可能会考的内容

第1章 概论

重要的缩写，比如：AST(抽象语法树)

RE：正则表达式，Transition Diagram：状态转换图

DFA：确定有穷自动机，NFA：非确定有穷自动机

CFG：上下文无关文法，parse tree：语法分析树，AST：抽象语法树（简称syntax tree：语法树）

SDD：语法制导定义，SDT：语法制导翻译

第3章 词法分析

1. 给出一个描述，写出其正则表达式
2. 给定一个正则表达式，描述它所定义的语言
3. 给定一个正则表达式，画出它的NFA，并且转换为DFA
4. **DFA状态最小化不考**

第4章 语法分析

综合考法：给定一个LL(1)文法：

1. 提取左公因子
2. 消除直接/多步左递归
3. 求出其First/Follow函数
4. 构造预测分析表，用其对串进行预测分析

给定一个LR文法：

1. 直接对串进行移入-规约分析
2. 构造其状态转移表
3. 根据状态转移表进行移入-规约

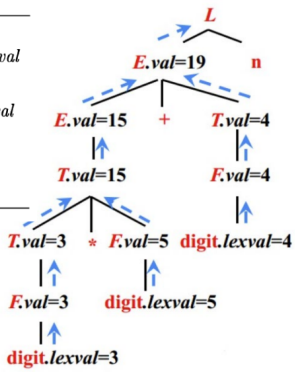
SLR、LR(1)、LALR不考

第5章 语法制导翻译

1. 给定产生式和语义规则，给出表达式的语法分析树

产生式	语义规则
1) $L \rightarrow E$ **n**	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow$ **digit**	$F.val =$ **digit**.lexval

- 对于输入3*5+4n
- n表示结束



2. 有区分度的考法：消除产生式中的左递归，并重写语义规则，再得出语法分析树。

产生式	语义规则
1) $T \rightarrow F\ T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow *\ F\ T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow \textbf{digit}$	$F.val = \textbf{digit}.lexval$

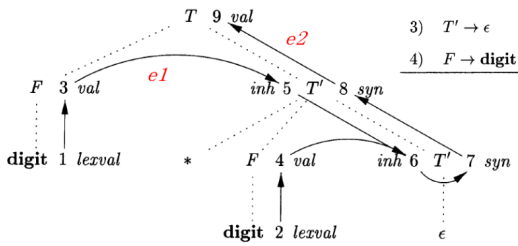
- 注意：T'的属性inh实际上继承了相应的*号的左运算分量。

3. 还可能考某注释分析树的依赖图，注意：从实例a1到实例a2的有向边表示计算a2时需要a1的值（必须先计算a2，再计算a1）——先算指向后算，表示的是数据流的传递。

   虚线表示注释语法分析树，实线表示依赖图关系。

- 3*2的注释分析树；
- T → FT' {T.val = T'.syn; T'.inh = F.val;}
  - 边e1、e2
- 可能的计算顺序：
  - 1,2,3,4,5,6,7,8.9

产生式	语义规则
1) $T \rightarrow F\ T'$	$T'.inh = F.val$ $T.val = T'.syn$
2) $T' \rightarrow *\ F\ T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
3) $T' \rightarrow \epsilon$	$T'.syn = T'.inh$
4) $F \rightarrow \textbf{digit}$	$F.val = \textbf{digit}.lexval$



第6章 中间代码生成

1. 构造表达式的有向无环图DAG

- 语法树中，公共子表达式每出现一次，就有一个对应的子树
- 表达式的有向无环图(Directed Acyclic Graph,DAG)能够指出表达式中的公共子表达式，更简洁地表示表达式

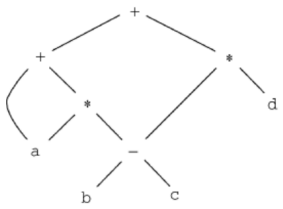


图 6-3 表达式 a+a*(b-c)+(b-c)*d 的 DAG

2. 将表达式转换为三地址代码
3. 给定一段代码，将其转换为静态单赋值形式（SSA）
4. 局部变量的存储布局（暂未给例题）
5. 类型检查与转换（考法见复习笔记）
6. 表达式翻译的SDT和控制流语句翻译的SDT
7. **回填（重点！）**

第7-9章，看例题