CHƯƠNG II
Phân tích từ vựng

Mục tiêu: Nắm được vai trò của giai đoạn phân tích từ vựng, sử dụng các khái niệm biểu thức chính qui (regular expression) và ô- tô- mát hứu hạn (finite automata) trong việc biểu diễn và nhận biết ngôn ngữ

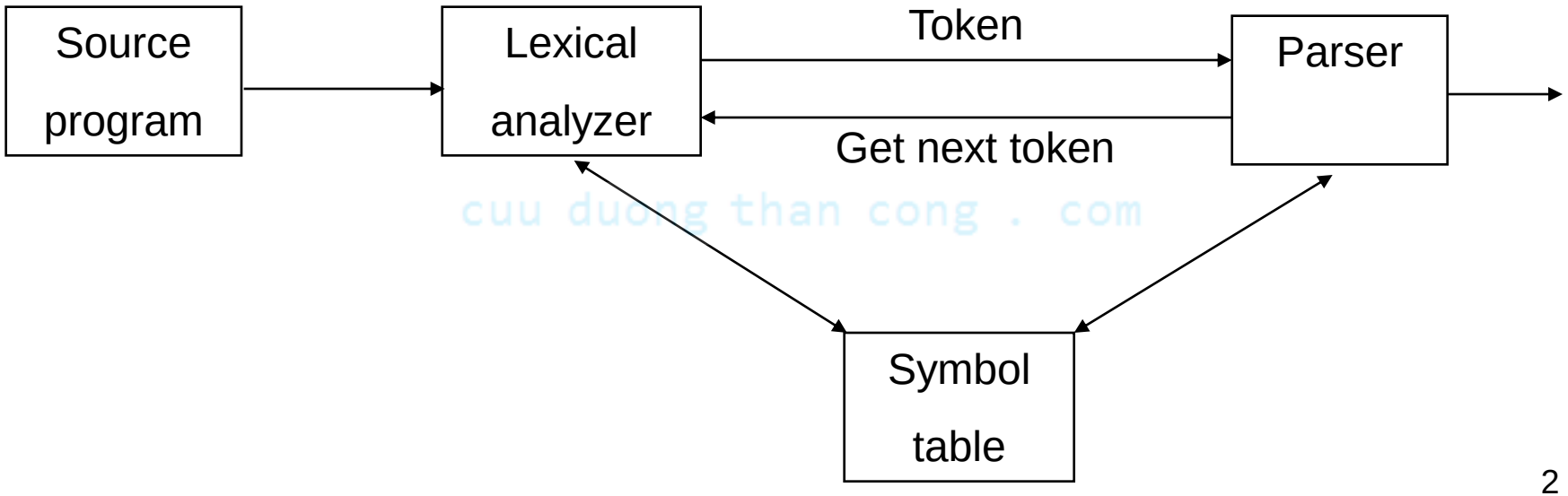
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Vai trò của phân tích từ vựng

- Đây là giai đoạn đầu tiên của quá trình biên dịch
- Nhiệm vụ chính: Đọc từng kí tự vào (input characters) từ chương trình nguồn và nhóm lại thành các token phục vụ cho giai đoạn phân tích cú pháp sau đó

cuu duong than cong . com



cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Phân tích từ vựng giúp cho các giai đoạn biên dịch tiếp theo dễ dàng hơn, ví dụ: Giai đoạn phân tích cú pháp không phải quan tâm đến các khoảng trắng cũng như các lời chú trích vì nó đã được loại bỏ khi khi phân tích từ vựng
- Giảm đáng kể thời gian đọc chương trình nguồn và nhóm thành các token nhờ một số chương trình xử lí chuyên dụng

cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Một số khái niệm

- Token: Một token là một tập hợp các xâu kí tự có một nghĩa xác định, ví dụ identifier token là tập hợp tất cả các identifier. Token chính là các kí hiệu kết thúc (terminal) trong định nghĩa văn phạm của một ngôn ngữ, ví dụ: Các từ khoá, định danh, toán tử, hằng, xâu kí tự, dấu ngoặc đơn, dấu phẩy, dấu chấm phẩy...
- Pattern: Pattern của một token là các qui tắc kết hợp các kí tự để tạo lên token đó
- Lexeme: Là một chuỗi các kí tự thoả mãn pattern của một token

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Bảng sau đưa ra các ví dụ về token, pattern và lexeme

Token	Lexeme	Thông tin mô tả của pattern
const	const	const
if	if	if
relation	<, <=, >, >=, =, <>	< hoặc <= hoặc > hoặc >= hoặc = hoặc <>
id	pi, count, d2	một kí tự, tiếp theo là các kí tự hoặc chữ số
num	3.1416, 0, 6.02E2	bất kì hằng số nào
literal	"computer"	các kí tự nằm giữa " và " ngoại trừ "

