Bài 6.
SINH MÃ TRUNG GIAN

Hoàng Anh Việt
Viện CNTT&TT - ĐHBKHN

Mô tả các bước dịch (1)

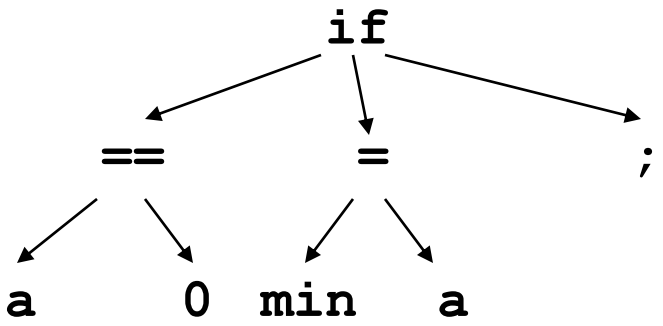
Mã nguồn (dãy các kí tự)

```
If (a == 0) min = a;
```

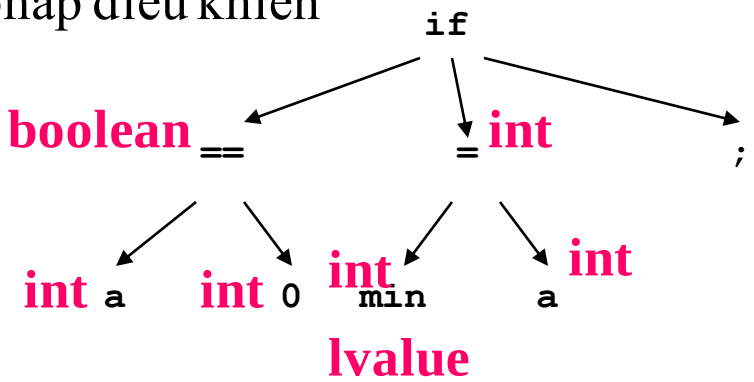
Dãy các từ tố (token)

If	(	Id:a	==	0	)	Id:min	=	Id:a	;

Cây cú pháp



Cây cú pháp điều khiển

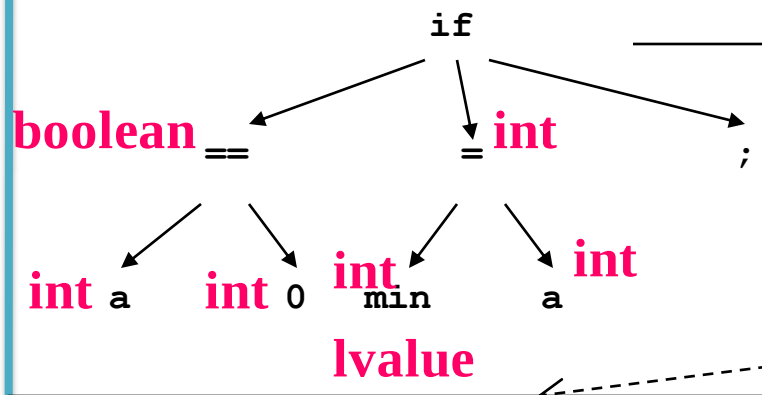


Phân tích từ vựng

Phân tích cú pháp

Phân tích ngữ nghĩa

Mô tả các bước dịch (2)



Sinh mã trung gian

```
SEQ(CJUMP(TEMP(a) == 0, L1, L2),
LABEL(L1),
TEMP(min) = TEMP(a)
LABEL(L2))
```

