CHƯƠNG TRÌNH DỊCH

Bài 3: Phân tích từ vựng

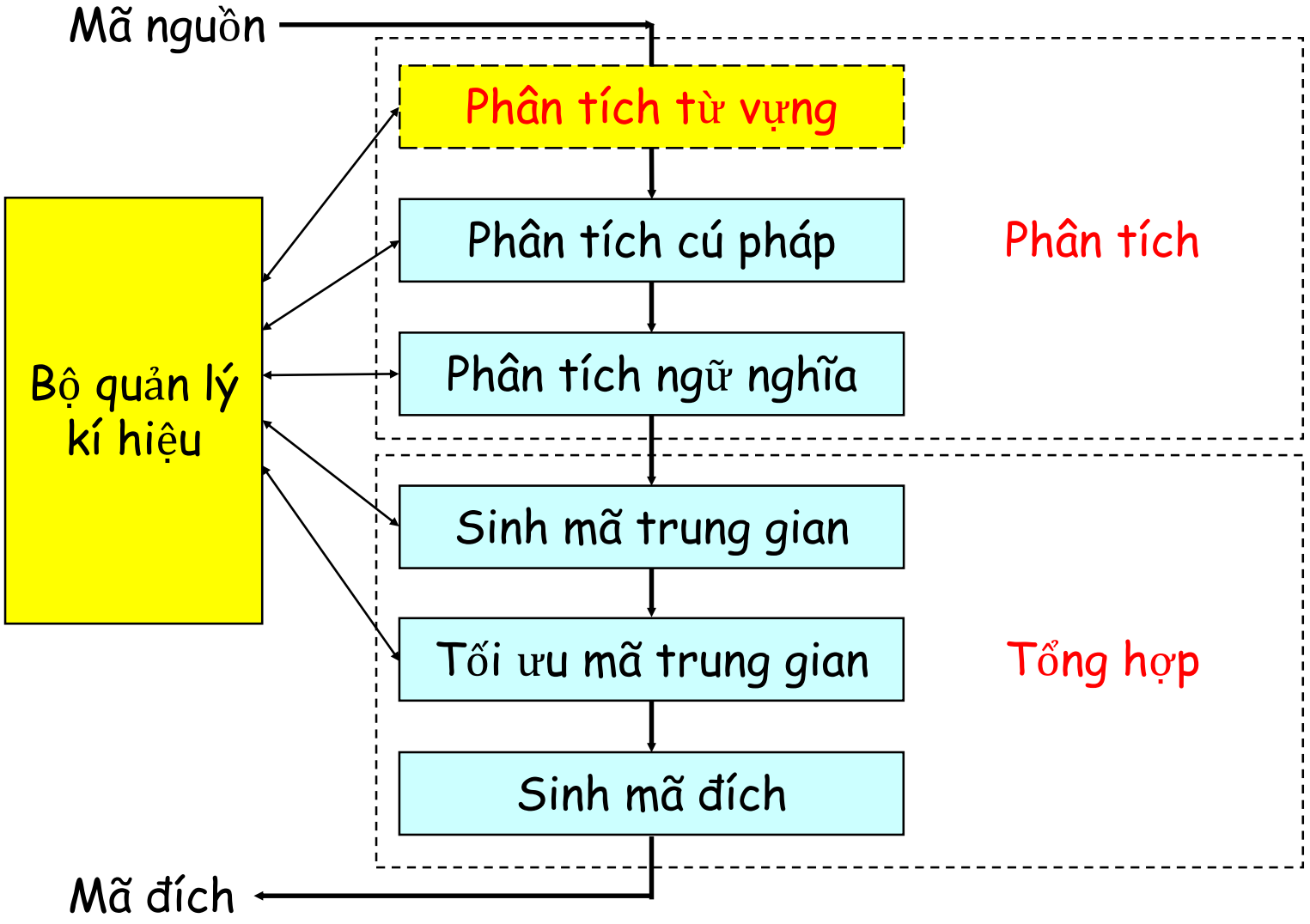
Nội dung

1. Vai trò của bộ phân tích từ vựng
2. Nhiệm vụ của phân tích từ vựng
3. Các mục tiêu của phân tích từ vựng
4. Đầu vào và đầu ra của phân tích từ vựng
5. Các bước xây dựng bộ phân tích từ vựng
6. Biểu diễn từ vựng bằng biểu thức chính quy
7. Lỗi và ngoại lệ khi phân tích từ vựng
8. Phân tích từ vựng cho một ngôn ngữ đơn giản
9. Bài tập và thảo luận