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Đặc tả Token

- Xâu kí tự (string): Là một chuỗi các kí tự từ một bảng chữ cái. Kí hiệu xâu rỗng là ε
- Một số khái niệm liên quan đến xâu kí tự: Tiền tố (prefix), hậu tố (suffix), xâu con (substring), tiền tố thực sự (proper prefix)....
- Ngôn ngữ (language): Là tập hợp các xâu kí tự được xây dựng từ một bảng chữ cái

cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Các phép toán trên ngôn ngữ: Giả sử L và M là hai ngôn ngữ khi đó

Hợp (union)của L và M: $L \cup M = \{s | s \in L \text{ hoặc } s \in M\}$

Ghép (concatenation) của L và M: $LM = \{ st \mid s \in L \text{ và } t \in M \}$

Bao đóng Kleen (kleene closure) L: $L^* = \cup_{i=0}^{\infty} L^i$

(Ghép của 0 hoặc nhiều L)

Bao đóng dương (positive closure) của L: $L^+ = \cup_{i=1}^{\infty} L^i$

(Ghép của 1 hoặc nhiều L)

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: L = {A, B, ..., Z, a, b, ..., z } và

D = { 0, 1, , ..., 9 }

1. $L \cup D$ là tập hợp các chữ cái và chữ số
2. LD là tập hợp các xâu bao gồm một chữ cái và một chữ số
3. L^4 là tập hợp tất cả các xâu 4 chữ cái
4. L * là tập hợp tất cả các xâu của các chữ cái bao gồm cả chuỗi rỗng
5. $L(L \cup D)^*$ là tập hợp tất cả các xâu mở đầu bằng một chữ cái theo sau là chữ cái hay chữ số
6. D^+ là tập hợp tất cả các xâu gồm một hoặc nhiều chữ số

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Biểu thức chính qui (regular expression)

- Mỗi biểu thức chính qui (BTCQ) r dùng để đặc tả một ngôn ngữ L(r).

  Ví dụ: Trong pascal mọi identifier được đặc tả bởi biểu thức chính qui letter(letter|digit)*

- Hai BTCQ r và s được gọi là tương đương (kí hiệu r=s) nếu chúng đặc tả chung một ngôn ngữ

  Ví dụ: (a|b)=(b|a)

- Một BTCQ được xây dựng từ những BTCQ đơn giản bằng cách sử dụng các luật

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Sau đây là các luật định nghĩa các BTCQ trên một bảng chữ cái Σ

  1. ε là một BTCQ đặc tả $\{\varepsilon\}$ (tập hợp chứa xâu rỗng)

  2. Nếu a $\in\Sigma$ thì a là BTCQ đặc tả {a}

  3. Giả sử r và s là các BTCQ đặc tả các ngôn ngữ L(r) và L(s), khi đó:

     a) (r)|(s) là một BTCQ đặc tả $L(r)\cup L(s)$

     b) (r)(s) là một BTCQ đặc tả L(r)L(s)

     c) $(r)^*$ là một BTCQ đặc tả $(L(r))^*$

     d) (r) là một BTCQ đặc tả L(r)

cuu duong than cong . com
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: Cho $\Sigma = \{ a, b \}$.

1. BTCQ a | b đặc tả {a, b}

2. BTCQ (a | b) (a | b) đặc tả {aa, ab, ba, bb}.Tập hợp này có thể được đặc tả bởi BTCQ tương đương sau: aa | ab | ba | bb

cuu duong than cong . com

3. BTCQ a* đặc tả {ε, a, aa, aaa, ... }

4. BTCQ (a | b)* đặc tả {a, b, aa,bb, ...}. Tập hợp này có thể đặc tả bởi (a*b*)*

cuu duong than cong . com

5. BTCQ a | a* b đặc tả {a, b, ab, aab,... }

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Định nghĩa chính qui (regular definition)

- Để thuận tiện về mặt kí hiệu, ta dùng định nghĩa chính qui (ĐNCQ) để đặt tên cho các BTCQ
- Một ĐNCQ là một dãy các định nghĩa có dạng

cuu duong than cong . com

$d_1 \rightarrow r_1$

$d_2 \rightarrow r_2$

.....

