编译原理模拟试题

Final exam demo

1、正则表达式和有限自动机

正则表达式可以转换为 NFA 和 DFA，也可以写出对应的 CFG。已知字母表{a,b,c}，任意构思一个正则式及其对应的 NFA 和 DFA，以及 CFG， 并给出 NFA 和 DFA 的状态数。（考核标准中，NFA 和 DFA 必须具有起始状态和接受状态，DFA 出项必须包含字母表中的每个符号。CFG 必须是标准的 BNF 形式）

2、CFG 的改写

a) 将二义性文法改造成为无二义性文法。

Rewrite the grammar into an unambiguous grammar.

文法 3-2

$E \rightarrow$ id
$E \rightarrow$ num
$E \rightarrow E * E$
$E \rightarrow E / E$
$E \rightarrow E + E$
$E \rightarrow E - E$
$E \rightarrow (E)$

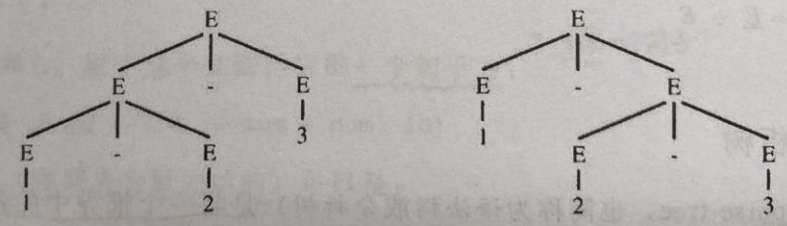

图 3-3 文法 3-2 的句子 1－2－3 的两棵语法分析树

b) 对以下文法提取左因子，消除左递归。

Left factor and eliminate left recursive

参考以前作业

3、自顶向下的分析

a. 计算下面文法的 nullable、FIRST 和 FOLLOW 集合：

$S \rightarrow u\ B\ D\ z$
$B \rightarrow B\ v$
$B \rightarrow w$
$D \rightarrow E\ F$
$E \rightarrow y$
$E \rightarrow$
$F \rightarrow x$
$F \rightarrow$

b. 构造 LL(1)分析表。

c. 给出证据说明该文法不是 LL(1)文法。

d. 尽可能少地修改该文法使它成为一个接收相同语言的 LL(1)文法。

a. Calculate nullable, FIRST, and FOLLOW for this grammar:

$S \rightarrow u\ B\ D\ z$
$B \rightarrow B\ v$
$B \rightarrow w$
$D \rightarrow E\ F$
$E \rightarrow y$
$E \rightarrow$
$F \rightarrow x$
$F \rightarrow$

b. Construct the LL(1) parsing table.

c. Give evidence that this grammar is not LL(1).

d. Modify the grammar **as little as possible** to make an LL(1) grammar that accepts the same language.

4、Bottom-Up Parsers

3.11 构造下面这个文法的 LR(0)状态，然后确定该文法是否为 SLR 文法。

0 $S \rightarrow B\ \$$

1 $B \rightarrow \text{id}\ P$
2 $B \rightarrow \text{id}\ (\ E\ ]$

3 $P \rightarrow$
4 $P \rightarrow (\ E\ )$

5 $E \rightarrow B$
6 $E \rightarrow B\ ,\ E$

Construct the LR(0) states for this grammar, and then determine whether it is an SLR grammar.

0 $S \rightarrow B\ \$$

1 $B \rightarrow \text{id}\ P$
2 $B \rightarrow \text{id}\ (\ E\ ]$

3 $P \rightarrow$
4 $P \rightarrow (\ E\ )$

5 $E \rightarrow B$
6 $E \rightarrow B\ ,\ E$

5、Syntax-Directed Translation

参考书上习题 6.6

Consider the following grammar for integer binary trees (in linearized form):

btree→**(number** *btree btree* **)|nil**

Write a **YACC specification** to check that a binary tree is ordered, that is, that the values of the numbers of the first subtree are ⩽ the value of the current number and the values of all the numbers of the second subtree are ⩾ the value of the current number. For example,**(2 (1 nil nil)(3 nil nil))**is ordered, but**(1 (2 nil nil)(3 nil nil))**is not. You may use the following attributes: **int num, bool order**. Use bison syntax: **$i.num** refers to the **num** attribute of the *ith* symbol of the production and **$$.num** refers to the num attribute of the production's result. Do not use global variables.