

编译原理Lab0

牛庆源 PB21111733

思考题

`\w+([-+.]\w+)*@\w+([-.\]\w+)*\.\w+([-.\]\w+)*` 正则表达式匹配的字符串的含义是什么？

`\w+` 匹配一个或多个字母、数字或者下划线

表示用户名

`([-+.\w+)*` 匹配连字符、加号、点中的一个后跟`\w+`即一个或者多个字母、数字或者下划线

表示用户名中的特殊字符

@ 匹配符号@

表示“@”

`\w+` 同上

表示域名

`([-.\w+)*` 同第二项，但不能有加号

表示域名中的特殊字符

\. 匹配符号.

表示“.”

`\w+` 同上

表示顶级域名

`([-.\w+)*` 同倒数第二项

表示顶级域名中的特殊字符

这个正则表达式匹配电子邮件地址。

匹配 HTML 注释：编写一个正则表达式，可以匹配 HTML 中的注释，例如 `<!-- This is a comment -->`。

`<!--(.*?)-->`

如果存在同时以下规则和动作，对于字符串 `+=`，哪条规则会被触发，并尝试解释理由。

```
%%
\+ { return ADD; }
= { return ASSIGN; }
```

```
\+= { return ASSIGNADD; }  
%%
```

return ASSIGNADD.

有多个匹配项匹配文本最多的项

如果存在同时以下规则和动作，对于字符串 ABC，哪条规则会被触发，并尝试解释理由。

```
%%  
ABC { return 1; }  
[a-zA-Z]+ {return 2; }  
%%
```

return 1

长度相同且先被匹配到

如果存在同时以下规则和动作，对于字符串 ABC，哪条规则会被触发，并尝试解释理由。

```
%%  
[a-zA-Z]+ {return 2; }  
ABC { return 1; }  
%%
```

return 2

长度相同且先被匹配到

上述计算器例子的文法中存在左递归，为什么 bison 可以处理？

bison 语法树自下而上建立，可以处理左递归

能否修改计算器例子的文法，使得它支持除数 0 规避功能？

可以，加入特判即可。如下：

```
case '/': if($3 == 0) {$$ = 0; printf("Error: Division by zero\n");} else $$ = $1 / $3;  
break;
```