$d_n \rightarrow r_n$

cuu duong than cong . com

Trong đó d_i là các tên, r_i là các BTCQ trên tập các kí hiệu $\Sigma \cup \{d_1, d_2, ...d_{i-1}\}$

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: ĐNCQ của các định danh trong pascal là

letter $\rightarrow$ A | B | ...| Z | a | b |...| z

digit $\rightarrow$ 0 | 1 | ...| 9

id $\rightarrow$ letter (letter | digit)*

Ví dụ: ĐNCQ của các số không dấu trong pascal như 3254, 23.243E5,16.264E-3... là

digit $\rightarrow$ 0 | 1 |...| 9

digits $\rightarrow$ digit digit*

optional_fraction $\rightarrow$. digits | e

optional_exponent $\rightarrow$ (E (+ | - | e) digits) | e

num $\rightarrow$ digits optional_fraction optional_exponent

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Các kí hiệu tắt được sử dụng trong các BTCQ

  +: Một hoặc nhiều

  ?: Không hoặc một

- Ví dụ: ĐNCQ cho tập số không dấu được viết lại như sau

  digit $\rightarrow$ 0 | 1 |...| 9

  digits $\rightarrow$ digit $^+$

  optional_fraction $\rightarrow$ (. digits) ?

  optional_exponent $\rightarrow$ (E (+ | -) ? digits) ?

  num $\rightarrow$ digits optional_fraction optional_exponent

cuu duong than cong . com

- Sử dụng các tập kí tự [abc]=a | b | c, [a-z]=a | b |..z ta có thể đặc tả các định danh bởi BTCQ

  [A - Z a - z] [A - Z a - z 0 - 9]*

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Nhận dạng token

- Làm thế nào để nhận dạng được token? Ta sẽ xét 1 ví dụ mang tính chất minh hoạ

Ví dụ: Cho ngữ pháp (grammar)

stmt → **if** *expr* **then** *stmt*
| **if** *expr* **then** *stmt* **else** *stmt*
| ε

expr → *term* **relop** *term*
| *term*

term → **id**
| **num**

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Trong đó các kí hiệu kết thúc (token) if, then, else, relop, id sinh ra tập các xâu kí tự theo các ĐNCQ sau:

  if $\rightarrow$ if

  then $\rightarrow$ then

  else $\rightarrow$ else

  relop $\rightarrow$ < | <= | = | <> | > | >=

  id $\rightarrow$ **letter** (**letter** | **digit**) *

  num $\rightarrow$ **digit** $^+$ (**.** **digit** +) ? (**E** (**+** | **-**) ? **digit** $^+$) ?

- Các kí tự khoảng trắng (loại bỏ khi phân tích từ vựng) được định nghĩa bởi ĐNCQ sau:

  delim $\rightarrow$ **blank** | **tab** | **newline**

  ws $\rightarrow$ **delim**$^+$

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Mục đích của lexical analyzer tạo ra output là các cặp <token, attribute-value>

Regular Expression	Token	Attribute-value
ws	-	-
if	**if**	-
then	**then**	-
else	**else**	-
id	**id**	pointer to table entry
num	**num**	pointer to table entry
<	**relop**	LT
<=	**relop**	LE
=	**relop**	EQ
<>	**relop**	NE
>	**relop**	GT
>=	**relop**	GE

cuu duong than cong . com

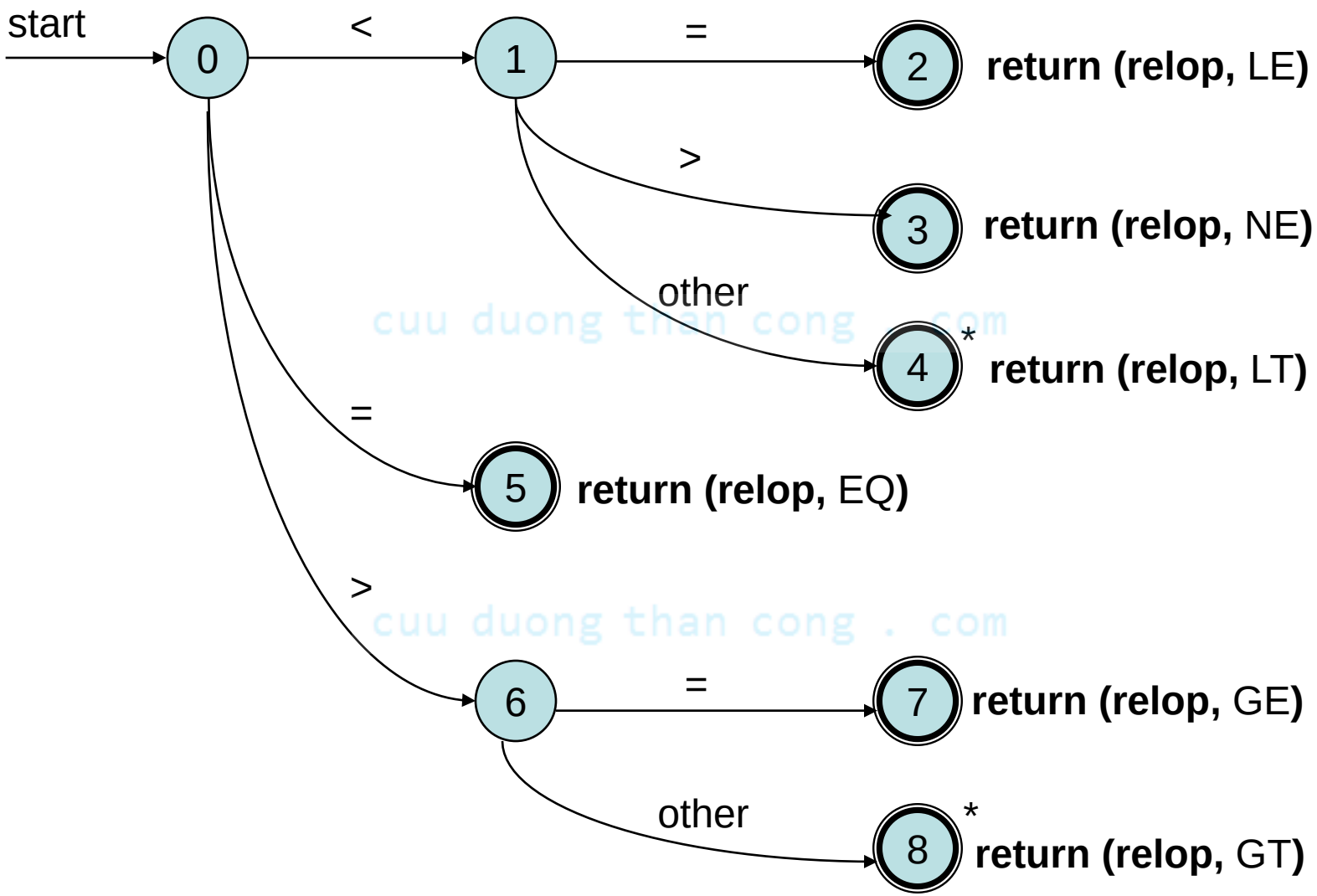
CuuDuongThanCong.com https://fb.com/tailieudientucntt

- Sơ đồ chuyển tiếp (transition diagram): Là bước trung gian minh hoạ tiến trình chuyển đổi trạng thái khi bộ phân tích từ vựng đọc lần lượt từng kí tự
- Ta phải xây dựng các sơ đồ chuyển tiếp để nhận biết từng loại token
- Một sơ đồ chuyển tiếp bao gồm các trạng thái (states) được vẽ bằng hình tròn, có 1 trạng thái bắt đầu (start state). Các trạng thái được nối với nhau bởi các mũi tên ta gọi là các cạnh (edges)
- Trạng thái được biểu diễn bởi vòng tròn kép là trạng thái được chấp nhận (accepting state) thông báo 1 token đã được nhận dạng

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

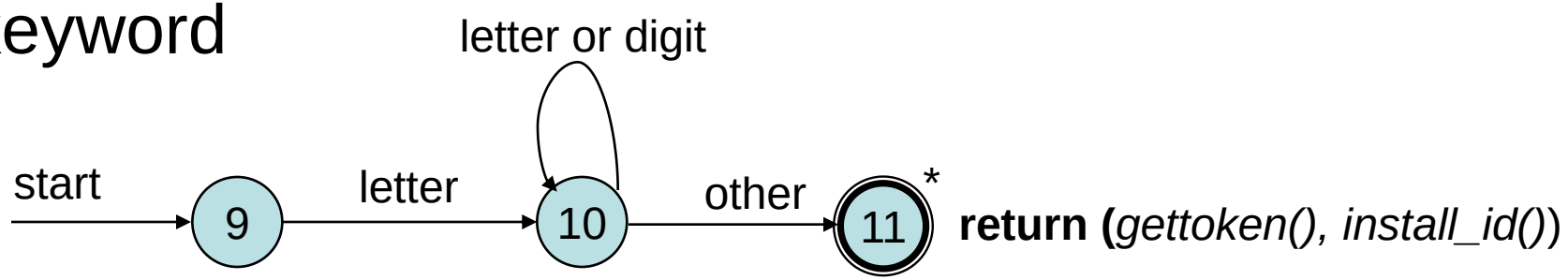
Ví dụ: Sơ đồ chuyển tiếp cho token các toán tử quan hệ **relop**



cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: Sơ đồ chuyển tiếp cho token các identifier và keyword



- Chú ý:

