

新疆大学《编译原理》试题

2019-2020 学年 第一学期

一、是非题（请在括号内，正确的划√，错误的划×）（每个 2 分，共 20 分）

1. 计算机高级语言翻译成低级语言只有解释一种方式。(×)
2. 在编译中进行语法检查的目的是为了发现程序中所有错误。(×)
3. 甲机上的某编译程序在乙机上能直接使用的必要条件是甲机和乙机的操作系统功能完全相同。(√)
4. 正则文法其产生式为 $A \rightarrow a$, $A \rightarrow Bb$, $A, B \in VN$, $a, b \in VT$ 。(×)
5. 每个文法都能改写为 LL(1) 文法。(√)
6. 递归下降法允许任一非终极符是直接左递归的。(√)
7. 算符优先关系表不一定存在对应的优先函数。(×)
8. 自底而上语法分析方法的主要问题是候选式的选择。(×)
9. LR 法是自顶向下语法分析方法。(×)
10. 简单优先文法允许任意两个产生式具有相同右部。(×)

二、选择题(请在前括号内选择最确切的一项作为答案划一个勾，多划按错论)(每个 4 分，共 40 分)

1. 一个编译程序中，不仅包含词法分析，，中间代码生成，代码优化，目标代码生成等五个部分。
A. () 语法分析 B. () 文法分析 C. () 语言分析 D. () 解释分析
2. 词法分析器用于识别 。
A. () 字符串 B. () 语句
C. () 单词 D. () 标识符
3. 语法分析器则可以发现源程序中的 。

A. () 语义错误 B. () 语法和语义错误

C. () 错误并校正 D. () 语法错误

4. 下面关于解释程序的描述正确的是 。

(1) 解释程序的特点是处理程序时不产生目标代码

(2) 解释程序适用于 COBOL 和 FORTRAN 语言

(3) 解释程序是为打开编译程序技术的僵局而开发的

A. () (1)(2) B. () (1) C. () (1)(2)(3) D. () (2)(3)

5. 解释程序处理语言时，大多数采用的是 方法。

A. () 源程序命令被逐个直接解释执行

B. () 先将源程序转化为中间代码，再解释执行

C. () 先将源程序解释转化为目标程序，再执行

D. () 以上方法都可以

6. 编译过程中，语法分析器的任务就是 。

(1) 分析单词是怎样构成的 (2) 分析单词串是如何构成语句和说明的

(3) 分析语句和说明是如何构成程序的 (4) 分析程序的结构

A. () (2)(3) B. () (2)(3)(4)

C. () (1)(2)(3) D. () (1)(2)(3)(4)

7. 编译程序是一种 。

A. () 汇编程序 B. () 翻译程序

C. () 解释程序 D. () 目标程序

8. 文法 G 所描述的语言是 的集合。

A. () 文法 G 的字母表 V 中所有符号组成的符号串

B. () 文法 G 的字母表 V 的闭包 V^* 中的所有符号串

C. () 由文法的开始符号推出的所有终极符串

D. () 由文法的开始符号推出的所有符号串

9. 文法分为四种类型, 即 0 型、1 型、2 型、3 型。其中 3 型文法是 。

A. () 短语文法

B. () 正则文法

C. () 上下文有关文法

D. () 上下文无关文法

10. 一个上下文无关文法 G 包括四个组成部分, 它们是: 一组非终结符号, 一组终结符号, 一个开始符号, 以及一组 。

A. () 句子

B. () 句型

C. () 单词

D. () 产生式

三、填空题(每空 1 分, 共 10 分)

1. 编译程序的工作过程一般可以划分为词法分析, 语法分析, 语义分析, 中间代码生成, 代码优化等几个基本阶段, 同时还会伴有__表格处理 和 __出错处理__。

2. 若源程序是用高级语言编写的, __目标程序__是机器语言程序或汇编程序, 则其翻译程序称为 __编译程序__。

3. 编译方式与解释方式的根本区别在于__是否生成目标代码 。

4. 对编译程序而言, 输入数据是 源程序__, 输出结果是__目标程序 。

5. 产生式是用于定义 语法成分__的一种书写规则。

6. 语法分析最常用的两类方法是__自上而下__和 自下而上__分析法。

四、简答题 (20 分)

1. 什么是句子? 什么是语言 ?

答: (1) 设 G 是一个给定的文法, S 是文法的开始符号, 如果 Sx (其中 $x \in VT^*$), 则称 x 是文法的一个句子。

(2) 设 $G[S]$ 是给定文法, 则由文法 G 所定义的语言 $L(G)$ 可描述为: $L(G) = \{x \mid Sx, x \in VT^*\}$ 。

参考答案:

(每个 2 分, 共 4 分)

答: (1) 设 G 是一个给定的文法, S 是文法的开始符号, 如果 Sx (其中 $x \in VT^*$), 则称 x 是文法的一个句子。

(2) 设 $G[S]$ 是给定文法, 则由文法 G 所定义的语言 $L(G)$ 可描述为: $L(G) = \{x \mid S \rightarrow x, x \in VT^*\}$ 。

2. 写一文法, 使其语言是偶正整数的集合, 要求:

(1) 允许 0 打头;

(2) 不允许 0 打头。

解: (1) $G[S] = (\{S, P, D, N\}, \{0, 1, 2, \dots, 9\}, P, S)$

P:

$S \rightarrow PD \mid D$

$P \rightarrow NP \mid N$

$D \rightarrow 0 \mid 2 \mid 4 \mid 6 \mid 8$

$N \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

(2) $G[S] = (\{S, P, R, D, N, Q\}, \{0, 1, 2, \dots, 9\}, P, S)$

P:

$S \rightarrow PD \mid P0 \mid D$

$P \rightarrow NR \mid N$

$R \rightarrow QR \mid Q$

$D \rightarrow 2 \mid 4 \mid 6 \mid 8$

$N \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$Q \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

3. 已知文法 $G[E]$ 为:

$E \rightarrow T \mid E + T \mid E - T$

$T \rightarrow F \mid T * F \mid T / F$

$F \rightarrow (E) \mid i$

① 该文法的开始符号 (识别符号) 是什么?

② 请给出该文法的终结符号集合 VT 和非终结符号集合 VN 。

③ 找出句型 $T+T^*F+i$ 的所有短语、简单短语和句柄。

解：① 该文法的开始符号（识别符号）是 E 。

② 该文法的终结符号集合 $VT=\{+, -, *, /, (,), i\}$ 。非终结符号集合 $VN=\{E, T, F\}$ 。

③ 句型 $T+T^*F+i$ 的短语为 i 、 T^*F 、第一个 T 、 $T+T^*F+i$ ；简单短语为 i 、 T^*F 、第一个 T ；句柄为第一个 T 。

4. 构造正规式相应的 NFA： $1(0|1)^*101$

解 $1(0|1)^*101$ 对应的 NFA 为

5. 写出表达式 $(a + b^*c)/(a + b) - d$ 的逆波兰表示和三元式序列。

逆波兰表示： $abc^* + ab + /d -$

三元式序列：
① $(*, b, c)$ ② $(+, a, ①)$ ③ $(+, a, b)$
④ $(/, ②, ③)$ ⑤ $(-, ④, d)$

五. 计算题 (10 分)

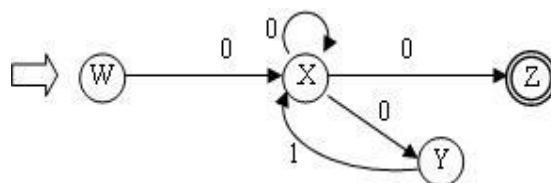
构造下述文法 $G[S]$ 的自动机：

$S \rightarrow A0$

$A \rightarrow A0|S1|0$

该自动机是确定的吗？若不确定，则对它确定化。

解：由于该文法的产生式 $S \rightarrow A0$ ， $A \rightarrow A0|S1|0$ 中没有字符集 VT 的输入，所以不是确定的自动机。要将其确定化，必须先用代入法得到它对应的正规式。把 $S \rightarrow A0$ 代入产生式 $A \rightarrow S1$ 有：
 $A = A0|A01|0 = A(0|01)^*|0 = 0(0|01)^*$ 。代入 $S \rightarrow A0$ 有该文法的正规式： $0(0|01)^*0$ ，所以，改写该文法为确

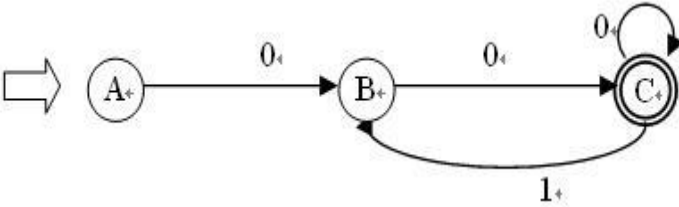


定的自动机为：

由于状态 A 有 3 次输入 0 的重复输入，所以上图只是 NFA，下面将它确定化：

I	$I_0 = \epsilon\text{-closure}(\text{MoveTo}(I, 0))$	$I_1 = \epsilon\text{-closure}(\text{MoveTo}(I, 1))$
A[w]	B[x]	
B[x]	C[x, y, z]	
C[x, y, z]	C[x, y, z]	B[x]

下表由子集法将 NFA 转换为 DFA:



由上表可知 DFA 为: