

Experiment in Compiler Construction

Phân tích ngữ nghĩa (4)

**Viện Công nghệ Thông tin và Truyền thông
Đại học Bách khoa Hà nội**

Nội dung

- Kiểm tra sự nhất quán về kiểu trong các cấu trúc chương trình
- Kiểm tra sự nhất quán của định nghĩa biến mảng và sử dụng biến mảng
- Kiểm tra sự nhất quán trong định nghĩa hàm và sử dụng hàm
- Kiểm tra sự nhất quán trong định nghĩa thủ tục và lời gọi thủ tục
- Kiểm tra sự nhất quán trong việc sử dụng tham biến

Kiểm tra sự nhất quán về kiểu

- So sánh kiểu
 - `checkIntType`
 - `checkCharType`
 - `checkArrayType`
 - `checkTypeEquality`

Kiểm tra sự nhất quán về kiểu (1)

- Duyệt hằng:
[+/-] <constant>
 - <constant> có kiểu nguyên

Kiểm tra sự nhất quán về kiểu (2)

- Duyệt lệnh gán:
 <LValue> := <Expr>;
 - Lvalue và Expr phải có cùng kiểu cơ bản

Kiểm tra sự nhất quán về kiểu (3)

- Duyệt lệnh for:

For <var> := <exp1> To <exp2> do <stmt>

- <var>, <exp1>, <exp2> phải có cùng kiểu cơ bản

Kiểm tra sự nhất quán về kiểu (4)

- Duyệt tham số hàm/thủ tục:
 - Tham số hình thức và tham số thực sự phải có cùng kiểu
 - Nếu tham số hình thức là tham biến thì tham số thực sự phải là một LValue

Kiểm tra sự nhất quán về kiểu (5)

- Duyệt điều kiện:
 <exp1> <op> <exp2>
 - exp1 và exp2 phải có cùng kiểu dữ liệu cơ bản

Kiểm tra sự nhất quán về kiểu (6)

- Duyệt biểu thức:

[+|-] <exp> \rightarrow exp : integer

[*|/] <term> \rightarrow term : integer

Kiểm tra sự nhất quán về kiểu (7)

- Duyệt chỉ số:
(. <exp> .) \rightarrow exp : integer
- Lưu ý đến số chiều của mảng khi duyệt chỉ số

Nội dung thực hành (1)

- Lập trình cho các hàm trong semantics.c
 - void checkIntType(Type* type);
 - void checkCharType(Type* type);
 - void checkArrayType(Type* type);
 - void checkBasicType(Type* type);
 - void checkTypeEquality(Type* type1, Type* type2);

Nội dung thực hành (2)

- Bổ sung các đoạn mã kiểm tra kiểu trong `parser.c` tương ứng với các luật kiểm tra kiểu kể trên
- Biên dịch và thử nghiệm với các ví dụ mẫu