

MỤC LỤC

LỜI NÓI ĐẦU	1
TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH VÀ CHƯƠNG TRÌNH DỊCH	2
TRÌNH BIÊN DỊCH.....	3
CHƯƠNG I.TỔNG QUAN.....	3
1.1. Các khái niệm liên quan.....	3
1.1.1. Trình biên dịch.....	3
1.1.2. Trình thông dịch:	3
1.2. Phân tích chương trình nguồn.....	4
1.2.1. Phân tích từ vựng.....	4
1.2.2. Phân tích cú pháp	5
1.2.3. Phân tích ngữ nghĩa.....	6
1.3. Các giai đoạn của trình biên dịch	7
1.3.1. Kỳ đầu.....	8
1.3.2. Kỳ sau	8
CHƯƠNG II. PHÂN TÍCH TỪ VỰNG.....	12
2.1.Vai trò của bộ phân tích từ vựng	14
2.1.1. Nhiệm vụ.....	14
2.1.2. Tiến trình phân tích từ vựng	15
2.1.3. Từ vị, từ tổ, mẫu.....	16
2.1.4. Thuộc tính của token	17
2.1.5. Lỗi từ vựng	17
2.2. Lưu trữ tạm thời trong trình nguồn.....	18
2.2.1. Cặp bộ đệm.....	18
2.2.2. Khóa cầm canh.....	19
2.3. Tính chất và nhận dạng token	20
2.3.1. Đặc tả token	20
2.3.2. Nhận dạng token	23
2.4. Các bước để xây dựng bộ phân tích từ vựng.	29
2.5. Ngôn ngữ và đặc tả cho bộ phân tích từ vựng	29
2.5.1. Bộ sinh bộ phân tích từ vựng	29
2.5.2. Đặc tả lex.....	30
BÀI TẬP CHƯƠNG II- PHÂN TÍCH TỪ VỰNG	32

CHƯƠNG III. PHÂN TÍCH CÚ PHÁP.....	33
3.1. Phương pháp phân tích cú pháp	35
3.1.1. Vai trò của bộ phân tích cú pháp	35
3.1.2. Văn phạm phi ngữ cảnh	36
3.1.3. Các phương pháp phân tích cú pháp	43
3.2. Các phương pháp phân tích tất định	64
3.2.1. Bộ phân tích LL	64
3.2.2. Biến đổi văn phạm mơ hồ	80
3.3. Cú pháp điều khiển	84
3.3.1. Định nghĩa điều khiển dựa cú pháp	84
3.3.2. Xây dựng cây phân tích cú pháp	88
3.3.3. Thứ tự đánh giá thuộc tính	92
BÀI TẬP CHƯƠNG III. PHÂN TÍCH CÚ PHÁP	103
CHƯƠNG IV. PHÂN TÍCH NGỮ NGHĨA VÀ BẢNG DANH BIỂU.....	108
4.1. Các hệ thống kiểu	110
4.1.1. Biểu thức kiểu.....	110
4.1.2. Hệ thống kiểu	110
4.1.3. Kiểm tra kiểu tĩnh và động	110
4.2. Các vấn đề của kiểm tra kiểu	110
4.2.1. Đặc tả một bộ kiểm tra kiểu đơn giản.....	110
4.2.2. Sự tương đương của các biểu thức kiểu.....	112
4.2.3. Chuyển đổi kiểu	113
4.3. Bảng danh biểu	115
4.3.1. Mục đích của bảng danh biểu.....	115
4.3.2. Các yêu cầu bảng danh biểu	115
4.3.3. Cấu trúc dữ liệu của bảng danh biểu.....	115
CHƯƠNG V. SINH MÃ	120
5.1. Sinh mã trung gian	121
5.1.1. Các ngôn ngữ trung gian.....	121
5.1.2. Khai báo.....	127
5.1.3. Lệnh gán	131
5.1.4. Biểu thức logic.....	134
5.1.5. Phát biểu Case.....	139
5.2. Sinh mã đích.....	140
5.2.1. Các dạng mã đối tượng	140
5.2.2. Các vấn đề thiết kế bộ sinh mã	141

5.2.3. Máy đích	144
5.2.4. Quản lý bộ nhớ trong thời gian thực hiện	146
5.2.5. Khối cơ bản và lưu đồ.....	152
5.2.6. Bộ sinh mã đơn giản	158
BÀI TẬP CHƯƠNG V. SINH MÃ	163
PHỤ LỤC A	165
PHỤ LỤC B.....	170
PHỤ LỤC C	177
TÀI LIỆU THAM KHẢO	183

Tailieu.vn

LỜI NÓI ĐẦU

Trong thời kỳ đầu của máy tính, cơ chế vận hành được lập trình viên chuyển từ mã nguồn sang mã máy để máy tính hiểu và thực hiện được công việc. Đến nay có nhiều ngôn ngữ lập trình được tạo ra với những trợ giúp và môi trường thuận tiện hơn để lập trình viên thiết kế các ứng dụng.

Nội dung chính của chương trình dịch là thực hiện việc chuyển đổi một chương trình hay đoạn chương trình từ ngôn ngữ nguồn mà con người hiểu sang ngôn ngữ đích (ngôn ngữ máy; mã máy) tương đương để máy tính hiểu và thực hiện, việc chuyển đổi này phụ thuộc hoàn toàn vào môi trường của máy tính (phần cứng, hệ điều hành). Trong đó chương trình dịch là một minh họa của lý thuyết ngôn ngữ hình thức và Automata, là kết quả của việc nghiên cứu về sự diễn dịch làm nền tảng của việc hình thành tri thức loài người như: diễn giải và biên dịch, thông dịch lại các thông tin nguồn, ngôn ngữ nguồn (của tự nhiên, của những đối tượng thể hiện sự tồn tại) sang ngôn ngữ mà một đối tượng khác hiểu và thực hiện các công việc.

Nội dung chính của giáo trình này giới thiệu cách tiếp cận sáu bước biên dịch của một ứng dụng tin học gọi là chương trình dịch (Trình biên dịch). Sáu bước biên dịch là: phân tích từ vựng, phân tích cú pháp, phân tích ngữ nghĩa, sinh mã trung gian, tối ưu hóa mã trung gian, sinh mã đích. Trong phần phụ lục tham khảo được trình bày kiến thức liên quan mà đường dẫn ở mỗi phần đã ghi rõ.

Nhóm tác giả đã nghiên cứu từ nhiều nguồn tài liệu về trình biên dịch, với mong muốn có một giáo trình “Trình Biên Dịch” đáp ứng phần nào nhu cầu mong đợi của sinh viên chuyên ngành.

