# Homework 3

1. Give the tokens (including names and attributes) of the following C++ code clip.

int CInt::nMulDiv(int n1, int n2) {

if(n3 == 0) return 0;

else return (n1 \* n2) / n3;

}

**Answer:**

1. <INT, -> 17. <num, point to entry in ST>

2. <ID, point to an entry in ST> 18. <), ->

3. <::, -> 19. <RETURN, ->

4. <ID, point to an entry in ST> 20. <num, point to entry in ST>

5. <(, -> 21. <;, ->

6. <INT, -> 22. <ELSE, ->

7. <ID, point to an entry in ST> 23. <RETURN, ->

8. <,, -> 24. <(, ->

9. <INT, -> 25. <ID, point to an entry in ST>

10. <ID, point to an entry in ST> 26. <MUL, ->

11. <), -> 27. < ID, point to an entry in ST >

12. <{, -> 28. <), ->

13. <IF, -> 29. <DIV, ->

14. <(, -> 30. < ID, point to an entry in ST >

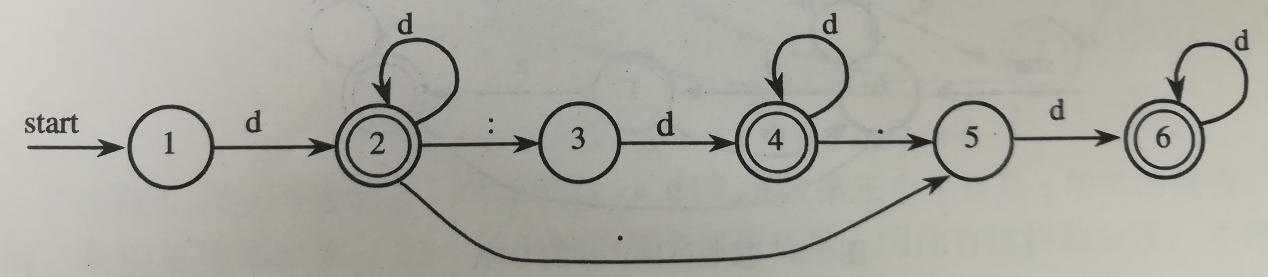
15. <ID, point to an entry in ST> 31. <;, ->

16. <RELOP, EQ> 32. <}, ->

1. The legal file name of a system has the form

device:name.extension

device and extension are optional. Simply, device, name and extension are strings without length limitation (but at least include one character). Draw a transition diagram to recognize the file names. (For simplicity, you can use a letter *d* for any legal input character)



1. Write the corresponding regular expressions for the following strings.

(1) Binary strings that end with 01

(2) Decimal integer strings than can be divided by five without remainder.

(3) Binary strings that include odd 1 or odd 0.

**Answers:**

(1) (0|1)\*01

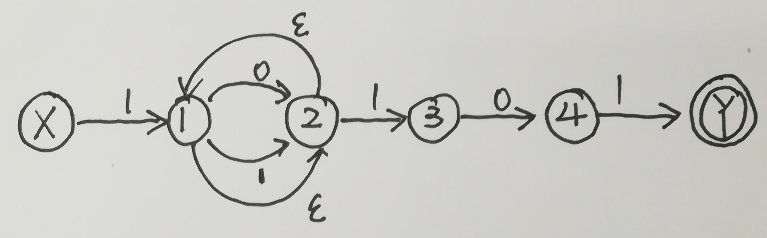
(2)0 | (+|-) 5 | ((1|2|3|4|5|6|7|8|9)(1|2|3|4|5|6|7|8|9|0)\*(0|5))

(3) (0\*1(0|10\*1)\*) | (1\*0(1|01\*0)\*)

