MIDTERM, MORNING CLASS

1.

1.1 2P

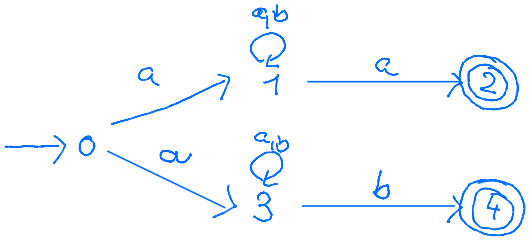
Write down a regular expression for the language over the alphabet {a,b} of all words where the last two letters are different.

1.2 2P

Write down an epsilon-NFA for the language of 1.1.

2. 2P

Consider the NFA given by the following state transition diagram.



Transform this NFA into a DFA. Write out the table of the DFA and its state-transition diagram.

3.

Consider the following C++ program and the grammar Cpp.cf we used as the starting point for Assignment 1.

```
void main () { bool x = 0 ; return x = x+++2 ; }
```

(Note that this question is only about parsing, do not worry about the fact that the program fails to type check.)

3.1 2P

Show the steps taken by a shift-reduce parser. Label the reduction steps by the corresponding name of the rule of the grammar.

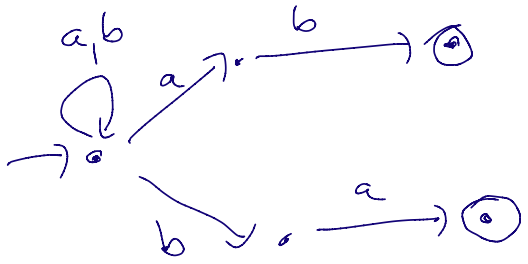
3.2 2P

Write out the abstract syntax tree of the program (in 2-dimensional, not in linearized, notation).

Answers:

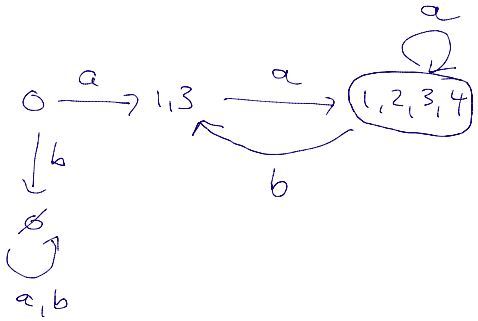
$$(a+b)^* ab + (a+b)^* ba$$

also: $(a+b)^* (ab + ba)$



"first and last either the same"

	State	a	b
initial	0	1,3	∅
	1,3	1,2,3,4	1,3
final	1,2,3,4	1,2,3,4	1,3
	∅	∅	∅



Output of the parser (not required for answer but useful to have at hand):

PDefs [DFun Type_void (Id "main") [] [SInit Type_bool (Id "x") (EInt 0),SReturn (EAss (EId (Id "x")) (EPlus (EPIncr (EId (Id "x"))) (EInt 2)))]]

Abstract Syntax Tree: