


# Nhập môn Chương Trình Dịch

Hoàng Anh Việt  
Viện CNTT&TT - ĐHBKHN



# Chương I: Giới thiệu

Chương trình dịch

Nguyên lý cơ bản của Ngôn ngữ lập trình  
và Thiết kế cấu tạo của chương trình  
dịch

# Chương trình dịch

## Vấn đề

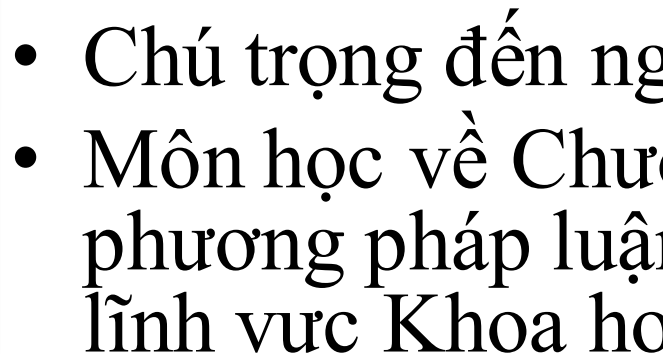
Người dùng : sử dụng basic, pascal ,c,java.....      Ngôn ngữ cấp cao

Diễn  
dịch



Biên dịch

Chỉ hiểu được mã nhị phân biểu thị chỉ lệnh và dữ liệu

- 
- Chú trọng đến nguyên lý và kỹ thuật liên quan.
  - Môn học về Chương trình dịch cung cấp phương pháp luận giải quyết những vấn đề của lĩnh vực Khoa học máy tính ở cấp độ vĩ mô.

### **Đặt vấn đề, giải quyết vấn đề**

- Môn học này có sự kết hợp của Lý luận và thực hành.

# Nội dung môn học

Chương I: Giới thiệu

Chương II: Văn phạm phi ngữ cảnh

Chương III Phân tích từ vựng

Chương IV Phân tích ngữ pháp

Chương V Phân tích ngữ nghĩa và sinh mã trung gian

Chương VI Sinh mã mục tiêu.

## §1.1 Lịch sử phát triển của Kỹ thuật dịch

- 1820—1850 Charles Babbage người Mỹ phát minh ra chiếc máy vi tính đầu tiên.
- Những năm 30 của thế kỷ này nhà toán học người Anh Turing đã đề xuất ra khái niệm máy Turing. Máy Turing trở thành mô hình toán học của máy tính hiện đại.
- 1946 A. Aiken của trường Đại học Harvard đã thiết kế thành công máy MARKI với khả năng tự động điều khiển đọc mã, trở thành chiếc máy tự động đầu tiên trên thế giới.
- Năm 1946 Chiếc máy tính điện tử đầu tiên (ENIAC) ra đời tại Mỹ.

# Sự hình thành ngôn ngữ.

- Ngôn ngữ máy và Hợp ngữ

Ngôn ngữ máy - Ngôn ngữ ký hiệu hợp ngữ - Ngôn ngữ Macro

- FORTRAN, ALGOL và COBOL

- FORTRAN(FORmulaTRANslation) Ngôn ngữ diễn dịch công thức, là ngôn ngữ cấp cao đầu tiên ra đời vào những năm 50

1954-1959 FORTRAN 0 Sự ra đời của FORTRAN 0 và Hệ thống biên dịch đánh dấu sự hình thành của kỹ thuật dịch.

# Sự hình thành ngôn ngữ.

## ➤ ALGOL(ALGOkithmic Language)

Ngôn ngữ Đại số toán học

ALGOL58    ALGOL60

SỬ dụng ký pháp BNF: hình thức hóa ngôn ngữ, tạo tiền đề cho lĩnh vực nghiên cứu về phân tích ngữ pháp của ngôn ngữ.

## ➤ COBOL

Đề xuất phương pháp mô tả dữ liệu độc lập với máy tính cụ thể, tạo tiền đề phát triển của hệ quản trị cơ sở dữ liệu.



# Sự hình thành ngôn ngữ.

- PASCAL

Do một tiểu nhóm của ALGOL60 phát minh chủ yếu dùng vào việc giảng dạy, và viết một số phần mềm hệ thống, PASCAL kế thừa ưu việt của ALGO60.

PASCAL có một vai trò rất lớn trong lịch sử phát triển của ngôn ngữ lập trình hướng cấu trúc.

- Ada

- PROLOG

- IDE(Interactive Development Environment)

# Sự hình thành ngôn ngữ.

- SIMULA 67

Từ ALGOL60 phát triển lên, phát triển class đánh dấu sự phát triển của dữ liệu trừu tượng

- Smalltalk 72-80

Kiến tạo giao diện người dùng: View

Đơn vị của chương trình là: Class. Đánh dấu sự thành thực của lập trình hướng đối tượng.

- C++

Cấu trúc và hướng đối tượng kết hợp toàn mỹ. Hệ thống biên dịch của C tính năng cao nhưng tính oan toàn còn hạn chế.

- Java Thế hệ ngôn ngữ phát triển cho Web, tính an toàn được nâng cao (so sánh với C)

## §1.2 Ngôn ngữ lập trình

- Ngôn ngữ cấp thấp: Phụ thuộc hệ máy Ngôn ngữ máy hợp ngữ
- Ngôn ngữ cấp cao Không phụ thuộc hệ máy cụ thể
  - FORTRAN——Ngôn ngữ thuật toán
  - BASIC——Ngôn ngữ tương tác
  - PASCAL——Ngôn ngữ cấu trúc
  - SQL —— Ngôn ngữ tìm kiếm
- CSDL
- 
- Ngôn ngữ cấp trung C Có chức năng của ngôn ngữ cấp thấp và cấp cao.

# 1. Ưu điểm của Ngôn ngữ cấp cao

- ❑ Hiệu suất cao (Lập trình), tính tương thích tốt
- ❑ Không cần quan tâm hệ máy cụ thể, như cấp phát bộ nhớ.
- ❑ Có cấu trúc dữ liệu phong phú
- ❑ Gần gũi ngôn ngữ tự nhiên, dễ học dễ nắm bắt.

## 2. Định nghĩa của Ngôn ngữ lập trình cấp cao.

Ngôn ngữ lập trình cấp cao, nói một cách đơn giản là một ngôn ngữ lập trình giúp người dùng dễ hiểu, nắm bắt. Hoặc theo cách khác có đầy đủ đặc điểm sau về: phương pháp biểu đạt, quy ước, quy tắc.

## 2. Định nghĩa ngôn ngữ lập trình (tiếp)

- (1) Không yêu cầu người lập trình phải nắm được những tri thức liên quan hệ máy cụ thể (Thanh ghi, dữ liệu biểu thị, I/O). Đặc trưng này không xét đến hiệu năng.
- (2) Độc lập với bất kỳ hệ máy cụ thể nào, dễ sử dụng để viết ra chương trình có thể chạy trên nhiều hệ máy khác nhau.
- (3) Mã nguồn được viết bằng ngôn ngữ lập trình cấp cao có thể được dịch thành ngôn ngữ máy chạy được trên các hệ máy khác nhau tương ứng.
- (4) Ngôn ngữ cấp cao có thể mô tả vấn đề một cách tự nhiên, là một ngôn ngữ hướng giải pháp.

