BỘ GIÁO DỤC VÀ ĐÀO TẠO

GIÁO TRÌNH CHƯƠNG TRÌNH DỊCH

NHÀ XUẤT BẢN KHOA HỌC XÃ HỘI

HÀ NỘI - 2009

MỤC TIÊU GIÁO TRÌNH

1.Cung cấp những kiến thức cơ bản về chương trình dịch
2.Cung cấp các phương pháp phân tích từ vựng, phân tích cú pháp.
3.Cơ sở cho việc tìm hiểu các ngôn ngữ lập trình.
4.Rèn luyện kỹ năng lập trình cho sinh viên

Nội dung giáo trình

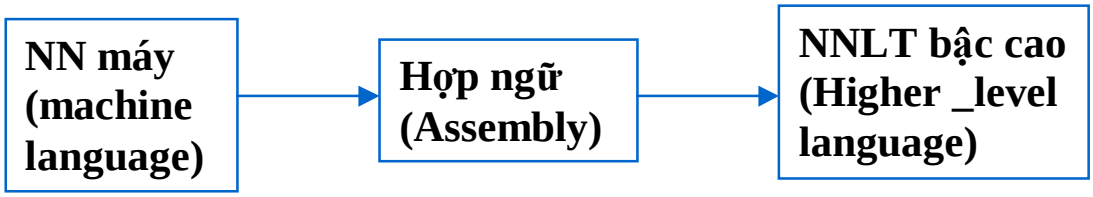
CHƯƠNG 1. NHẬP MÔN CHƯƠNG TRÌNH DỊCH

1. Các khái niệm cơ bản
2. Đặc trưng của ngôn ngữ lập trình (NNLT) bậc cao
3. Các qui tắc từ vựng và cú pháp
4. Các chức năng của một trình biên dịch

1. Các khái niệm cơ bản

1.1. Sự phát triển của ngôn ngữ lập trình
1.2. Khái niệm chương trình dịch
1.3. Phân loại chương trình dịch
1.4. Các ứng dụng khác của kỹ thuật dịch

1.1. Sự phát triển của ngôn ngữ lập trình

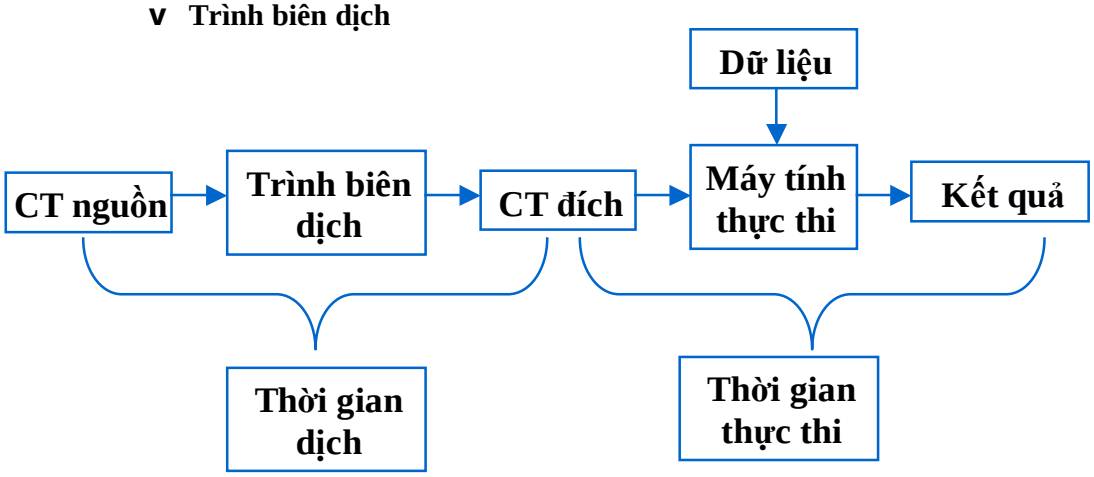


1.2. Khái niệm chương trình dịch

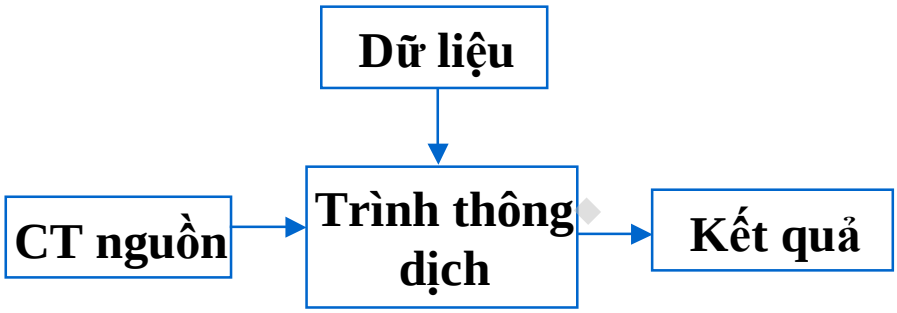
Chương trình dịch là chương trình dùng để dịch một chương trình (CT nguồn) viết trên NNLT nào đó (NN nguồn) sang một chương trình tương đương (CT đích) trên một NN khác (NN đích)

1.3. Phân loại chương trình dịch

- **Trình biên dịch**



- **Trình thông dịch**



1.4. Các ứng dụng khác của kỹ thuật dịch

- Trong các hệ thống: phần giao tiếp giữa người và máy thông qua các câu lệnh.
- Hệ thống xử lý NN tự nhiên: dịch thuật, tóm tắt văn bản.

2. Đặc trưng của NNLT bậc cao

- Tính tự nhiên
- Tính thích nghi
- Tính hiệu quả
- Tính đa dạng

3. Các qui tắc từ vựng và cú pháp

3.1. Bản chữ cái

- Gồm những ký hiệu được phép sử dụng để viết chương trình
- Số lượng, ý nghĩa sử dụng của các ký tự trong bản chữ cái của các NN là khác nhau.
- Nhìn chung bản chữ cái của các NNLT:
  + 52 chữ cái: A -->Z, a-->z

+ 10 chữ số: 0 -->9
+ Các ký hiệu khác:*, /, +, -, …

3.2. Từ tố (Token)

- Từ tố là đơn vị nhỏ nhất có nghĩa
- Từ tố được xây dựng từ bản chữ cái
- **Ví dụ**: hằng, biến, từ khoá, các phép toán,…

3.3. Phạm trù cú pháp

- Phạm trù cú pháp là một dãy từ tố kết hợp theo một qui luật nào đó
- Các cách biểu diễn cú pháp thông thường
  + BNF(Backus Naus Form):
    <lệnhgán>::=<tên biến>:=<biểu thức>
  + Biểu đồ cú pháp:

Chương trình-->Program -->Danh biểu--> Khối
Khối -->- var…
- procedure --> Danh biểu -->Khối
- begin -->lệnh --> end -->.

- Mục tiêu của phạm trù cú pháp là việc định nghĩa được khái niệm chương trình đến mức độ tự có

3.4. Các qui tắc từ vựng thông dụng

- Cách sử dụng khoảng trống(dấu trắng), dấu tab('\t'), dấu sang dòng('\n')
- Đối với liên kết tự do, có thể sử dụng nhiều khoảng trống thay vì một khoảng trống.

- Một khoảng trống là bắt buộc giữa các từ tố: từ khoá và tên,…
  Ví dụ: program tenct;
- Khoảng trống không bắt buộc: số và các phép toán, tên biến và các phép toán
  Ví dụ: x:=x+3*3;
- Cách sử dụng chú thích và xâu ký tự

4. Các chức năng của một chương trình biên dịch

- Phân tích từ vựng
- Phân tích cú pháp
- Phân tích ngữ nghĩa
- Xử lý lỗi
- Sinh mã trung gian
- Tối ưu mã trung gian
- Sinh mã đối tượng

4.1. Phân tích từ vựng

- CT nguồn là một dãy các ký tự.
- Phân tích từ vựng là phân tích CT nguồn thành các từ tố (Token).
- Các Token này sẽ là dữ liệu đầu vào của phân tích cú pháp.

4.2. Phân tích cú pháp

- Đầu vào sẽ là dãy các Token nối nhau bằng một qui tắc nào đó.
- Phân tích xem các Token có tuân theo qui tắc cú pháp của ngôn ngữ không

4.3. Phân tích ngữ nghĩa

- Kiểm tra tính hợp lệ của các phép toán và các phép xử lý
- **Ví dụ**:
  - Biến phải khai báo trước khi sử dụng (Pascal)
  - Kiểm tra tính tương thích kiểu dữ liệu của biến và biểu thức

4.4. Xử lý lỗi

- CT nguồn vẫn có thể xảy ra lỗi.
- Phần xử lý lỗi sẽ thông báo lỗi cho NSD
- Lỗi ở phần nào báo ở phần đó.

