



2020~2021第二学期

系统软件开发实践

计算机科学与技术学院

张博

Email: zbcumt@163.com

课程资源下载



•链接:

https://pan.baidu.com/s/14Rh_16ak04-R1JwV9FP5Bg


• 提取码:

•3z8f

课程内容

利用Flex/Bison构造编译器

(实验文件夹里有课程PPT和参考资料)

- 1. Flex 理论与练习(8学时)
- 2. Bison理论与练习(8学时)
- 综合实验：
 - 3. 使用Flex和Bison开发了一个具有全部功能的桌面计算器 (16学时) 
 - 4. 编译器后端实现及目标代码生成 (16学时)

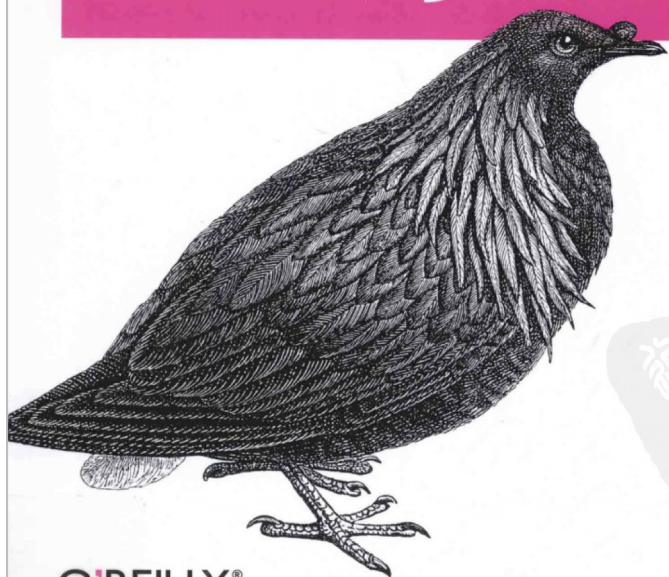
参考书

flex & bison



flex 与 bison

(中文版)



O'REILLY®

东南大学出版社

John Levine 著

陆军 译

参考书
PDF

Flex&Bison综合实验

- 本实验分为 4 个部分。

- 实验内容：

- 阅读《Flex/Bison.pdf》第三章。使用Flex和

- Bison开发了一个具有全部功能的桌面计算器：

- a) 支持变量；
 - b) 实现赋值功能；
 - c) 实现比较表达式（大于、小于等）；
 - d) 实现if/then/else和do/while的流程控制；
 - e) 用户可以自定义函数；
 - f) 简单的错误恢复机制。

- 实验目标：

- 重点学习抽象语法树的用法。

Flex&Bison综合实验

实验3-3（4学时）

•实验内容：

- 1、阅读《Flex&Bison》第三章 P79习题1，学习抽象语法树；
- 2、修改fb3-2的相关代码，实现以下自定义函数，并保存为fb3-3。
- 3、撰写实验报告，结合实验结果，给出抽象语法树的构建过程。。

•实验成果：

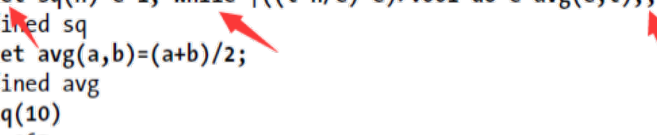
- 提交实验报告。

Flex&Bison综合实验

实验3-3（4学时）

Example 3-9. Computing square roots with the calculator

```
> let sq(n)=e=1; while |((t=n/e)-e)>.001 do e=avg(e,t);;
Defined sq
> let avg(a,b)=(a+b)/2;
Defined avg
> sq(10)
= 3.162
> sqrt(10)
= 3.162
> sq(10)-sqrt(10)
= 0.000178 accurate to better than the .001 cutoff
```



