn là **tính được theo Turing** (Turing-computable), nếu tồn tại một máy Turing dừng M sao cho hàm tương ứng với nó F_M trùng với hàm f, tức $F_M \backsimeq f$ theo nghĩa, đối với mỗi $\omega \in \Sigma^*$, nếu một trong hai hàm xác định thì hàm kia cũng xác định và chúng nhận cùng một giá trị. Khi đó ta nói rằng "máy Turing dừng M tính hàm f" hay "hàm f tính được bởi máy Turing dừng M".

Bài tập 1.7 Cho hàm $f : \{0,1\}^* \to \{0,1\}^*$, trong đó

$$f(\omega) = \begin{cases} 0^{ik}, & \text{nếu } \omega = 0^i 1 0^k \text{ với mọi } i,k \geq 1, \\ \textit{không xác định} & \text{nếu ngược lại.} \end{cases}$$

Chứng minh rằng f là hàm tính được.

Máy Turing tính các hàm

Trong Bài tập 1.7, ta có thể xây dựng một máy Turing dừng M theo ý tưởng sau đây.

Ý tưởng ban đầu:

- ⋆ Trước tiên loại bỏ các từ vào không có dạng $0^i10^k(i,k \geq 1)$ và ngăn cách phần dữ liệu đầu vào với kết quả tính toán bởi dấu $\sharp$; đồng thời, tiến hành chuyển xâu 0^k sang bên phải dấu $\sharp$ bằng cách lần lượt chuyển từng kí tự một (kí tự nào chuyển đi thì được đánh dấu bởi $\otimes$).
- ⋆ Quá trình chuyển xâu 0^k được diễn ra i lần, mỗi lần chuyển như vậy một kí tự 0 trong xâu 0^i (bên trái kí tự 1) được xóa đi. Quá trình này kết thúc khi không còn kí tự 0 nào bên trái kí tự 1.
- ⋆ Cuối cùng, trước khi kết thúc quá trình tính toán, xóa các kí tự bắt đầu từ 1 cho đến dấu $\sharp$ và khi đó trên băng chỉ còn lại đúng ik kí tự 0.

Máy Turing không tất định

Định nghĩa 1.10 Một máy Turing không đơn định/không tất định (Nondeterministic Turing Machine) N là một bộ năm $(\mathcal{K}, \Sigma, \Delta, s, \mathcal{H})$, với

- $\mathcal{K}, \Sigma, s, \mathcal{H}$ được định nghĩa tương tự các thành phần $\mathcal{K}, \Sigma, s, \mathcal{H}$ trong máy Turing tiêu chuẩn (Định nghĩa 1.2);
- Δ là một bộ phận của tập hợp $((\mathcal{K} - \mathcal{H}) \times \Sigma) \times (\mathcal{K} \times (\Sigma \cup \{\leftarrow, \rightarrow\}))$ - Trong định nghĩa máy Turing tiêu chuẩn, thành phần này là một hàm từ $(\mathcal{K} - \mathcal{H}) \times \Sigma$ sang $\mathcal{K} \times (\Sigma \cup \{\leftarrow, \rightarrow\})$.

⋆ Lưu ý: Các cấu hình, quan hệ $\vdash_N$ và quan hệ $\vdash_N^*$ được định nghĩa tương tự như đối với máy Turing tiêu chuẩn. Tuy nhiên $\vdash_N$ không nhất thiết đơn trị. Nghĩa là, một cấu hình có thể sản sinh ra một số cấu hình khác trong một bước chuyển.

⋆ Quá trình tính toán của máy Turing N trên mỗi từ vào ω được biểu diễn bằng **một cây tính toán** (computation tree), kí hiệu là $\mathcal{T}_N(\omega)$.

Máy Turing không tất định

Định nghĩa 1.11 Cho máy Turing không tất định N với bảng chữ vào Σ. Ta nói rằng máy Turing N **chấp nhận** (một cách không tất định) đầu vào ω nếu trong cây tính toán $\mathcal{T}_N(\omega)$ của N trên ω có nhánh tính toán dẫn đến cấu hình chấp nhận.