  - Các từ khoá là các từ được bảo vệ và được lưu trữ sẵn trong symbol-table

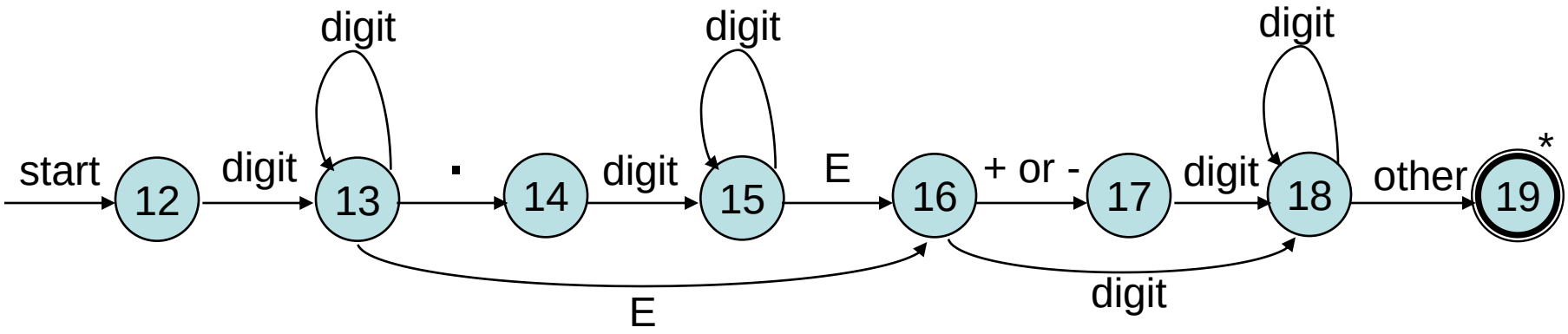
  - Thủ tục *gettoken()* tra cứu lexeme trong symbol-table nếu là 1 keyword thì token tương ứng được trả về còn ngược lại token **id** được trả về

  - Thủ tục *install_id()* tra cứu lexeme trong symbol-table nếu là 1 keyword thì trả lại giá trị 0, nếu là một biến đã có thì trả lại một con trỏ tới vị trí trong symbol-table. Nếu lexeme không có thì tạo một phần tử mới trong symbol-table và trả về con trỏ tới thành phần mới vừa được tạo

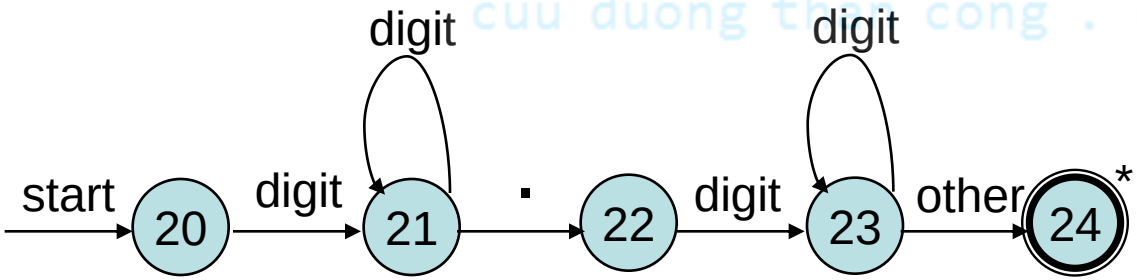
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

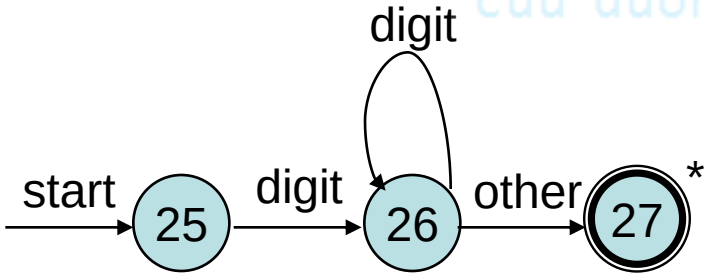
Ví dụ: Sơ đồ chuyển tiếp cho token các unsigned numbers trong pascal



cuu duong than cong . com

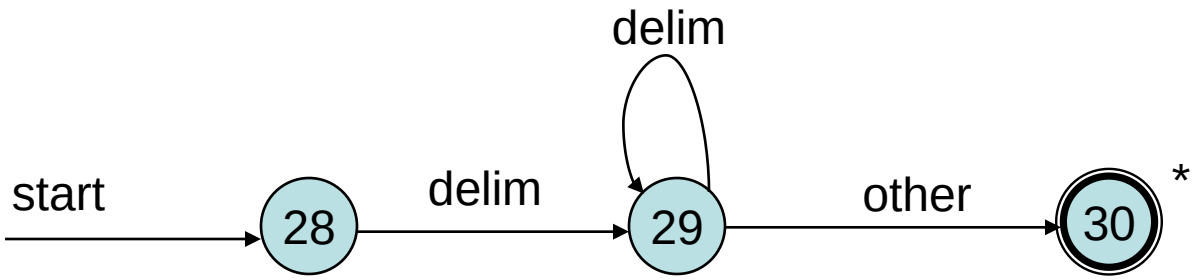


cuu duong than cong . com



CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: Sơ đồ chuyển tiếp cho token các ws



