

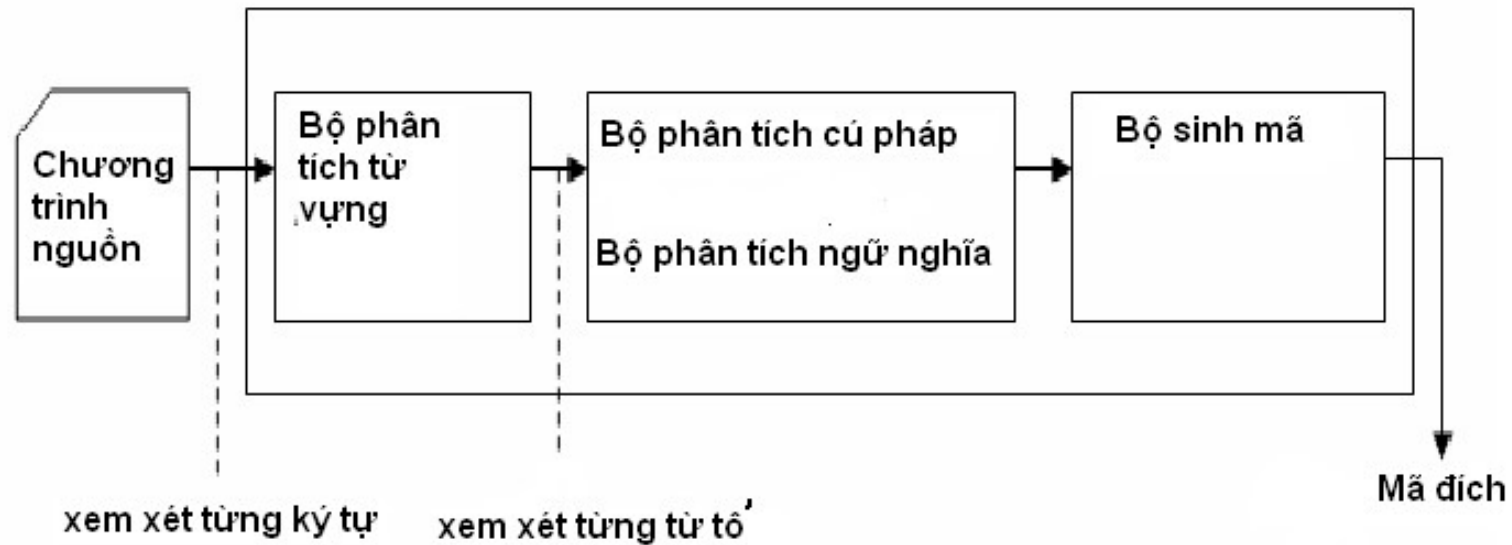


Bài 2.

Các pha chính của chương trình dịch

ONE LOVE. ONE FUTURE.

Các thành phần chính của trình biên dịch



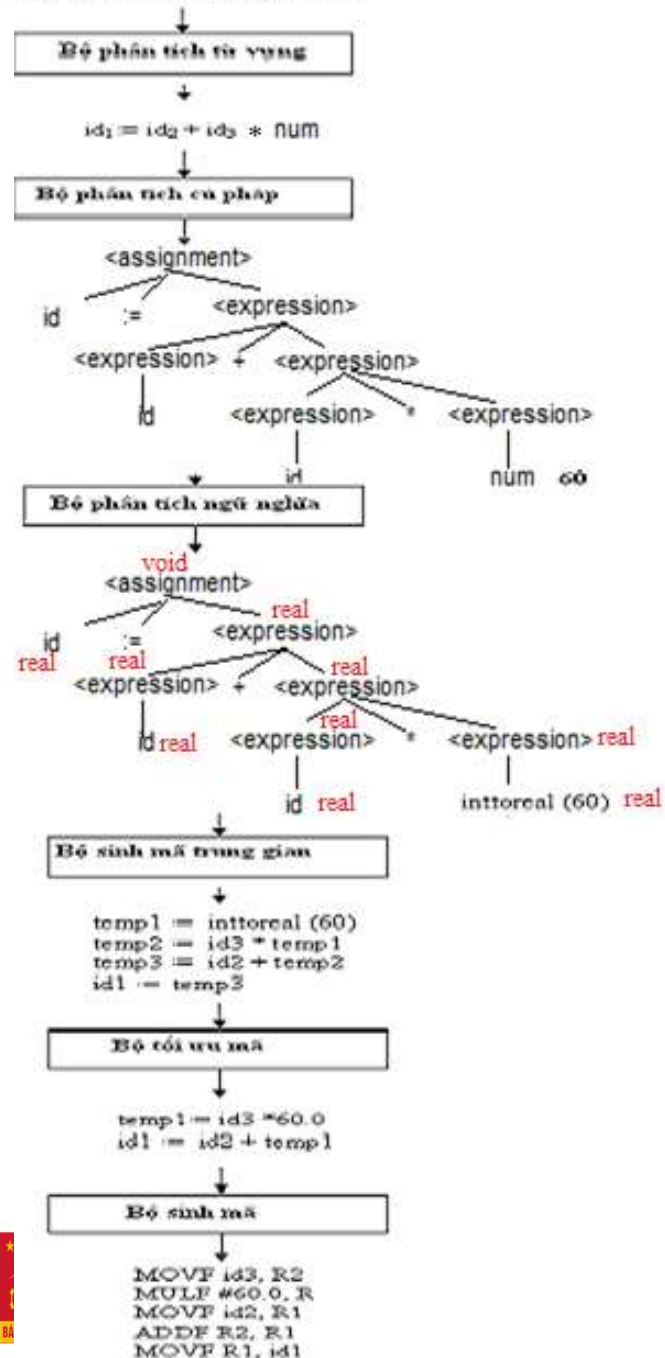
- Phân tích từ vựng (Lexical Analysis - Scanner)
Lần lượt xem xét từng ký tự của chương trình nguồn, phân nhóm chúng thành những đơn vị cú pháp gọi là từ tố (token)
- Phân tích cú pháp (Syntax Analysis)
Dãy token do bộ phân tích từ vựng đưa ra được kiểm tra xem có đúng cú pháp không?
- Phân tích ngữ nghĩa (Semantic Analysis)
Phân tích ý nghĩa từng lệnh của ngôn ngữ nguồn.

Các pha trong trình biên dịch

- Sinh mã trung gian (Intermediate Code Generation) thường là mã 3 địa chỉ. Mã trung gian không phụ thuộc máy nên dễ tối ưu.
- Sinh mã đích: Sinh ra các lệnh máy để thực hiện thao tác.
- Tối ưu mã: Thực hiện với mã trung gian và cả mã đích nhằm làm cho chương trình hiệu quả hơn.

Quá trình dịch một câu lệnh

Position := Initial + Rate * 60



Tập luật của văn phạm

`<assignment> → id := <expression>`

`<expression> → <expression> + <expression>`

`<expression> → <expression> * <expression>`

`<expression> → id`

`<expression> → num`

Pha 1:Phân tích từ vựng

- Bộ từ vựng:Chương trình làm nhiệm vụ phân tích từ vựng
- Các công việc của bộ từ vựng
 - Nhóm các ký tự thành từ tố
 - Từ tố :đơn vị cú pháp được xử lý trong quá trình dịch như một thực thể không thể chia nhỏ hơn nữa
 - Nhóm các từ tố theo loại.

Một số loại từ tố

- Loại từ tố (token)
 - Định danh
 - Từ khóa
 - Số
 - Toán tử
 - Dấu phân cách
- Thể hiện (Lexeme)
 - a, chuongtrinh, x1
 - if, else, for
 - -1, -2.3E10
 - +, -, *, /, =, ==, >>
 - :, ;, (,)



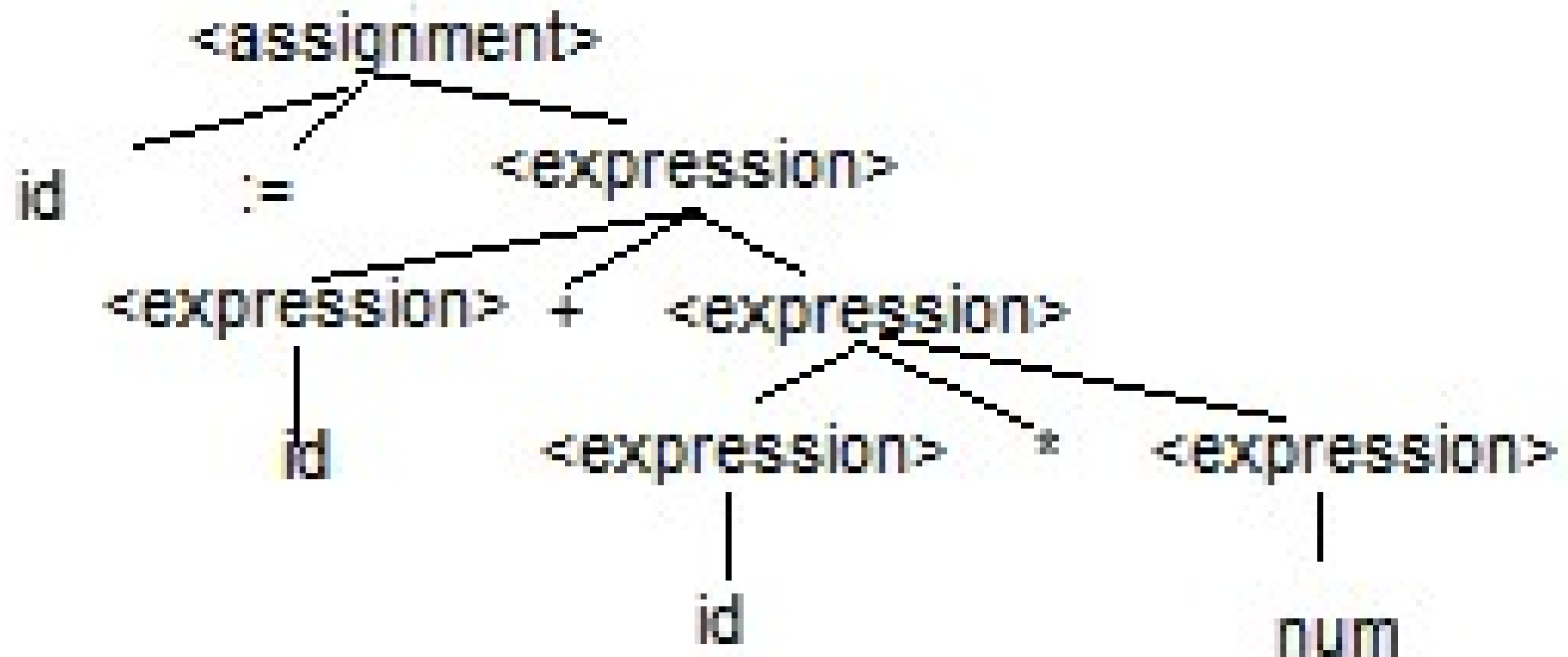
Pha 2: Phân tích cú pháp

- Trình biên dịch kiểm tra xem những từ tố mà bộ từ vựng nhận biết được có kết hợp thành những câu lệnh đúng cú pháp không
- Do bộ phân tích cú pháp đảm nhận

Pha 2: Phân tích cú pháp

- Đầu ra của bộ phân tích cú pháp:
 - Cây phân tích cú pháp (nếu có)
 - Thông báo lỗi nếu ngược lại
- Việc xây dựng được cây phân tích cú pháp chứng tỏ chương trình đúng về cú pháp

- Cây phân tích cú pháp của câu lệnh
Position := Initon + Rate * 60



Biểu diễn văn phạm bằng BNF

- Các luật của BNF cũng như văn phạm hình thức sử dụng 2 loại ký hiệu ở vế phải
- Ký hiệu kết thúc :
 - Từ tố của ngôn ngữ
 - Không xuất hiện ở vế trái
- Ký hiệu không kết thúc
 - Ký hiệu trung gian của văn phạm để mô tả cấu trúc ngôn ngữ
 - Cần xuất hiện ở vế trái của ít nhất một luật
 - Bao trong cặp <>

- Ký hiệu đầu :
 - Ký hiệu không kết thúc ở mức cao nhất
 - Xuất hiện ở gốc cây phân tích cú pháp

Ví dụ: xét tập luật BNF

$\langle \text{assignment} \rangle \rightarrow \text{id} := \langle \text{expression} \rangle$

$\langle \text{expression} \rangle \rightarrow \langle \text{expression} \rangle + \langle \text{expression} \rangle$

$\langle \text{expression} \rangle \rightarrow \langle \text{expression} \rangle * \langle \text{expression} \rangle$

$\langle \text{expression} \rangle \rightarrow \text{id}$

$\langle \text{expression} \rangle \rightarrow \text{num}$

Cho biết tập ký hiệu kết thúc, không kết thúc, ký hiệu đầu?

- Bằng cách áp dụng liên tục các luật mô tả văn phạm
- Nếu bộ PTCP chuyển thành công từ xâu vào thành ký hiệu đầu thì xâu vào đúng cú pháp
- Ngược lại, câu được xem xét không đúng cú pháp

Khái niệm và kỹ thuật phân tích cú pháp

- Vấn đề quan trọng nhất khi xây dựng trình biên dịch là xây dựng một văn phạm
- Bao gồm đầy đủ các cấu trúc của một chương trình
- Không thể tạo nên một luật nào khác
- Văn phạm phải không nhập nhằng
- Nếu văn phạm nhập nhằng, xây dựng được nhiều hơn 1 cây cho mỗi câu được đưa ra phân tích

Pha 3: Phân tích ngữ nghĩa

- Duyệt cây cú pháp của chương trình để xem mọi cấu trúc ngữ nghĩa có đúng không
- Chương trình đúng cả về cú pháp và ngữ nghĩa mới sinh mã được

Pha 4: Sinh mã trung gian

- Chương trình với mã nguồn được chuyển sang chương trình tương đương trong ngôn ngữ trung gian bằng bộ sinh mã trung gian.
- *Mã trung gian* là mã máy độc lập tương tự với tập lệnh trong máy.
- Có thể tối ưu hóa mã độc lập với máy đích cho dạng biểu diễn trung gian.
- Giảm thời gian thực thi chương trình đích vì mã trung gian có thể được tối ưu

Pha 5: Sinh mã đích

- Vào: biểu diễn trung gian của chương trình nguồn
- Ra: chương trình đích
 - Mã Assembly
 - Mã mô phỏng trên máy đích ảo