

中山大学移动信息工程学院本科生实验报告

(2014-2015 学年春季学期)

课程名称: 计算机语言及其处理实验

任课教师:

已拒绝

年级&班级		已拒绝	专业(方向)	软件工程(移动信息工程)
学号			姓名	
完成日期	2015-06-07		Email	已拒绝 @sina.com.cn

比较两个不同 yacc 族工具 (JavaCUP 和 GNU Bison) 语法规则定义差异

GNU bison 是一个自由软件, 用于自动生成语法分析器程序, 实际上可用于所有常见的操作系统。Bison 把 LALR 形式的上下文无关文法描述转换为可做语法分析的 C/C++ 程序。一般情况下配合 FLEX 使用。

➤ JavaCUP 与 Bison 的文件结构不同

JavaCUP 的文件结构

.cup

Package and import declarations

Action code

Parser code

Init with and scan with

Symbol and grammar

Bison 的文件结构

.y

%{

Prologue

%}

Bison declarations

%%

Grammar rules

%%

Epilogue

Cup 文件中各部分介绍

1. *Package and import declarations* 部分主要是 package 和 import 的声明部分, 和 Jflex 的 flex 文件相同。
2. *Action code* 部分主要是提供类来封装语法中用户嵌入的代码, 并且提供一系列的数据结构和语义动作。
3. *Parser code* 部分代码主要包括 parser 类本身, 并且重写默认的错误。
4. *Init with and scan with* 这部分主要是一小部分代码的初始化, 以及显示如何调用 scanner。
5. *Symbol and grammar* 在这一部分声明所有的终结符和非终结符, 运行时表示为 symbol 类, 并且包含语法本身。

Y 文件各部分介绍

1. *Prologue* 部分包括宏定义和在语法规则动作中使用的函数和变量的声明。和一般 C/C++ 文件一样，可以使用 `include<>` 或是 `include ""`，如果不需要可以忽略。这部分将被复制到生成文件的头部。
2. *Bison declarations* 部分包含了定义终结符和非终结符的声明，优先级等等。在一些简单的语法中，可以不需要任何声明。
3. *Grammar rules* 部分包含了一个或多个 Bison 语法规则，在这里至少应该有一个语法规则，并且第一个 `%%` (先于语法规则的那个) 绝对不能省略，解释它在文件的最开头。
4. *Epilogue* 这部分将被复制到生成文件的尾部，如果最后一部分为空，你可以省略分隔它的分隔符 `%%`。

➤ 在 Bison 中语义动作的形式和 JavaCUP 是不相同的

在 Bison 中如下面的例子

这个规则表明一个表达式可由两个表达式相加而成。

expression: expression '+' expression { \$\$ = \$1 + \$3; };

而在 JavaCUP 中确实下面的形式

expression ::= expression:e1 PLUS expression:e2

(:

RESULT = new Integer (e1.intValue() + e2.intValue())

.);

Bison 中是通过 `$` 符号来返回，而在 JavaCUP 中是通过 `RESULT` 来返回。

➤ 优先级

在 JavaCUP 中是通过下面的形式指定的

`precedence right non_terminal name`

`precedence left non_terminal name`

在 Bison 中如下

`%left non_terminal`

`%right non_terminal`

在 JavaCUP 中后面是非终结符的名字，而在 Bison 中是符号，如加减的优先级的定义分别如下

JavaCUP

`precedence left PLUS, MINUS;`

Bison

`%left '+' '-'`

当然在不同的 yacc 工具族间 (JavaCUP 和 Bison) 有很多不同，但是由于很多因素，在此就简单的列出一些比较显而易见的差别。