Sinh mã assembly

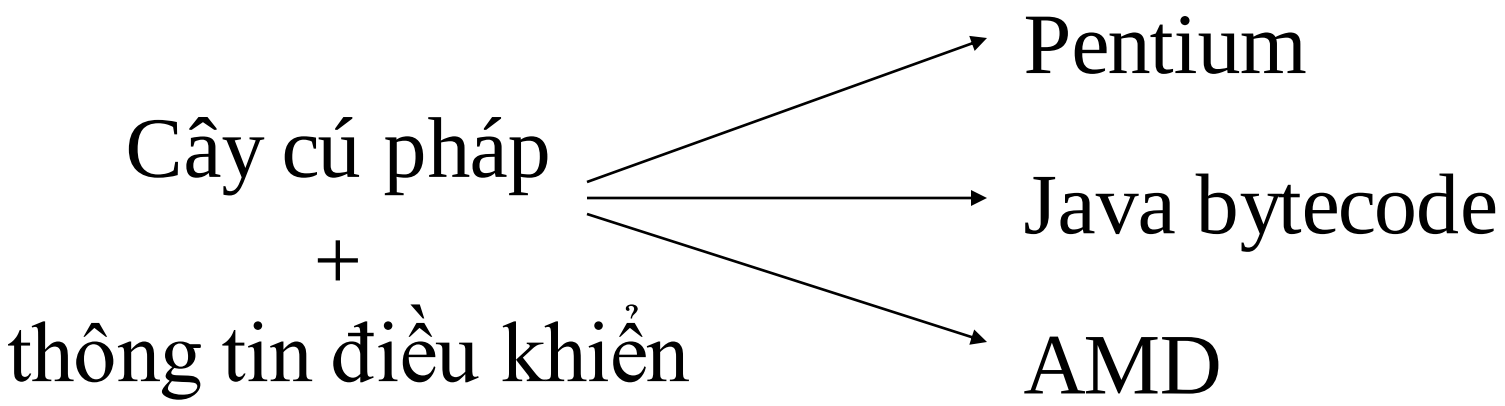
```
cmp rb, 0
jnz L2
L1: mov ra, rb
L2:
```

Tối ưu mã

```
cmp ecx, 0
cmovz edx,ecx
```

Ngôn ngữ trung gian

- Là ngôn ngữ cho một loại máy *trừu tượng*
- Cho phép sinh mã không phụ thuộc vào máy đích
- Cho phép tối ưu mã trước khi sinh mã máy thật sự



Ngôn ngữ trung gian

- Dễ sinh ra từ cây cú pháp
- Dễ sinh mã máy
- Số lượng lệnh nhỏ, gọn
  - Dễ tối ưu mã
  - Dễ chuyển sang loại mã máy khác

Cây cú pháp (>40 nút)

↓

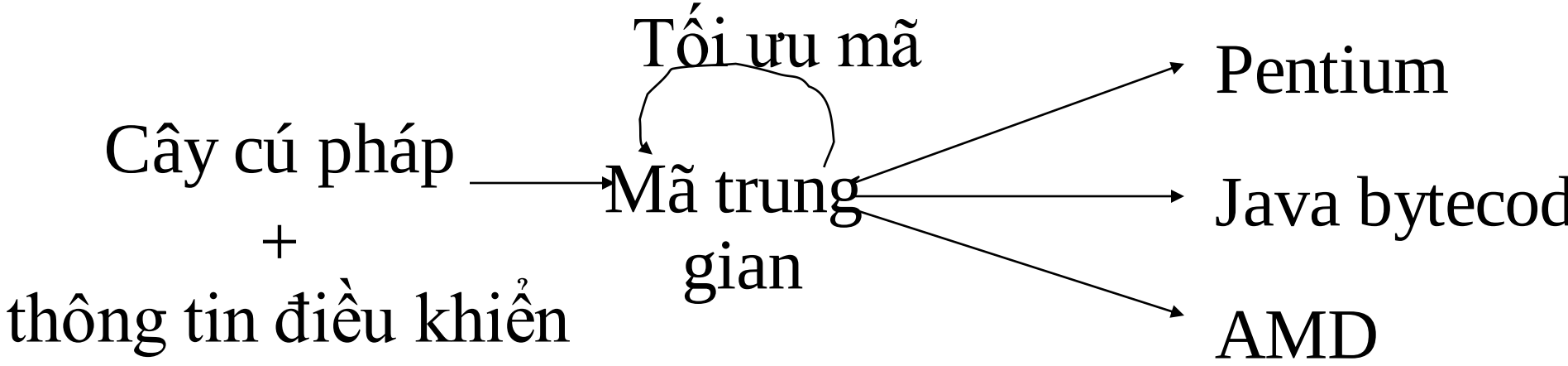
Mã trung gian (13 nút)

↓

Pentium (>200 lệnh)

Ngôn ngữ trung gian

- Một dạng thể hiện của chương trình nằm giữa cây cú pháp điều khiển và mã máy
- Sử dụng
  - Lệnh nhảy
  - Thanh ghi
  - Vị trí trên bộ nhớ