# §1.3 Chương trình biên dịch, Chương trình Hợp Ngữ, Chương trình Diễn dịch

## 1. CT Phiên dịch

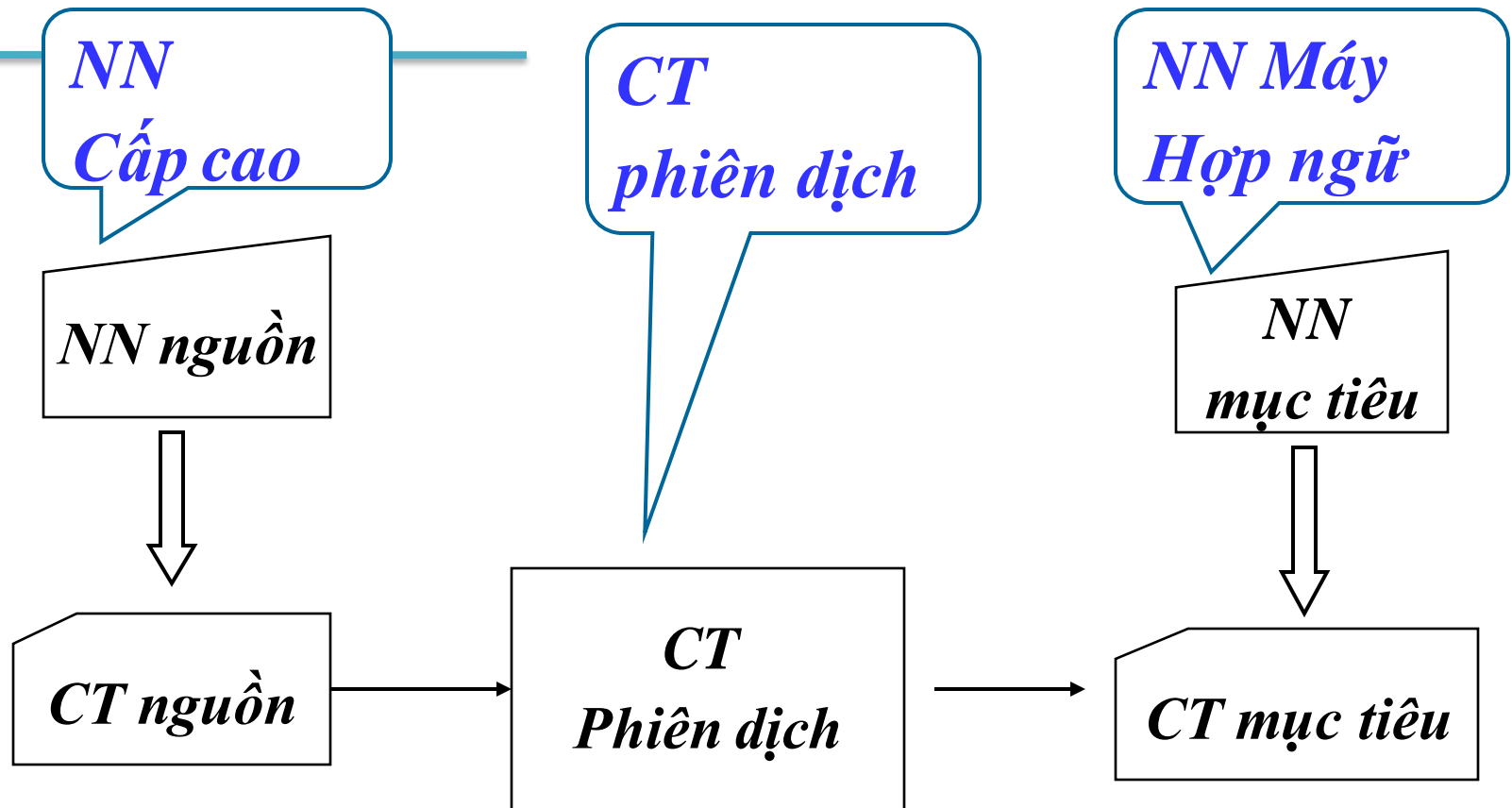
Là một chương trình có thể phiên dịch mã CT viết bằng ngôn ngữ A sang mã CT tương đương viết bằng ngôn ngữ B.

Ngôn ngữ A: *Ngôn ngữ nguồn của.*

Ngôn ngữ B: *Ngôn ngữ mục tiêu.*

Mã CT viết bằng ngôn ngữ nguồn: *Mã nguồn.*

Mã CT viết bằng ngôn ngữ mục tiêu: *Mã mục tiêu.*





## **2. Chương trình Biên dịch**

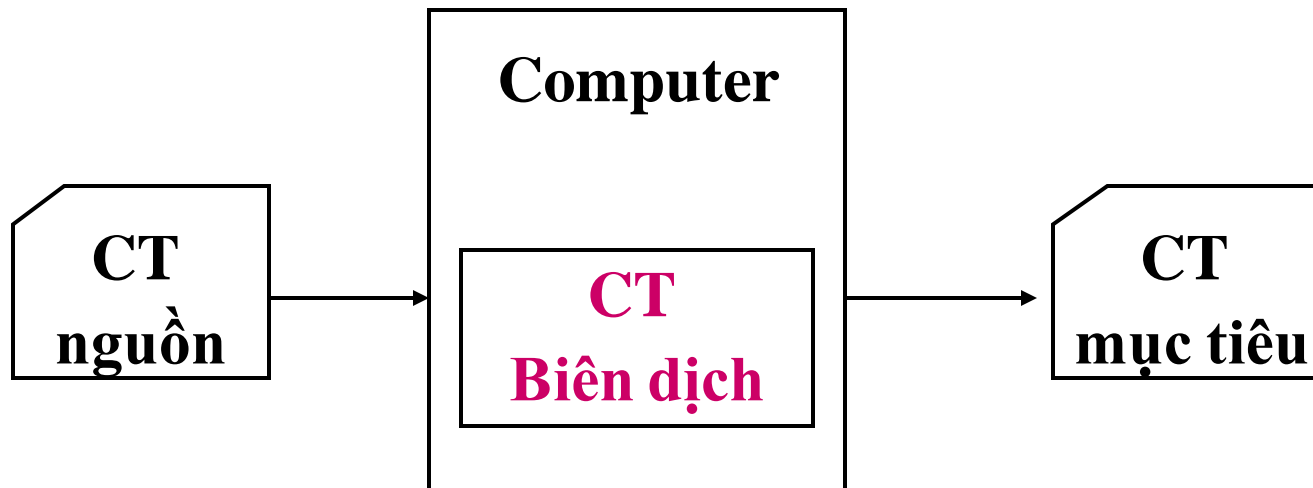
Nếu ngôn ngữ nguồn là ngôn ngữ cấp cao, ngôn ngữ mục tiêu là ngôn ngữ cấp thấp (ngôn ngữ máy, hợp ngữ), thì CT phiên dịch gọi là CT Biên dịch

