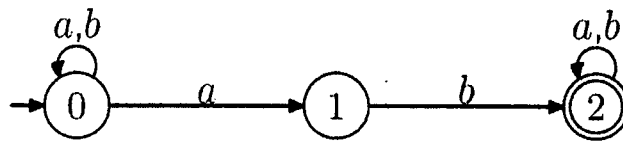


1. Write the number -0.3 in IEEE single precision floating point format. Give your answer in binary. (10%)
2. Recall that \oplus is "exclusive or". Define $y = \overline{a \oplus b} \cdot c$.
 - a) Rewrite y as a sum-of-products where each product depends on all three variables a , b , and c . Use a truth table to compute your solution. (5%)
 - b) Draw a circuit for the sum-of-products in (a). (5%)
3. Draw a parse tree for the derivation of "bbbbaaa" using the grammar below. (10%)

$A \rightarrow BbAa \mid aa$
 $B \rightarrow BB \mid b \mid \epsilon$
4. Design a Turing Machine that accepts the following language: $L = \{ab(a + b)^*\}$. (10%)
5. In RSA algorithm, if $p = 7$ and $q = 13$, what are the five smallest possible numbers for e ? (10%)
6. Convert the following NFA to a DFA using the subset construction. (10%)



7. Let G be the grammar on the right.
 - a) Draw the parse tree for the string "xxy" (5%)
 - b) Can the language of this grammar be captured by a regular expression? If so, give the regular expression. If not, give a short argument why not. (5%)

$S \rightarrow AB$
 $A \rightarrow Ax$
 $A \rightarrow \lambda$
 $B \rightarrow yB$
 $B \rightarrow y$

8. What is the output of the following C++ code? (10%)

```

1  int x = 35;
2  int y = 46;
3  int *p = &x;
4  int *q = &y;
5  *p = 97;
6  p = q;
7  *p = 27;
8  cout<<x<<" "<<y<<endl;
9  cout<<*p<<" "<<*q<<endl;
    
```

見背面

9. The following C program supposes to calculate and print the square root of every second integer number from 1 to 101 inclusive beginning with 1 in two decimal points accuracy. There are syntax or logical errors in this program. For each line that has error(s), state the line number that error(s), and give the correct statement. (10%)

1	#include <stdio.h>
2	#include <math.h>
3	
4	int main(void) {
5	double root;
6	
7	for (int number = 1; number < 101; number += 2) {
8	number = sqrt((double) root);
9	
10	printf("The Square Root of %0.2f is %.1f\n", number, root);
11	}
12	return (EXIT_DENIED);
13	}

10. Show the values of registers R0 and R1 after the each of the instructions in the assembly code below. (10%)

1	MOV R0, #11
2	MOV R1, #13
3	EOR R0, R0, R1
4	EOR R1, R0, R1
5	EOR R0, R0, R1

試題隨卷繳回