Một ngôn ngữ trung gian

- **IR** (**I**ntermediate **R**epresentation) là một cây thể hiện các lệnh của một loại máy trừu tượng
- Nút lệnh không trả lại giá trị, được thực hiện theo thứ tự nhất định
  - Ví dụ: MOVE, SEQ, CJUMP
- Nút biểu thức trả lại giá trị, các nút con có thể thực hiện theo thứ tự bất kì
  - Ví dụ: ADD, SUB
  - Cho phép tối ưu mã

Mô tả các nút biểu thức của IR

- CONST(i): hằng số nguyên i
- TEMP(t): thanh ghi t, máy trừu tượng có vô hạn thanh ghi.
- OP(e_1, e_2): các phép toán
  - Số học: ADD, SUB, MUL, DIV, MOD
  - Logic: AND, OR, XOR, LSHIFT, RSHIFT
  - So sánh: EQ, NEQ, LT, GT, LEQ, GEQ
- MEM(e): giá trị bộ nhớ ở vị trí e
- CALL(f, a_0, a_1, …): giá trị của hàm f với các tham số a_0, a_1, …
- NAME(n): địa chỉ của lệnh hoặc dữ liệu có tên là n
- ESEQ(s, e): giá trị của e sau khi lệnh s được thực hiện

CONST

- Nút CONST đại diện cho hằng số

- Giá trị của nút là i

TEMP

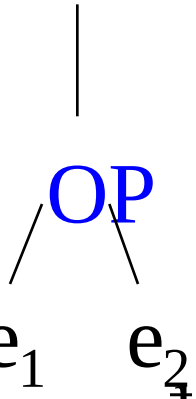
- Nút TEMP đại diện cho một thanh ghi trong số vô hạn các thanh ghi của máy trừu tượng
- Các biến cục bộ và các biến tạm
- Để dễ viết, ký hiệu FP = TEMP(FP) là địa chỉ bắt đầu bộ nhớ của hàm
- Giá trị của nút là giá trị của thanh ghi tại thời điểm tính toán

TEMP(t)

Toán tử

- Máy trừu tượng có nhiều phép toán

$OP(e_1, e_2)$



- Tính giá trị của e_1 và e_2, sau đó áp dụng phép toán với các giá trị này
- e_1 và e_2 phải là hai nút có giá trị
- Có thể tính giá trị e_1 và e_2 theo thứ tự bất kì

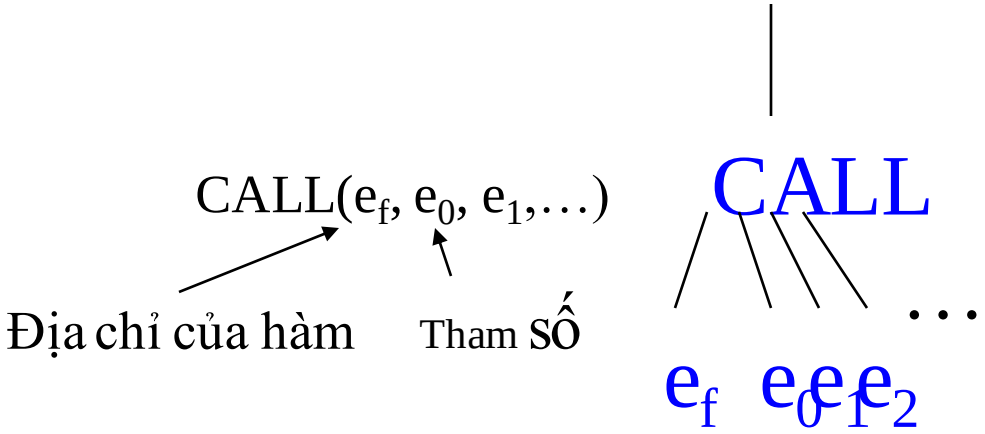
MEM

- Nút MEM đại diện cho một vị trí trong bộ nhớ
- Giá trị của nút là giá trị tại vị trí e trong bộ nhớ

MEM(e)

CALL

- Nút CALL đại diện cho một lời gọi hàm



- Không định nghĩa cách cài đặt việc truyền tham số, quản lý ngăn xếp
- Giá trị của nút là giá trị của hàm

NAME

- Nút NAME đại diện cho địa chỉ của một tên trên bộ nhớ
- VD: địa chỉ của một nhãn nhảy

NAME(n)

ESEQ

- Nút ESEQ tính toán giá trị của biểu thức e sau khi thực hiện lệnh s

ESEQ(s, e)

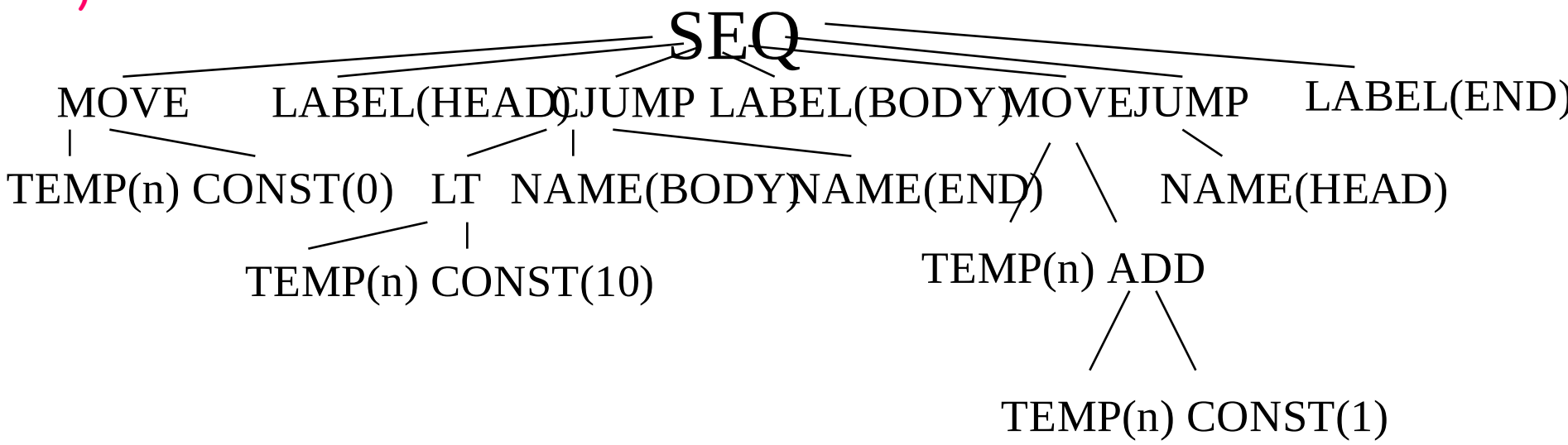
Mô tả các nút lệnh của IR

- MOVE(dest, e): chuyển giá trị của e vào dest
- EXP(e): tính toán giá trị của e, không cần lưu lại kết quả
- SEQ(s_1, s_2, … s_n): thực hiện các lệnh theo thứ tự
- JUMP(e): nhảy đến địa chỉ e
- CJUMP(e, l_1, l_2): nhảy đến l_1 hoặc l_2 tuỳ thuộc vào giá trị của e là true hoặc false
- LABEL(n): tạo ra nhãn có tên n

Ví dụ

```
n = 0;
while (n < 10)
{
   n = n + 1;
}
```

```
SEQ(
  MOVE(TEMP(n), CONST(0)),
  LABEL(HEAD),
  CJUMP(LT(TEMP(n), CONST(10)), NAME(BODY), NAME(END)),
  LABEL(BODY),
  MOVE(TEMP(n), ADD(TEMP(n), CONST(1))),
  JUMP(NAME(HEAD)),
  LABEL(END)
)
```



Cấu trúc của IR

- Gốc của cây là một nút lệnh
- Các nút biểu thức nằm dưới nút lệnh
- Chỉ có nút biểu thức ESEQ có nút lệnh nằm dưới
- Có thể duyệt cây IR để chạy chương trình

Sinh cây IR (mã trung gian)

- Kỹ thuật: phương pháp dịch sử dụng cú pháp điều khiển (giống kiểm tra kiểu)
- Chuyển cây cú pháp điều khiển thành cây IR
- Mỗi cây con của cây cú pháp được chuyển thành một cây con dạng IR có cùng giá trị