## **3. Chương trình Hợp ngữ**

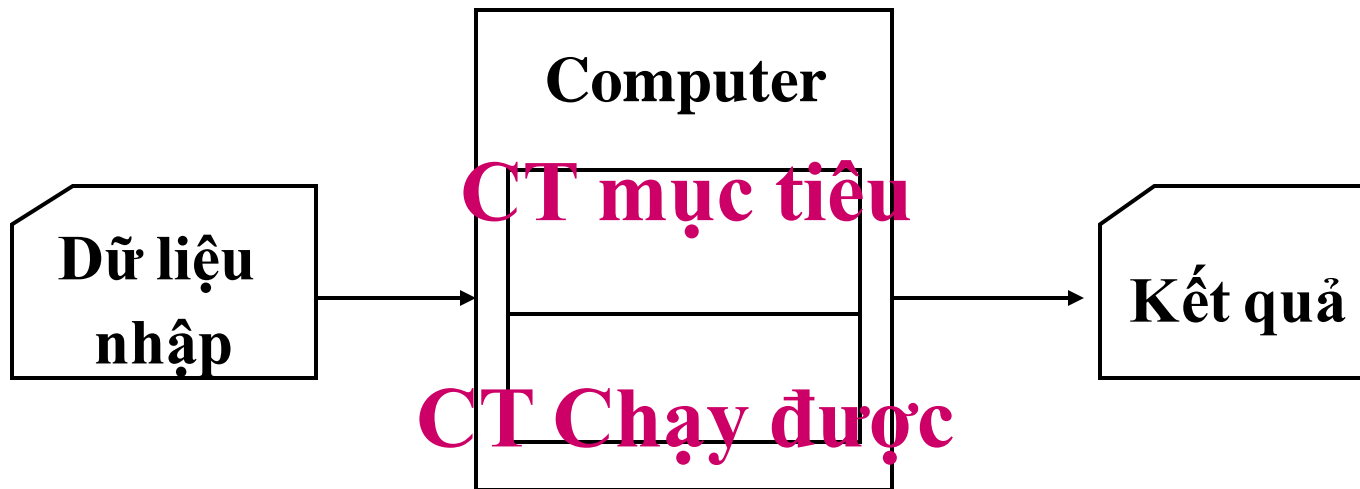
Nếu ngôn ngữ nguồn là Hợp ngữ, ngôn ngữ mục tiêu là ngôn ngữ Máy, thì CT phiên dịch gọi là CT Hợp dịch

# Thuyết minh :

- CT Biên dịch, Ngôn ngữ nguồn và Máy vi tính là những khái niệm liên quan mật thiết với nhau:
  - Ngôn ngữ nguồn khác nhau có CT Biên dịch khác nhau.
  - Một ngôn ngữ nguồn có thể có nhiều CT Biên dịch khác nhau.
- Giai đoạn biên dịch sinh ra CT mục tiêu không phải CT mã máy, mà là CT Hợp ngữ, quá trình thực thi CT nguồn phân thành 3 giai đoạn: Biên dịch, Hợp dịch, Thực thi (chạy)



**Giai đoạn Biên dịch**



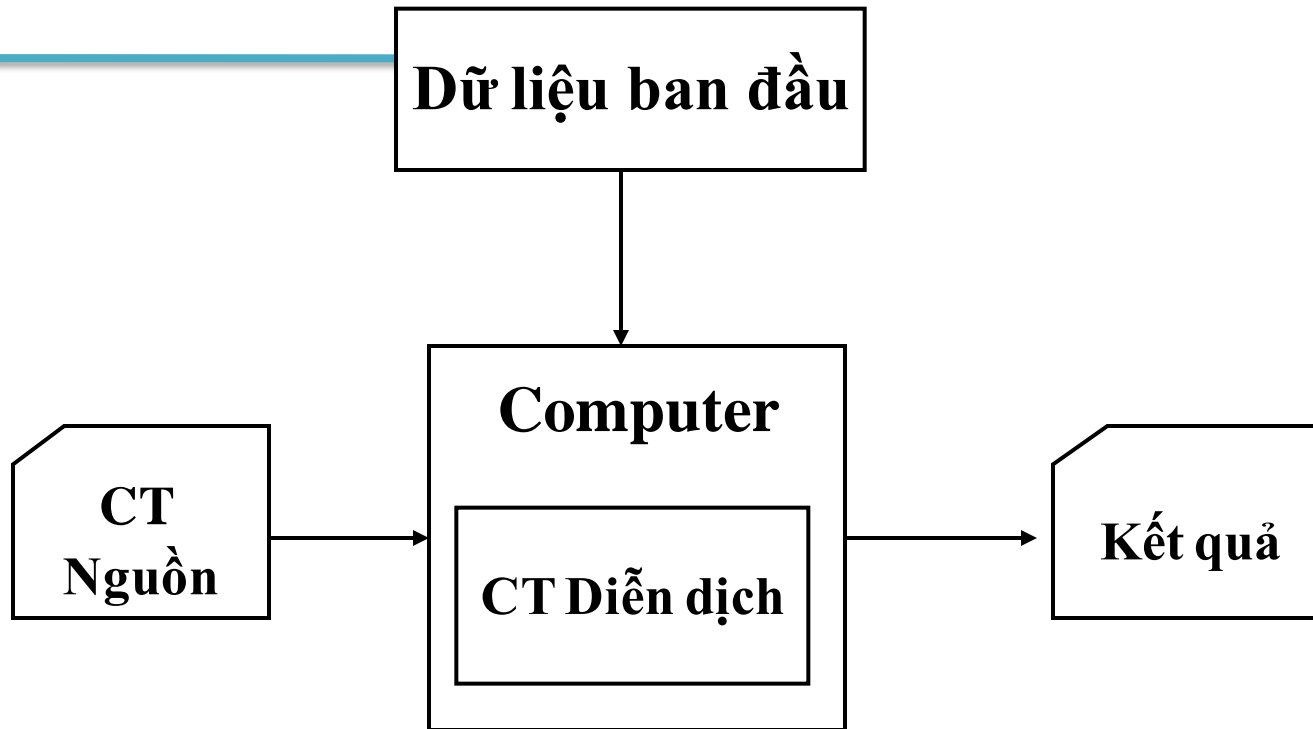
**Giai đoạn thực thi**

## 4. Chương trình Diễn dịch

Dựa vào thứ tự động của câu lệnh tiến hành phân tích tuần tự đồng thời thực thi ngay câu lệnh cho đến khi CT kết thúc (không còn câu lệnh nào)