Mặc dù nhiều cố gắng trong quá trình biên soạn nhưng vẫn còn hạn chế, rất mong nhận được nhiều ý kiến đóng góp của quý đồng nghiệp và bạn đọc gần xa, để lần tái bản sau hoàn chỉnh hơn.

TỔNG QUAN VỀ NGÔN NGỮ LẬP TRÌNH VÀ CHƯƠNG TRÌNH DỊCH

Nội dung chính:

Chương này cung cấp những kiến thức cơ bản giúp cho bạn đọc hiểu sáu bước biên dịch là: bước 1: phân tích từ vựng; bước 2: phân tích cú pháp; bước 3: phân tích ngữ nghĩa; bước 4: sinh mã trung gian; bước 5: tối ưu hóa mã trung gian; bước 6: sinh mã đích.

Mục tiêu

Giúp cho bạn đọc hiểu cách thức một chương trình dịch nhận dạng mã nguồn, phân tích mã nguồn, xử lý mã nguồn. Nếu xử lý thành công: sinh mã đích, nếu xử lý không thành công: thông báo lỗi.

Lĩnh vực ứng dụng

Mô hình hóa việc xử lý thông tin đầu vào, thông tin đầu vào ở đây là đối tượng ngôn ngữ nguồn.

Yêu cầu chung

Đọc giả đã hiểu về ngôn ngữ lập trình, cấu trúc máy tính, lý thuyết ngôn ngữ, cấu trúc dữ liệu, phân tích thiết kế giải thuật và công nghệ phần mềm.

TÀI LIỆU THAM KHẢO CHƯƠNG 1:

- [1] **Dương Tuấn Anh** (1986), “*Giáo trình biên dịch*” - NXB Đại Học Bách Khoa TPHCM
- [2] **Phạm Hồng Nguyên**, “*Giáo trình Chương trình Dịch*” - NXB Đại Học Quốc Gia Hà Nội
- [3] **Phan Thị Tươi (2011)**, “*Trình Biên Dịch*” - (Trường Đại học kỹ thuật Tp.HCM) - NXB Giáo dục
- [4] **Compilers : Principles, Technique and Tools** - Alfred V.Aho, Jeffrey D.Ullman - Addison - Wesley Publishing Company, 1986.
- [5] **Compiler Design** – Reinhard Wilhelm, Dieter Maurer - Addison – Wesley Publishing Company, 1996.

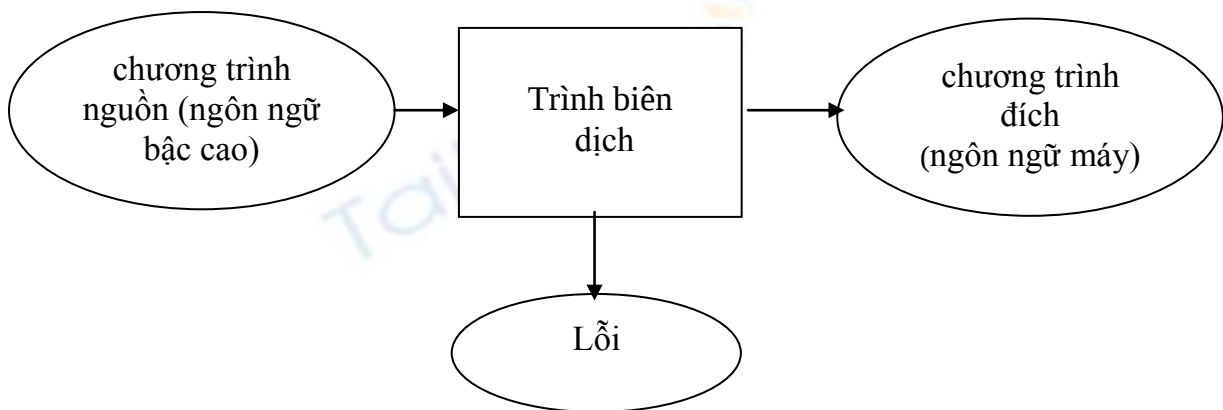
TRÌNH BIÊN DỊCH

CHƯƠNG I.TỔNG QUAN

1.1. Các khái niệm liên quan

1.1.1. Trình biên dịch

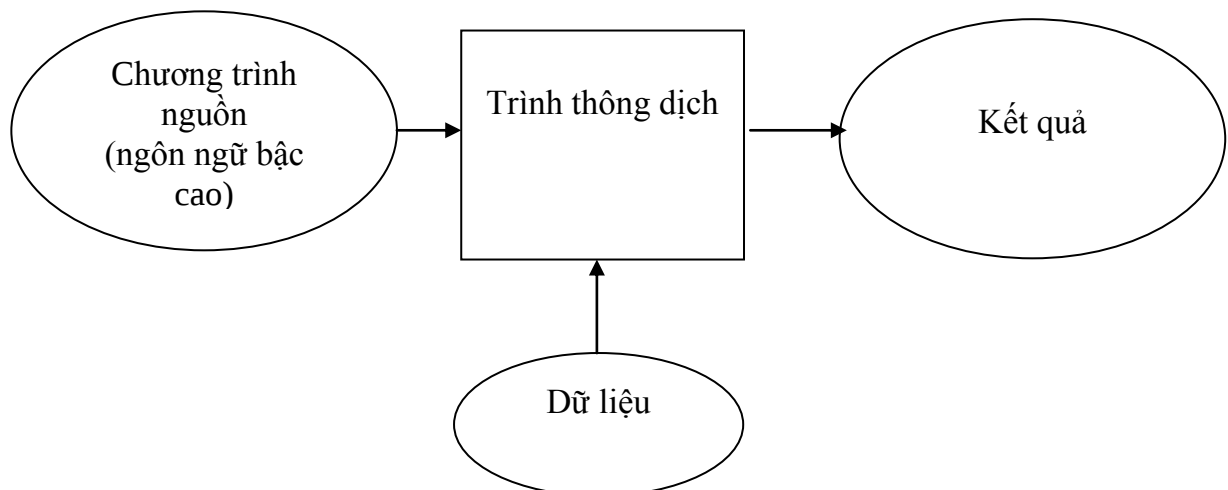
Trình biên dịch là một chương trình thực hiện việc chuyển đổi một chương trình hay đoạn chương trình từ ngôn ngữ này (gọi là ngôn ngữ nguồn) sang ngôn ngữ khác (gọi là ngôn ngữ đích) tương đương. Một phần quan trọng trong quá trình dịch là ghi nhận lại các lỗi có trong chương trình nguồn để thông báo lại cho người viết chương trình.



Hình 1.1: Sơ đồ một trình biên dịch

1.1.2. Trình thông dịch:

Trình thông dịch là quá trình xử lý dạng bên trong của chương trình nguồn và dữ liệu cùng một thời gian. Trong thông dịch thì mã nguồn không được dịch trước thành ngôn ngữ máy mà mỗi lần cần chạy chương trình thì mã nguồn mới được dịch để thực thi từng dòng một (line by line). Tất cả các ngôn ngữ không biên dịch ra mã máy đều phải sử dụng trình thông dịch (PHP, WScripts, Perl, Linux Shell, Python....).



