中间代码生成

梁宇钦 151180076

2018.06.21

1. 功能实现

1.1 功能实现情况

实验实现了基本要求以及选择要求的所有要求。对于给出的所有基础例子和选做例子都能正确生成中间代码，并且生成的中间代码进行了优化，具有较高的效率。

1.2 功能实现方法

出于对代码结构和维护的方便性考虑，中间代码生成是在完成了语义分析之后再进行的。

1.2.1 函数的翻译

函数的翻译主要就是要生成函数名以及参数的中间代码，而函数体的内容无外乎是局部变量的声明以及 Statement 和表达式（包括布尔表达式）的翻译。

对于函数名以及参数的翻译很简单，因为已经有了完整的符号表，所以直接在符号表找到函数，就可知道它的参数了，生成相应的中间代码即可。

1.2.2 变量声明的翻译(varDec)

同样的，由于已经有了完整的符号表，所以可以直接在符号表里面找到对应的变量名，就可得到变量的属性，然后根据属性生成相应的中间代码即可。

1.2.3 Statement 的翻译

Statement 主要包括 if (else)语句和 while 语句。对它们的翻译，采用继承属性。它们的代码布局分别如下：

if stmt(if else stmt) code: Stmt: IF LP Exp RP Stmt (ELSE Stmt)
// Exp
// Truelabel
// stmt
// (goto label)
// falselabel
// (else stmt)
// label
while loop code:Stmt: WHILE LP Exp RP Stmt
// label
// Exp

```
// turelabel
// Stmt
// goto label
// falselabel
```

1.2.4 普通表达式(Exp)的翻译

这个部分主要难度在与高维数组以及结构体的翻译。

不过由于符号表符号类型的统一，高维数组以及结构的引用的翻译采用递归就可以较好的解决。

总的思路就是沿着数组链以及当前的位置变量（常量）计算偏移以及生成相应的代码。具体细节可以参看代码。

1.2.5 布尔表达式的翻译

布尔表达式的翻译没有特别的难度，直接生成相应的 IF GOTO 代码以及 GOTO 代码就可以了。

1.3 中间代码数据结构

中间代码的数据结构采用双项链表。

2. 编译和测试

这部分内容在文件“README.MD”中有详细说明, 这里仅做简要说明。

2.1 目录树

```
IntermediateCode.d
├── 3rdparty
│   └── uthash.h
├── IntermedCode.c
├── IntermedCode.h
├── IRCodeOutput.d
├── irsim
├── lexical.l
├── main.c
├── Makefile
├── README.md
├── runtests.sh
├── sample.d
├── semantic.c
├── semantic.h
├── SymbolTab.c
```

```
├── SymbolTab.h
├── syntaxtree.c
├── syntaxtree.h
└── syntax.y
```

本次实验的所有内容都在 IntermediateCode.d 目录下面。主要文件包括 lexical.l, syntax.y, syntaxtree.h, syntaxtree.c , SymbolTab.c, SymbolTab.h, semantic.h, semantic.c, IntermedCode.c, IntermedCode.h 和 main.c 等文件。其中 lexical.l 是词法文件, 用 flex 编译；Syntax.y 是文法文件, 用 bison 编译；syntaxtree.h 和 syntaxtree.c 是语法树的数据结构文件；SymbolTab.c 和 SymbolTab.h 是符号表数据结构文件；semantic.h 和 semantic.c 是语义分析文件；IntermedCode.c 和 IntermedCode.h 是中间代码生成文件；sample.d 下面是样例。

2.2 编译

直接在终端输入命令:

```
make
```

就可以了编译了。

2.3 测试

2.3.1 测试已经提供的样例

要测试实验指导提供的样例, 也是非常简单的. 直接运行脚本 “runtests.sh” 即可. 建议采用以下的命令运行脚本:

```
sh runtests.sh
```

其结果会会打印屏幕上的同时也会存到 IRCodeOuput.d 目录下面。

2.3.1 测试其他未提供的样例

命令格式如下:

```
./parser testfilename IRCodeOutputfilename
```

其结果会打印屏幕上的同时也会存到 IRCodeOuput.d 目录下面。