Sinh cây IR

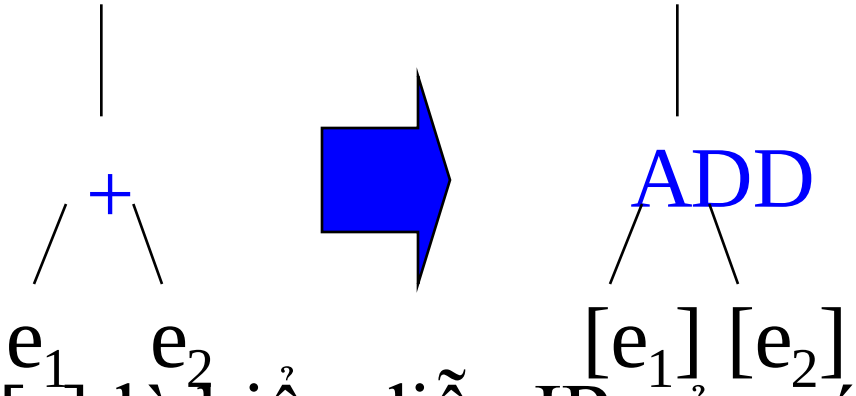
- Giống kiểm tra kiểu: thêm một phương thức vào nút tương ứng trong cây cú pháp

```
abstract class ASTNode {
  IRNode translate(SymTab A) { … }
}
```

- Cài đặt kiểu đệ quy
- Vấn đề: giống như kiểm tra kiểu, cần mô tả chính xác cách viết hàm translate()

Biểu thức

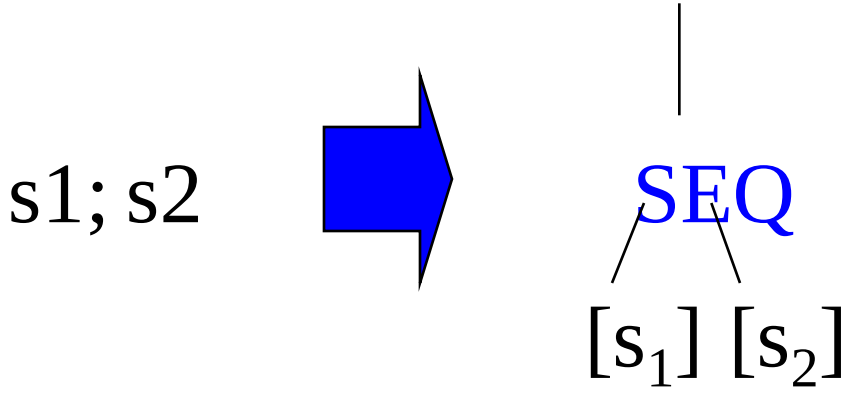
- Các nút của cây cú pháp thể hiện biểu thức được chuyển thành nút IR tương ứng



- Kí hiệu [e] là biểu diễn IR của nút e trong cây cú pháp

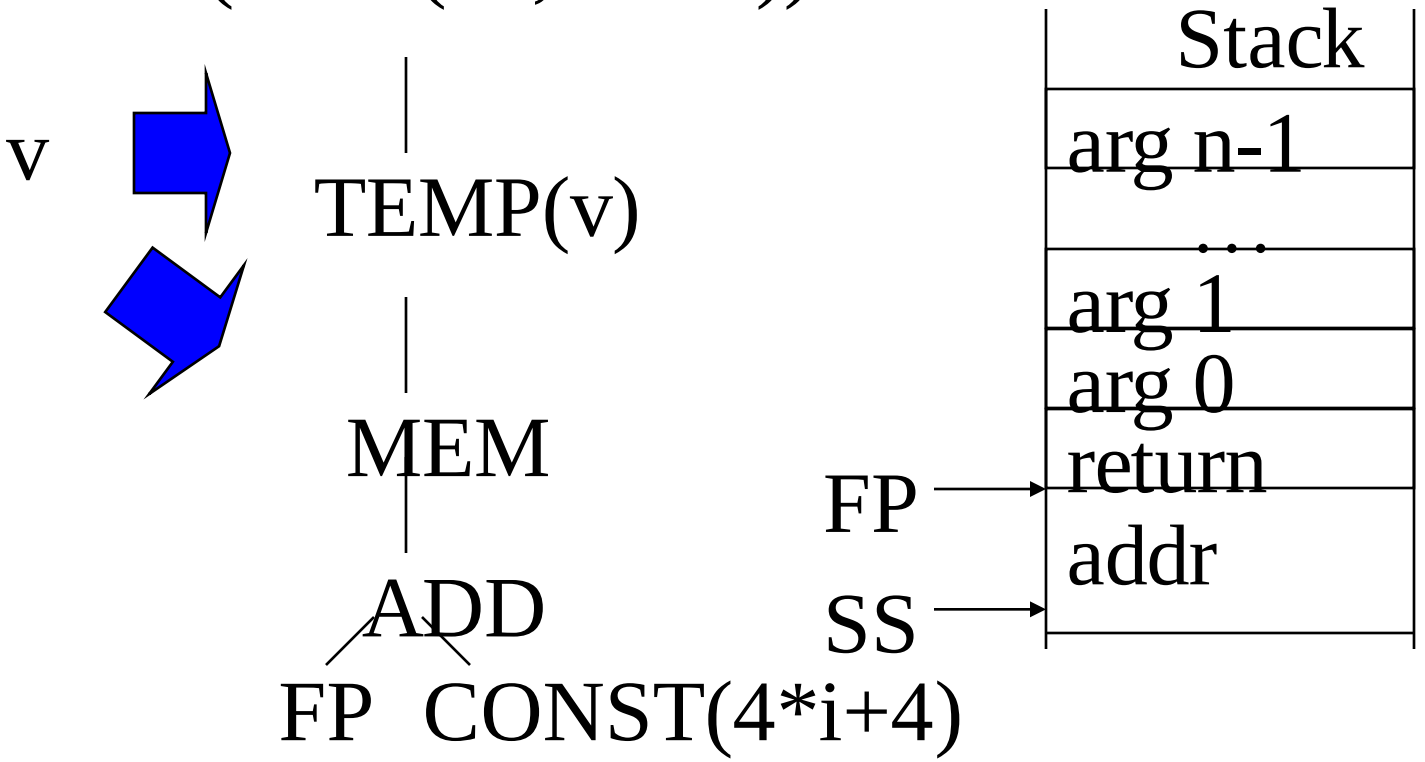
Câu lệnh

- Dãy các lệnh được biểu diễn bằng nút SEQ trong biểu diễn IR
- Nếu $[s_1]$ và $[s_2]$ là biểu diễn IR của nút s_1 và s_2
- thì SEQ($[s_1]$, $[s_2]$) là biểu diễn IR của s_1; s_2



Biến

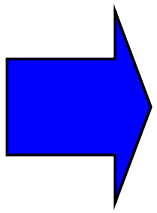
- Biến cục bộ v chuyển thành nút TEMP(v)
- Tham số thứ i nằm ở vị trí MEM(ADD(FP,4*i+4))

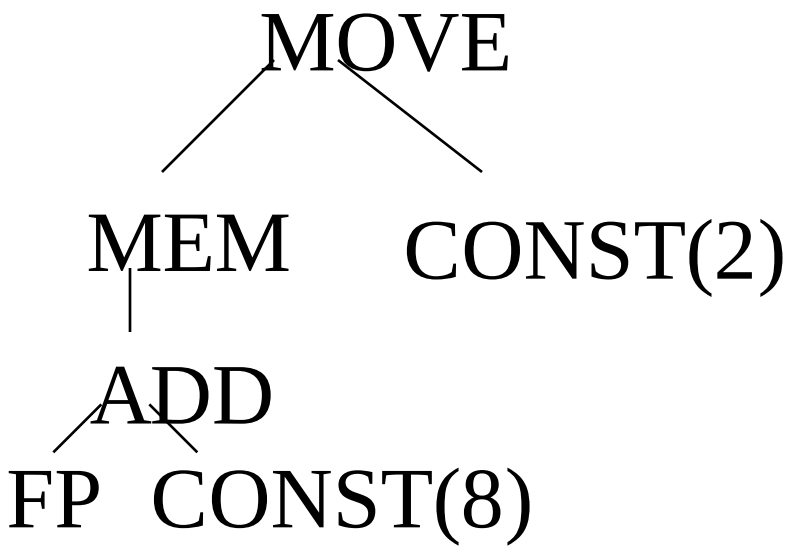


Phép gán

- Phép gán v = e chuyển thành nút MOVE(dest, [e]) với dest là địa chỉ của v, [e] là biểu diễn IR của e
- Ví dụ

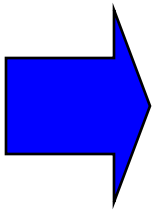
x = 2

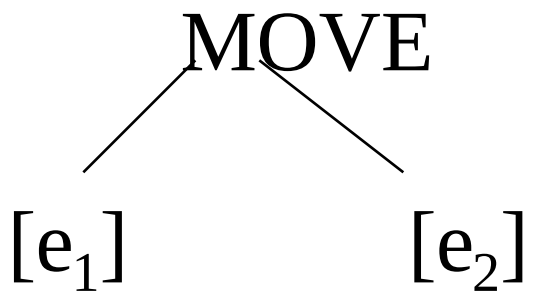




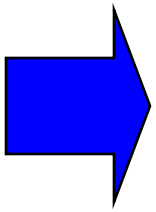
Phép gán

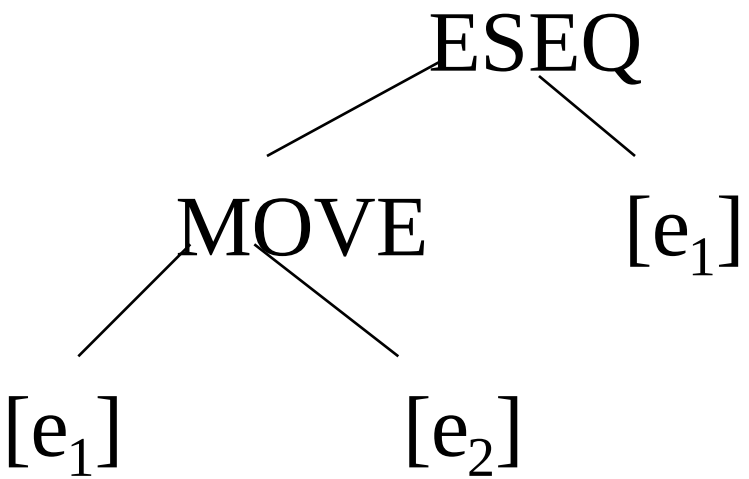
- Cách dịch

$e_1 = e_2$ 



- Vấn đề: nút MOVE không có giá trị, làm thế nào để dịch x = (y = 2)?

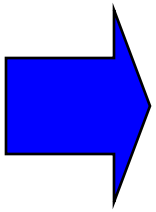
$e_1 = e_2$ 

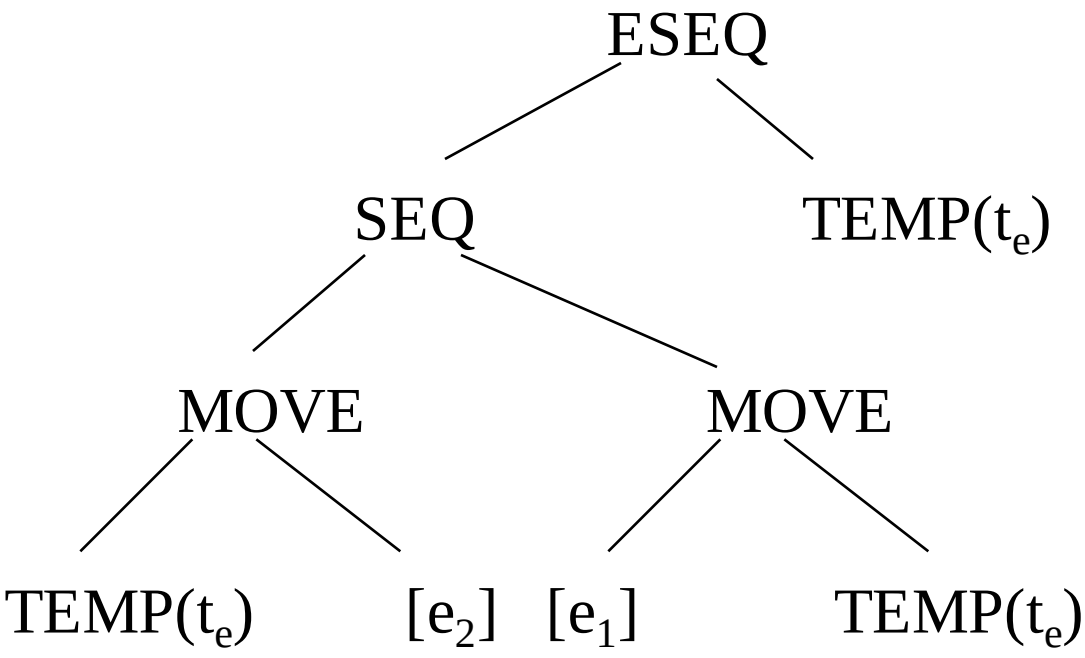


Phép gán

- Như vậy, [e_1] phải chạy 2 lần, cần lưu lại giá trị của [e_1]

$e_1 = e_2$





Thảo luận