cuu duong than cong . com

cuu duong than cong . com

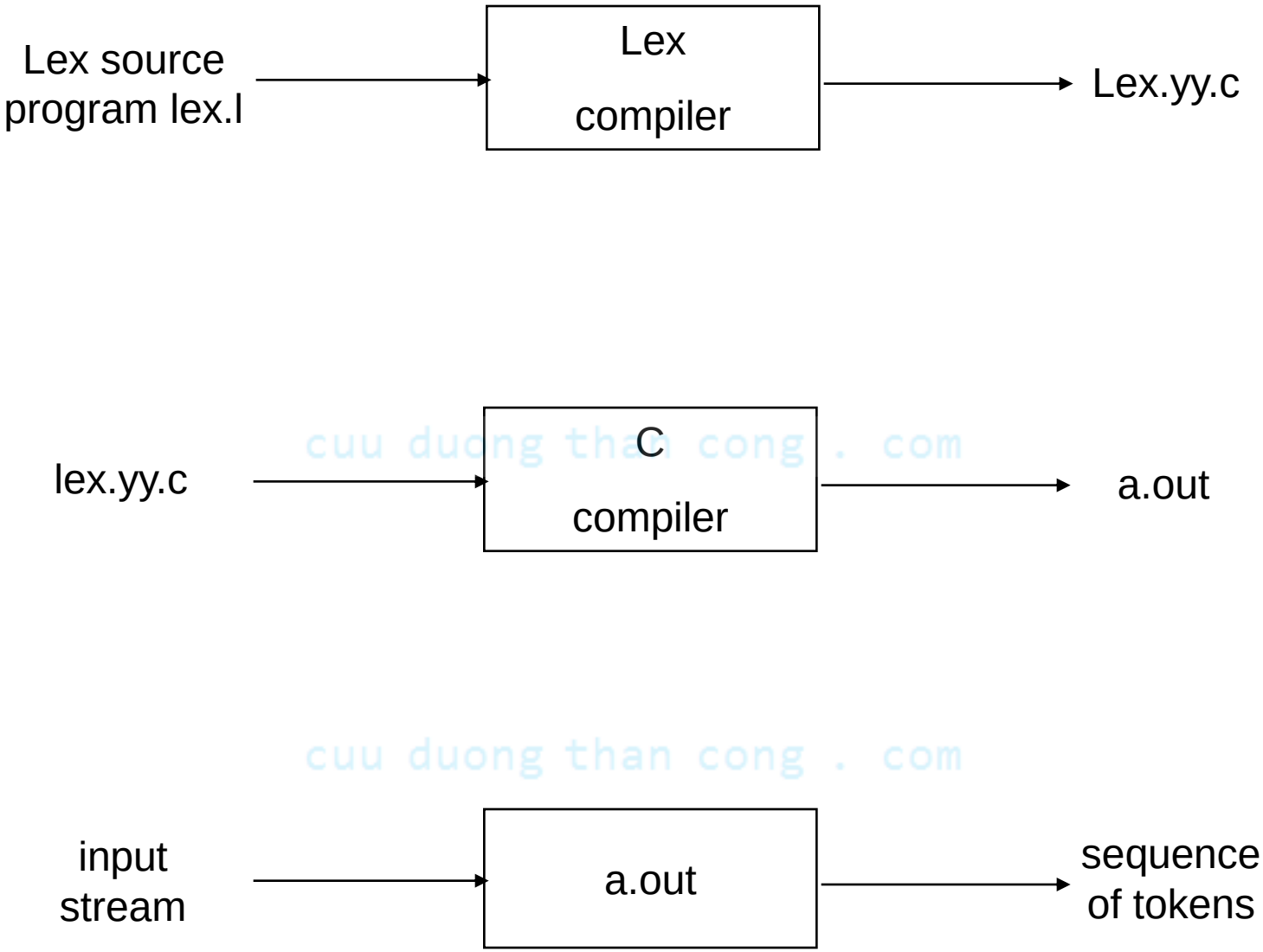
CuuDuongThanCong.com https://fb.com/tailieudientucntt

Công cụ phân tích từ vựng Lex

- Một số công cụ có sẵn cho phép xây dựng một bộ phân tích từ vựng dựa trên các biểu thức chính qui
- Một trong số những công cụ đó là Lex compiler (một công cụ có sẵn của Unix)
- Sơ đồ sau chỉ rõ cách tạo một bộ phân tích từ vựng bằng cách sử dụng Lex

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt



cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ô- tô- mát hữu hạn (finite automata)