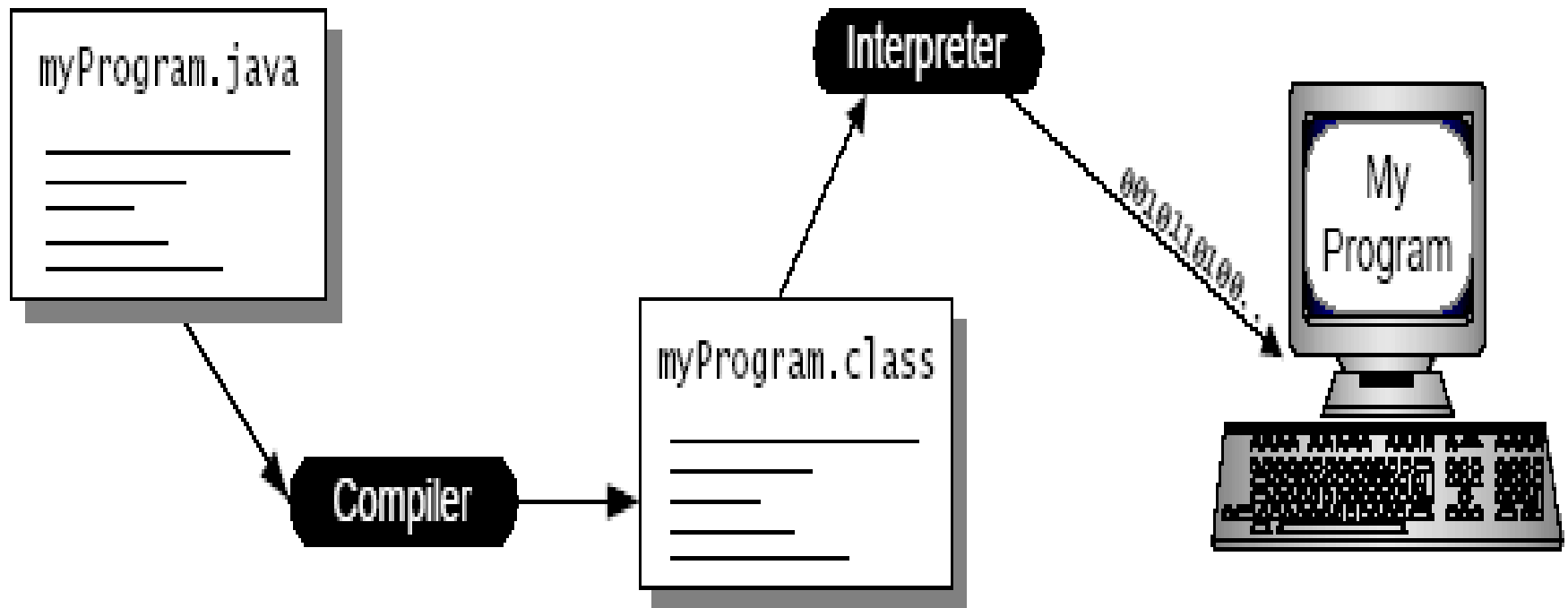
CT Diễn dịch vừa biên dịch vừa thực thi, không sinh ra CT mục tiêu. Vận hành theo phương thức tương tác (với người dùng), thuận tiện cho debug, nhưng hiệu năng thấp.

CT Biên dịch sinh ra CT mục tiêu, sau khi liên kết trở thành File chạy, tất cả công việc biên dịch được hoàn thành trước khi thực thi (chạy CT).  
Hiệu năng cao.



