Văn phạm và Ngôn ngữ hình thức

Ngôn ngữ và văn phạm

- **Ngôn ngữ tự nhiên (Natural language)**
  - Cú pháp (syntax) và ngữ nghĩa (semantic)
    - "Colorless green ideas sleep furiously" - N. Chomsky, 1957
  - Rất phức tạp

- **Ngôn ngữ hình thức (Formal language)**
  - Có một tập hoàn toàn xác định các quy tắc cú pháp
    - Làm thế nào để xác định một tổ hợp từ có là một câu hợp lệ trong một ngôn ngữ hình thức hay không?
    - Làm thế nào để sinh ra các câu hợp lệ trong một ngôn ngữ hình thức?

Ngôn ngữ và Văn phạm

1. Một **câu** được tạo bởi một **cụm_danh_từ** theo sau là một **cụm_động_từ**;
2. Một **cụm_danh_từ** được tạo bởi một **mạo_từ** theo sau là một **tính_từ** theo sau là một **danh_từ**, hoặc
3. Một **cụm_danh_từ** được tạo bởi một **mạo_từ** theo sau là một **danh_từ**;
4. Một **cụm_động_từ** được tạo bởi một **động_từ** theo sau là một **trạng_từ**, hoặc
5. Một **cụm_động_từ** được tạo bởi một **động_từ**;
6. Một **mạo_từ** là "a", hoặc
7. Một **mạo_từ** là "the";
8. Một **tính_từ** là "large", hoặc
9. Một **tính_từ** là "hungry";
10. Một **danh_từ** là "rabbit", hoặc
11. Một **danh_từ** là "mathematician";
12. Một **động_từ** là "eats", hoặc
13. Một **động_từ** là "hops";
14. Một **trạng_từ** là "quickly", hoặc
15. Một **trạng_từ** là "wildly".

Ngôn ngữ và Văn phạm

câu

cụm_danh_từ cụm_động_từ

mạo_từ tính_từ danh_từ cụm_động_từ

mạo_từ tính_từ danh_từ động_từ trạng_từ

the **tính_từ danh_từ động_từ trạng_từ**

the large **danh_từ động_từ trạng_từ**

the large rabbit **động_từ trạng_từ**

the large rabbit hops **trạng_từ**

the large rabbit hops quickly

Văn phạm cấu trúc đoạn

Một ***từ vựng (vocabulary)*** V là một tập hữu hạn, không rỗng của các phần tử gọi là các ***ký hiệu (symbol)***. Một ***từ (word)*** trên V là một xâu hữu hạn các phần tử của V. ***Xâu rỗng (empty string)*** được ký hiệu là λ, là xâu không chứa một ký hiệu nào. Tập tất cả các từ trên V được ký hiệu là V^*. Một ***ngôn ngữ (language)*** trên V là tập con của V^*.

- **Ký hiệu kết thúc (terminal)**
  - Là một phần tử của từ vựng không thể thay thế bằng các ký hiệu khác
- **Ký hiệu xuất phát (start symbol – S)**
  - Là phần tử của ngữ vựng mà ta luôn bắt đầu từ nó
- **Luật sinh (production) của văn phạm (grammar)**
  - Các quy tắc chỉ rõ khi nào ta có thể thay thế một xâu trong V^* bởi một xâu khác

Văn phạm cấu trúc đoạn

Một ***văn phạm cấu trúc đoạn (phrase-structure grammar)*** $G = (V, T, S, P)$ gồm một từ vựng V, một tập con T của V gồm các phần tử kết thúc, một ký hiệu xuất phát S và tập các luật sinh P. Tập $V - T$ được ký hiệu là N. Các phần tử thuộc N được gọi là các ký hiệu ***không kết thúc (nonterminal)***. Mỗi luật sinh trong P cần phải chứa ít nhất một ký hiệu không kết thúc ở vế trái của nó.

- Ví dụ: Cho $G = (V, T, S, P)$, trong đó $V = \{a, b, A, B, S\}$, $T = \{a, b\}$, S là ký hiệu xuất phát và tập các luật sinh $P = \{S \Rightarrow ABa, A \rightarrow BB, B \rightarrow ab, AB \rightarrow B\}$. G là một ví dụ về văn phạm cấu trúc câu.

Văn phạm cấu trúc câu

Cho $G = (V, T, S, P)$ là một văn phạm cấu trúc câu. Cho $w_0 = Iz_0r$ (tức là phép ghép của I, z_0 và r) và $w_1 = Iz_1r$ là các xâu trên V. Nếu $z_0 \rightarrow z_1$ là một luật sinh của G, thì ta nói rằng w_1 được ***dẫn xuất trực tiếp (directly derivable)*** từ w_0 và viết $w_0 \Rightarrow w_1$. Nếu $w_0, w_1, \ldots w_n$ với $n \geq 0$ là các xâu trên V sao cho $w_0 \Rightarrow w_1, w_1 \Rightarrow w_2, \ldots, w_{n-1} \Rightarrow w_n$, thì ta nói rằng w_n được dẫn xuất từ w_0 và viết $w_0 \Rightarrow w_n$. Dãy các bước được dùng để nhận được w_n từ w_0 được gọi là một dẫn xuất.

Văn phạm cấu trúc câu

Cho $G = (V, T, S, P)$ là một văn phạm cấu trúc câu. Ngôn ngữ được sinh bởi văn phạm G (hay ngôn ngữ của G), được ký hiệu là $L(G)$, là tập hợp tất cả các xâu ký hiệu kết thúc được dẫn xuất từ ký hiệu xuất phát S. Nói các khác,

$$L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$$

Văn phạm cấu trúc câu

❑ Ví dụ:

Cho G là văn phạm với từ vựng $V = \{S, A, a, b\}$, tập cá ký hiệu kết thúc T={a,b}, ký hiệu bắt đầu S và tập các luật sinh $P = \{S \to aA, S \to b,\ A \to aa\}$. Xác định ngôn ngữ $L(G)$ của văn phạm đó.

❑ Lời giải:

Từ ký hiệu xuất phát S ta có thể dẫn ra aA hoặc b bằng các dùng hai luật sinh $S \to aA$ hoặc $S \to b$ tương ứng. Từ aA có thể dẫn xuất ra aaa. Ngoài ra, không thể thêm một từ nào nữa. Do đó

$$L(G) = \{b, aaa\}$$

Các loại văn phạm cấu trúc đoạn

❑ Các loại văn phạm cấu trúc đoạn

- Dựa theo các loại luật sinh
  - **Loại 0**: không có các hạn chế đối với các luật sinh của chúng
  - **Loại 1**: các luật sinh có dạng $lAr \rightarrow lwr$
    - Với A là ký hiệu không kết thúc; l và r là các xâu không hoặc nhiều ký hiệu kết thúc hoặc không kết thúc; và w là một xâu không rỗng của các ký tự kết thúc hoặc không kết thúc.
    - Luật sinh $S \rightarrow \lambda$ cũng như S không xuất hiện trong vé phải của bất kỳ luật sinh nào khác.
    - Văn phạm cảm ngữ cảnh

Các loại văn phạm cấu trúc đoạn

- Các loại văn phạm cấu trúc đoạn
  - Dựa theo các loại luật sinh
    - **Loại 2**: các luật sinh chỉ có dạng $w_1 \rightarrow w_2$
      - w_1 là ký hiệu đơn không phải ký hiệu kết thúc
      - Văn phạm phi ngữ cảnh
    - **Loại 3**: các luật sinh chỉ có dạng $w_1 \rightarrow w_2$
      - Với $w_1 = A$ và $w_2 = aB$ hoặc $w_2 = a$, trong đó A và B là các ký hiệu không kết thúc và a là ký hiệu kết thúc, hoặc với $w_1 = S$ và $w_2 = \lambda$.
      - Văn phạm chính quy

Các loại văn phạm cấu trúc đoạn

- Luật sinh không ràng buộc (noncontracting)
  - Dạng $w_1 \rightarrow w_2$ với độ dài của w_1 nhỏ hơn hoặc bằng độ dài của w_2
  - Văn phạm không ràng buộc (noncontracting) hoặc văn phạm đơn điệu (monotonic)

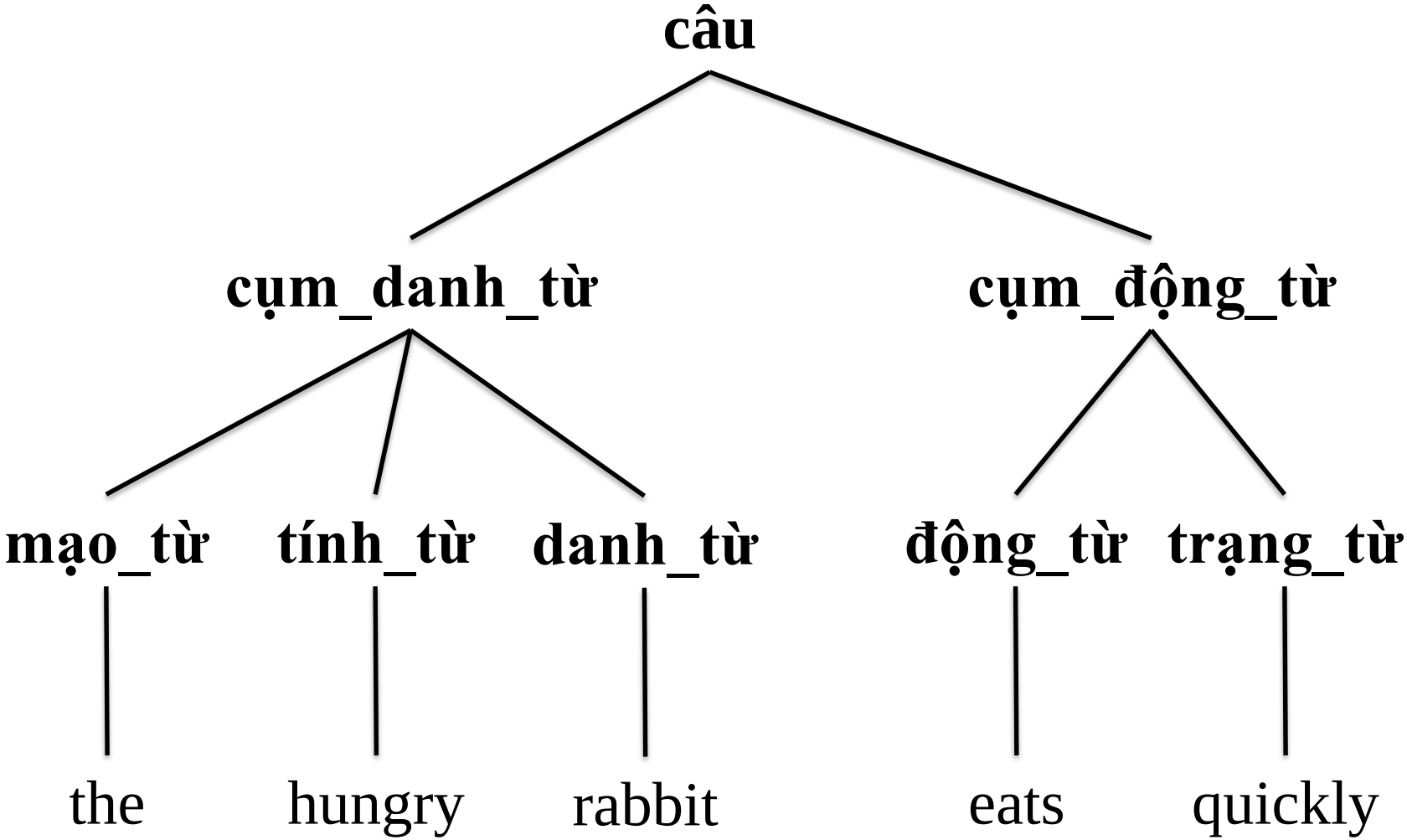
BẢNG 1. CÁC LOẠI VĂN PHẠM

Loại	Những hạn chế đối với các luật sinh $w_1 \rightarrow w_2$
0	Không có hạn chế nào
1	$w_1 = lAr$ và $w_2 = lwr$, trong đó $A \in N, l, r, w \in (N \cup T)^*$ và $w \neq \lambda$; hoặc $w_1 = S$ và $w_2 = \lambda$ cũng như S không xuất hiện trong vế phải của bất kỳ luật sinh nào khác
2	$w_1 = A$, trong đó A là ký hiệu không kết thúc
3	$w_1 = A$ và $w_2 = aB$ hoặc $w_2 = a$, trong đó $A \in N, B \in N$, và $a \in T$; hoặc $w_1 = S$ và $w_2 = \lambda$

Cây dẫn xuất

- Cây dẫn xuất (derivation tree), hoặc cây phân tích cú pháp (syntax parsing tree)
  - Biểu diễn một dẫn xuất trong ngôn ngữ được sinh bởi một văn phạm phi ngữ cảnh
    - **Gốc**: ký hiệu xuất phát.
    - **Các đỉnh trong**: các ký hiệu không kết thúc xuất hiện trong dẫn xuất.
    - **Các lá**: các ký hiệu kết thúc.

Cây dẫn xuất



HÌNH 1. CÂY DẪN XUẤT

Dạng Backus – Naur

- Được đặt theo tên của John Backus – người phát minh ra nó – và Peter Naur người đã cải tiến nó để dùng đặc tả ngôn ngữ lập trình ALGOL.
  - Đáng ngạc nhiên là, ký hiệu tương tự đã được sử dụng khoảng 2500 năm trước để mô tả ngữ pháp của tiếng Phạn.