Phần 1

Vai trò của bộ phân tích từ vựng (PTTV)

Cấu trúc một chương trình dịch



Vai trò của bộ phân tích từ vựng

```
program gcd (input, output);
var i, j : integer;
begin
  read (i, j);
  while i <> j do
    if i > j then i := i - j else j := j - i;
  writeln (i)
end.
```

program	gcd	(	input	,	output	)	;
var	i	,	j	:	integer	;	begin
read	(	i	,	j	)	;	while
i	<>	j	do	if	i	>	j
then	i	:=	i	-	j	else	j
:=	i	-	i	;	writeln	(	i
)	end	.					

Vai trò của bộ phân tích từ vựng

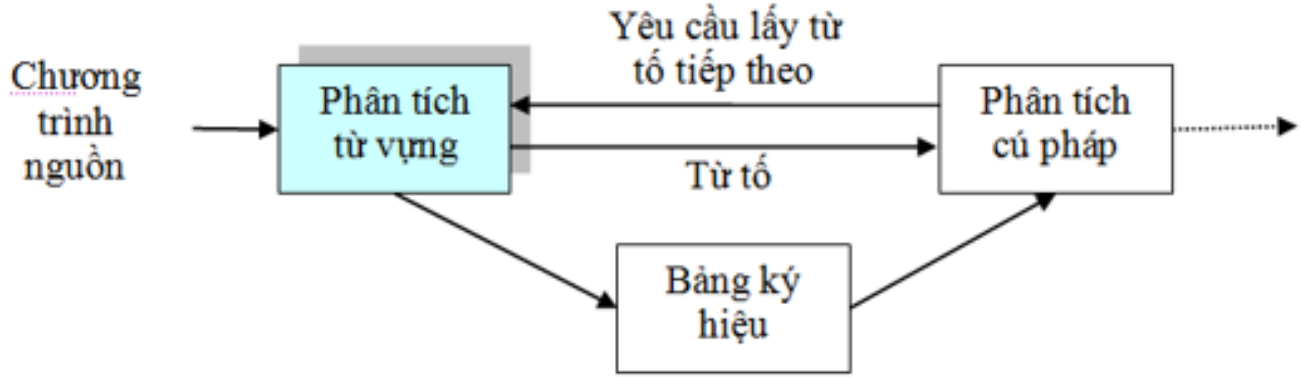
- Phân tích từ vựng (lexical analysis) là bước đầu tiên của trình dịch
  - Còn gọi là scanning hoặc lexing, bộ phân tích từ vựng là scanner hoặc lexer
- Khối phân tích từ vựng (PTTV):
  - Nhận dữ liệu đầu vào là mã nguồn cần dịch
  - Loại bỏ các đoạn mã không cần thiết
  - Chia đoạn mã còn lại thành dãy các từ tố (token)
  - Chuyển kết quả cho khối phân tích cú pháp (PTCP)
- Tương tác giữa PTTV và PTCP như thế nào?

Vai trò của bộ phân tích từ vựng

- Có nhiều quan điểm về sự tương tác giữa bộ PTTV và bộ phân tích cú pháp
  - **Thiết kế cổ điển**: coi PTTV như một tiến trình song song và phụ thuộc vào bộ phân tích cú pháp, quá trình phân tích cú pháp điều khiển việc phân tích từ vựng
  - **Thiết kế hiện đại**: tách PTTV thành một module độc lập, kết quả đầu ra của PTTV được tiêu chuẩn hóa để có thể được ghi ra file hoặc sử dụng bởi các mục đích khác
- Chú ý: việc chọn cách thiết kế là do mục tiêu xây dựng chương trình, không có nghĩa là thiết kế hiện đại thì tốt hơn thiết kế cổ điển