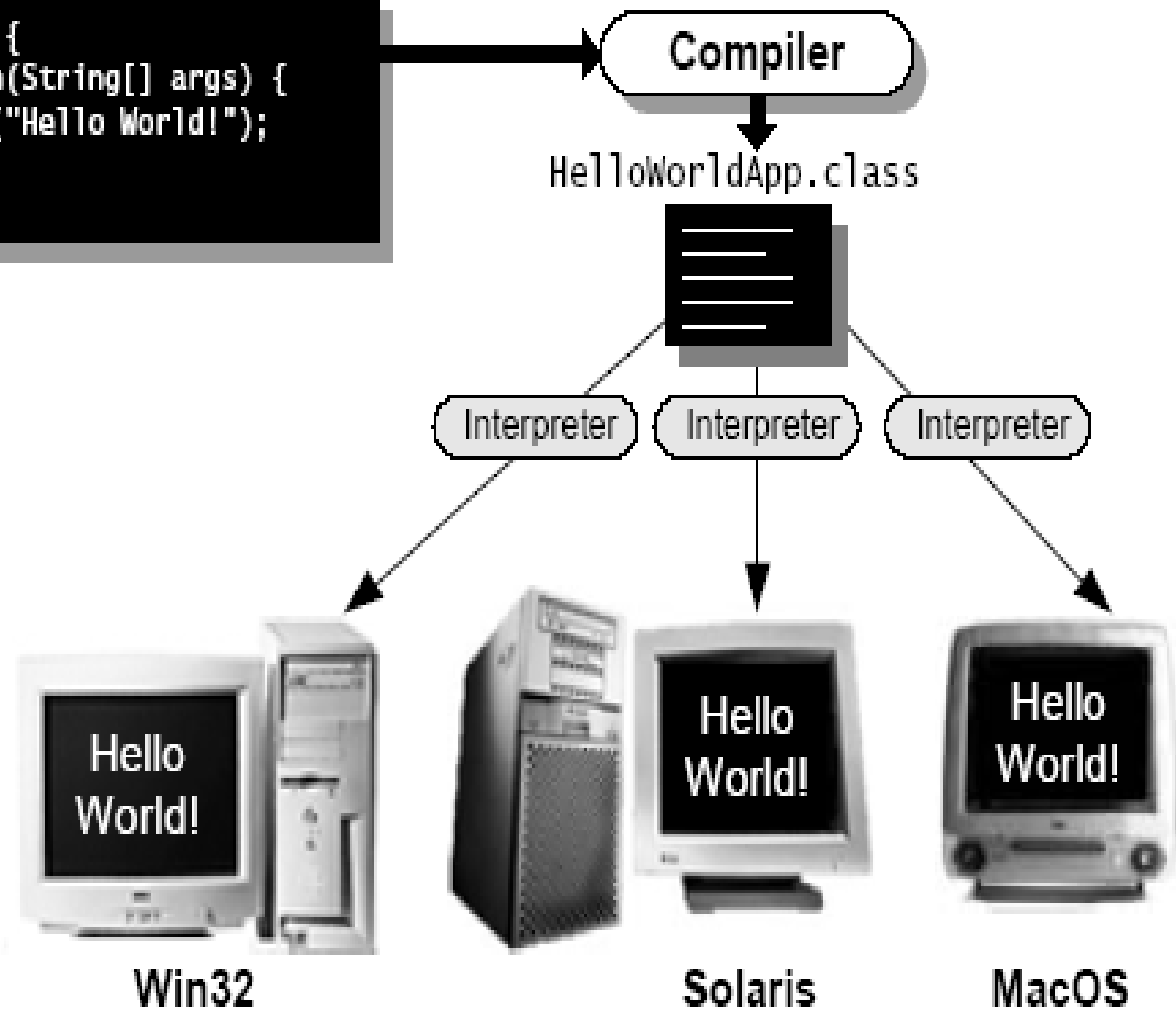
**CT Diễn dịch**

# Quá trình biên dịch của Ngôn ngữ Java



HelloWorldApp.java

```
public class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```





## §1.4 Khái quát quá trình Biên dịch

### ● Quá trình Biên dịch điển hình phân thành 5 giai đoạn:

- Phân tích từ vựng
  - Phân tích ngữ pháp (cú pháp)
  - Phân tích ngữ nghĩa và sinh mã trung gian
  - Ưu hóa mã
  - Sinh mã mục tiêu (ưu hóa mã mục tiêu)
- } Phân tích mã nguồn  
(analysis)
- } Tổng hợp  
mã mục tiêu  
(synthesis)

# 1. Phân tích từ vựng (Scanner)

## —Bộ phân tích từ vựng

Nhiệm vụ chủ yếu: quét sâu (string) mã nguồn của CT nguồn, phân tách thành các *đơn từ* là đơn vị nhỏ nhất của ngữ pháp ngôn ngữ mà mang một ý nghĩa độc lập nhất định

**Vấn đề:**

**Trong ngôn ngữ đâu là đơn từ ?**