- Đôi khi được sử dụng để xác định văn phạm loại 2
- Được sử dụng để xác định các quy tắc cú pháp của một số ngôn ngữ máy tính, bao gồm Java.

Dạng Backus – Naur

- Luật sinh của văn phạm loại 2
  - Gộp tất cả các luật sinh có cùng ký hiệu không kết thúc ở vế trái thành một mệnh đề.
  - Sử dụng ký hiệu ::=
  - Đưa tất cả các ký hiệu không kết thúc vào trong dấu ngoặc, ⟨ ⟩
  - Liệt kê tất cả vế phải của các luật sinh trong một mệnh đề, ngăn cách giữa chúng bởi một gạch đứng

Ví dụ: luật sinh $A \rightarrow Aa$, $A \rightarrow a$, và $A \rightarrow AB$ có thể kết hợp thành $\langle A \rangle ::= \langle A \rangle a | a | \langle A \rangle \langle B \rangle$

Automat hữu hạn

Máy hữu hạn trạng thái có đầu ra

- Máy hữu hạn trạng thái (Automat)
  - Có thể được sử dụng để mô hình hoá nhiều loại máy vật lý, bao gồm các linh kiện trong các máy tính.
  - Được sử dụng rộng rãi trong các ứng dụng trong khoa học máy tính và mạng dữ liệu
    - Kiểm tra chính tả, kiểm tra ngữ pháp, lập chỉ mục hoặc tìm kiếm nội dung của văn bản, nhận dạng giọng nói, chuyển đổi văn bản sử dụng các ngôn ngữ đánh dấu như XML và HTML, và các giao thức mạng xác định cách các máy tính giao tiếp
  - Có một trạng thái ban đầu được chỉ định, một bảng chữ cái đầu vào, và một hàm chuyển tiếp gán một trạng thái tiếp theo cho mỗi cặp trạng thái và đầu vào.

