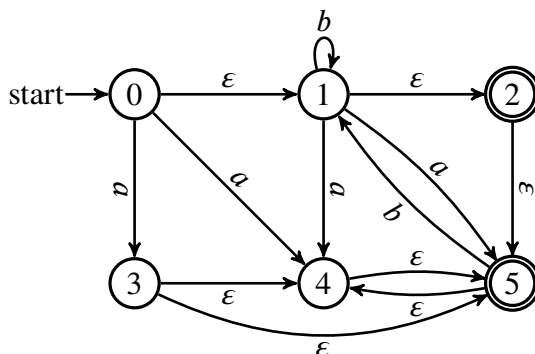


武汉大学计算机学院
2015-2016 学年第一学期 2013 级
《编译原理》期末考试试卷 (A)

学号: _____ 姓名: _____ 专业: _____ 成绩: _____

(注: ①考试时间为 120 分钟; ②所有的解答必须写在答题纸上, 并注明题号。)

一、 设 NFA N 的状态转换图如下所示: (25 分, 每小题 5 分)



- (1) 试写出 NFA N 接受字符串 “babbbab” 的过程;
- (2) 设用子集构造法求出的与 NFA N 等价的 DFA M 有 4 个状态 A, B, C 和 D , 其中 $A = \epsilon\text{-closure}(\{0\})$, 状态转换函数 $\text{Dtrans}(A, a) = B$, $\text{Dtrans}(A, b) = C$ 试求与状态 A, B, C 和 D 所对应的 NFA N 的状态集, 并画出 DFA M 的状态转换图;
- (3) 求 DFA M 的最小状态自动机;
- (4) 试用自然语言描述 NFA N 所接受的语言;
- (5) 求正则表达式 r , 使得 $L(r) = L(N)$.

二、 设 Json 语言的文法 $G(S)$ 定义如下: (25 分, 每小题 5 分)

$$\begin{aligned} S &\rightarrow \{I\} \\ I &\rightarrow I, I \mid C \\ C &\rightarrow i : E \\ E &\rightarrow i \mid S \end{aligned}$$

其中: ‘{’, ‘}’, ‘,’ 和 ‘:’ 为终结符, ‘ S ’, ‘ I ’, ‘ C ’ 和 ‘ E ’ 是非终结符, S 是文法开始符号.

- (1) 试写出语句 “ $\{i : i, i : i\}$ ” 的一个最左推导;
- (2) 试消除文法 $G(S)$ 中的左递归;
- (3) 试对消除左递归后的文法所有非终结符求 First 集和 Follow 集;
- (4) 试对消除左递归后的文法构造 LL(1) 分析表, 从而说明消除左递归后的文法不是 LL(1) 文法;

(5) 试利用你的分析表写出语句 “ $\{i : i\}$ ” 的一个正确的分析过程。

三、 设文法 $G(S)$ 如题二所示： (10 分, 5+5)

- (1) 试对语句 “ $\{i : i, i : i, i : i\}$ ” 画出两颗不同的语法树，从而说明该文法为二义文法；
- (2) 试设计一个与文法 $G(S)$ 等价的无二义的文法，使得列表链接运算 (I, I) 为左结合运算。

四、 设题二文法 $G(S)$ 的拓广文法 $G(S')$ 如下所示： (20 分, 5+5+5+5)

$$S' \rightarrow S \quad (0)$$

$$S \rightarrow \{ I \} \quad (1)$$

$$I \rightarrow I, I \quad (2)$$

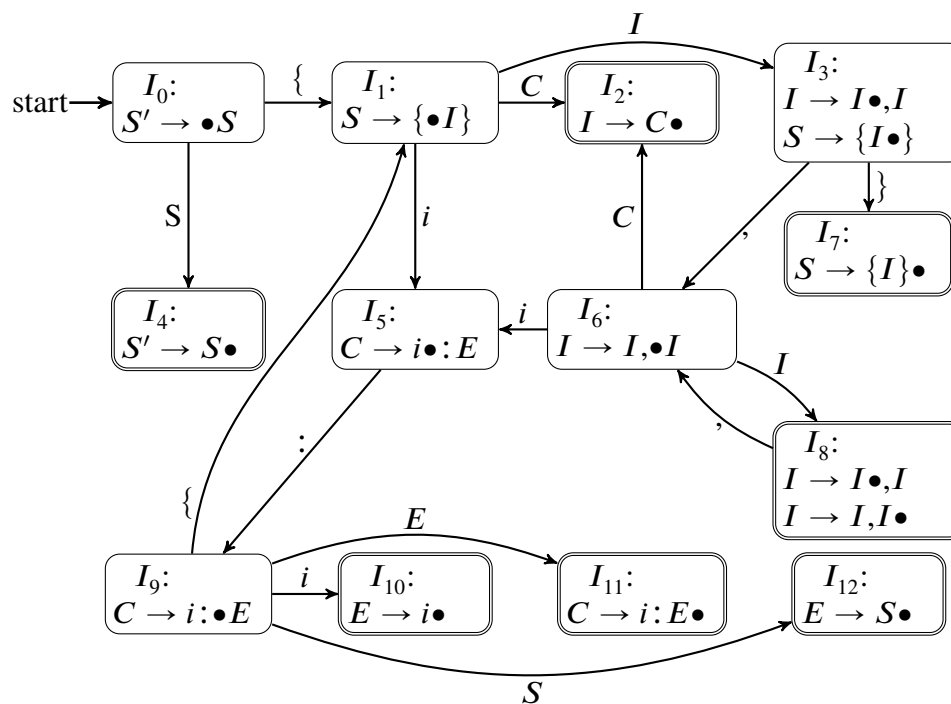
$$I \rightarrow C \quad (3)$$

$$C \rightarrow i : E \quad (4)$$

$$E \rightarrow i \quad (5)$$

$$| \quad S \quad (6)$$

文法 $G(S')$ 的识别活前缀 LR(0) 项目自动机如下图所示 (注意每个状态仅列出了核心项目)：



- (1) 试求状态 I_1 所对应的 LR(0) 项目集；
- (2) 试求状态 I_1 所接受的所有的仅以终结符组成的活前缀对应的正则表达式；
- (3) 试构造该文法的 SLR 分析表，并使得运算的结合次序与题三所规定的一致；

(4) 试利用你的分析表写出语句 “ $\{i : i\}$ ” 的分析过程.

五、Json 是一种轻量级的数据交换格式. 易于人阅读和编写, 同时也易于机器解析和生成. 由于其格式简单, 且易于 XML 相互转化, 因此常用于移动设备, 以减少网络流量. 现需对题四的 Json 文法设计一个 Json 到 XML 转换的语法制导语义定义. 其转换规则如下: 对每个 C 成份中的 i 转换为 XML 的嵌套标签, E 转换为所嵌套的内容. 如: “A:B” 转换为 “<A> B ”, “C:{D:E}” 转换为 “<C> <D> E </D> </C>”. I 是 C 的列表, 转换为每个翻译后的 XML 元素的列表. 如: “A:B,C:{D:E}” 转换为 “<A> B <C> <D> E </D> </C>”. 为此对每个非终结符设计属性 `xml`, 其取值为字符串, 终结符 i 具有属性 `lexval`, 为 i 所对应的词形. (10pt, 5+5)

(1) 完成文法 $G(S')$ 所生成 Json 语句按上述规则转换 XML 语言的语法制导语义定义;

(2) 试写出下述 Json 语句所对应的 XML 语言.

```
{
  "glossary": {
    "title": "example glossary",
    "GlossDiv": {
      "title": "S",
      "GlossList": {
        "GlossEntry": {
          "ID": "SGML",
          "Abbrev": "ISO 8879:1986",
          "GlossDef": {
            "para": "A meta-markup language."
          },
          "GlossSee": "markup"
        }
      }
    }
  }
}
```

六、 设有如下 Pascal 程序片段:

(5 分)

```
repeat
  x := x + 1;
  if not a > b or c > d then break;
  else x := x + 2;
until e > f and not (g > h and i > k);
```

其对应的三地址码如下所示

L0: t0 := x + 1		x := t1
x := t0		[] (e > f) goto L__
[] (a > b) goto L__		[] (g > h) goto L__
[] (c > d) goto L__		[] (i > k) goto L__
t1 := x + 2		L1:

试为其中空白 “__” 填上正确的标号编号, 并为空白 “[]” 填上 if 或 ifnot.

七、 设有如下 GCC 程序:

(5 分)

```
#include <stdio.h>
void foo(char dest[20], char src[20])
{
  int i;
  for (i = 0; i < sizeof(src) && src[i] != '\0'; i++)
    dest[i] = src[i];
  for ( ; i < sizeof(src); i++)
    dest[i] = '\0';
  return;
}

int main(void)
{
  char s[20] = "happy 2016";
  char t[20];
  foo(t, s);
  printf("%s, %d\n", t, sizeof(t));
  return 0;
}
```

该程序在 X86 下用 GCC 能正确编译, 但运行时输出如下乱码:

happ[?][?][?][?], 20

试分析原因.