Hình 1.2: Sơ đồ một trình thông dịch

1.2. Phân tích chương trình nguồn

1.2.1. Phân tích từ vựng

Trong một trình biên dịch, giai đoạn phân tích từ vựng sẽ đọc chương trình nguồn từ trái sang phải (scanning) để tách ra thành các thẻ từ ngữ (token). Các thẻ từ ngữ này thuộc dạng nào trong những dạng đã định nghĩa trước của trình biên dịch: danh biểu, ký hiệu, số. Có hai dạng danh biểu: danh biểu cài đặt sẵn (từ khóa, ví dụ Dim trong Visual Basic), danh biểu người dùng định nghĩa.

Ví dụ 1.1a: Quá trình phân tích từ vựng cho câu lệnh gán

position := initial + rate * 10

sẽ tách thành các token như sau:

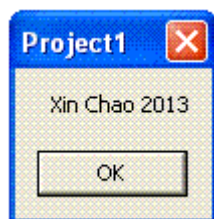
- | | |
|--------------------------|--------------------------|
| 1. Danh biểu position | 5. Danh biểu rate |
| 2. Ký hiệu phép gán:= | 6. Ký hiệu phép nhân (*) |
| 3. Danh biểu initial | 7. Số 10 |
| 4. Ký hiệu phép cộng (+) | |

Trong quá trình phân tích từ vựng các khoảng trắng (blank) sẽ bị bỏ qua.

Ví dụ 1.1b: Xem chương trình nguồn được tạo bằng Visual Basic 6.0

```
Private Sub Form_Load()  
    Dim namtam As Integer  
    namtam = Year(Date)  
    MsgBox (" Xin Chao " & namtam)  
End Sub
```

Với kết quả hiển thị



sẽ tách thành các token như sau:

1. Danh biểu Private
2. Danh biểu Sub
3. Danh biểu Form_Load
4. Ký hiệu (
5. Ký hiệu)
6. Danh biểu Dim

7. Danh biểu namtam
8. Danh biểu As
9. Danh biểu Integer
10. Ký hiệu phép gán =
11. Danh biểu Year
12. Danh biểu Date
13. Danh biểu MsgBox
14. Ký hiệu “
14. Ký hiệu ”
16. Danh biểu Xin
17. Danh biểu Chao
18. Ký hiệu &
19. Danh biểu End
20. Danh biểu Sub

Trong quá trình phân tích từ vựng các khoảng trắng (blank) sẽ bị bỏ qua. Gặp Private Sub, End Sub bộ phân tích từ vựng nhóm lại thành một danh biểu cài đặt sẵn. Khi đọc ký hiệu “ ; bộ phân tích từ vựng sinh một danh biểu String1 và String1 lưu giữ chuỗi kết thúc bằng ký hiệu ”. Đó là chuỗi “Xin Chào” có cả khoảng trắng.

Trong quá trình phân tích từ vựng các khoảng trắng (blank) sẽ bị bỏ qua.

1.2.2. Phân tích cú pháp

Giai đoạn phân tích cú pháp thực hiện công việc xem chuỗi token có thể sinh ra từ văn phạm cho trước không. Đó cũng là quá trình xây dựng cây phân tích cho chuỗi token.

Thông thường, các ngữ đoạn văn phạm này được biểu diễn bằng dạng *cây phân tích cú pháp* với :

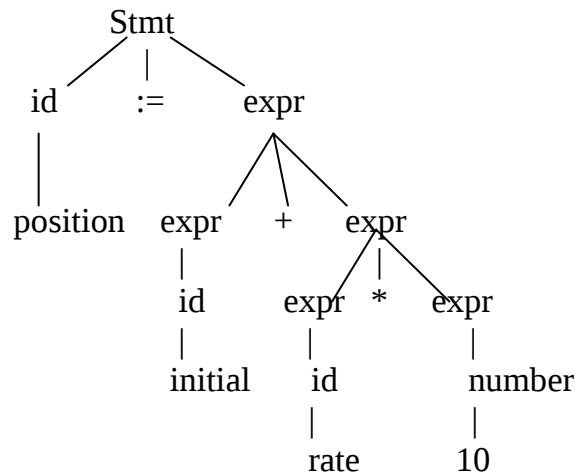
- Ngôn ngữ được đặc tả bởi các luật sinh.
- Phân tích cú pháp dựa vào luật sinh để xây dựng cây phân tích cú pháp.

Ví dụ 1.2: Giả sử ngôn ngữ đặc tả bởi các luật sinh sau:

Stmt $\rightarrow$ id := expr

expr $\rightarrow$ expr + expr | expr * expr | id | number

Với câu nhập: position := initial + rate * 10, cây phân tích cú pháp được xây dựng như sau :



Hình 1.3 - Cây phân tích cú pháp

Ví dụ 1.3:

- 1) Danh biểu (identifier) là một biểu thức (expr).
- 2) Số (number) là một biểu thức.
- 3) Nếu expr1 và expr2 là các biểu thức thì:
 $\text{expr1} + \text{expr2}$, $\text{expr1} * \text{expr2}$, (expr) được xem là biểu thức

Câu lệnh (statement) cũng có thể định nghĩa đệ qui :

- 1) Nếu id1 là một danh biểu và expr2 là một biểu thức thì
 $\text{id1} := \text{expr2}$ là một lệnh (stmt).
- 2) Nếu expr1 là một biểu thức và stmt2 là một lệnh
 thì while (expr1) do stmt2
 if (expr1) then stmt2
 đều là các lệnh.

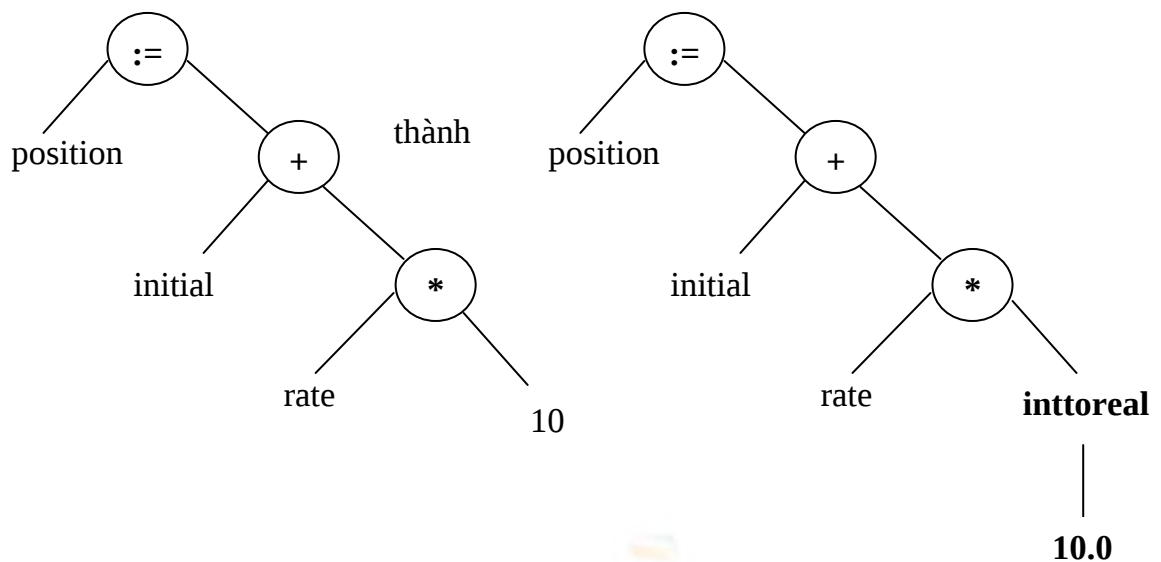
Người ta dùng các qui tắc đệ qui như trên để đặc tả luật sinh (production) cho ngôn ngữ. Sự phân chia giữa quá trình phân tích từ vựng và phân tích cú pháp cũng tùy theo công việc thực hiện.

1.2.3. Phân tích ngữ nghĩa

Giai đoạn phân tích ngữ nghĩa sẽ thực hiện việc kiểm tra xem chương trình nguồn có chứa lỗi về ngữ nghĩa hay không và tập hợp thông tin về kiểu cho giai đoạn sinh mã về sau. Một phần quan trọng trong giai đoạn phân tích ngữ nghĩa là kiểm tra kiểu và ép chuyển đổi kiểu.

Ví dụ 1.4: Trong biểu thức $\text{position} := \text{initial} + \text{rate} * 10$

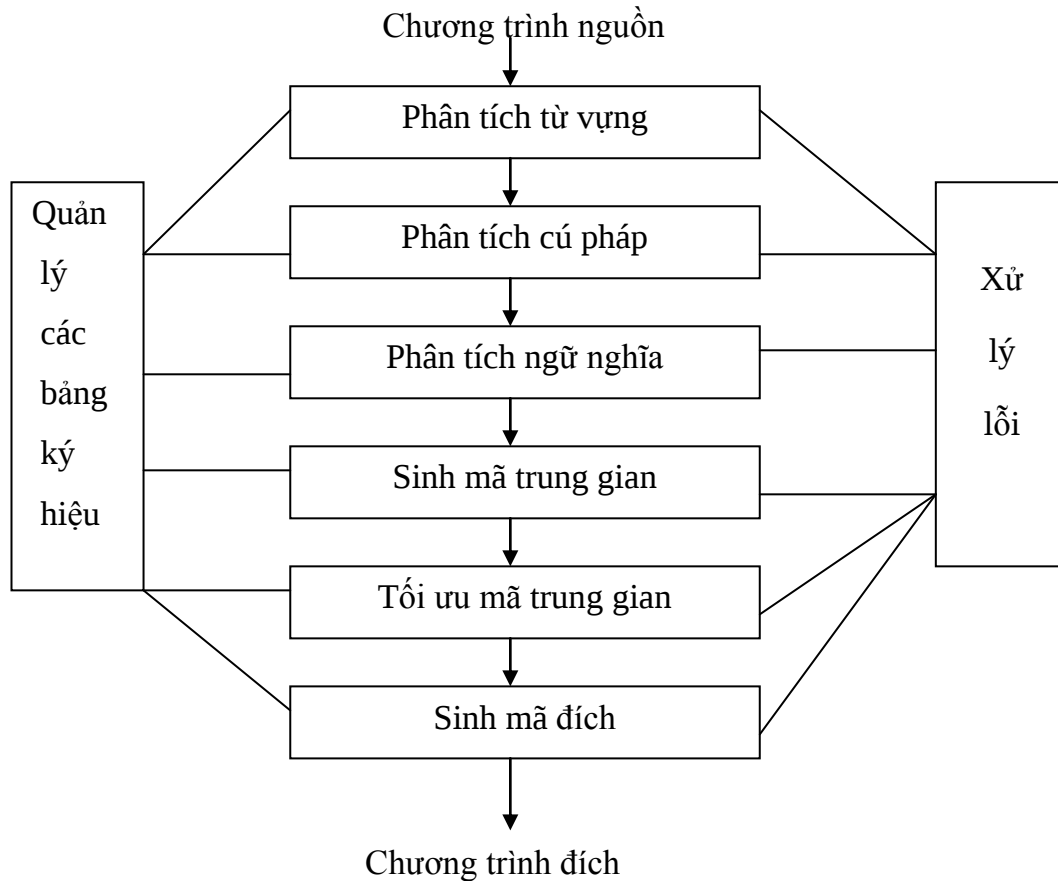
Các danh biểu (tên biến) được khai báo là real, 10 là số integer vì vậy trình biên dịch đổi số nguyên 10 thành số thực 10.0



Hình 1.4 - Chuyển đổi kiểu trên cây phân tích cú pháp

1.3. Các giai đoạn của trình biên dịch

Để dễ hình dung, một trình biên dịch được chia thành các giai đoạn, mỗi giai đoạn chuyển chương trình nguồn từ một dạng biểu diễn này sang một dạng biểu diễn khác. Một cách phân rõ diễn hình trình biên dịch được trình bày trong hình sau.



Hình 1.5 - Các giai đoạn trình biên dịch

Các giai đoạn mà chúng ta đề cập ở trên là thực hiện theo trình tự logic của một trình biên dịch. Nhưng trong thực tế, cài đặt các hoạt động của nhiều hơn một giai đoạn có thể được nhóm lại với nhau. Thông thường chúng được nhóm thành hai nhóm cơ bản, gọi là: kỳ đầu (Front end) và kỳ sau (Back end).

1.3.1. Kỳ đầu

Kỳ đầu bao gồm các giai đoạn hoặc các phần giai đoạn phụ thuộc nhiều vào ngôn ngữ nguồn và hầu như độc lập với máy đích. Thông thường, nó chứa các giai đoạn sau: phân tích từ vựng, phân tích cú pháp, phân tích ngữ nghĩa và sinh mã trung gian.

1.3.2. Kỳ sau

Kỳ sau bao gồm một số phần nào đó của trình biên dịch phụ thuộc vào máy đích và nói chung các phần này không phụ thuộc vào ngôn ngữ nguồn mà là ngôn ngữ trung gian. Trong kỳ sau, chúng ta gặp một số vấn đề tối ưu hoá mã, phát sinh mã đích cùng với việc xử lý lỗi và các thao tác trên Bảng danh biểu.

Ví dụ 1.5 : các giai đoạn biên dịch một biểu thức

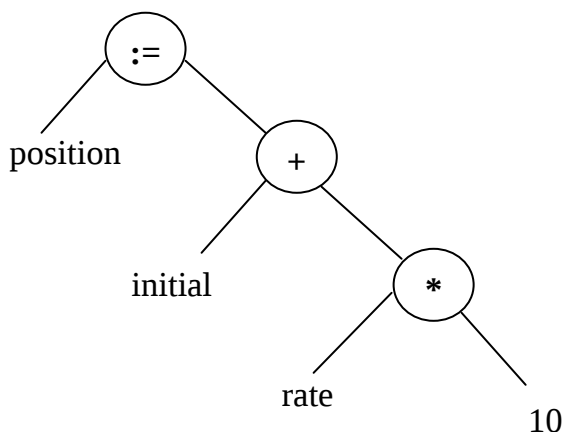
Sáu giai đoạn biên dịch cho biểu thức $position := initial + rate * 10$.

position := initial + rate * 10

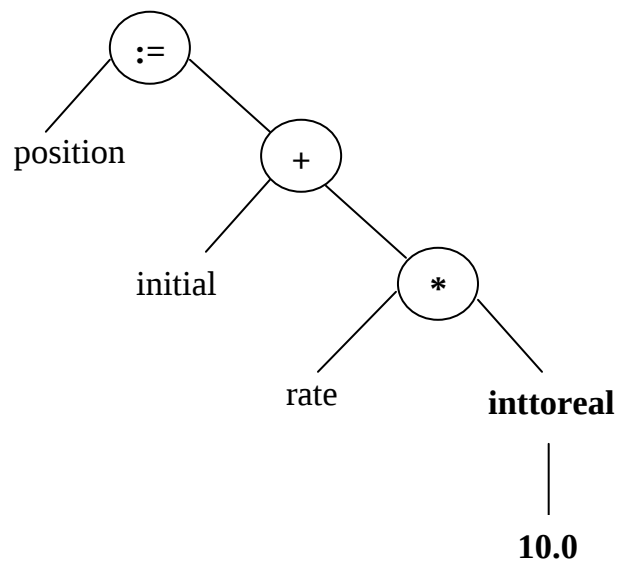
Giai đoạn 1- phân tích từ vựng $id1 := id2 + id3 * 10$

Giai đoạn 2- phân tích cú pháp

$id1 := id2 + id3 * 10$



Giai đoạn 3 - phân tích ngữ nghĩa



Hình 1.7 - Minh họa giai đoạn biên dịch một biểu thức

Giai đoạn 4 - Sinh mã trung gian

t1 := inttoreal (10)

t2:= id3 * t1

t3 := id2 + t2

id1 := t3

Giai đoạn 5 - Tối ưu hóa mã

t1 := id3 * 10.0

id1 := id2 + t1

Giai đoạn 6 - Phát sinh mã đích

MOVF id3, R2

MULF #10.0, R2

MOVF id2, R1

ADDF R2, R1

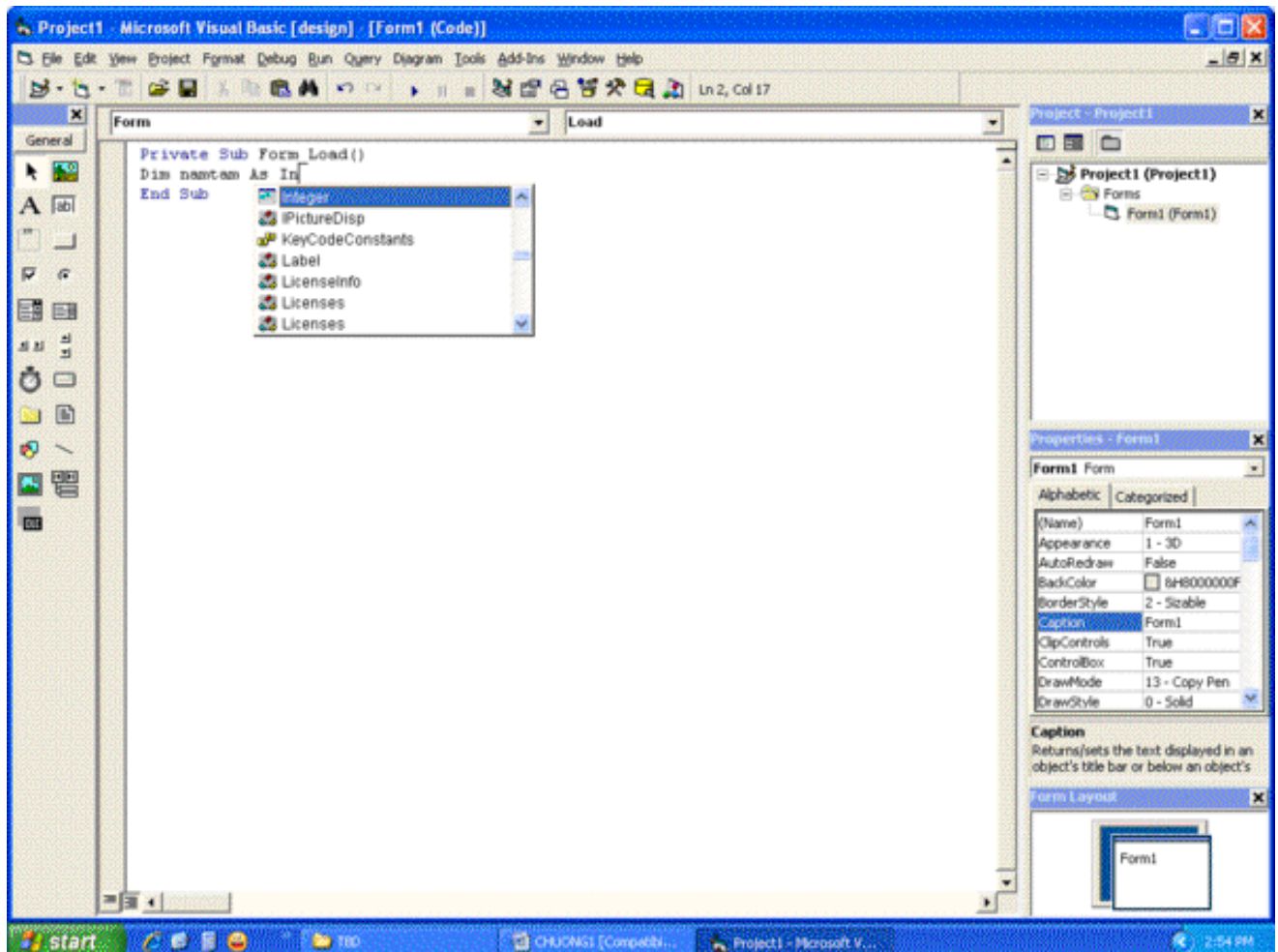
MOVF R1, id1

BÀI TẬP CHƯƠNG 1:

1. Bạn hãy lập bảng danh biểu được cài đặt sẵn (từ khóa, danh biểu, toán tử, dấu câu, hằng, chuỗi, ...) của một ngôn ngữ lập trình họ Pascal, một ngôn ngữ lập trình họ C, một ngôn ngữ lập trình họ Basic, một ngôn ngữ lập trình họ Java.
2. Với mỗi ngôn ngữ đã lập bảng danh biểu được cài đặt sẵn, hãy viết một câu lệnh đúng và thực hiện hiệu sáu bước biên dịch (bước 1: phân tích từ vựng; bước 2: phân tích cú pháp; bước 3: phân tích ngữ nghĩa; bước 4: sinh mã trung gian; bước 5: tối ưu hóa mã; bước 6: sinh mã đích).
3. Với mỗi ngôn ngữ đã lập bảng danh biểu được cài đặt sẵn, hãy viết một câu lệnh sai từ vựng và một câu lệnh sai cú pháp. Thực hiện hiệu sáu bước biên dịch. Chỉ ra nếu viết sai từ vựng và sai cú pháp thì thông báo lỗi xảy ra lúc nào, vì sao ?

Bài thực hành chương 1:

Bạn hãy sử dụng ngôn ngữ lập trình Visual Basic 6.0 hay phiên bản mới hơn và tìm hiểu:



1. Nhập lệnh và dùng tổ hợp phím Ctrl + Spacebar để trợ giúp nhập lệnh để không phạm lỗi từ vựng.
2. Tìm hiểu từ vựng của Visual Basic 6.0
3. Ghi ra các kinh nghiệm sử dụng và trên cơ sở này tự xây dựng phương pháp tìm hiểu thêm về các ngôn ngữ lập trình khác.

TaiLieu.vn

CHƯƠNG II. PHÂN TÍCH TỪ VỰNG

Nội dung chính:

Chương này giúp cho bạn đọc hiểu vai trò của bộ phân tích từ vựng.

Bộ vi xử lý của máy vi tính thực ra chỉ là máy nghiền số với tốc độ điện tử, chỉ biết hai bit 0 và 1, nôm na là có điện hay không có điện, và hai luật cho nó vận hành: Luật + và Luật -, nhân và chia là việc cộng hay trừ nhiều lần với số lần xác định rõ ràng. Để đưa số và luật cho nó nghiền cần thanh ghi với cấu trúc và cách thức vận hành riêng, với 2 mẫu: là lệnh hay dữ liệu, đều biểu diễn qua mẫu với tiền tố đầu xác định loại, tiền tố sau là giá trị. Người lập trình thiết kế một ứng dụng multimedia có video, âm thanh, hình ảnh, ký tự thì với bộ vi xử lý đều là một dãy bit 1001....0110

Để biên dịch những gì lập trình viên tạo ra sang mã máy, việc đầu tiên mà trình biên dịch yêu cầu là lập trình viên phải hiểu từ vựng nào máy tính hiểu thông qua bảng danh biểu cài đặt sẵn (từ khóa, các tên hàm được cài đặt sẵn) trong ngôn ngữ lập trình, nếu không nhớ rõ thì ngôn ngữ lập trình hiện đại có module trợ giúp. Đây là bảng danh biểu được cài đặt sẵn với từ ngữ đã được định nghĩa. Bộ phân tích từ vựng chỉ thực hiện việc nhóm các ký tự nhập thành ra chuỗi các từ ngữ (token), phân tích ngôn ngữ nguồn có bao nhiêu từ ngữ, từ ngữ nào đã được định nghĩa trước, từ ngữ nào người sử dụng định nghĩa. Kết thúc công việc của bộ phân tích từ vựng chính là hình thành được chuỗi các từ ngữ (token), xác định được nó, hình thành bảng danh biểu người sử dụng định nghĩa (tên biến, tên hàm ...), chuyển ký tự người dùng nhập sang dãy bit 01.....10

Ngoài ra, chương này giúp bạn đọc hiểu việc thiết kế bộ phân tích từ vựng của thế hệ đi trước. Thiết kế bộ phân tích từ vựng chính là bước mở đường cho bài toán nhận dạng. Hiện nay, lập trình để nhận dạng dòng ký tự nguồn có bao nhiêu từ tổ đã được thiết lập sẵn không còn quá khó, khó chăng là những bài toán khác như nhận dạng một bộ phận để tìm ra trong 1 cơ sở dữ liệu lớn như nhận dạng hình ảnh (ví dụ bài toán nhận dạng vân tay, nhận dạng chuỗi gen), nhận dạng âm thanh. Nhưng trong thời kỳ đầu của vi tính, việc phải tính đến là tiết kiệm từng bit nhớ, tiết kiệm thời gian vận hành, đã là bài toán khó và đã nảy sinh các giải thuật thông minh. Nội dung chương này nêu hai vấn đề chính là giải thuật xử lý trong tình huống tiết kiệm từng bit nhớ, tiết kiệm thời gian dịch và việc dùng thuật toán định nghĩa ngôn ngữ, định nghĩa cú pháp để làm cơ sở cho việc nhận dạng thay vì như hiện nay lưu thành cơ sở dữ liệu các từ tổ, các cấu trúc, các danh biểu xác định sẵn trong ngôn ngữ.

Mục tiêu

Giúp cho bạn đọc hiểu rằng máy vi tính chỉ hiểu ngôn ngữ nguồn với việc sử

dụng từ ngữ đã được định nghĩa trước trong trình biên dịch, từ ngữ mà trình biên dịch giúp người sử dụng định nghĩa. Người sử dụng muốn sử dụng, đang sử dụng và sẽ sử dụng từ ngữ trong tiến trình trình biên dịch dịch mã nguồn sang mã đích và trong lúc mã đích vận hành rất quan trọng, với kết quả đúng sai rõ ràng (từ ngữ dùng lập trình, từ ngữ phải nhập khi chương trình biên dịch xong và đang thực thi). Bộ phân tích từ vựng tùy theo thiết kế riêng của từng ngôn ngữ lập trình, nếu xử lý thành công: sinh các bảng danh biểu lưu giữ các từ mã theo mẫu (Tên_từ_mã; Loại_từ_mã ; Vị_trí_từ_mã); hay chuỗi token; nếu xử lý không thành công thì thông báo lỗi.