Máy hữu hạn trạng thái có đầu ra

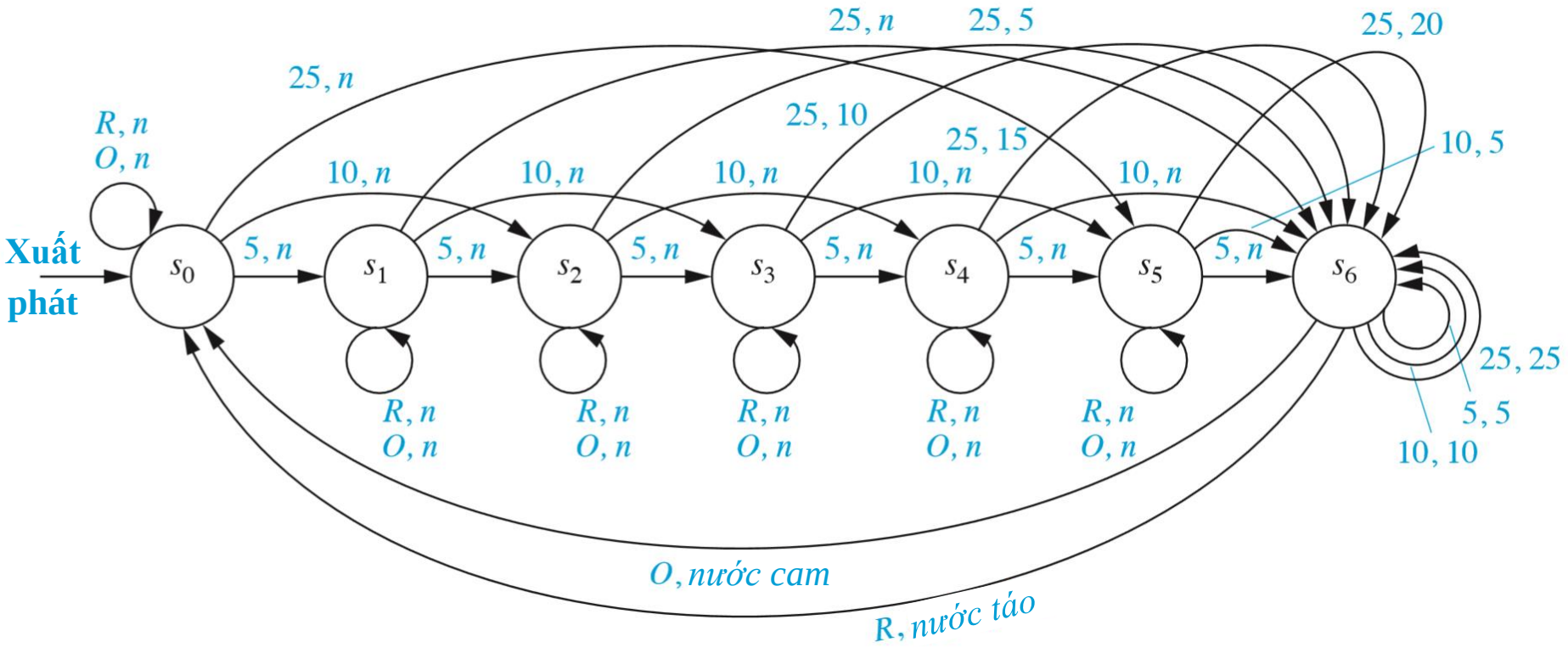
BẢNG 1. BẢNG TRẠNG THÁI CỦA MỘT MÁY BÁN HÀNG

	Trạng thái tiếp theo					Đầu ra				
Trạng thái	Đầu vào					Đầu vào				
	5	10	25	O	R	5	10	25	O	R
s_0	s_1	s_2	s_5	s_0	s_0	n	n	n	n	n
s_1	s_2	s_3	s_6	s_1	s_1	n	n	n	n	n
s_2	s_3	s_4	s_6	s_2	s_2	n	n	5	n	n
s_3	s_4	s_5	s_6	s_3	s_3	n	n	10	n	n
s_4	s_5	s_6	s_6	s_4	s_4	n	n	15	n	n
s_5	s_6	s_6	s_6	s_5	s_5	n	5	20	n	n
s_6	s_6	s_6	s_6	s_0	s_0	5	10	25	OJ	AJ