```
Program ged(input,output) ;  
Var num1,num2:integer;  
Begin  
    read(num1,num2) ;  
    num2:=num1*2+num2 ;  
    Writeln(num2)  
End.
```

**program**      **Key word**

**ged**              **Định danh**

**(**              **Dấu phép toán**

**input**      **Tiêu chuẩn ĐD**

**integer**      **Tiêu chuẩn ĐD**

**,**              **Dấu phân cách**

**num1**              **Định danh**

**:=**              **Phép toán**

## VD. Một câu lệnh của Pascal

if A=B then X :=Y ;

Kết quả  
phân tích  
từ vựng:

<b>If</b>	<b>Từ khóa</b>
<b>A</b>	<b>Biến</b>
<b>=</b>	<b>Phép toán</b>
<b>B</b>	<b>Biến</b>
<b>Then</b>	<b>Từ khóa</b>
<b>X</b>	<b>Biến</b>
<b>:=</b>	<b>Phép toán</b>
<b>Y</b>	<b>Biến</b>
<b>;</b>	<b>Phân cách</b>

# Thuyết minh

- Quy tắc phân tích dựa vào quy tắc từ vựng của ngôn ngữ.
- Đơn từ : Hằng số, Tên biến, Từ khóa, Phép toán...。
- Đơn từ có thể được đánh số bằng số nguyên, hoặc theo một cách khác nào đó.
- QT phân tích từ vựng phải chỉ ra được các đơn từ sai quy tắc, sai quy ước.

## 2. Phân tích ngữ pháp (Parser)

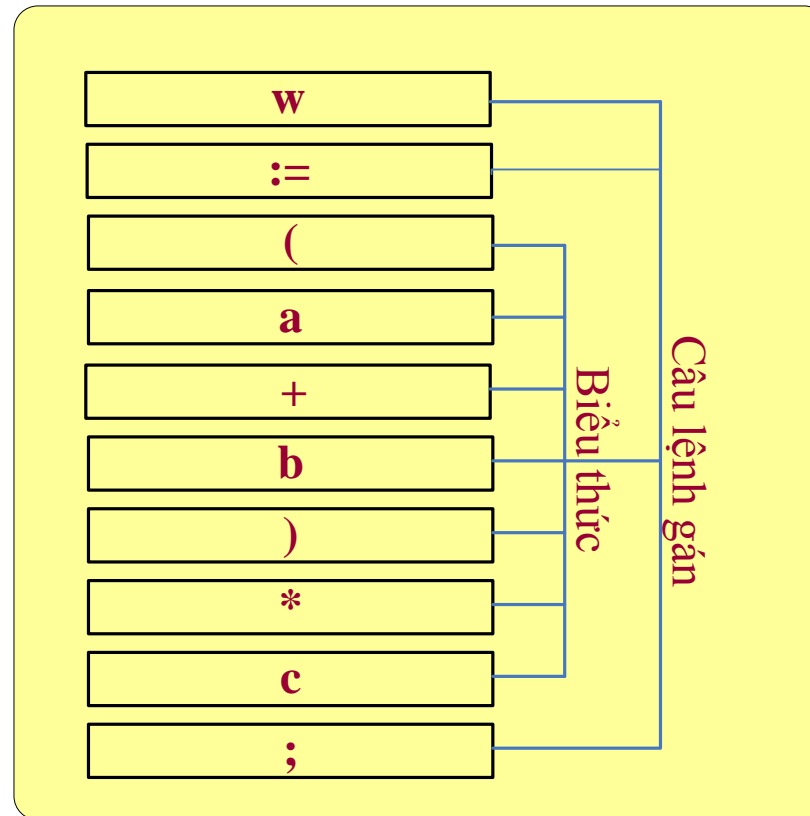
### —Bộ phân tích ngữ pháp

Nhiệm vụ chủ yếu: nhận đơn từ từ kết quả của quá trình phân tích từ vựng, nhóm các đơn từ thành các **lớp ngữ pháp**

- Quy tắc của phân tích ngữ pháp là văn phạm (quy tắc ngữ pháp ) của ngôn ngữ
- **Lớp ngữ pháp** : Biểu thức, câu lệnh , CT con....
- Phân tích ngữ pháp phải chỉ ra được các câu lệnh sai ngữ pháp.

## VD2 Một câu lệnh của Pascal

**w := (a + b) \* c ;**



### 3. Phân tích ngữ nghĩa và sinh mã trung gian (semantic routine)

#### —Bộ phân tích ngữ nghĩa

Nhiệm vụ chủ yếu: xác định ý nghĩa (ngữ nghĩa) mã nguồn, đối với những lớp ngữ pháp khác nhau tiến hành phiên dịch sơ bộ, bao gồm ***phân tích tĩnh ngữ nghĩa và sinh mã trung gian***

**Phân tích tĩnh ngữ nghĩa** : đối với lớp ngữ pháp khác tiến hành kiểm tra ngữ nghĩa (biến đã được khai báo, định nghĩa hay chưa, kiểu dữ liệu có thống nhất hay không...) 。

**Sinh mã trung gian** : tiến hành phiên dịch sơ bộ, sinh mã trung gian (intermediate representation, IR) 。



## Thuyết minh:

- Cơ sở của PT ngữ nghĩa là quy tắc ngữ nghĩa。
- Mã trung gian là một loại mã có kết cấu đơn giản hàm ý (ngữ nghĩa) rõ ràng, có đặc điểm độc lập với phần cứng (hệ máy), ở mức độ nào đó tương tự hệ thống chỉ lệnh của hệ máy, nên rất dễ dàng chuyển từ mã trung gian sang mã máy.
- Phân tích ngữ nghĩa và phân tích ngữ pháp là hai khái niệm khác nhau, nhưng trong quá trình biên dịch cụ thể, hai quá trình trên kết hợp khăng khít, thông thường được hoàn thành một cách đồng thời.

## VD3. Câu lệnh Pascal

**$w := (a+b)*c ;$**

**Phân tích ngữ nghĩa quyết định cộng trước nhân sau, và sinh mã trung gian**

Mã trung gian của câu lệnh trên.

(1) (+,a,b)

(2) (\*,(1),c)

(3) (:=,w,(2))

## **4. Ưu hóa mã (Optimizer)**

### **—Bộ tối ưu mã**

Nhiệm vụ chủ yếu: đối với mã trung gian tiến hành biến đổi tương đương về mặt thuật giải để thu được mã mục tiêu hữu hiệu hơn.

- Hữu hiệu chỉ: có hiệu lực về không gian và thời gian tính toán.
- Quá trình tối ưu hóa có thể hoàn thành trước hoặc sau giai đoạn sinh mã mục tiêu.

## 5. Sinh mã mục tiêu (code generator)

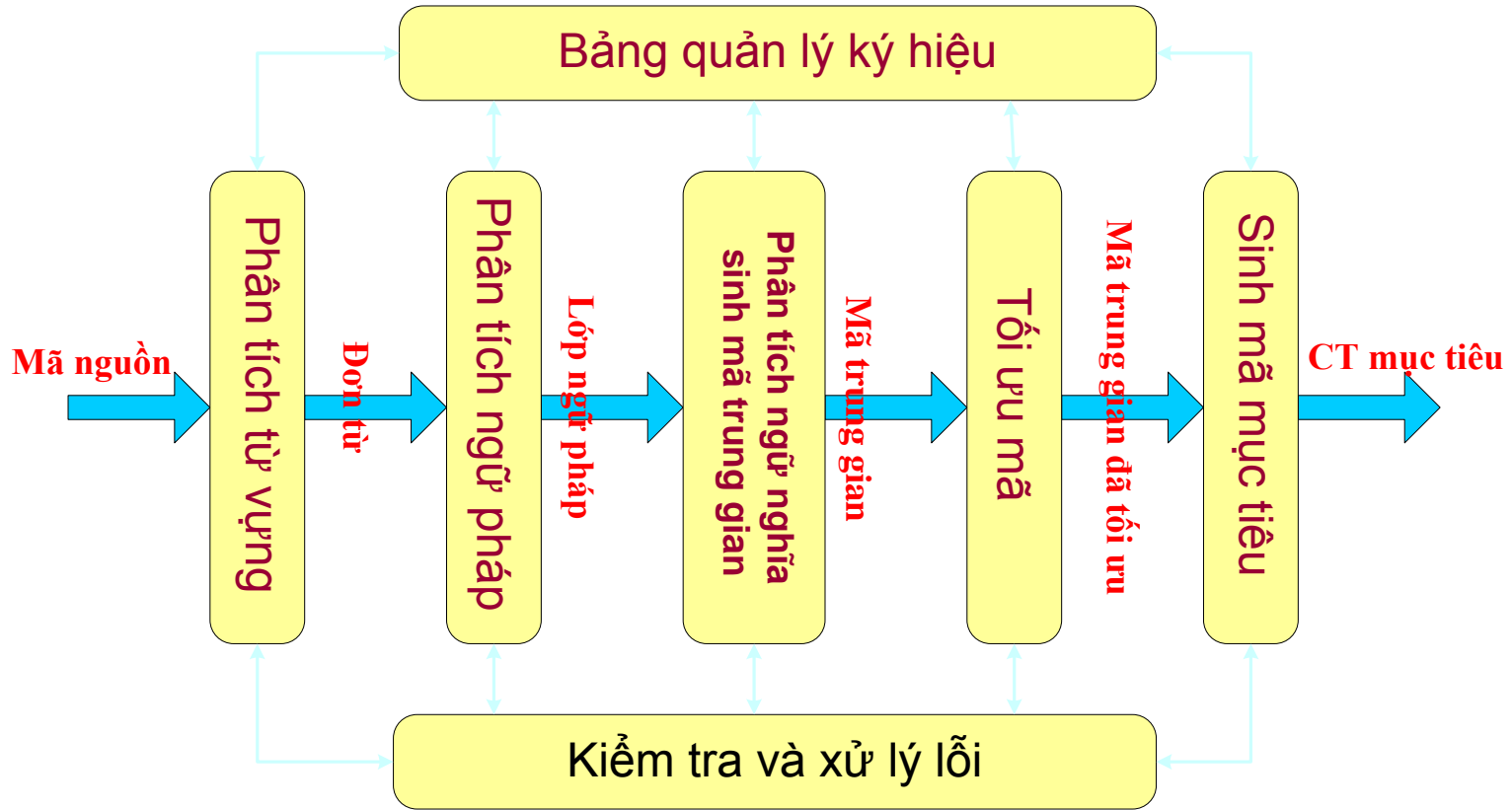
### —Bộ sinh mã

Nhiệm vụ chủ yếu: dựa vào hệ máy cụ thể biến đổi mã trung gian thành ngôn ngữ máy hay hợp ngữ của hệ máy tương ứng.

### Thuyết minh

- Không phải tất cả các chương trình biên dịch đều tuân theo mô hình 5 giai đoạn.
- Một CT biên dịch đầy đủ còn bao hàm, **bảng quản lý ký hiệu (symbol) và xử lý lỗi.**

## *Cấu thành của một CT Biên dịch điển hình*



**Vai trò của mã trung gian: dễ tương thích, tiện ưu hóa, dễ sinh mã mục tiêu**

## II. Cấu thành của CT Biên dịch

- Bộ phân tích từ vựng
- Bộ phân tích ngữ pháp
- Bộ phân tích ngữ nghĩa và sinh mã trung gian
- Bộ tối ưu mã
- Bộ sinh mã mục tiêu
- Bộ quản lý bảng ký hiệu
- Bộ xử lý lỗi

# 1.Quản lý bảng ký hiệu

Bảng ký hiệu là một cấu trúc dữ liệu lưu giữ các định danh và thuộc tính tương ứng, phục vụ cho quá trình phân tích ngữ pháp và sinh mã trung gian.

**Định danh** : tên biến, tên hàm số, tên thủ tục...

**Thuộc tính** : cấp phát bộ nhớ cho định danh, định kiểu, không gian sống (scope)

- *Thiết kế một cách hợp lý bảng quản lý ký hiệu là vấn đề quan trọng của quá trình xây dựng chương trình dịch*

## 2 . Kiểm tra và xử lý lỗi

Mỗi giai đoạn biên dịch đều có thể phát sinh lỗi, phải lập tức xử lý để công tác biên dịch có thể tiếp tục, đồng thời tiếp tục kiểm soát lỗi có thể có ở các giai đoạn sau.

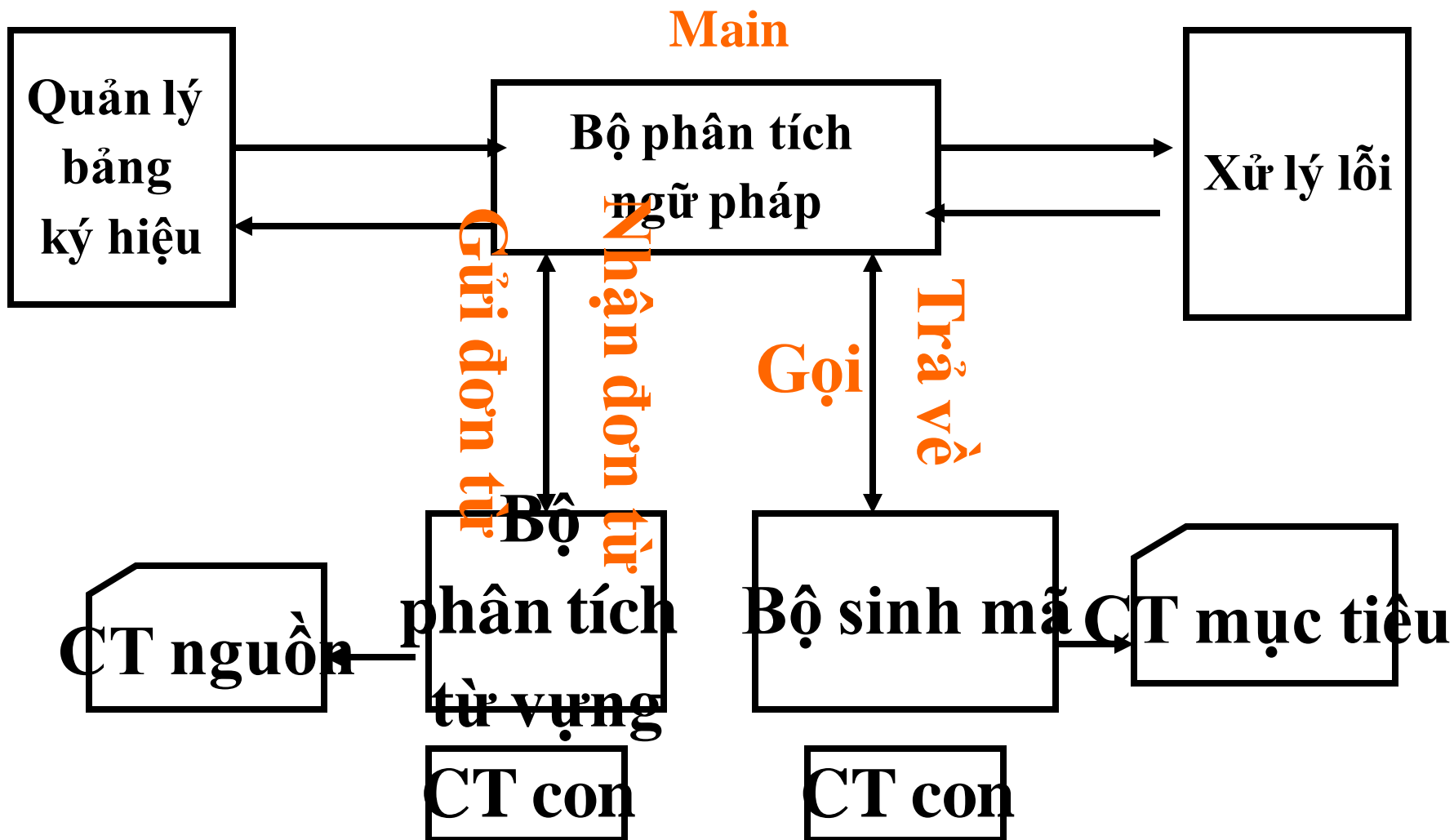
- Thông thường giai đoạn phân tích ngữ pháp, ngữ nghĩa có thể tìm ra đại bộ phận lỗi.
- Một CT biên dịch tốt phải là CT tìm được các loại lỗi khác nhau trong mã nguồn, đồng thời hạn chế ảnh hưởng của lỗi đến mức độ nhỏ nhất.



### 3. Lượt

Là một chu kỳ đầy đủ của quá trình xử lý dữ liệu: chỉ quá trình quét từ ký tự đầu đến ký tự cuối của mã nguồn đồng thời tiến hành gia công, hoặc quá trình sinh ra dạng thức trung gian của mã nguồn hay mã mục tiêu.

- Có thể coi 5 giai đoạn biên dịch là một lượt
- Cũng có thể coi mỗi giai đoạn là một lượt
- Căn cứ phân lượt phụ thuộc nhiều yếu tố cụ thể.



**Kết cấu của CT Biên dịch Ngôn ngữ PL/0 (Một lượt)**

**Pha trước**

↓  
**Tiền xử lý**

↓  
**Mã nguồn**

↓  
**CT biên dịch**  
**CT mục tiêu**

↓  
**CT hợp dịch**

↓  
**Khả định vị mã máy**

**Pha sau**

↓  
**Load/Link**

↓  
**Mã máy tuyệt đối**

← {  
**Macro define**  
**Bao hàm include**  
**Phần mở rộng**

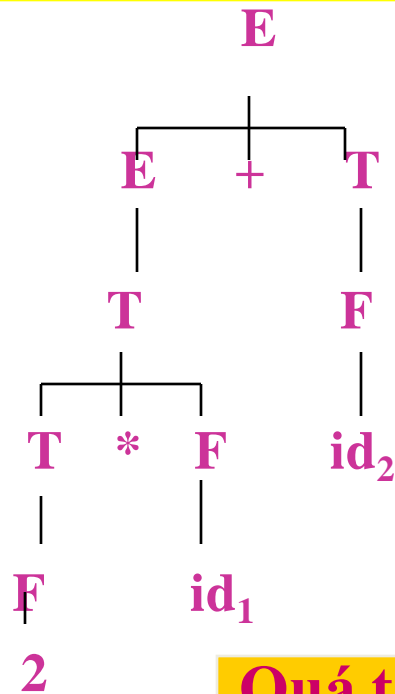
← {  
**Thư viện**  
**Mã mục tiêu tương đối**

$x := 2 * x + y$

Phân tích từ vựng

$id_1 := 2 * id_1 + id_2$

Phân tích ngữ pháp



Quá trình biên dịch một câu

Phân tích ngữ nghĩa

$T_1 = \text{int to real}(2)$   
 $T_2 := id_1 * T_1$   
 $T_3 := id_2 + T_2$   
 $id_1 := T_3$

Ưu hóa

$T_1 := id_1 * 2.0$   
 $id_1 := id_2 + T_1$

Sinh mã

MOV R<sub>2</sub> id<sub>1</sub>  
MUL R<sub>2</sub> 2.0  
MOV R<sub>1</sub> id<sub>2</sub>  
ADD R<sub>1</sub> R<sub>2</sub>  
MOV id<sub>2</sub> R<sub>1</sub>

# Kỹ thuật dịch và Kỹ thuật phần mềm

- CT biên tập ngôn ngữ hướng kết cấu
- Công cụ debug
- Công cụ Test
- Biến đổi tương đương giữa các ngôn ngữ cấp cao
- Ngôn ngữ song song, biên dịch song song
- .....

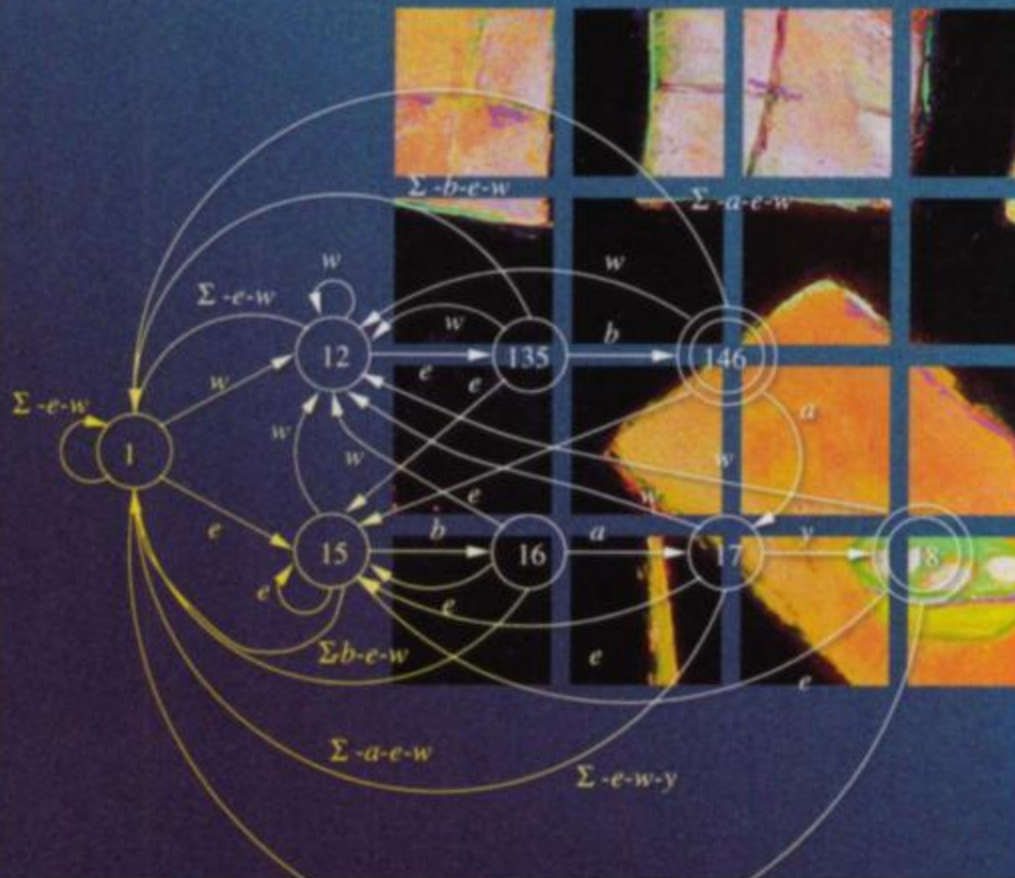
*Mục tiêu của sự phát triển của Máy (CT) biên dịch là sự nỗ lực trong giải thuật tối ưu và sinh mã.*

# Quá trình học tập cần chú ý

- Môn học phân lý thuyết và thực hành:  
    Nắm được phương pháp  
    Tư duy trừu tượng, hình thức hóa mô tả,  
    có cái nhìn tổng thể
- Nắm được cơ chế xây dựng một ngôn ngữ  
    Nguyên lý → Kỹ thuật → Cài đặt

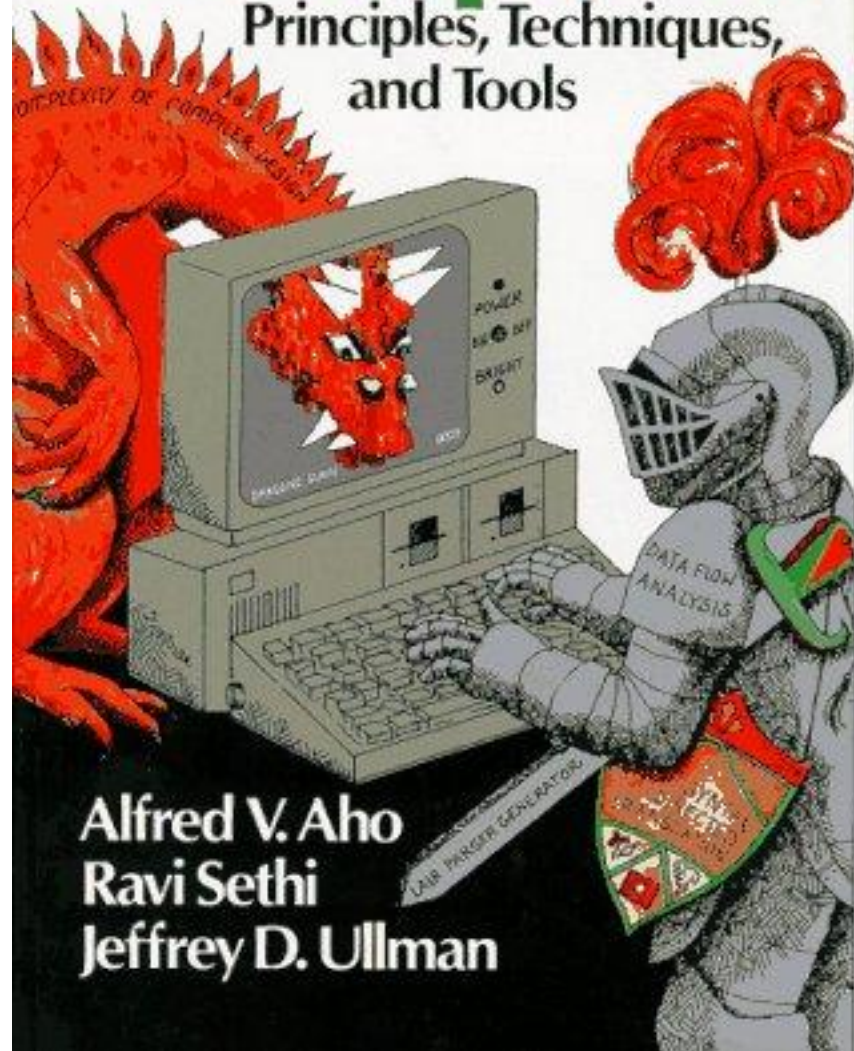
# Introduction to Automata Theory, Languages, and Computation

JOHN E. HOPCROFT  
RAJEEV MOTWANI  
JEFFREY D. ULLMAN



# Compilers

Principles, Techniques, and Tools



Alfred V. Aho  
Ravi Sethi  
Jeffrey D. Ullman

# Câu hỏi