Vai trò của bộ phân tích từ vựng

- Trong thiết kế cổ điển, PTTV đóng vai trò như bộ cung cấp dữ liệu cho bộ phân tích cú pháp
  - Bộ phân tích cú pháp yêu cầu PTTV lấy từ tố tiếp theo
  - Bộ PTTV đọc chương trình nguồn từ đầu hoặc từ vị trí đang phân tích trong lần gọi trước, tách lấy từ tố tiếp theo trả lại cho bộ phân tích cú pháp
  - Quá trình lặp lại cho đến khi hết mã nguồn hoặc gặp lỗi



Vai trò của bộ phân tích từ vựng

- Trong các thiết kế mới hơn, bộ PTTV có xu hướng đứng tách ra độc lập, việc này có nhiều lợi ích:
  - Thiết kế theo hướng module hóa, đơn giản hơn
  - Tăng hiệu quả hoạt động của bộ PTTV, chẳng hạn như PTTV có thể độc lập xử lý các macro, xử lý khoảng trắng, ghi chú,…
  - Tối ưu hoạt động của trình dịch, bộ PTTV sau khi hoạt động có thể giải phóng các tài nguyên mà nó sử dụng thay vì giữ lại cùng lúc với bộ phân tích cú pháp
  - Xử lý được ngay lập tức một số lỗi cơ bản về từ vựng mà không cần phân tích cú pháp

Phần 2

Nhiệm vụ của phân tích từ vựng

Nhiệm vụ của phân tích từ vựng

- PTTV đóng vai trò như một bộ chuẩn hóa dữ liệu đầu vào, ngoài ra nó cũng giúp hạn chế các lỗi cơ bản (viết sai luật, sai từ khóa, sai cấu trúc,…)
- Các nhiệm vụ chính (nhất thiết phải có để đảm bảo hoạt động của chương trình dịch):
  - Đọc chương trình nguồn, loại bỏ các kí hiệu vô ích (khoảng trắng, dấu tab, xuống dòng, ghi chú,…)
  - Phát hiện một số lỗi cơ bản về từ vựng
  - Xác định nội dung của từ vựng
  - Xác định từ loại của từ vựng đó
  - Đưa ra một số thông tin thuộc tính của từ vựng

Nhiệm vụ của phân tích từ vựng

- Để hỗ trợ cho việc báo lỗi nếu có, PTTV còn ghi nhận lại các thông tin ngữ cảnh để giúp thông báo lỗi trực quan hơn (chẳng hạn như ghi lại số dòng số cột của từ vựng, giúp báo lỗi chính xác hơn)
- Bộ PTTV trong nhiều thiết kế còn thực hiện các công việc hỗ trợ cho bộ soạn thảo mã nguồn
  - Hỗ trợ các hàm tiền xử lý (các macro văn bản)
  - Hỗ trợ việc định dạng mã nguồn, khiến việc viết mã trở nên trực quan hơn
  - Hỗ trợ các tính năng gợi nhớ khi viết mã, giúp việc viết mã ít sai sót hơn

Phần 3

Các mục tiêu của phân tích từ vựng

Các mục tiêu của PTTV

- **Chính xác**: đây là mục tiêu quan trọng nhất, kết quả phân tích cần trả về chính xác dãy các từ vựng
- **Tốc độ**: các bộ PTTV cần hoạt động ở độ phức tạp tuyến tính theo độ dài của mã nguồn cần phân tích
- **Đầy đủ**: bộ PTTV cung cấp càng chi tiết về từ vựng thì công việc của các pha sau càng nhanh chóng
  - Nhiều bộ PTTV hiện đại hiểu gần như chính xác ý nghĩa của từ vựng trong ngữ cảnh (chẳng hạn phân biệt được tên biến và tên hàm)
- **Chịu lỗi**: bộ PTTV cần có khả năng chịu lỗi và có chiến lược khắc phục lỗi phù hợp

Phần 4

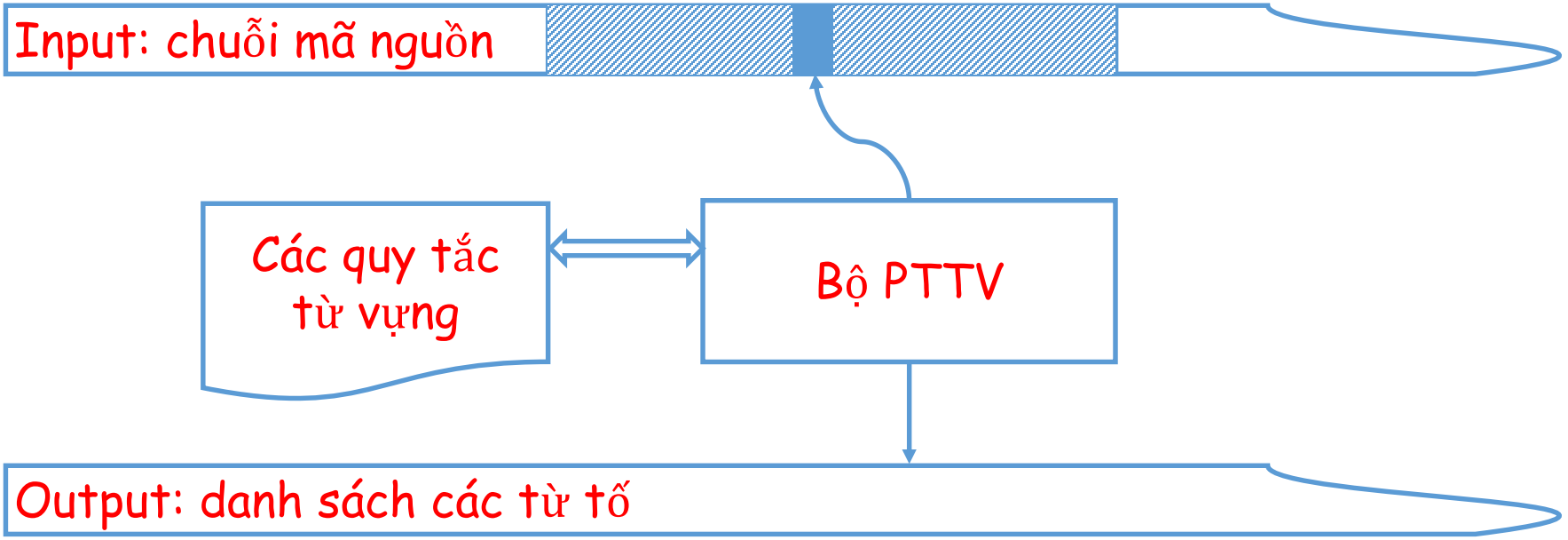
Đầu vào và đầu ra của phân tích từ vựng

Đầu vào của bộ PTTV

- Trong trường hợp tổng quát nhất, đầu vào của bộ PTTV là mã nguồn cần phân tích, không có bất kì ràng buộc nào
- Đối với đa số các ngôn ngữ lập trình, do các quy tắc về từ vựng, đầu vào của bộ PTTV có thể được ngắt thành các dòng (và xử lý đặc biệt trong trường hợp từ vựng đầu vào chiếm nhiều dòng)
- Trong nhiều trường hợp, dữ liệu đầu vào thường là một stream đi kèm là một vùng đệm để lưu dữ liệu hiện đang xử lý (phòng khi bộ PTTV có thể phải đọc lại nhiều lần dữ liệu đã xử lý)

Đầu vào của bộ PTTV

- Với những ngôn ngữ có cấu trúc từ vựng đơn giản, các bộ PTTV có thể hoạt động tốt mà không cần sử dụng bất kì một vùng nhớ đệm tạm thời nào
- Hình dưới minh họa đầu vào của bộ PTTV



Đầu ra của bộ PTTV

- Đầu ra của bộ PTTV phụ thuộc vào các đặc điểm của ngôn ngữ nguồn và bộ phân tích cú pháp
- Trong hầu hết các tình huống, bộ PTTV thường trả về kết quả ở dạng sau:
  - Danh sách các từ vựng ứng theo mã nguồn (thường là một danh sách liên kết, chẳng hạn – List<Word>)
  - Với mỗi từ vựng, thông tin bao gồm:
    - Từ loại của từ vựng
    - Giá trị chính xác của từ vựng
    - Giá trị mã hóa của từ vựng
    - Vị trí của từ vựng trong mã nguồn

Bộ PTTV đơn giản (mã giả C#)

```
// chứa thông tin về một từ tố
class Word {
    public int wordType;          // chứa từ loại của từ
    public string wordContent;    // chứa nội dung của từ
}
// bộ phân tích từ vựng
class PTTV {
    // phân tích chuỗi S thành dãy các từ tố
    public List<Word> process(string S) { … }
    // lấy ra từ tố tiếp theo
    Word getNextWord() { … }
}
```

Phần 5

Các bước xây dựng bộ phân tích từ vựng

Các bước xây dựng bộ PTTV

- Có 3 phương án thông dụng khi viết bộ PTTV
  - Tự viết (hard coding) bộ PTTV: tự xử lý từng loại từ tố và các tình huống có thể xảy ra (nhập nhằng, lỗi,…)
  - Viết một automat phân tích từ vựng: hoạt động của máy automat phân tích từ vựng luôn giống nhau (như vậy có thể tham khảo mã nguồn khác), nạp đồ thị chuyển của ngôn ngữ nào thì automat sẽ phân tích từ vựng của ngôn ngữ đó
  - Tự động sinh (auto-generated code) bộ PTTV: chỉ mô tả các loại từ vựng và cách xử lý tương ứng, công cụ sẽ tự động làm hết các phần còn lại

Các bước xây dựng bộ PTTV

1. Mô tả các loại từ vựng của ngôn ngữ nguồn, các mô tả này có thể ở dạng ngôn ngữ tự nhiên
2. Đặc tả các từ loại bằng biểu thức chính quy
3. Lựa chọn cách xử lý trong tình huống lỗi, ngoại lệ
4. Lựa chọn phương án phù hợp để tạo mã cho bộ PTTV, dù chọn cách nào trong 3 cách trên thì cũng cần thực hiện các việc sau:
   - Xây dựng đồ thị chuyển cho từng biểu thức chính quy
   - Kết hợp chúng thành một đồ thị chuyển duy nhất
   - Tối ưu hóa trạng thái của đồ thị chuyển cuối cùng

Phần 6

Biểu diễn từ vựng bằng biểu thức chính quy

Từ vựng

- Từ loại (hay từ tố) cần được mô tả một cách chặt chẽ để tránh nhập nhằng
- Ví dụ:
  - Chữ cái: các chữ cái trong bảng chữ cái tiếng Anh
  - Chữ số: các chữ số trong hệ thập phân
  - Tên: một chuỗi, bắt đầu bằng chữ cái, có thể có chữ cái hoặc chữ số theo sau
  - Số nguyên: một dãy các chữ số
  - Phép toán: +, -, *, /
  - Phép so sánh: <, <=, >, >=, ==, !=

Biểu thức chính quy (BTCQ)

- Biểu thức chính quy (regular expression) là lựa chọn phổ biến để mô tả từ vựng
  - Phương pháp này mô tả ngôn ngữ một cách chặt chẽ
  - Có thuật toán xử lý hiệu quả để kiểm tra từ vựng có thuộc ngôn ngữ sinh bởi biểu thức chính quy hay không
- Kí hiệu quy ước

|	lựa chọn
()	nhóm các thành phần
*	lặp lại không, một hoặc nhiều lần
+	lặp lại một hoặc nhiều lần
?	lặp lại không hoặc một lần

Biểu diễn từ vựng bằng BTCQ

- Ví dụ sử dụng BTCQ để mô tả từ vựng:

  Chữ cái = A | B | … | Z | a | b | … | z

  Chữ số = 0 | 1 | … | 9

  Tên = Chữ cái (Chữ cái | Chữ số)*

  Số nguyên = (Chữ số)+

  Phép toán = + | - | * | /

  Phép so sánh = < | <= | > | >= | == | !=

- Cách ghi BTCQ đôi lúc còn mở rộng như sau:

  [abcd] tương đương với (a|b|c|d)

  [a-z] tương đương với (a|b|c|…|z)

  [^a-c] bất kì kí hiệu nào không phải a, b hoặc c

Phần 7

Lỗi và ngoại lệ khi phân tích từ vựng

Xử lý ngoại lệ

- Nhiều ngôn ngữ lập trình phải lựa chọn giữa việc thân thiện với người viết và tính chặt chẽ
  - Nếu ngôn ngữ chặt chẽ, không có ngoại lệ thì lại kém thân thiện với lập trình viên
  - Nếu ngôn ngữ thân thiện với lập trình viên thì dễ nhập nhằng
- Chẳng hạn:
  - Chuỗi "`a >= b;`"
  - Nên tách thành "`a | >= | b | ;`"
  - Hay tách thành "`a | > | = | b | ;`"