1. Construct an NFA for regular expression 1(0|1)\*101, then convert the NFA to DFA.

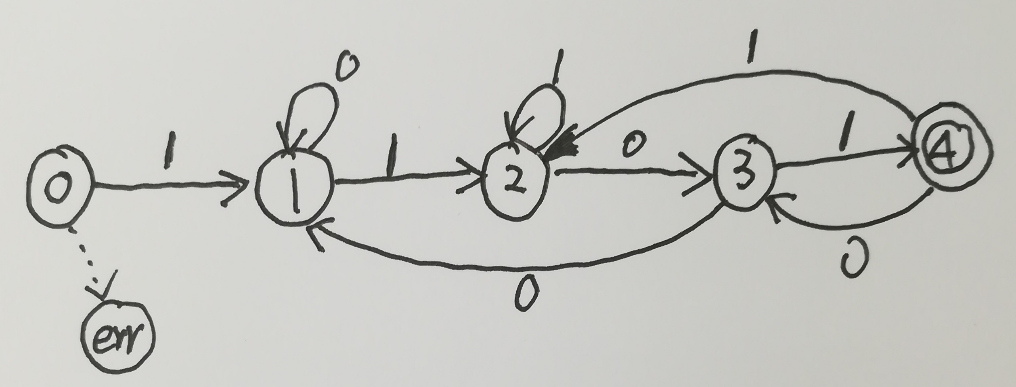
**Answer:**

NFA:



|  |  |  |
| --- | --- | --- |
| I | I0=ε\_closure(move(I, 0)) | I1=ε\_closure(move(I, 1)) |
| {X} |  | {1,2} |
| {1, 2} | {1, 2} | {1,2,3} |
| {1,2,3} | {1,2,4} | {1,2,3} |
| {1,2,4} | {1,2} | {1,2,3,Y} |
| {1,2,3,Y} | {1,2,4} | {1,2,3} |

DFA:



1. Construct a DFA that can accept strings on  that include even numbers of 0 and even numbers of 1.

**Answer:**

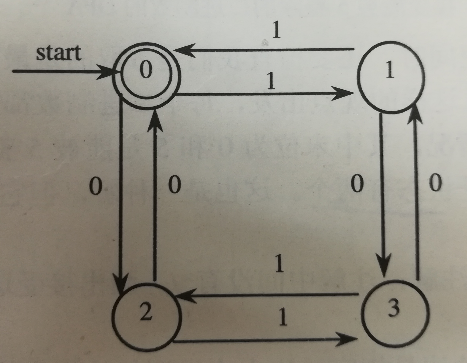
Any string belong to one of the following four types:

0: even 0 and even 1

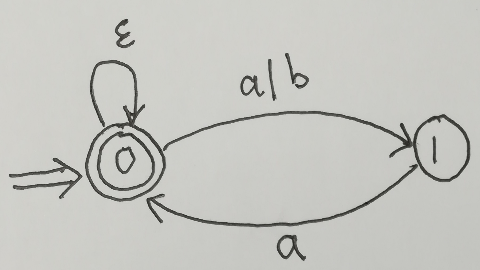
1: even 0 and odd 1

2: odd 0 and even 1

3: odd 0 and odd 1



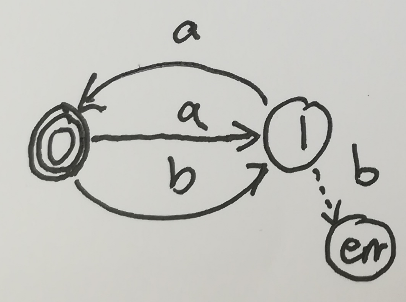
1. Convert the following NFA to DFA



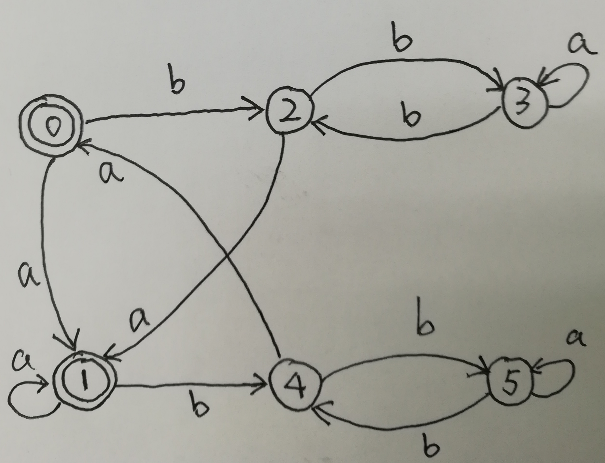
**Answer:**

|  |  |  |
| --- | --- | --- |
| I | I0=ε\_closure(move(I, 0)) | I1=ε\_closure(move(I, 1)) |
| {0} | {1} | {1} |
| {1} | {0} | - |

DFA:



1. Minimize the following DFA



**Answer:**

The first partitions are {0, 1} and {2, 3, 4, 5}



{0, 1} should not be divided.



Finally, we have 0={0, 1}, 1={2, 4}, 2={3, 5}

Minimum DFA:

