**样题B参考答案**

# 1填空题（20分）

**1.1 符号表管理 ， 出错处理 1.2 词法（或构词） ， 语法**

**1.3 语法， 语义 1.4 {a, b}**

**1.5 上下文无关文法 (CFG)， 下推自动机(PDA)**

**1.6 低 ，右 1.7 上下文有关文法(CSG)**

**1.8 匹配（终结符） ， 调用相应子程序（展开非终结符）**

**1.9 符号表， 翻译成中间代码（或三地址码）**

**1.10短路 1.11 类型转换 1.12 arr + 6 \* d**

2简答题（30分）

**2.1 共同点： 均完成对源程序的翻译；**

**差异：编译器采用先翻译后执行，解释器采用边翻译边执行。**

**2.2 （1）DFA上没有 ε转移；**

**（2）在任一状态下，对于任一输入，其下一状态最多仅有一个。**

**【可选】这两方面使得NFA的最大特点是其不确定性，而DFA的最大特点是确定性。**

**2.3 R1: aaa, aba；首尾均为a的ab串。**

**R2: abb, aabb；至少包含1个子串 abb 的ab串。**

**2.4 对于句子 id or id or id，存在以下两棵不同分析树，因此G是二义的。**

****

**消除文法二义性的方法主要有：（1）改写文法；（2）为文法中的符号规定优先级与结合性。**

**2.5 因为文法中存在直接左递归，所以G不是LL(1)的。改写后的文法为：**

**E → TE’ E’ → \*TE’ |ε T → num T’ T’ → / num T’ |ε**

**2.6 好处主要有两点：一是便于编译程序的开发和移植；**

**二是便于对代码进行优化处理。**

**特点：便于语法制导翻译；既与机器指令的结构相近，**

**又与具体机器无关。**

**2.7后缀式为（2分）： a~bc~&%, 语法树如右图。**

3.计算题（4小题，50分）

**3.1 （1） 一个NFA如下图：**

****

**（2） 用子集法构造的DFA状态转换矩阵如下：**

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **{ 0,1,2,4,7 }** | **{ 3, 6,7,1,2,4 }** | **{ 5, 8 , 6,7,1,2,4, 9, 11 }** |
| **{ 3, 6,7,1,2,4}** | **{ 3, 6,7,1,2,4 }** | **{ 5, 8 , 6,7,1,2,4, 9, 11 }** |
| **{ 5, 8 , 6,7,1,2,4, 9, 11 }** | **{ 3, 10, 6,7,1,2,4, 14}** | **{ 5, 8, 12, 6,7,1,2,4, 9, 11,14 }** |
| **{ 3, 10, 6,7,1,2,4, 14}** | **{ 3, 6,7,1,2,4 }** | **{ 5, 8 , 6,7,1,2,4, 9, 11 }** |
| **{ 5, 8, 12, 6,7,1,2,4, 9, 11,14 }** | **{ 3, 10, 6,7,1,2,4, 14}** | **{ 5, 8, 12, 6,7,1,2,4, 9, 11,14 }** |

**分别令 A={ 0,1,2,4,7 } , B={ 3, 6,7,1,2,4}, C={ 5, 8 , 6,7,1,2,4, 9, 11 },D={ 3, 10,6,7,1,2,4, 14},**

**E={ 5, 8, 12, 6,7,1,2,4, 9, 11,14 }，则上述矩阵可表示如下，其中A为初态，D和E为终态：**

|  |  |  |
| --- | --- | --- |
|  | **0** | **1** |
| **A** | **B** | **C** |
| **B** | **B** | **C** |
| **C** | **D** | **E** |
| **D** | **B** | **C** |
| **E** | **D** | **E** |

**（3） 状态数最少的DFA的状态转换矩阵和状态转换图分别如下。**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  | **0** | **1** | | **A** | **A** | **C** | | **C** | **D** | **E** | | **D** | **A** | **C** | | **E** | **D** | **E** | |  |

**3.2**

**（1）FIRST(B) = {c, ε} FIRST(A) = {a, b, ε} FIRST(S) = {a, b, c, ε}**

**FOLLOW(S) = {#} FOLLOW(A) = {c, #} FOLLOW(B) = {#}**

**（2）若输入为a则按照 aA展开；若输入为b则按照bA；若输入为c或#则按照ε展开。**

**3.3（1）拓广文法G' = G∪{S'→S}，构造的DFA如下，其中I0为初态，I1为终态：**

**（2）G是SLR(1)的。在状态I4中存在移进/归约冲突，但是FOLLOW(S)={#}，FIRST(c)={c}，两个集合相交为空，故冲突可通过简单向前看一个终结符解决，因此该文法是SLR(1)的。**