Máy hữu hạn trạng thái có đầu ra

Một ***máy hữu hạn trạng thái (finite-state machine)*** $M = (S, I, O, f, g, s_0)$ gồm một tập hữu hạn S ***các trạng thái***, một ***bộ chữ cái hữu hạn đầu vào*** I, một ***bộ chữ cái hữu hạn đầu ra*** O, một ***hàm chuyển*** f gán cho mỗi cặp gồm một trạng thái và một đầu vào một trạng thái mới, một ***hàm đầu ra*** g gán cho mỗi cặp gồm một trạng thái và một đầu vào một đầu ra, và một ***trạng thái ban đầu*** s_0.

Máy hữu hạn trạng thái có đầu ra



HÌNH 1. MÁY BÁN HÀNG TỰ ĐỘNG

Máy hữu hạn trạng thái có đầu ra

Với $M = (S, I, O, f, g, s_0)$ là một máy hữu hạn trạng thái và $L \subseteq I^*$. Ta gọi M ***chấp nhận*** L nếu một xâu đầu vào x thuộc L khi và chỉ khi bit đầu ra cuối cùng sinh ra bởi M với đầu vào x là bit 1.

Các loại máy hữu hạn trạng thái

- Nhiều loại máy hữu hạn trạng thái đã được phát triển để mô hình hoá các máy tính toán
  - Máy Mealy
    - Nghiên cứu đầu tiên bởi G. H. Mealy năm 1955
    - Đầu ra tương ứng với sự dịch chuyển giữa các trạng thái
  - Máy Moore
    - Giới thiệu bởi E. F. Moore năm 1956
    - Đầu ra chỉ được xác định bởi trạng thái

Máy hữu hạn trạng thái không có đầu ra

- Đoán nhận ngôn ngữ (language recognition)
  - Một trong những ứng dụng quan trọng nhất của máy hữu hạn trạng thái
  - Đóng vai trò cơ bản trong việc thiết kế và xây dựng các trình biên dịch cho các ngôn ngữ lập trình
  - Có một loại máy hữu hạn trạng thái khác được thiết kế chuyên để đoán nhận ngôn ngữ
    - Thay vì tạo ra đầu ra, các máy này có những trạng thái kết thúc. Một xâu được chấp nhận khi và chỉ khi nó đưa trạng thái xuất phát tới một trạng thái kết thúc

Tập các xâu

Cho A và B là hai tập con của V^*, với V là một từ vựng. ***Phép ghép (concaternation)*** của A và B, được ký hiệu bởi AB, là tập tất cả các xâu có dạng xy trong đó x là xâu thuộc A và y là xâu thuộc B.

Cho A là một tập con của V^*. Khi đó ***bao đóng Kleene (Kleene closure)*** của A – được ký hiệu là A^* – tập gồm các phép ghép một số tuỳ ý các xâu thuộc A. Điều này có nghĩa là $A^* = \cup_{k=0}^{\infty} A^k$

Tập các xâu

❑ Ví dụ:

Tìm bao đóng Kleene của các tập $A = \{0\}$, $B = \{0,1\}$ và $C = \{11\}$

❑ Lời giải:

$A^* = \{0^n \mid n \in \mathbb{N}\}$

$B^* = V^*$ là tập tất cả các xâu nhị phân

$C^* = \{1^{2n} \mid n \in \mathbb{N}\}$

Automat hữu hạn trạng thái

Một ***automat hữu hạn (finite-state automat)*** $M = (S, I, f, s_0, F)$ gồm một tập hữu hạn S các ***trạng thái***, một ***bộ chữ cái đầu vào*** I, một ***hàm chuyển (transition function)*** f gán trạng thái tiếp theo cho mỗi cặp trạng thái đầu vào, ***trạng thái xuất phát*** s_0 và một tập con F của S gồm các ***trạng thái kết thúc***.