- Có các loại lỗi:
  - Lỗi từ vựng (trong Pascal sử dụng biến mà chưa khai báo)
  - Lỗi cú pháp ((a+5; lỗi thiếu dấu ‘)’)
  - Lỗi ngữ nghĩa (x=3.5; nhưng khai báo int x)
  - Lỗi thực hiện (phép chia 0)

4.5. sinh mã trung gian

- Sau giai đoạn phân tích ngữ nghĩa
- Mã trung gian là một dạng trung gian của CT nguồn có 2 đặc điểm:
  - Dễ được sinh ra
  - Dễ dịch sang ngôn ngữ đích

4.6. Tối ưu mã trung gian

- Bỏ bớt các lệnh thừa.
- Cải tiến lại mã trung gian để khi sinh mã đối tượng thì thời gian thực thi mã đối tượng sẽ ngắn hơn

4.7. Sinh mã đối tượng

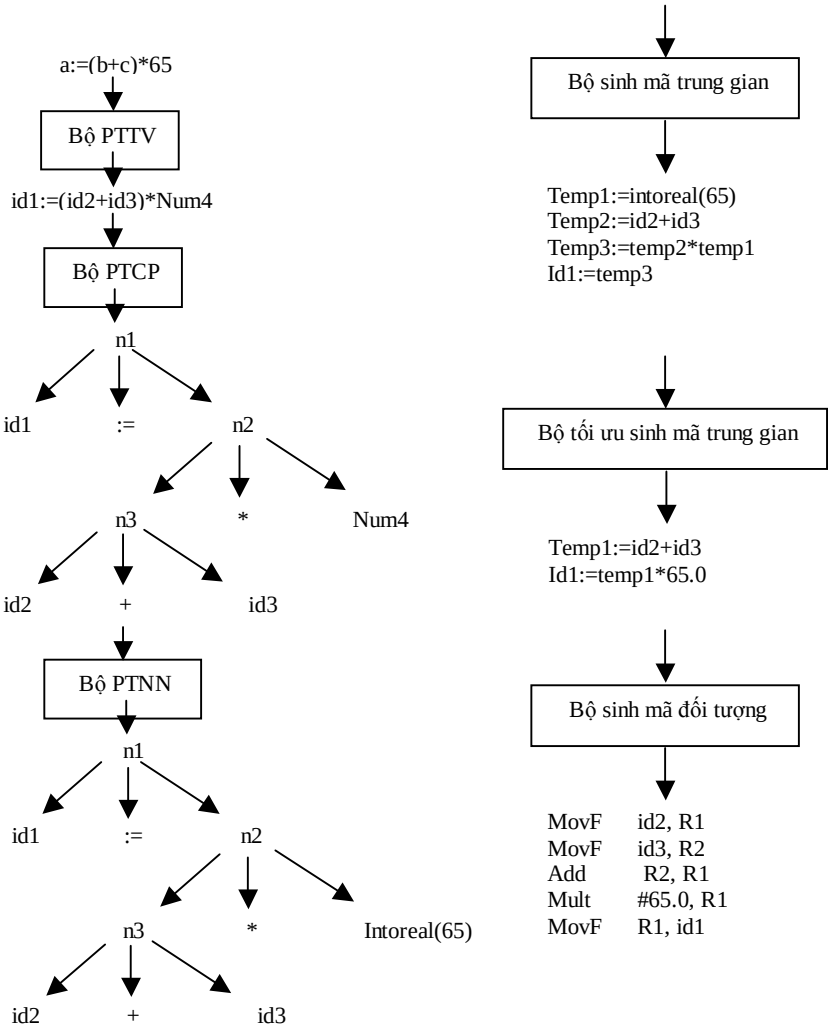
- Giai đoạn cuối của trình biên dịch.
- Mã đối tượng có thể là mã máy, hợp ngữ hay một ngôn ngữ khác ngôn ngữ nguồn.

Ø Các pha (giai đoạn) có thể thực hiện song hành

Ø Một vài pha có thể ghép lại thành lượt (chuyến)

Ø Một lượt sẽ đọc toàn bộ CT nguồn hay một dạng trung gian của CT nguồn, sau đó ghi kết quả để lượt sau đọc và xử lý tiếp.