- ⋆ Tập các từ vào mà máy Turing N chấp nhận tạo thành *ngôn ngữ chấp nhận được* của N, được gọi là **ngôn ngữ của máy Turing không tất định** N và được kí hiệu bởi L_N.

Máy Turing không tất định

Định nghĩa 1.12 Cho ngôn ngữ L và một máy Turing không tất định N. Ta nói rằng máy Turing N đoán nhận ngôn ngữ L hay ngôn ngữ L được đoán nhận bởi máy Turing N, nếu $L = L_N$.

Định lý 1.1 Mỗi máy Turing không tất định đều có máy Turing tất định một băng tương đương.

- ⋆ **Hệ quả 1.1** Một ngôn ngữ là đoán nhận được khi và chỉ khi nó được đoán nhận bởi máy Turing không tất định nào đó.
- ⋆ **Hệ quả 1.2** Một ngôn ngữ là khẳng định được khi và chỉ khi nó được đoán nhận bởi máy Turing không tất định *dừng* nào đó.

Máy Turing không tất định

Ví dụ 1.6 Một hợp số là tích của hai số tự nhiên lớn hơn 1 (ví dụ, 4,6,8,12 là các hợp số, còn 1,2,3,5,7 thì không là hợp số). Đặt $C = \{100, 110, 1000, 1001, \ldots, 1011011, \ldots\}$ là tập các xâu nhị phân biểu diễn các hợp số. Hãy thiết kế một thuật toán "hiệu quả" khẳng định C.

Ta có thể xây dựng một máy Turing N kiểm tra xem một số nguyên n biểu diễn dưới dạng xâu nhị phân cho trước có là hợp số không theo ý tưởng sau đây:

- ⋆ Chọn tùy ý hai số nhị phân p, q lớn hơn một và ghi xâu biểu diễn nhị phân của chúng lên băng vào.
- ⋆ Thiết kế một máy Turing đơn định thực hiện nhân hai số p, q nói trên và thay thế p và q bởi tích của chúng.
- ⋆ Kiểm tra xem hai số nguyên n và $p.q$ có bằng nhau không. Việc này có thể thực hiện đơn giản bằng cách so sánh từng bit một. Dừng nếu hai số nguyên bằng nhau, ngược lại tiếp tục thực hiện công việc.

Một vài bài toán không giải được

Bài toán chấp nhận (acceptance problem) đối với máy Turing:

$$A_{TM} = \{\langle M, \omega \rangle | \ M \text{ là máy Turing và } M \text{ chấp nhận } \omega\}.$$

Định lý 1.2 Ngôn ngữ A_{TM} là không khẳng định được.

Một vài bài toán không giải được

Bài toán dừng (halting problem) đối với máy Turing:

$HALT_{TM} = \{\langle M, \omega \rangle | \ M$ là máy Turing và M dừng trên đầu vào $\omega\}$.

Định lý 1.3 $HALT_{TM}$ là không khẳng định được.

Một vài bài toán không giải được

Bài toán tương ứng Post (PCP - Post correspondence problem):

Mỗi dữ kiện của bài toán PCP là một cặp (A, B), trong đó A và B là hai danh sách chứa k từ trên một bảng chữ nào đó:

$$A = (a_1, a_2, \ldots, a_k),$$
$$B = (b_1, b_2, \ldots, b_k).$$

Nội dung của PCP bao gồm việc kiểm tra xem, đối với mỗi dữ kiện (A, B), liệu có hay không một dãy các chỉ số $(i_1, i_2, \ldots, i_h)$ với $1 \leq i_j \leq k$ và *không nhất thiết khác nhau*, sao cho:

$$a_{i_1} a_{i_2} \ldots a_{i_h} = b_{i_1} b_{i_2} \ldots b_{i_h}$$

Một vài bài toán không giải được

Bài toán tương ứng Post (PCP - Post correspondence problem):

$$PCP = \{\langle P \rangle | \ P \text{ là dữ kiện ghép được của PCP}\}.$$

Định lý 1.4 PCP là không khẳng định được.

TRÂN TRỌNG CẢM ƠN!