

编译原理PA1-A实验报告

提前上学2018 李嘉图

2019年1月6日

1 主要任务

PA1-A的主要任务是在已有Decaf编译器的框架下，借助Yacc和Lex，新增浅拷贝、数组操作等语法特性。在实验的过程中，主要难点在于对实验框架的理解和对Yacc、Lex的使用。

2 具体流程

要完成一种新的语句或表达式的词法分析和语法分析，主要需进行以下操作：

1. 在Lex中增加新的关键字。可以通过返回Parser.NAME与Yacc进行交互，即源代码经过Lex处理变成一个个终结符构成的串交由Yacc处理。
2. 在Yacc中增加新的文法。Yacc使用BNF范式描述文法，使用LALR(1)分析文法。因此在增加文法的过程中要注意消除出现的移进/规约冲突。通常可以通过更改运算符的优先级解决冲突。
3. 在SemValue.java中加入需要的变量以方便Yacc构建语法树，还要在开头建立编号，以标识其类型。
4. 在Tree.java中加入新语法对应的语法树节点。在当前阶段，要增加构造函数和输出语法树的函数。

3 实验操作

3.1 支持对象复制语句

在frontend/Parser.y中加入关键字SCOPY("scopy")，在frontend/Lexer.l中

新增关键字的词法；在tree/Tree.java中加入AST节点类Tree/Scopy，并在frontend/Parser.y中加入对象复制语句的文法：

```
Scopy : SCOPY '(' IDENTIFIER ',' Expr ')'  
Stmt : Scopy ';' | ...
```

最后加入语义动作构造AST节点。

3.2 加入sealed修饰词

在frontend/Parser.y中加入关键字SEALED("sealed")，在frontend/Lexer.l中新增关键字的词法；在tree/Tree.java中的类Tree/ClassDef中新增一个变量boolean sealed，表示这个类是否是不可被继承的，并修改类的构造函数；在frontend/SemValue.java中新增一个变量boolean sealed，改写frontend/Parser.y中关于ClassDef的文法：

```
ClassDef : ClassClause ExtendsClause '{' FieldList '}'  
ClassClause : CLASS IDENTIFIER | SEALED CLASS IDENTIFIER
```

并增加相应的语义动作，调用构造函数来声明一个类是否是sealed。

之所以要专门创建一个ClassClause，是为了之后为类增加更多修饰词时的方便。

3.3 支持串行条件卫士

在frontend/Parser.y中加入关键字GUARDED("|||")，在frontend/Lexer.l中新增关键字的词法，将':'归入frontend/Lexer.l的简单运算符中；在tree/Tree.java中新建类Tree/IfSubStmt包含条件Expr expr和跟随的语句Tree stmt，表示一个子语句；新建类Tree/GuardedStmt包含一个链表，记录所有的子语句；在frontend/Parser.y中新增文法：

```
Stmt : GuardedStmt | ...  
GuardedStmt : IF '{' IfBranch '}'  
IfBranch : IfBranch GUARDED IfSubStmt | IfSubStmt | <Empty>  
IfSubStmt : Expr ':' Stmt
```

并调用构造函数建立节点。

3.4 支持简单的自动类型推导

在frontend/Parser.y中加入关键字VAR("var")，在frontend/Lexer.l中

新增关键字的词法，在tree/Tree.java中新建类LValue/Var包含变量名称的名称，在frontend/Parser.y中新增文法：

```
Var : VAR IDENTIFIER
```

调用构造函数建立节点。

3.5 数组常量

在tree/Tree.java中建立一个抽象类Expr/Constant，将类Expr/Literal改为此类的子类Constant/Literal，并新建类Constant/ArrayConstant，包含一个List<Constant> calist，表示这个数组常量。在frontend/Parser.y中新增文法：

```
Expr : Constant | ...
```

```
Constant : ArrayConstant | LITERAL | NULL
```

```
ArrayConstant : '[' ConstantList ']'
```

```
ConstantList : ConstantList ',' Constant | Constant | <Empty>
```

并调用构造函数建立对应的节点。

这样设计的好处，是允许了“数组常量的数组常量”的存在，即'[[], [], []]'。

3.6 数组初始化常量表达式

在frontend/Parser.y中新增关键字ARRAYINIT("%%")，在frontend/Lexer.l中建立词法，在tree/Tree.java头部增加标识符编号，修改类Expr/Binary的函数Binary.printTo使其可以正确输出，增加文法：

```
Expr : Expr ARRAYINIT Expr
```

并调用Expr/Binary的构造函数建立节点。

3.7 数组拼接表达式

在frontend/Parser.y中新增关键字ARRAYLINK("++")，在frontend/Lexer.l中建立词法，在tree/Tree.java头部增加标识符编号，修改类Expr/Binary的函数Binary.printTo使其可以正确输出，增加文法：

```
Expr : Expr ARRAYLINK Expr
```

并调用Expr/Binary的构造函数建立节点。

注意，为了使以上两个的优先级和结合性正确，需要在frontend/Parser.y中修改优先级，具体方法是按照优先级从高到低定义标识符，并用'left, right'标出结合性。

3.8 取子数组表达式

在tree/Tree.java中新建类ArrayRange/Expr，内含三个变量Expr name, left, right，分别是数组、开始位置和结束为止，增加文法：

```
Expr : Expr '[' Expr ':' Expr ']' | ...
```

并调用Expr/Binary的构造函数建立节点。

3.9 数组下标动态访问表达式

在frontend/Parser.y中新建关键字DEFAULT("default")，在frontend/Lexer.l中建立词法，在tree/Tree.java中新建类Expr/ArrayDefault，内含变量Expr name, index, def，分别表示数组，下标和默认值，在frontend/Parser.y中增加文法：

```
Expr : Expr '[' Expr ']' DEFAULT Expr
```

并调用构造函数建立节点。

3.10 数组生成式

在frontend/Parser.y中新建关键字IN("in")，在frontend/Lexer.l中建立词法，在tree/Tree.java中新建类，内含变量Expr fun, array, condition和String ident，分别表示产生函数，数组，产生条件和迭代变量名，并新增文法：

```
Expr : '[' Expr FOR IDENTIFIER IN Expr ']' Expr : '[' Expr  
FOR IDENTIFIER IN Expr IF Expr ']'
```

并调用构造函数建立节点，其中，第一种没有产生条件，可以创建一个Tree.Literal("true")占位。

3.11 数组迭代语句

在frontend/Parser.y中新建关键字FOREACH("foreach")，在frontend/Lexer.l中建立词法，在tree/Tree.java中新建类FOREACH("foreach")，含变量Expr array, condition, boolean varTag, String name, TypeLiteral type, Tree stmt，分别表示数组，访问条件，是否需要类型推导，变量名，(如

果不需要推导) 类型名是什么, 和语句体, 在frontend/Parser.y中新增文法:

```
Stmt : ForeachStmt
ForeachStmt : FOREACH '(' VAR BoundVariable IN Expr ')' Stmt
ForeachStmt : FOREACH '(' VAR BoundVariable IN Expr ')' WHITE
Expr Stmt
BoundVariable : VAR IDENTIFIER | Type IDENTIFIER
并调用构造函数建立节点。
```