- Một bộ nhận dạng ngôn ngữ (recognizer for a language) là một chương trình nhận đầu vào là một xâu kí tự x, trả lời "Yes" nếu x thuộc ngôn ngữ và trả lời "No" nếu ngược lại
- Ô- tô- mát hữu hạn là một sơ đồ chuyển tiếp được khái quát hoá, đóng vai trò là recognizer cho các biểu thức chính qui
- Một Ô- tô- mát hữu hạn có thể là deterministic finite automata (DFA) hoặc nondeterministic finite automata (NFA)

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

- "Nondeterministic" nghĩa là khi một kí tự được đọc vào thì sơ đồ chuyển tiếp có thể chuyển đến nhiều hơn một trạng thái tiếp theo
- Cả hai DFA và NFA đều có khả năng nhận dạng các biểu thức chính qui
- DFA có thể nhận dạng nhanh hơn nhưng cũng có kích thước lớn hơn NFA tương đương

cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

NFA

- Một NFA là một mô hình toán học bao gồm:

  - Một tập hợp trạng thái S

  - Một trạng thái bắt đầu $s_0 \in S$

  - Một tập hợp các trạng thái chấp nhận (hoặc trạng thái kết thúc) $F \subseteq S$

  - Một tập hợp kí tự vào của bảng chữ cái Σ

  - Một hàm chuyển đổi trạng thái move: $S \times (\Sigma \cup \{\varepsilon\}) \rightarrow S$

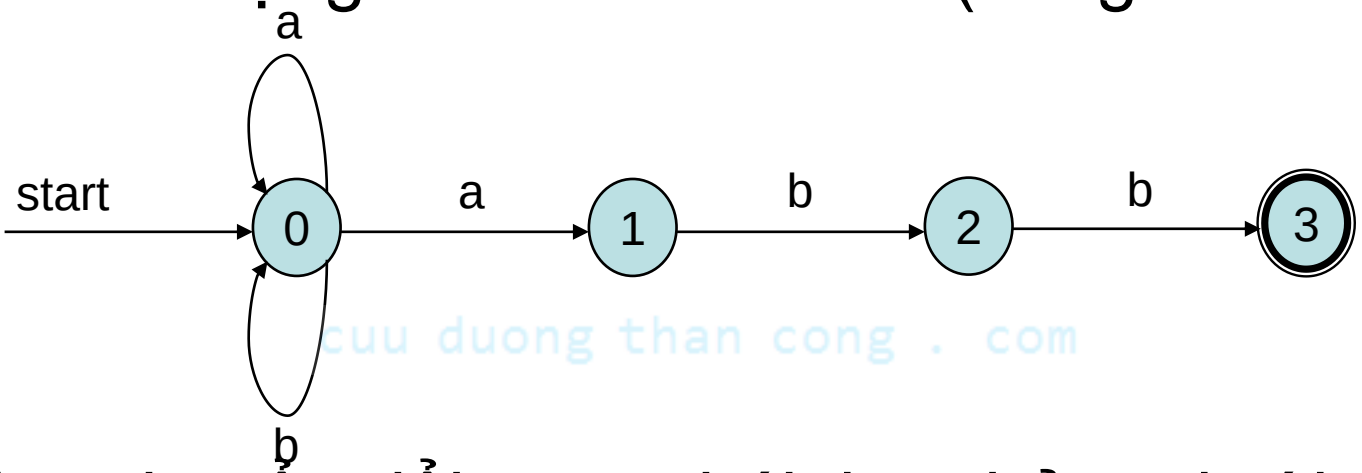
  (Một phép chuyển đổi $(s_k, \varepsilon) \rightarrow s_j$ nghĩa là chuyển từ s_k sang s_j mặc dù không có kí tự nào được đọc vào)

- NFA được biểu diễn trực tiếp bằng sơ đồ chuyển tiếp

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: NFA nhận biết BTCQ $(a|b)^*abb$ được mô tả dưới đây trong đó tập S={0, 1, 2, 3}, Σ={a, b}, s_0=0 và trạng thái kết thúc là 3 (vòng tròn kép)



