

编译原理

2014

本来想照题的，做完感觉没啥新意，就不照了。

考点还是那几个。

LL(1), LR(0), SLR(1)和LR(1)的状态机和分析表构建。

短语、直接短语、句柄的定义。

L模式和S模式的翻译模式（给例子写拓展）。

UD链和DU链，以及活跃变量的计算。

实验相关考了类的实例的成员变量地址，VTable的（详细具体）内容（offset和function的对应关系啥的），考察方式是给TAC代码补Decaf代码。

以及给lex和yacc代码求匹配结果。

特殊的一题是要构造一个串使得同一文法（悬挂else打最近匹配补丁）的SLR(1)自动机和LR(1)自动机的匹配失败时ACTION的长度不一样。

认真/好好刷下几次作业题，然后做两套题就行了（e_n的06-07啥的）。

大约复习时间是（全力）6-8h.....（个人）

2013

顺手发一下编译好了。我依然记得的不多了。大家补充。

实验题：

1 一段lex代码，问作用是什么，答案应该是统计单词个数吧。反正就是每一个单词就count++。

2 3 4 记不得了= =，有一个是关于变量作用域的，好像是给了一句话里面引用了一个参数，然后问这个参数可能出现在哪个参数表里，我记得有好多种情况吧，可以是global也可以是某一个函数的formal之类的。

5 给一段代码，两个类，一个继承另一个的。好像介绍了一下vtable，然后问每一个类的vtable和每一个类的一个实例分别占多大空间。这道题考完之后大家对答案就没有任意两个人是一样的。。

其它：

常规的什么控制流啊，还有句柄啊，是不是LR(0)算法啊，画FSM啊，有什么冲突啊，之类的。最后那道题考的是continue，要求写一个合法性的判断语句，一个属性文法，一个拉链回填。

2012

题目2大张，不过代码很多

需要速度

一、decaf

1. BYACC中函数的表示为

Function: Attributes Type Ident blabla...

其中Attributes表示函数性质，如static, private, public等，EBNF为

Attributes: Attribute*

为方便起见，假定每个属性可以出现多次，且private和public可重复出现，如public static public private void f()

//这都什么玩意儿.....

假定static=1,public=2,private=4,那么Attributes=5=101表示private static。

写出Attributes的BYACC代码:

Attributes:_____

```
{
    _____
}
///*empty*/{
/*nothing*/
}
```

2.在判断BadArgCountError中有这样一段代码:

argNum= func.isStatic()? func.getArglist().size():func.getArglist().size()+1;

请解释为什么一个+1一个不+1.

3.TAC码

```
class Main{
    int x;
    int y;
    int f(int a){return a;}
    int g(int a){return a*a;}

    static void main(){
        class Main a;
        _____
    }
}
```

还有简写的vtable

然后给了一段TAC码

//完全就是天书.....全忘了

然后填答案

估计答案是a.x = a.f(a.y) 四个中的一个

a.y a.x

把TAC反着写出来也就会了

二、

PPT10 53页的图, a:=0变成a:=3

1.编译到第8行时, 作用域栈的所有内容; 以及每个作用域的符号表

2.第2次运行到L行时, Display表中的内容 (要求回答是第几次调用)

3.静态作用域和动态, 在运行时结果有什么不同?

三、

PPT10 20页

1.B5的支配节点, 回边, 自然循环

2.B2的livein, liveout

3.d7的UD链

4.d5的DU链

四、

S→M{P.y:=M.x}P{S.x:=M.x+P.x}

P→{A.y:=0}A{P1.y:=A.x}P1{P.x:=A.x+P1.x}

P→{A.y:=P.y}A{P.x:=A.x}

A→({P.y:=A.y}P{A.x:=P.x})

$A \rightarrow a\{A.x := A.y + 1\}$

$M \rightarrow \varepsilon\{M.x := 0\}$

1. 给出(a)a的语法树，计算顶层节点的属性值
2. $M(P)A$ 和 $M(A)a$ 的短语，直接短语，句柄
3. 改写上述文法，使得可以用top分析，写出代码

五、

$P \rightarrow D;E$

$D \rightarrow D;D$

$D \rightarrow d$

$E \rightarrow a$

1. 作LR(0) FSM, 一部分已画好
2. 上述发生了什么冲突，在哪个状态？
3. 是否是SLR(1)的
4. 做LR(1) FSM, 一部分已画好
5. 是否是LR(1)的
6. 添加规则使文法可以使用LR(1)分析；并画状态表，一部分已画好

六、

$S \rightarrow \{P.y := 0\}P\{S.x := P.x\}$

$S \rightarrow \varepsilon\{S.x := 0\}$

$P \rightarrow \{P1.y := P.y\}P1\{P2.y := P1.y\}P2\{P.x := P1.x + P2.x\}$

$P \rightarrow a\{P.x := P.y + 1\}$

1. 求first集，follow集, PS集
2. 是否是LL(1)的
3. 画预测表
4. 写出递归下降的语法代码。

七、

每年都有的很长一堆code，今年是gen形式的

要求实现switch语句和break语句

switch:

$S \rightarrow \text{case } A \text{ is } L$

$L \rightarrow V:S L$

$L \rightarrow \text{default } S$

break:

$S \rightarrow \text{while } E \text{ do } S$

$S \rightarrow S1;S2$

$S \rightarrow \text{break}$

--

风萧萧兮易水寒，DB作业兮做不完！

反正这个月的生活只要改中间的名词就行了.....

2010

我目前记得的就是最后一道翻译模式的题了：

$S \rightarrow \text{if } G \text{ fi}$

$G \rightarrow E:S$

$G \rightarrow E:S \$ G$

其中if $E_1:S_1 E_2:S_2 \dots fi$

表示如果 E_1 为真则执行 S_1 并跳出，否则继续判断 E_2 ，直到块结束

用拉链与代码回填技术

我的做法，不一定对。。

【在 jesu (被学弟鄙视~~~) 的大作中提到:】

我目前记得的就是最后一道翻译模式的题了:

S->if G fi

```
{  
    S.nextlist = G.nextlist  
}
```

G-> E:MSN

```
{  
    backpatch(E.truelist,M.gotostm)  
    G.nextlist = merge(E.falselist,merge(S.nextlist,N.nextlist))  
}
```

G-> E:M_1SN \$ M_2G_1

```
{  
    backpatch(E.truelist,M_1.gotostm)  
    backpatch(E.falselist,M_2.gotostm)  
    G.nextlist = merge(S.nextlist,merge(N.nextlist,G_1.nextlist))  
}
```

其中if E_1:S_1 E2:S_2 ... fi

表示如果E_1为真则执行S_1并跳出，否则继续判断E2，直到块结束

用拉链与代码回填技术