Mở rộng của hàm chuyển

- Một máy hữu hạn trạng thái $M = (S, I, f, s_0, F)$
  - Hàm chuyển f có thể được mở rộng thành hàm:

$$f: S \times I^* \rightarrow S$$

    - Có thể định nghĩa hàm chuyển mở rộng f bằng đệ quy

(*i*) $f(s, \lambda) = s$ cho tất cả các trạng thái $s \in S$; và

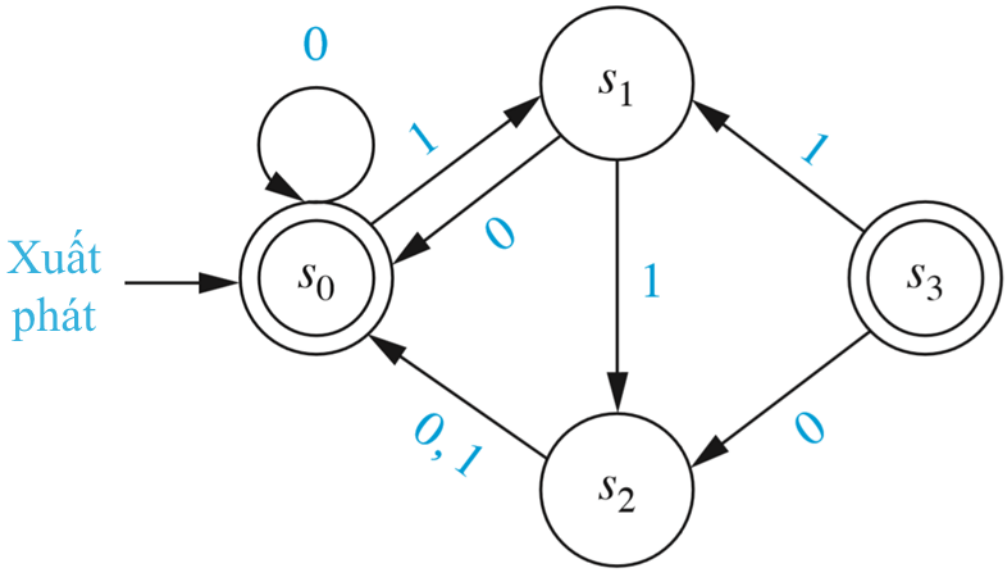
(*ii*) $f(s, xa) = f(f(s, x), a)$ với mọi $s \in S$, $x \in I^*$, và $a \in I$.

Automat hữu hạn trạng thái

❑ Ví dụ

Dựng giản đồ trạng thái của một automat hữu hạn $M = (S, I, f, s_0, F)$ với $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0,1\}$, $F = \{s_0, s_3\}$ và hàm chuyển f được cho bởi bảng:

Trạng thái	f	
	Đầu vào	
	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1



Đoán nhận ngôn ngữ bởi máy hữu hạn trạng thái

Một xâu x được gọi là ***được chấp nhận (recognize, accepte)*** bởi máy $M = (S, I, f, s_0, F)$ nếu nó đưa trạng thái xuất phát tới một trạng thái kết thúc, tức là $f(s_0, x)$ là một trạng thái thuộc F. ***Ngôn ngữ được chấp nhận*** bởi máy M, được ký hiệu là $L(M)$, là tập tất cả các xâu được chấp nhận bởi M. Hai automat hữu hạn được gọi là ***tương đương***, nếu chúng cùng chấp nhận một ngôn ngữ.