Xử lý nhập nhằng

- Chiến lược xử lý nhập nhằng đơn giản nhất: ưu tiên từ vựng dài (chiến lược tham lam)
- Không phải lúc nào chiến lược này cũng đúng
- Ngôn ngữ C++:
  - "`Vector<Array<string>> v(10);`"
  - Trong tình huống này nếu ưu tiên chuỗi dài thì từ vựng nhận được sẽ là ">>"
  - Trong thực tế đây là 2 dấu ">" thì hợp lý hơn
- Trong nhiều tình huống bộ PTTV sẽ sử dụng thông tin ngữ cảnh để có quyết định phù hợp

Xử lý khi gặp lỗi

- Bộ PTTV sẽ xử lý thế nào khi phát hiện ra lỗi?
  - Cách xử lý đơn giản nhất là ngừng lại và báo lỗi cho người sử dụng
  - Nhưng nếu bộ PTTV tiếp tục hoạt động thì tốt hơn, chẳng hạn có thể phát hiện thêm nhiều lỗi và nhắc người dùng sửa cùng một lượt
- Nhiều chiến lược khi muốn xây dựng bộ PTTV có khả năng chịu lỗi. Chiến lược được áp dụng phổ biến nhất là xóa hoặc bỏ qua các kí tự lỗi cho tới khi gặp một kí hiệu bắt đầu phù hợp (dấu trống, dấu xuống dòng,…)

Phần 8

Phân tích từ vựng cho một ngôn ngữ đơn giản

Ngôn ngữ A

Ngôn ngữ lập trình A chuyên thực hiện các phép toán số

1. Mỗi lệnh viết trên 1 dòng
2. Lệnh bao giờ cũng có dạng <biến> = <biểu thức>
3. <biến> là một tên riêng, không cần khai báo trước, giống quy cách tên biến thông dụng, biến luôn là số
4. <biểu thức> được viết theo quy cách biểu thức số học, có thể gồm:
   - Số nguyên, số thực, biến
   - Lời gọi hàm toán học thông dụng: sqrt, log, exp, power,...
   - Các phép toán thông dụng + - * / %
   - Các cặp ngoặc tròn

Phần 9

Bài tập và thảo luận

Bài tập và thảo luận

1. Viết biểu thức chính quy để mô tả:
   - Số thực viết ở dạng khoa học (-1.2e+1, 2.3E-10,...)
   - Số thực (viết ở tất cả các dạng)
   - Các số nhị phân lẻ
   - Các số nhị phân lẻ lớn hơn 5
   - Các chuỗi số nhị phân không có chuỗi con 101
2. Viết biểu thức chính quy để mô tả:
   - Tên file hoặc folder trong máy tính
   - Đường dẫn (directory) trong máy tính
   - Tên miền internet
   - Địa chỉ liên kết (link address)

Bài tập và thảo luận

3. Viết biểu thức chính quy để mô tả:
   - Địa chỉ IP4
   - Địa chỉ email
   - Số điện thoại của Việt Nam
   - Vị trí tọa độ trên Google Maps: (vĩ độ, kinh độ)
4. Mô tả ngôn ngữ sinh bởi các biểu thức chính quy:
   - 1(0|1)+1
   - ((ε|1)0*)+
   - (0|1)*0(0|1)(0|1)
   - 0*10*10*10*
   - (00|11)*((01|10)(00|11)*(01|10)(00|11*))*

Bài tập và thảo luận

5. (sử dụng ngôn ngữ C++, C# hoặc java) Viết khai báo class phù hợp cho việc mô tả thông tin về một từ vựng được đề cập trong slide 19.
6. Từ class kết quả của bài trên, hãy viết khai báo hợp lý cho đầu ra của bộ phân tích từ vựng
   - Áp dụng PTTV với ngôn ngữ A mô tả ở slide 31, với biểu thức "**a=100/(1+2)**", kết quả trả về của bộ PTTV sẽ như thế nào?
7. Hãy chỉ ra những lỗi có thể gặp phải khi viết lệnh cho ngôn ngữ A mà bộ PTTV có thể phát hiện ra.