练习

1. 请在高级计算器中尝试一些不同的语法。在前例中，函数`sq`不得不使用两个分号来分别结束`while`循环和`let`语句，这有些臃肿。你可以修改这个语法使它变得更为直观吗？如果你可以添加结束符号至条件语句`if/then/else/fi`和循环语句`while/do/done`，你能够使语句列表的语法更加灵活吗？



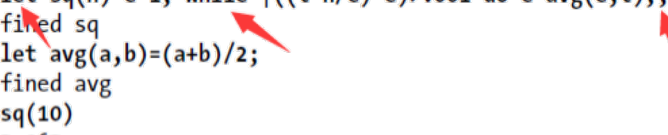
```
let sq(n)
{
  e=1;
  while (|((t=n/e)-e)>.001) do
  {
    e=avg(e,t);
  }
}
```

Flex&Bison综合实验

实验3-3（4学时）

Example 3-9. Computing square roots with the calculator

```
> let sq(n)=e=1; while |((t=n/e)-e)>.001 do e=avg(e,t);;
Defined sq
> let avg(a,b)=(a+b)/2;
Defined avg
> sq(10)
= 3.162
> sqrt(10)
= 3.162
> sq(10)-sqrt(10)
= 0.000178 accurate to better than the .001 cutoff
```



练习

1. 请在高级计算器中尝试一些不同的语法。在前例中，函数`sq`不得不使用两个分号来分别结束`while`循环和`let`语句，这有些臃肿。你可以修改这个语法使它变得更为直观吗？如果你可以添加结束符号至条件语句`if/then/else/fi`和循环语句`while/do/done`，你能够使语句列表的语法更加灵活吗？



```
let avg(a,b) { (a+b)/2; }
```

```
let max(a,b) { if(a>b) then a; else b; }
```

```
let max3(a,b,c)
```

```
{
  if(a>b) then { if(a>c) then a; else c; }
  else { if(b>c) then b; else c; }
}
```


步骤1： 执行计算任务

定义函数 $\text{sq}(n)$ 、 $\text{avg}(a, b)$ ，用于计算平方根

```
C:\GnuWin32\bin>fb3-2.tab.exe
```

```
> let avg(a,b) { (a+b)/2; }
```

```
Defined avg
```

```
> let sq(n) { e=1; while (|((t=n/e)-e)|>.001) do { e=avg(e,t); } }
```

```
Defined sq
```

```
> sq(10);
```

```
= 3.162
```

```
> sq(10)-sqrt(10);
```

```
= 0.000178
```

步骤2： 执行计算任务

定义函数 $\max(a, b)$ ，计算 a, b 的最大值

```
C:\GnuWin32\bin>fb3-2.tab.exe
```

```
> let max(a,b) { if(a>b) then a; else b; }
```

```
Defined max
```

```
> max(4,5);
```

```
= 5
```

```
> max(1,5);
```

```
= 5
```

```
> max(9,5);
```

```
= 9
```

步骤3： 执行计算任务

定义函数 $\text{max3}(a, b, c)$ ，计算 a, b, c 的最大值

```
> let max3(a,b,c) { if(a>b) then { if(a>c) then a; else c; } else  
{ if(b>c) then b; else c; } }
```

Defined max3

```
> max3(3,4,5);
```

= 5

```
> max3(3,9,5);
```

= 9

```
> max3(10,9,5);
```

= 10

提交成果

- 1、提交Windows和Linux环境下的、**步骤1~3**的计算结果截图到[SPOC](#)。（说明：**每种环境各一张截图**）。
- 2、提交修改后的Bison源代码（ fb3-3.y ）、分析Flex源代码（ fb3-3.l ）、 fb3-3funcs.c、fb3-3.h。
- 3、实验报告
 - 3.1 报告中阐述修改的思路，并给出相关代码；
 - 3.2 **结合实验结果，如avg(3,5)，给出抽象语法树的构建过程。**
 - 3.3 你在编程过程中遇到了哪些难题？你是怎么克服的？你的收获有哪些？