Ví dụ:


a:=(b+c)*65
Bộ PTTV
id1:=(id2+id3)*Num4
Bộ PTCP
n1
id1
:=
n2
n3
*
Num4
id2
+
id3
Bộ PTNN
n1
id1
:=
n2
n3
*
Intoreal(65)
id2
+
id3
Bộ sinh mã trung gian
Temp1:=intoreal(65)
Temp2:=id2+id3
Temp3:=temp2*temp1
Id1:=temp3
Bộ tối ưu sinh mã trung gian
Temp1:=id2+id3
Id1:=temp1*65.0
Bộ sinh mã đối tượng
MovF id2, R1
MovF id3, R2
Add R2, R1
Mult #65.0, R1
MovF R1, id1

CHƯƠNG 2. PHÂN TÍCH TỪ VỰNG

- Mục đích
- Nội dung
- Otomat hữu hạn đơn định
- Bộ phân tích từ vựng
- Bảng danh biểu

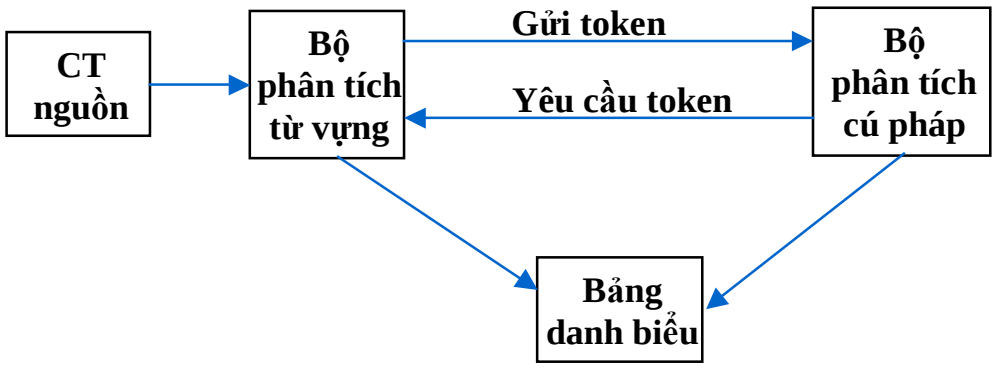
1.Mục đích

- Chia cắt xâu vào (CT nguồn) thành dãy các từ tố.
- Hai cách cài đặt
  - Sử dụng một lượt cho việc phân tích từ vựng ◊ dãy các token -->phân tích cú pháp.
  - Phân tích từ vựng dùng chung một lượt với phân tích cú pháp. Một lần chỉ phát hiện 1 token gọi là từ tố tiếp đến.

2.Nội dung

- Đọc xâu vào từng ký tự một --> gom lại thành token đến khi gặp ký tự không thể kết hợp thành token.
- Luôn luôn đọc trước một ký tự.
- Loại bỏ các ký tự trống và chú thích.
- Chuyển những thông tin của những từ tố (văn bản, mã phân loại) vừa phát hiện cho bộ phân tích cú pháp.
- Phát hiện lỗi.

- Sự giao tiếp giữa bộ phân tích từ vựng và bộ phân tích cú pháp



3.Otomat hữu hạn đơn định

3.1. Định nghĩa: M(Σ, Q, δ, q0, F)

Σ: bộ chữ vào

Q: tập hữu hạn các trạng thái

q0 ∈ Q: trạng thái đầu

F ⊆ Q: tập các trạng thái kết thúc

δ: hàm chuyển trạng thái có dạng δ(q,a)=p Với q,p ∈ Q, a ∈ Σ

Ø δ(q,a)=p: nghĩa là ở trạng thái q, đọc a, chuyển sang trạng thái p

3.2. Biểu diễn các hàm chuyển trạng thái

v Dùng bảng: sử dụng ma trận δ có:

- Chỉ số hàng: trạng thái
- Chỉ số cột: ký hiệu vào
- Giá trị tại hàng q, cột a là trạng thái p, sao cho δ(q,a)=p

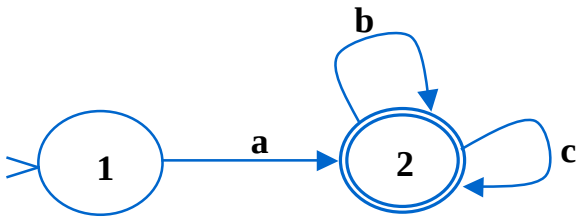
Ví dụ: có hàm chuyển của một Otomat như sau: δ(1,a)=2, δ(2,b)=2, δ(2,c)=2

δ	a	b	c
1	2		
2		2	2

v Hình vẽ:

- mỗi trạng thái q∈Q được đặt trong các vòng tròn.
- Trạng thái bắt đầu q0 có thêm dấu '>' ở đầu.
- Trạng thái kết thúc q∈F được đặt trong vòng tròn kép.
- Các cung nối từ trạng thái q sang trạng thái p có mang các nhãn a∈Σ, có nghĩa δ(q,a)=p

