



MÔN HỌC

TRÌNH BIÊN DỊCH



MỤC LỤC

- **CHƯƠNG 1**

Giới thiệu về trình biên dịch

- **CHƯƠNG 2**

Trình biên dịch đơn giản

- **CHƯƠNG 3**

Phân tích từ vựng

- **CHƯƠNG 4**

Phân tích cú pháp

- **CHƯƠNG 5**

Trình biên dịch trực tiếp cú pháp

- **CHƯƠNG 6**

Xử lý ngữ nghĩa

- **CHƯƠNG 7**

Quản lý bộ nhớ trong thời gian thực thi

- **CHƯƠNG 8**

Tổ chức bảng danh biểu

- **CHƯƠNG 9**

Sinh mã đối tượng

- **CHƯƠNG 10**

Tối ưu mã



TÀI LIỆU THAM KHẢO

- 1) Alfred V.Aho, Jeffrey D.Ullman (1986). *Compilers, Principles techniques, and tools*. Addison – Wesley Publishing Company.
- 2) Alfred V.Aho, Jeffrey D.Ullman (1972). *The theory of parsing, translation and compiling*. Prentice – Hall, inc.
- 3) Terrence W. Pratt. *Programming Languages: design and implementation second edition*. Prentice – Hall International editions.
- 4) Allen I. Holub. *Compiler design in C*. Prentice – Hall International editions.
- 5) D. Gries (1976). *Compiler construction*. Springer – Verlag.
- 6) Jeffrey D. Ullman (1977). *Fundamental concepts of programming system*. Addison – Wesley Publishing Company.
- 7) Dương Tuấn Anh (1986) *Giáo trình Trình biên dịch*. Đại học Bách Khoa TP. Hồ Chí Minh.
- 8) Nicklaus Wirth (1976), *Algorithms + Data Structure = program*. Prentice – Hall International editions.
- 9) Alfred V.Aho, Jeffrey D. Ullman (1977). *Principles of compiler design*. Addison – Wesley, Reading, Mass.
- 10) Lê Hồng Sơn, *Luận văn tốt nghiệp “Xây dựng giải thuật tối ưu mã trung gian của trình biên dịch”* – Khoa CNTT Trường ĐH Bách khoa 2002.
- 11) Phan Thị Tươi (2001). *Trình Biên Dịch*. Đại học Bách Khoa TP. Hồ Chí Minh



YÊU CẦU

- Phần Lý thuyết:
SV học 42 tiết lý thuyết
- Phần Thực hành:
SV tham dự thực hành – thực hiện Bài tập Môn học 14t
(1 Bài tập Môn học / 1 SV)
- Hình thức đánh giá:
 - Kiểm tra Bài tập Môn học → Điểm TH
 - Thi viết Lý thuyết cuối kỳ → Điểm LT
- Cách tính điểm:
Điểm tổng kết môn = LT * 60% + BTTH * 40%

CHƯƠNG 1

GIỚI THIỆU VỀ TRÌNH BIÊN DỊCH

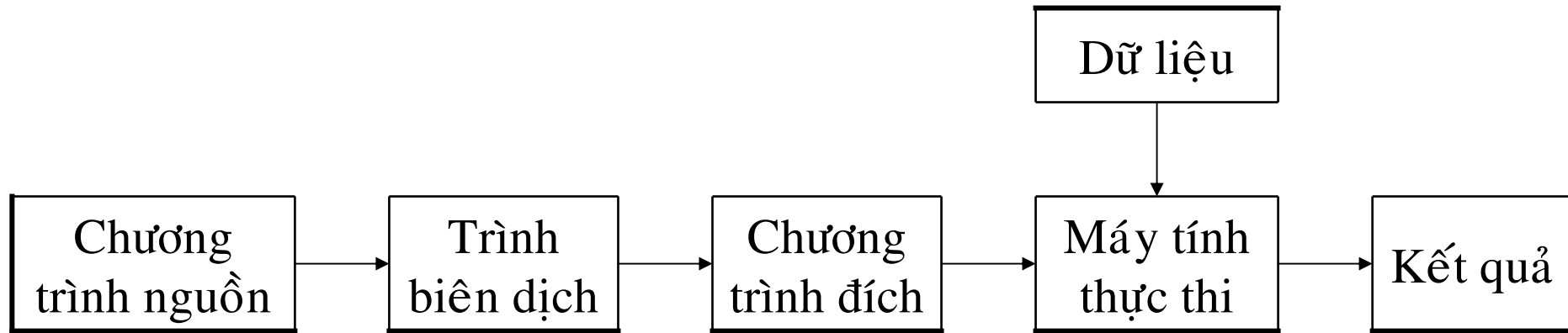
1.1. Ngôn ngữ lập trình

1. Giới thiệu

Phân loại

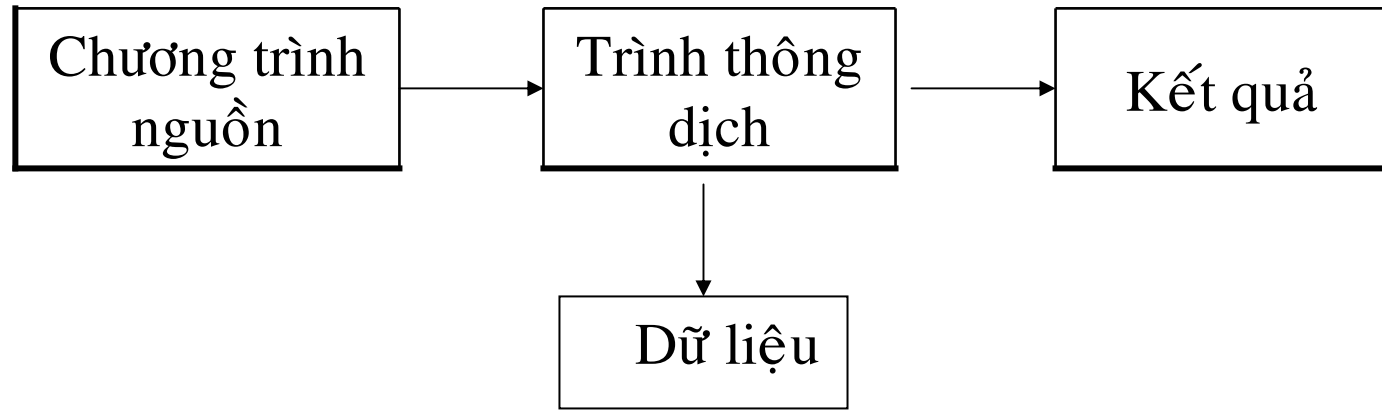
Chương trình dịch

- Trình biên dịch



Hình 1.1. Chương trình thực thi theo cơ chế dịch của trình biên dịch

- *Trình thông dịch*



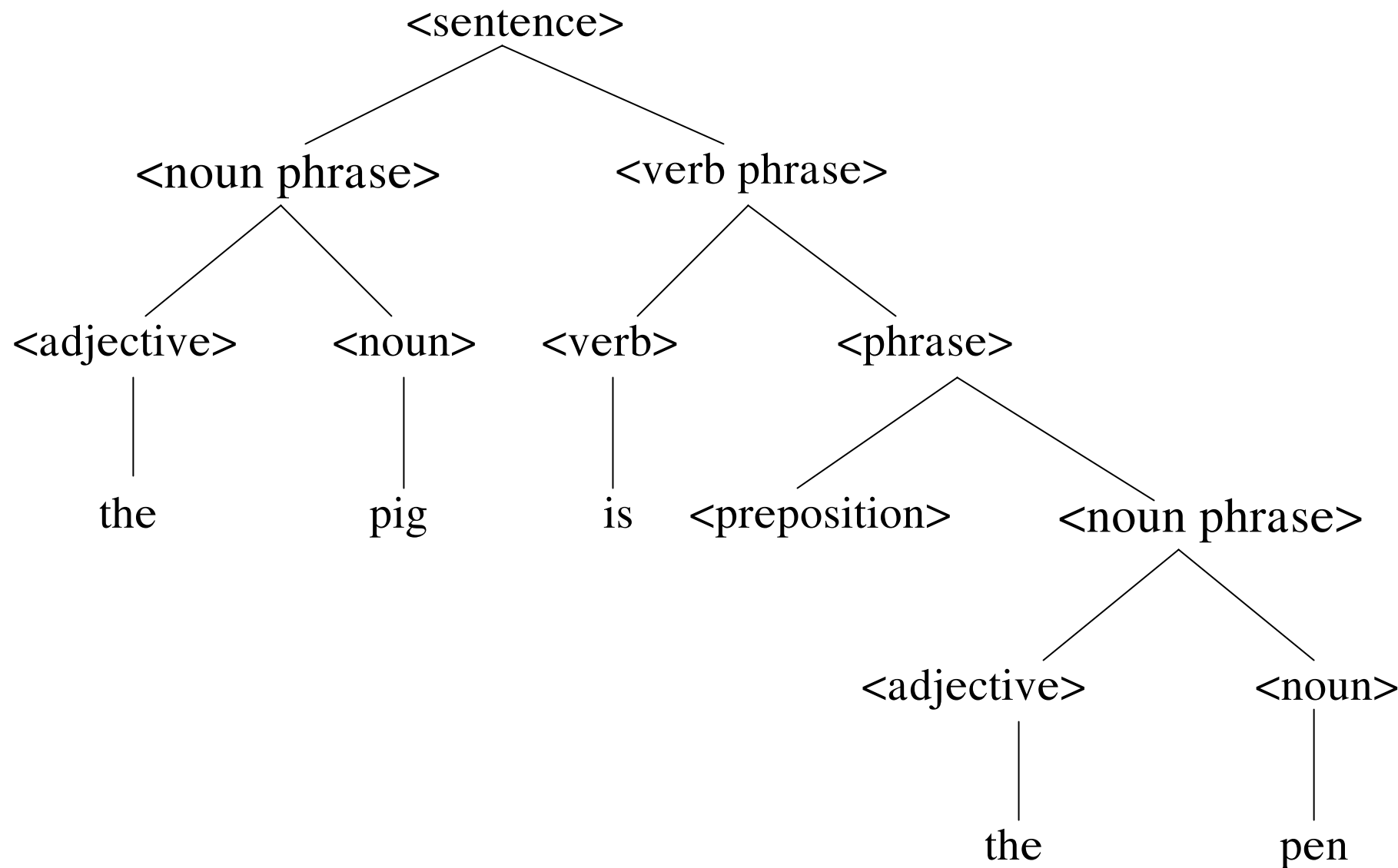
Hình 1.2. Chương trình thực thi theo cơ chế dịch của trình thông dịch

Đặc tả ngôn ngữ lập trình

1. Tập các ký hiệu cần dùng trong các chương trình hợp lệ
2. Tập các chương trình hợp lệ
3. Nghĩa của chương trình hợp lệ
 - Phương pháp thứ nhất là định nghĩa bằng phép ánh xạ. Sử dụng phép toán hàm: hàm Lamda.
 - Phương pháp thứ hai: Máy trừu tượng.
 - Phương pháp thứ ba: Tập (x,y) là sự biên dịch.

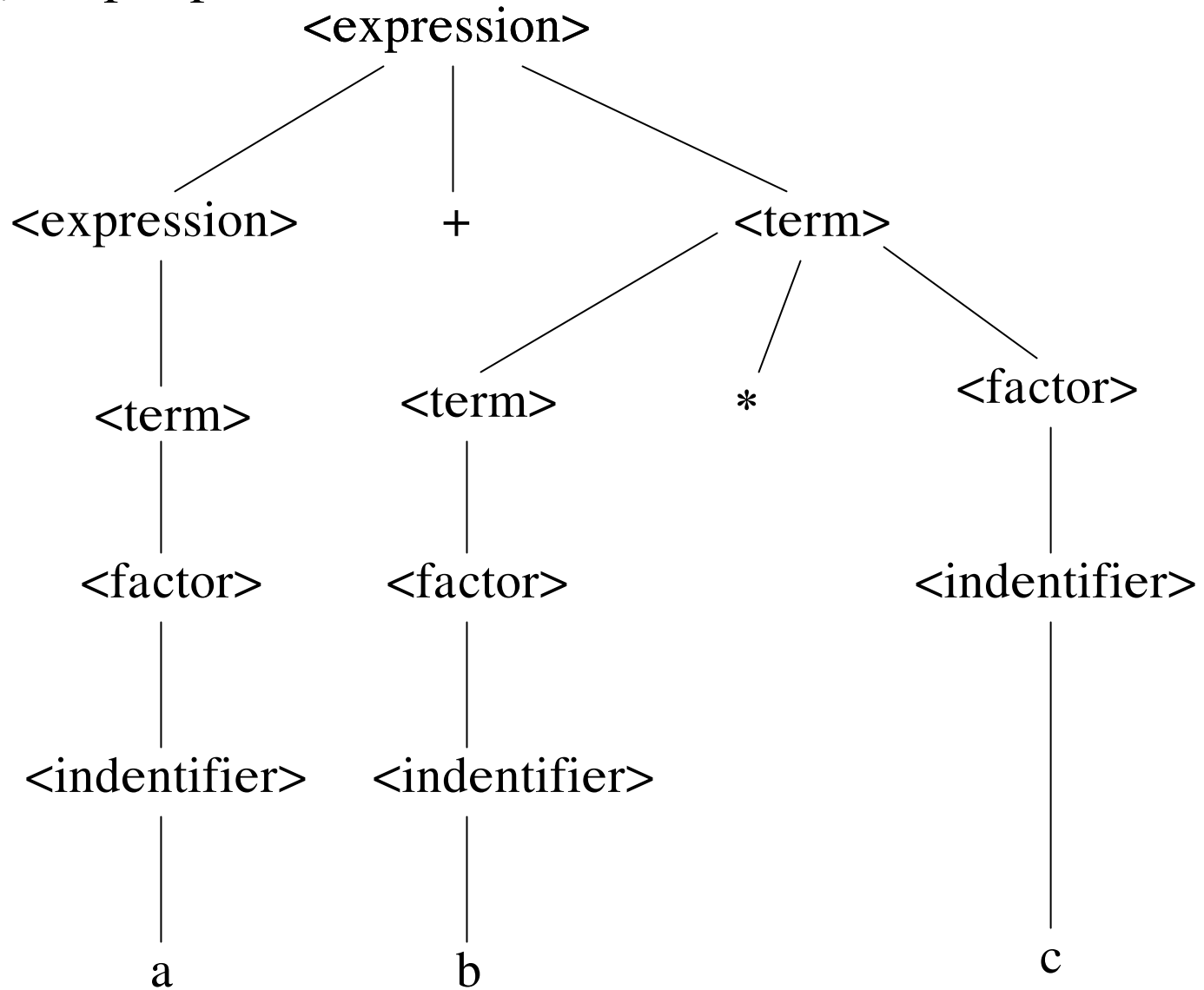
2. Cú pháp và ngữ nghĩa

- Ánh xạ cú pháp (syntactic mapping)



Hình 1.3 Cấu trúc cây của câu tiếng Anh: *the pig is in the pen*

- Ánh xạ cú pháp



Hình 1.4. Cây cú pháp của biểu thức số học $a + b * c$

1.2. Trình biên dịch

1. Các thành phần của trình biên dịch

1. Phân tích từ vựng

Nhận dạng token.

Token: danh biểu, hằng, từ khóa, các toán tử phép toán, các ký hiệu phân cách, khoảng trắng, các ký hiệu đặc biệt

Ví dụ:

$COST := (PRICE + TAX) * 65$

Đầu ra của bộ phân tích từ vựng:

$(\langle id, 1 \rangle) := ((\langle id, 2 \rangle) + (\langle id, 3 \rangle)) * (\langle num \rangle, 4)$

Viết gọn : $id1 := (id2 + id3) * num4$

Bộ phân tích từ vựng thao tác trực tiếp

Bộ phân tích từ vựng thao tác không trực tiếp

2. Bảng danh biểu

Ví dụ: $COST := (PRICE + TAX) * 65$

Bảng 1.1 Bảng danh biểu

Chỉ số	token	lexeme	Các thông tin khác
1	id	COST	biến thực
2	id	PRICE	biến thực
3	id	TAX	biến thực
4	num	65	hằng số nguyên

3. Phát hiện và thông báo lỗi

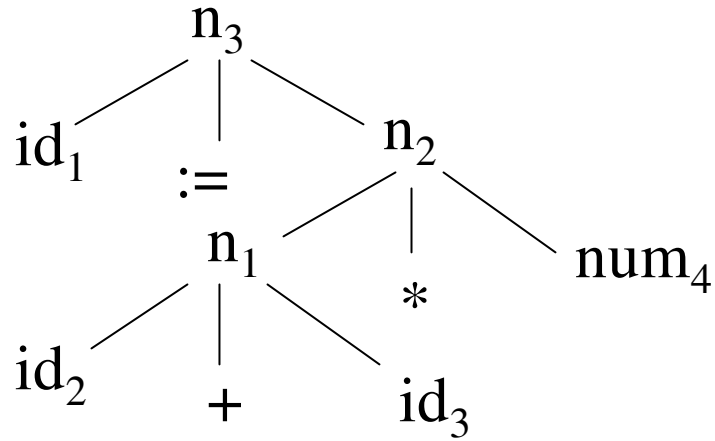
4. Phân tích cú pháp

Ví dụ: $COST := (PRICE + TAX) * 65$

Kết quả phân tích từ vựng:

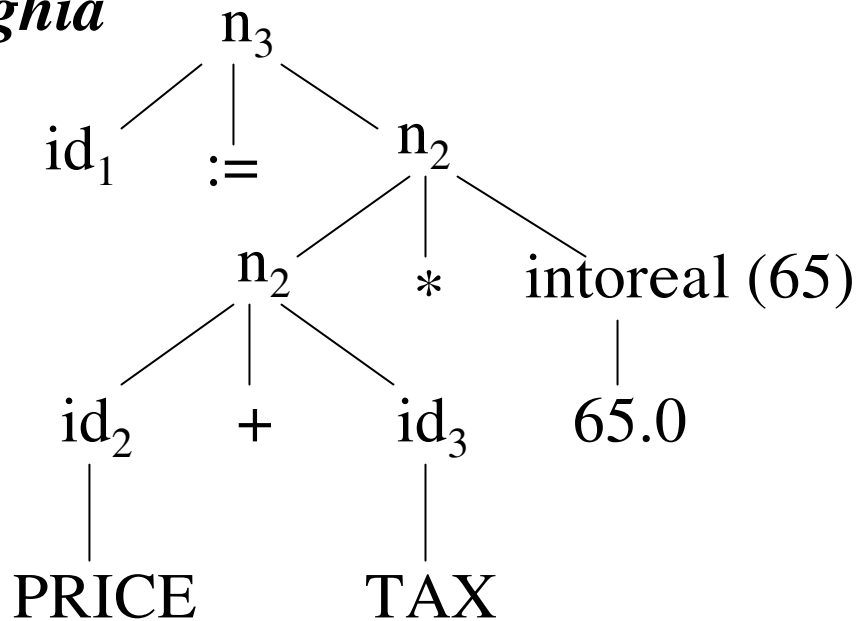
$id1 := (id2 + id3) * num4$

Kết quả phân tích cú pháp:



Hình 1.6. Cây cú pháp của phát biểu

5. Phân tích ngữ nghĩa



Hình 1.7. Cây cú pháp có xử lý ngữ nghĩa

6. Sinh mã trung gian

temp₁ := intoreal (65)

temp₂ := id₂ + id₃

temp₃ := temp₂ * temp₁

id₁ := temp₃

7. Tối ưu mã trung gian

temp₁ := id₂ + id₃

id₁ := temp₁ + 65.0

8. Sinh mã đối tượng

movF id₂, R₁

movF id₃, R₂

addF R₂, R₁

multF # 65.0, R₁

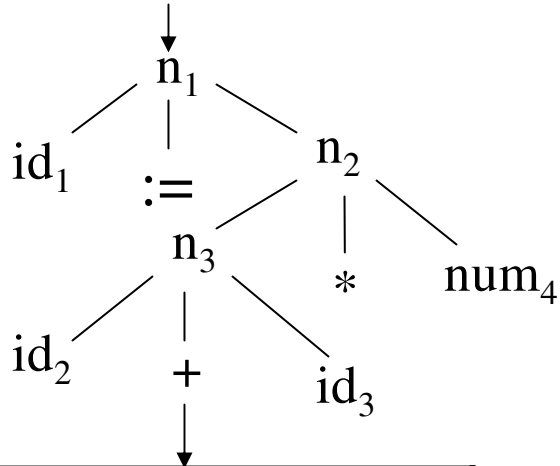
movF R₁, id₁

$COST := (PRICE + TAX) * 65$

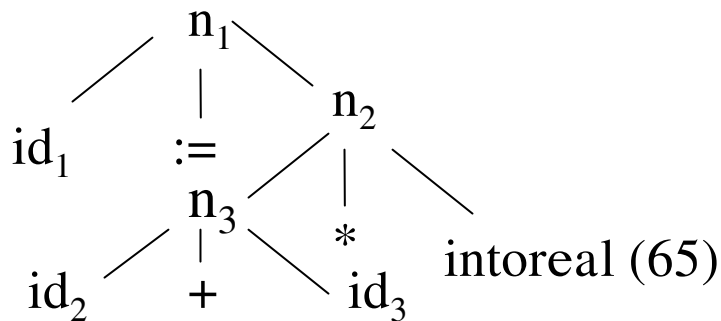
Bộ phân tích từ vựng

$id := (id_2 + id_3) * num_4$

Bộ phân tích cú pháp



Bộ phân tích ngữ nghĩa



Bộ sinh mã trung gian

temp₁ := intoreal (65)

temp₂ := id₃ + id₃

temp₃ := temp₂ * temp₁

id₁ := temp₃

Bộ tối ưu trung gian

temp₁ := id₂ + id₃

id₁ := temp₁ * 65.0

Bộ sinh mã đối tượng

movF id₂, R₁

movF id₃, R₂

ADDF R₂, R₁

mulF # 65.0, R₁

movF R₁, id₁

Hình 1.8. Biên dịch phát biểu

1.3. Các mối liên quan với trình biên dịch

1. Bộ tiền xử lý

- Xử lý macro (macro processing)
- Chêm tập tin (file inclusion)
- Bộ xử lý hoà hợp (rational processor)
- Mở rộng ngôn ngữ (language extension)

Thí dụ về xử lý macro:

- Hệ thống máy đánh chữ typesetting:

```
\define <macro name><template> {<body>}
```

Thí dụ macro định nghĩa về sự trích dẫn của tạp chí ACM:

```
\define\JACM # 1; #2; #3  
{\S1J.ACM}{\bf #1}: #2, pp.#3}
```

Khi dùng macro:

```
\JACM 17; 4; 715-728
```

Sẽ được hiểu như sau:

```
J.ACM 17 : 4 , pp. 715-728
```

2. *Trình biên dịch hợp ngữ*

Phát biểu gán $b := a + 2$ được dịch ra mã hợp ngữ

MOV a, R₁

ADD #2, R₁

MOV R₁, b

3. *Trình biên dịch hợp ngữ hai chuyển*

- Chuyển thứ nhất: đọc mã hợp ngữ và tạo bảng danh biểu

Danh biểu	Địa chỉ tương đối
-----------	-------------------

a	0
---	---

b	4
---	---

- Chuyển thứ hai: đọc mã hợp ngữ và dịch sang mã máy khả định vị địa chỉ:

MOV a, R ₁	0001 010000000000*
-----------------------	--------------------

ADD #2, R ₁	0010 0110 00000010	(1.6)
------------------------	--------------------	-------

MOV R ₁ , b	0100 010000000100*
------------------------	--------------------

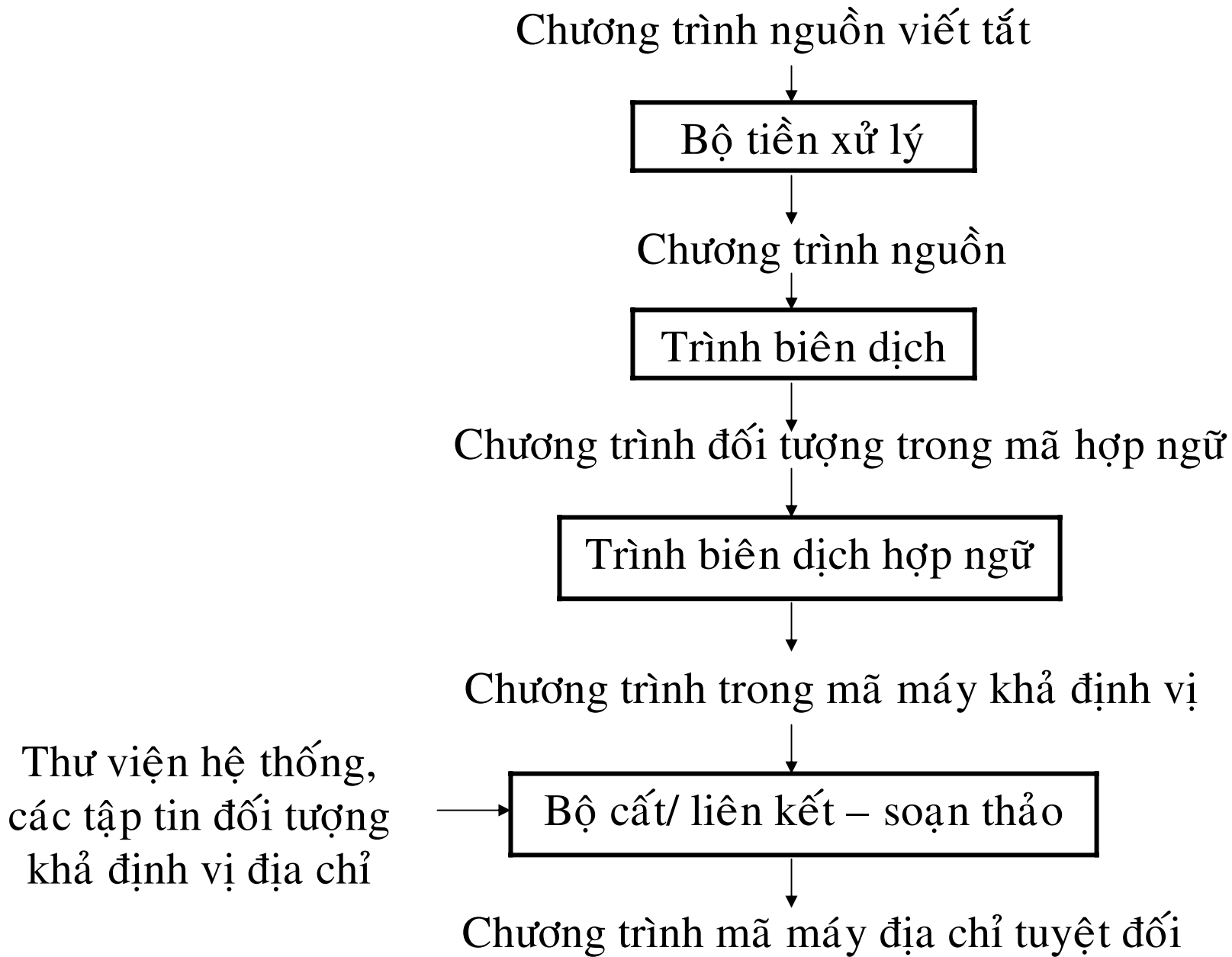
4. Bộ cất liên kết soạn thảo

Loader là chương trình thực hiện hai nhiệm vụ: cất và soạn thảo liên kết. Quá trình cất bao gồm lấy mã máy khả định vị tính lại thành địa chỉ tuyệt đối.

Như ở ví dụ phần 3: Giả sử mã máy được cất trong bộ nhớ trong tại địa chỉ $L = 00001111$; địa chỉ tuyệt đối của a , b là 00001111 và 00010011 . Ba chỉ thị (1.6) được viết lại dưới dạng mã máy tuyệt đối:

$$\begin{array}{l} 0001010000001111 \\ 0011011000000010 \\ 0010010000010011 \end{array} \quad (1.7)$$

Link-editor cho phép tạo một chương trình duy nhất từ nhiều tập tin ở dạng mã máy khả định vị của những lần biên dịch riêng biệt và từ các tập tin thư viện do hệ thống cung cấp.



Hình 1.19. Hệ thống xử lý ngôn ngữ

1.4. Nhóm các giai đoạn của trình biên dịch

- *Giai đoạn trước và giai đoạn sau* (front end and back end)
- *Các chuyển*
- *Thu giảm số lượng các chuyển*

Thí dụ: goto L

 :

 goto L

 :

 L : a = b + c