Thực hành
Xây dựng chương trình dịch

Bài 4: Kiểm tra kiểu

ONE LOVE. ONE FUTURE.

Nội dung thực hành

- 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

Cấu trúc của bộ phân tích ngữ nghĩa (chưa hoàn thiện)

#	Tên tệp	Nhiệm vụ
1	Makefile	Quản lý project. Sinh viên tự make
2	scanner.c, scanner.h	Phân tích từ vựng
3	reader.h, reader.c	Đọc từng ký tự của chương trình nguồn
4	charcode.h, charcode.c	Phân loại các ký tự
5	token.h, token.c	Phát hiện các từ tố
6	error.h, error.c	Xử lý các loại lỗi
7	parser.c, parser.h	Bộ phân tích cú pháp
8	debug.c, debug.h	In ấn
9	symtab.c symtab.h	Xây dựng bảng ký hiệu
10	semantics.c. semantics.h	Các hàm hỗ trợ cho phân tích ngữ nghĩa
11	main.c	Chương trình chính

Vấn đề kiểm tra kiểu

- Định nghĩa một hệ thống kiểu cho ngôn ngữ KPL
- Các kiểu được chấp nhận trong ngôn ngữ:
  - Kiểu cơ sở: integer, char
  - Kiểu do người sử dụng tự đặt: biểu diễn qua định danh
  - Kiểu có cấu trúc: array, đặc trưng bởi
    - Miền chỉ số: chỉ quản lý kích thước tối đa. Chỉ số bắt đầu từ 1
    - Kiểu phần tử
- Để thục hiện hệ thống kiểu đã định nghĩa, cần tác động tới 2 module: semantics và parser.

Các hàm trong semantics hỗ trợ kiểm tra sự nhất quán kiểu

- checkIntType
- checkCharType
- checkArrayType
- checkTypeEquality

Kiểm tra sự nhất quán về kiểu

- Duyệt hằng:
  - [+/-] <constant>: <constant> có kiểu nguyên
  - Việc xử lý dấu của hằng đã được thực hiện tại hàm compileConstant

Kiểm tra sự nhất quán về kiểu trong lệnh gán

- Duyệt lệnh gán:

  <LValue> := <Expr>;

  - Lvalue và Expr phải có cùng kiểu cơ bản

Kiểm tra sự nhất quán kiểu trong lệnh for

- Duyệt lệnh for:

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

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

Kiểm tra sụ nhất quán về kiểu trong lệnh if và lệnh while

- Điều kiện là thành phần quan trọng nhất trong 2 lệnh trên, với cú pháp:

  <exp1> <op> <exp2>

  - <exp1> và <exp2> phải có cùng kiểu dữ liệu cơ bản
- Ngoài ra các lệnh nằm trong nhánh then, else và thân vòng lặp while cũng cần tương ứng về kiểu

Đánh giá kiểu cho biểu thức số học

- Duyệt biểu thức: Nếu biểu thức có toán tử -> chỉ có thể có kiểu integer
- [+ | -] <exp> → exp : integer
- [* | /] <term> → term : integer
- Nếu biểu thức không có toán tử, vẫn có thể trả ra kiểu char

Kiểm tra sự nhất quán về kiểu khi gọi hàm và thủ tục

- Duyệt tham số hàm/thủ tục:
  - Danh sách tham số hình thức và tham số thực sự phải đồng nhất về số lượng, thứ tự và kiểu.
  - Chú ý đến đặc điểm về cú pháp khi danh sách tham số hình thức hay tham số thực sự là rỗng (không có cặp ())
  - Cần kiểm tra tương ứng kiểu giữa tham số hình thức và tham số thực sự.
  - Nếu tham số hình thức là tham biến thì tham số thực sự phải có địa chỉ (là một Lvalue)

Kiểm tra kiểu trên mảng

- Để xem xét kiểu phần tử của mảng cần duyệt chỉ số.
- Xét hàm compileIndexes

  (. <exp> .) → 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 checkBasicType(Type* type);
void checkTypeEquality(Type* type1, Type*
type2); //chú ý đã có hàm compareType
```

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