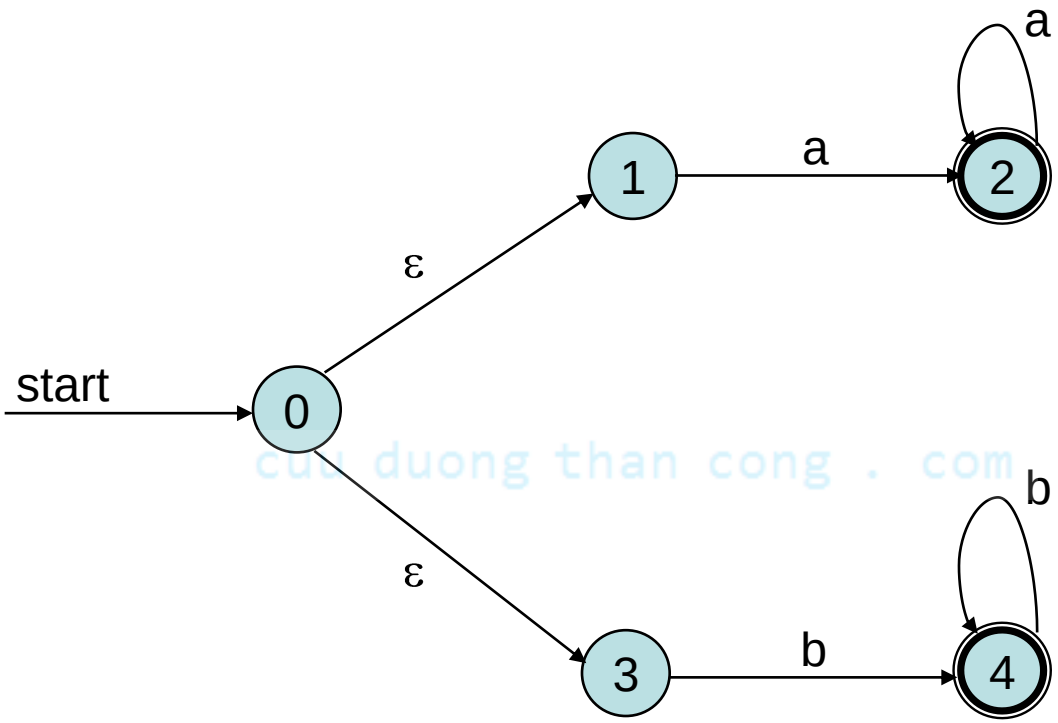
cuu duong than cong . com

- Hàm chuyển đổi trạng thái theo bảng dưới đây

Trạng thái	Kí tự vào	
	a	b
0	{0, 1}	{0}
1	-	{2}
2	-	{3}

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: NFA nhận biết biểu thức chính qui $aa^* \mid bb^*$



cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

DFA

- DFA là một trường hợp đặc biệt của NFA, DFA có thêm các dặc diểm sau:

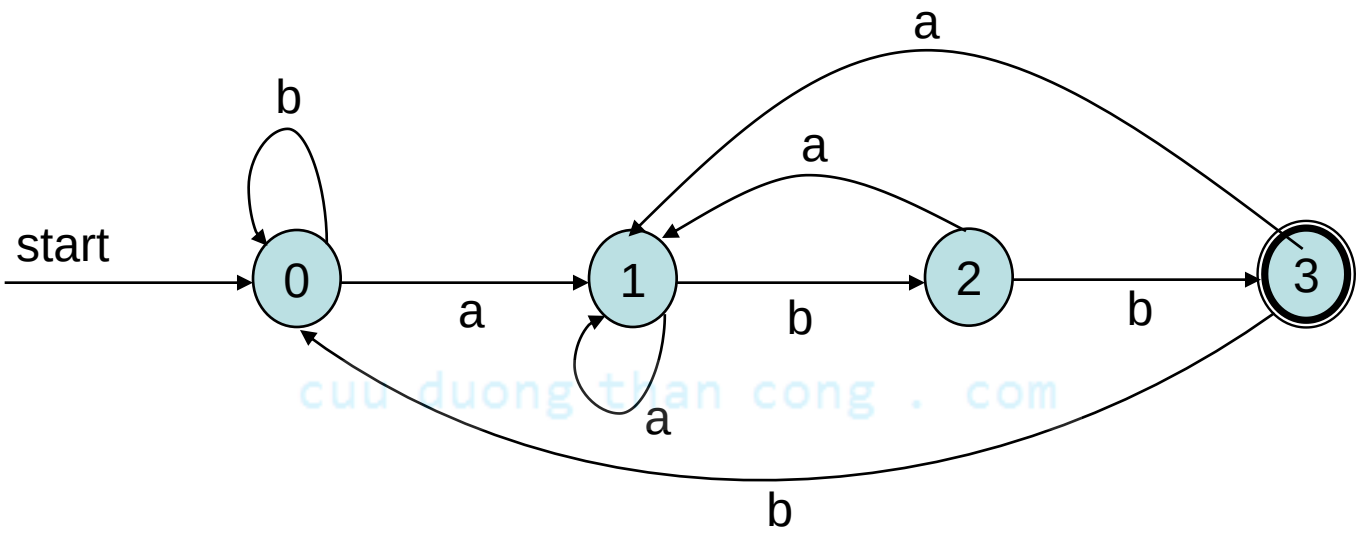
  - Không có chuyển đổi trạng thái ứng với kí tự rỗng ε

  - Từ một trạng thái s khi có một kí tự x được đọc vào, DFA sẽ chuyển sang một trạng thái s' duy nhất

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Ví dụ: DFA nhận biết BTCQ $(a|b)^*abb$



cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt