**编译原理第一次作业**

**任凯 2020141460080**

1. 已知  上正则表达式为 
2. 该正则表达式所定义的语言是什么？

**答：**以 b 结尾的，且含有一个不包括结尾b的“abb”子串的a，b构成的串。

1. 画出接收该语言的NFA

**答：**该接收语言的NFA如下图所示：



1. 把该 NFA 转化成等价的 DFA

**答：**步骤如下

|  |  |  |  |
| --- | --- | --- | --- |
| **M** | **Set of NFA States** | **a** | **b** |
| {0} | {0,1,2,3,5,6,7,9,10,11} | {4,12} | {8} |
| {4,12} | {1,2,3,4,5,6,7,9,10,11,12} | {4, 12} | {8,13} |
| {8} | {1,2,3,5,6,7,8,9,10,11} | {4, 12} | {8} |
| {8, 13} | {1,2,3,5,6,7,8,9,10,11,13} | {4, 12} | {8,14} |
| {8, 14} | {1,2,3,5,6,7,8,9,10,11,14,15,16,18,21} | {4,12,17} | {8,19,22} |
| {4,12,17} | {1,2,3,4,5,6,7,9,10,11,12,15,16,17,18,20,21} | {4,12,17} | {8,13,19,22} |
| {8,19,22} | {1,2,3,5,6,7,8,9,10,11,15,16,18,19,20,21,22} | {4,12,17} | {8,19,22} |
| {8,13,19,22} | {1,2,3,5,6,7,8,9,10,11,13,15,16,18,19,20,21,22} | {4,12,17} | {8,14,19,22} |
| {8,14,19,22} | {1,2,3,5,6,7,8,9,10,11,14,15,16,18,19,20,21,22} | {4,12,17} | {8,19,22} |

|  |  |  |  |
| --- | --- | --- | --- |
| **NFA State** | **DFA State** | **a** | **b** |
| {0} | A | B | C |
| {4,12} | B | B | D |
| {8} | C | B | C |
| {8, 13} | D | B | E |
| {8, 14} | E | F | G |
| {4,12,17} | F | F | H |
| {8,19,22} | G | F | G |
| {8,13,19,22} | H | F | I |
| {8,14,19,22} | I | F | G |

进而可得等价的 DFA 如下图：



1. 对该 DFA 进行状态最小化

**答：**

* {A,B,C,D,E,F} {G,H,I}
* {A,B,C,D} {E,F} {G,H,I}
* {A,B,C}{D}{E,F}{G,H,I}
* {A,C}(0) {B}(1) {D}(2) {E,F}(3) {G,H,I}(4)

最小化 DFA 如下图：



1. 用双层 case 嵌套实现（4）中的 DFA

**答：**代码如下

1. state = 0; *// {start}*
2. while(1){
3. switch(state){
4. case 0:
5. ch = nextChar();
6. switch(ch){
7. case 'a': state = 1;
8. case 'b': state = 0;
9. else fail();
10. break;
11. }
12. case 1:
13. ch = nextChar();
14. switch(ch){
15. case 'a': state = 1;
16. case 'b': state = 2;
17. else fail();
18. break;
19. }
20. case 2:
21. ch = nextChar();
22. switch(ch){
23. case 'a': state = 1;
24. case 'b': state = 3;
25. else fail();
26. break;
27. }
28. case 3:
29. ch = nextChar();
30. switch(ch){
31. case 'a': state = 3;
32. case 'b': state = 4;
33. else fail();
34. break;
35. }
36. case 4:
37. ch = nextChar();
38. switch(ch){
39. case 'a': state = 3;
40. case 'b': state = 4;
41. else fail();
42. break;
43. }
44. if state == 4 {
45. return(accept);
46. }
47. }
48. }
49. 构造一个最小 DFA，能够识别被 3 整除的无符号十进制整数

**答：**类比上课所提到的整除题目，被能够识别被三整除的无符号十进制整数的DFA应该有以下三种状态：

1. ***S0***：除3余0状态；
2. ***S1*：**除3余1状态；
3. ***S2*：**除3余2状态；

将转移条件分为以下三类：

1. ***T0***：0或3或6或9
2. ***T1*：**1或4或7
3. ***T2*：**2或5或8

**根据以上假设可以得到如下最小DFA：**



1. 编写一个程序，将一个C语言源程序中位于注释外的所有保留字全部大写
2. 构建相应的 DFA

**答：**如下图



1. 用表驱动方法实现（1）中的 DFA （写出伪代码）

**答：**驱动表格如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **input char**  **state** | **‘/’** | **‘\*’** | **‘\n’** | **other** | **Accepting** |
| 0 | 1 |  |  |  | no |
| 1 | 4 | 2 |  |  | no |
| 2 | 2 | 3 | 2 | 2 | no |
| 3 | 5 | 3 | 2 | 2 | no |
| 4 | 4 | 4 | 5 | 4 | no |
| 5 |  |  |  |  | yes |

1. state := 0;
2. ch := next input char;
3. while not Accept[state] and not error(state) do
4. newstate := T[state, ch];
5. if Advance[state, ch] then
6. ch := next input char;
7. state := newstate;
8. end while;
9. if Accept[state] then accept;