Ví dụ: có hàm chuyển của một Otomat như sau: δ(1,a)=2, δ(2,b)=2, δ(2,c)=2



- **Nhận xét:**

- Biểu diễn hàm chuyển trạng thái bằng hình vẽ có ưu điểm hơn. Trong hình vẽ ta xác định đầy đủ tất cả các thành phần của Otomat.
- Biểu diễn bằng bảng xác định hàm chuyển trạng thái, tập các trạng thái, bộ chữ vào nhưng không phân biệt được trạng thái bắt đầu và trạng

3.3. Hoạt động của Otomat

- Đọc các ký hiệu của xâu vào từ trái sang phải, bắt đầu từ trạng thái q0.
- Mỗi bước đọc một ký hiệu thì chuyển sang trạng thái theo δ. Có thể đọc xong hay không đọc xong xâu vào.
- Đọc xong xâu vào đến một trạng thái p∈ F thì xâu vào được đoán nhận (xâu đúng).
- Đọc xong xâu vào mà rơi vào trạng thái p∉ F thì xâu vào không được đoán nhận.
- Không đọc xong xâu vào (do δ rơi vào điểm không xác định) thì xâu vào không được đoán nhận.

3.4. Ví dụ: Xác định Otomat đoán nhận số nhị phân. M(Σ, Q, δ, q0, F)

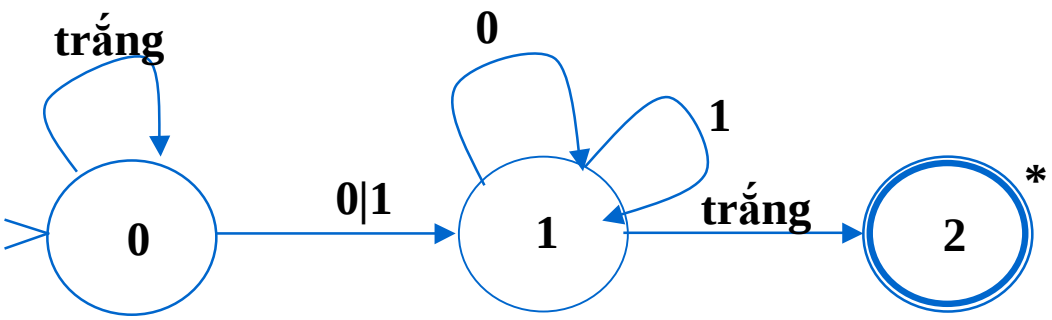
Σ: {0, 1, trắng}

Q: {0, 1, 2}

q0: 0

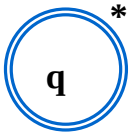
F : {2}

δ: δ(0,trắng)=0, δ(0,0)=1, δ(0,1)=1, δ(1,0)=1, δ(1,1)=1, δ(1,trắng)=2



4.Lập bộ phân tích từ vựng

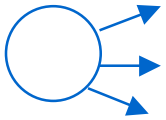
Ngoài các hình qui ước của Otomat thông thường lại có thêm:



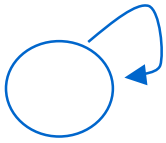
Trạng thái kết thúc và trả lui ký tự vừa đọc

4.1. Phương pháp mô phỏng

- **Mỗi trạng thái: tương ứng với một đoạn chương trình**
- **Nối tiếp các trạng thái: nối tiếp 2 đoạn chương trình tương ứng**
-
-



Lệnh rẽ nhánh



Lệnh lặp

```
Max=10; {độ dài tối đa của 1 danh biểu}
Type
Loaikytu=(conso,cham, Ttu, trang, Ccai);
Loaituto=(nguyen,thuc,Toantu,
Danhbieu);
Xau=Array[1..max] of char ;
Var       Kytutiep:char;
Procedure Dockytu(var c:char);
…{Đọc ký tự tiếp, ký tự này luôn luôn
được đọc trước}
Function LoaiKT(c:char):Loaikytu;
…{Cho biết loại của ký tự c}
Procedure Baoloi;
…{Cho một thông báo lỗi}
```

```
Procedure Tuvung(var ma:Loaituto;var
x:xau);
  Var i:0..max;
  Begin
    For i:=1 to max do x[i]:=’’;
    I:=0;
    While loaikytu(kytutiep)=trang do
          Dockytu(kytutiep);
    Case loaikytu(kytutiep) of
      Conso:  Begin
        Repeat  I:=i+1;
            x[i]:=kytutiep;
            Dockytu(kytutiep);
        Until
Loaikytu(kytutiep)<>conso;
        Ma:=nguyen;
```