Automat hữu hạn không tất định

- Các automat đã được xem xét đều là các automat ***tất định (deterministic)***
  - Mỗi cặp trạng thái và giá trị đầu vào có một trạng thái tiếp theo duy nhất được cho bởi hàm chuyển.
- Automat ***không tất định (nondeterministic)***
  - Có thể có nhiều trạng thái tiếp theo
  - Quan trọng trong việc xác định ngôn ngữ nào có thể đoán nhận được bởi automat hữu hạn

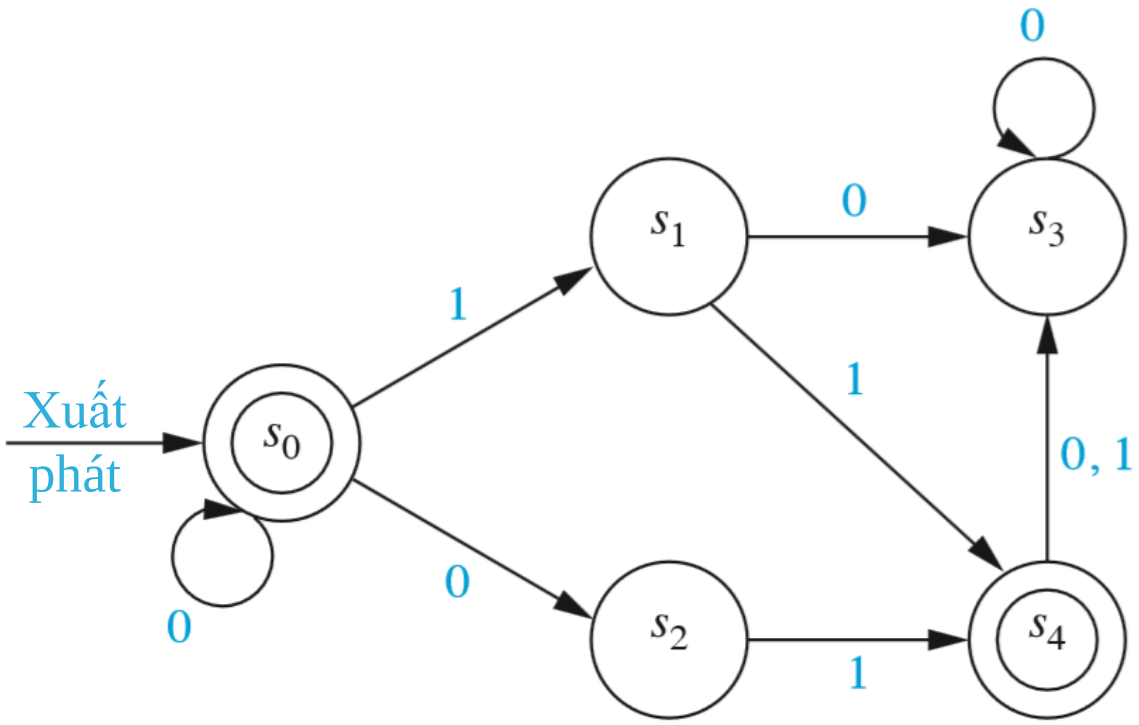
Automat hữu hạn không tất định

Automat hữu hạn không tất định $M = (S, I, f, s_0, F)$ gồm tập S các trạng thái, một bộ chữ cái I, môt hàm chuyển f gán cho mỗi cặp trạng thái và đầu vào một tập các trạng thái, trạng thái xuất phát s_0 và tập con F của S gồm các trạng thái kết thúc.

Automat hữu hạn không tất định

❑ Ví dụ

Lập bảng trạng thái cho automat hữu hạn không tất định có giản đồ trạng thái như hình dưới:



Trạng thái	f	
	Đầu vào	
	0	1
s_0	s_0, s_2	s_1
s_1	s_3	s_4
s_2		s_4
s_3	s_3	
s_4	s_3	s_3

Automat hữu hạn không tất định

Định lý 1: Nếu ngôn ngữ L được chấp nhận bởi một automat hữu hạn không tất định M_0 thì L cũng được chấp nhận bởi một automat tất định M_1

- Chứng minh: trong giáo trình

Automat hữu hạn không tất định

❑ Ví dụ

Tìm một automat hữu hạn tất định chấp nhận cùng một ngôn ngữ như automat hữu hạn không tất định trong ví dụ trước.