Giúp cho bạn đọc hiểu cách thiết kế bộ phân tích từ vựng nhận dạng máy ngôn ngữ nguồn với việc sử dụng từ ngữ đã được định nghĩa trước trong trình biên dịch, từ ngữ mà trình biên dịch giúp người sử dụng định nghĩa .

Yêu cầu chung

Đọc giả đã hiểu:

- Về tiến trình xử lý thông tin với đầu vào – đầu ra.
- Về tiến trình phân tích và thiết kế giải thuật.

TÀI LIỆU THAM KHẢO CHƯƠNG 2:

- [1] **Dương Tuấn Anh** (1986), “*Giáo trình trình biên dịch*” - NXB Đại Học Bách Khoa TP HCM
- [2] **Phạm Hồng Nguyên**, “*Giáo trình Chương trình Dịch*” - NXB Đại Học Quốc Gia Hà Nội
- [3] **Phan Thị Tươi (2011)**, “*Trình Biên Dịch*” - (Trường Đại học kỹ thuật Tp.HCM) - NXB Giáo dục
- [4] **Automata and Formal Language. An Introduction** – Dean Kelley – Prentice Hall, Englewood Cliffs, New Jersey 07632.
- [5] **Compilers : Principles, Technique and Tools** - Alfred V.Aho, Jeffrey D.Ullman - Addison - Wesley Publishing Company, 1986.
- [6] **Compiler Design** – Reinhard Wilhelm, Dieter Maurer - Addison – Wesley Publishing Company, 1996.
- [7] **Design of Compilers : Techniques of Programming Language Translation** - Karen A. Lemone - CRC Press, Inc, 1992.
- [8] **Modern Compiler Implementation in C** - Andrew W. Appel – Cambridge University Press, 1997.

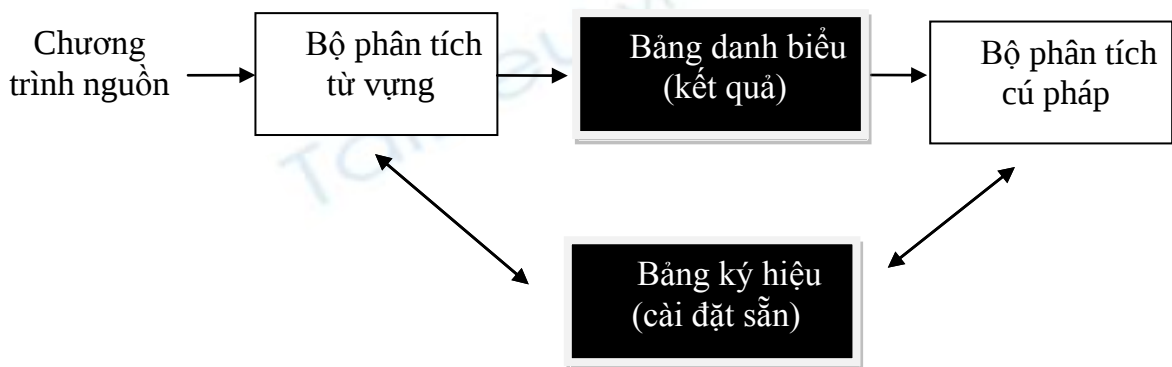
2.1.Vai trò của bộ phân tích từ vựng

2.1.1. Nhiệm vụ.

Bộ phân tích từ vựng có nhiệm vụ là đọc các kí tự nhập vào từ chương trình nguồn và phân tích đưa ra danh sách các từ tố (từ vựng và phân loại cú pháp của nó) cùng một số thông tin thuộc tính, lưu vào một bảng tạm gọi là bảng danh biểu.

Bộ phân tích từ vựng được thiết kế với hai ý tưởng:

- Xử lý hết từ ngữ đã nhập của ngôn ngữ nguồn lưu vào một bảng gọi là bảng danh biểu
- Hay xử lý từng từ ngữ đã nhập của ngôn ngữ nguồn, bộ phân tích từ vựng được thiết kế như một thủ tục được gọi bởi bộ phân tích cú pháp.



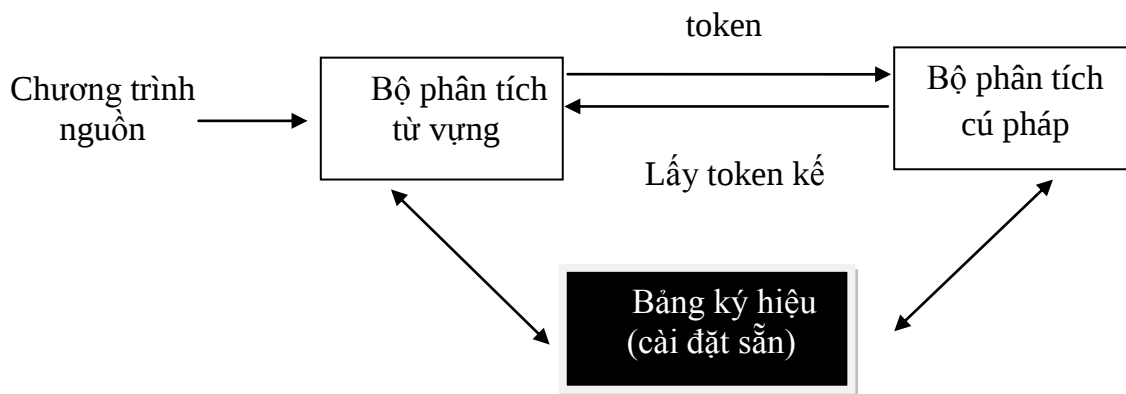
Hình 2.1a - *Giao diện của bộ phân tích từ vựng*

Đầu ra của bộ phân tích từ vựng là danh sách các từ tố đã lưu vào một bảng tạm gọi là bảng danh biểu và là đầu vào cho phân tích cú pháp.

Thực tế thì bộ phân tích cú pháp sẽ gọi lần lượt mỗi từ tố từ bộ phân tích để xử lý, chứ không gọi một lúc toàn bộ danh sách từ tố của cả chương trình nguồn.

Khi nhận được yêu cầu lấy một từ tố tiếp theo từ bộ phân tích cú pháp, bộ phân tích từ vựng sẽ đọc kí tự vào và tra trong bảng danh biểu cài đặt sẵn cho đến khi đưa ra được một từ tố.

Sự tương tác này được thể hiện như hình sau, trong đó bộ phân tích từ vựng được thiết kế như một thủ tục được gọi bởi bộ phân tích cú pháp, trả về một token khi được gọi.



Hình 2.1 b- *Giao diện của bộ phân tích từ vựng*

2.1.2. Tiến trình phân tích từ vựng

1) Xóa bỏ kí tự không có nghĩa (các chú thích, dòng trống, kí hiệu xuống dòng, kí tự trống không cần thiết)

Quá trình dịch sẽ xem xét tất cả các ký tự trong dòng nhập nên những ký tự không có nghĩa (khoảng trắng (blanks, tabs, newlines)) hoặc lời chú thích phải bị bỏ qua. Khi bộ phân tích từ vựng bỏ qua các khoảng trắng này thì bộ phân tích cú pháp không bao giờ quan tâm đến nó nữa.

2) Nhận dạng các kí hiệu: nhận dạng các từ tố.

Ví dụ ghép các chữ số để được một số và sử dụng nó như một đơn vị trong suốt quá trình dịch. Đặt num là một token biểu diễn cho một số nguyên. Khi một chuỗi các chữ số xuất hiện trong dòng nhập thì bộ phân tích sẽ gửi cho bộ phân tích cú pháp num. Giá trị của số nguyên đã được chuyển cho bộ phân tích cú pháp như là một thuộc tính của token num.

3) Số hoá các kí hiệu: Do con số xử lý dễ dàng hơn các xâu, từ khoá, tên, nên xâu thay bằng số, các chữ số được đổi thành số thực sự biểu diễn trong máy. Các tên được cất trong danh sách tên, các xâu cất trong danh sách xâu, các chuỗi số trong danh sách hằng số.

Các vấn đề của giai đoạn phân tích từ vựng

Nhiều lý do để tách riêng giai đoạn phân tích từ vựng với giai đoạn phân tích cú pháp:

1. Thứ nhất, nó làm cho việc thiết kế đơn giản và dễ hiểu hơn. Chẳng hạn, bộ phân tích cú pháp sẽ không phải xử lý các khoảng trắng hay các lời chú thích nữa vì chúng đã được bộ phân tích từ vựng loại bỏ.

2. Hiệu quả của trình biên dịch cũng sẽ được cải thiện, nhờ vào một số chương trình xử lý chuyên dụng sẽ làm giảm đáng kể thời gian đọc dữ liệu từ chương trình

nguồn và nhóm các token.

3. Tính đa tương thích (mang đi dễ dàng) của trình biên dịch cũng được cải thiện. Đặc tính của bộ ký tự nhập và những khác biệt của từng loại thiết bị có thể được giới hạn trong bước phân tích từ vựng. Dạng biểu diễn của các ký hiệu đặc biệt hoặc là những ký hiệu không chuẩn, chẳng hạn như ký hiệu (trong Pascal có thể được cô lập trong bộ phân tích từ vựng.

2.1.3. Từ vị, từ tố, mẫu

Khi làm việc bộ phân tích từ vựng, ta sẽ sử dụng các thuật ngữ *từ tố* (thẻ từ, token), *mẫu từ vựng* (pattern) và *trị từ vựng* (lexeme) với nghĩa cụ thể như sau:

1) Từ tố (token) là các ký hiệu kết thúc trong văn phạm đối với một ngôn ngữ nguồn, chẳng hạn như: từ khóa, danh biểu, toán tử, dấu câu, hằng, chuỗi, ...

- Đối với ngôn ngữ lập trình thì từ tố có thể được phân vào các loại sau:

+ từ khoá

+ tên của hằng, hàm, biến

+ số

+ xâu ký tự

+ các toán tử

+ các ký hiệu.

Ví dụ 2.1: `position := initial + 10 * rate ;`

Ta có các từ vựng position, :=, initial, +, 10, *, rate, ; Trong đó position, initial, rate là các từ vựng có cùng ý nghĩa cú pháp là các tên.

:= là phép gán

+ là phép cộng

* là phép nhân

10 là một con số

; là dấu chấm phẩy

Như vậy trong câu lệnh trên có 8 từ vựng thuộc 6 từ tố.

Phân tích cú pháp sẽ làm việc trên các từ tố chứ không phải từ vựng, ví dụ như là làm việc trên khái niệm một số chứ không phải trên 5 hay 2; làm việc trên khái niệm tên chứ không phải là a, b hay c.

2) Trị từ vựng (lexeme) của một token là một chuỗi ký tự biểu diễn cho token đó.

3) Mẫu từ vựng (pattern) là qui luật mô tả một tập các trị từ vựng kết hợp với một token nào đó.

Một số ví dụ về cách dùng của các thuật ngữ này được trình bày trong bảng sau:

Token	Trị từ vựng minh họa	Mô tả của mẫu từ vựng
const	const	const
if	if	if
relation	<, <=, =, < >, >, >=	< hoặc <= hoặc = hoặc <> hoặc > hoặc >=
id	pi, count, d2	Mở đầu là chữ cái theo sau là chữ cái, chữ số
num	3.1416, 0, 5	Bất kỳ hằng số nào
literal	“ hello ”	Mọi chữ cái nằm giữa “ và “ ngoại trừ “

Bảng 2.1 - Các ví dụ về token

2.1.4. Thuộc tính của token

Khi có nhiều mẫu từ vựng khớp với một trị từ vựng, bộ phân tích từ vựng trong trường hợp này phải cung cấp thêm một số thông tin khác cho các bước biên dịch sau đó. Do đó đối với mỗi token, bộ phân tích từ vựng sẽ đưa thông tin về các token vào các thuộc tính đi kèm của chúng. Các token có ảnh hưởng đến các quyết định phân tích cú pháp; các thuộc tính ảnh hưởng đến việc phiên dịch các thẻ từ. Token kết hợp với thuộc tính của nó tạo thành một bộ $\langle token, tokenval \rangle$.

Ví dụ 2.2 : Token và giá trị thuộc tính đi kèm của câu lệnh `position := initial + rate*10` được viết như một dãy các bộ sau:

< **tên**, con trỏ đến position trong bảng danh biểu >

< **phép_gán**, >

< **tên**, con trỏ đến initial trong bảng danh biểu >

< **toán_tử_cộng**, >

< **tên**, con trỏ đến rate trong bảng danh biểu >

< **toán_tử_nhân**, >

< **số_nguyên**, giá trị nguyên 10 >

Chú ý rằng một số bộ không cần giá trị thuộc tính, thành phần đầu tiên là đủ để nhận dạng trị từ vựng.

2.1.5. Lỗi từ vựng

Chỉ một số ít lỗi được phát hiện tại bước phân tích từ vựng, bởi vì bộ phân tích từ vựng có nhiều cách nhìn nhận chương trình nguồn. Ví dụ chuỗi **fi** được nhìn thấy lần đầu tiên trong một chương trình C với ngữ cảnh : **fi (a == f (x)) ...** Bộ phân tích từ vựng không thể biết đây là lỗi không viết đúng từ khóa **if** hay một danh biểu chưa được khai báo. Vì fi là một danh biểu hợp lệ nên bộ phân tích từ vựng phải