

## E02: Regular expressions and scanning

### solutions

Note that there are often several equivalent ways of writing a regular expression.

**E02-1: Solution:**  $[0-9]^+$

**E02-2: Solution:**  $0|1(0|1)^*$

Alternative solution:  $0|1[0-1]^*$

**E02-3: Solution:**  $[0-9]^+(\backslash+\backslash^*[0-9]^+)^*$

Some examples: 0, 44+9, 3+56\*23, 70\*4\*156

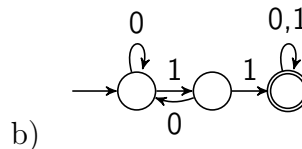
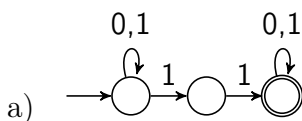
**E02-4: Solution:**

a)  $[0-1]^*11[0-1]^*$

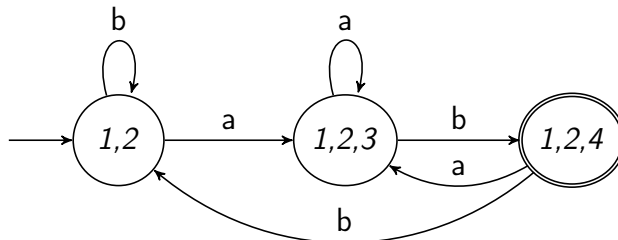
b)  $0^*(10^+)^*(1|\epsilon)$

Alternative solution to b):  $0^*(10^+)^*1?$

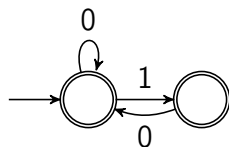
**E02-5: Solution:**



**E02-6: Solution:**



**E02-7: Solution:**



Note that both states are final states.

**E02-8: Solution:**

DFA:

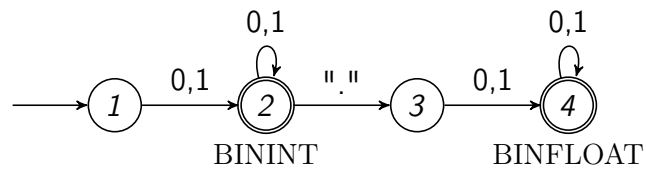


TABLE:

state	0	1	.	kind	final
0	0	0	0	ERROR	yes
1	2	2	0		no
2	2	2	3	BININT	yes
3	4	4	0		no
4	4	4	0	BINFLOAT	yes

The initial state is 1.

**E02-9: Solution:**

2: ID = [a-z][a-z0-9]\*

3: PLUS = "+"

4: MINUS = "-"

5: INT = [0-9]+

8: INT-WITH-EXPONENT = [0-9]+ "e" [+]?[0-9]+

**E02-10: Solution:**

The string 1e+a will require a 3 character lookahead before accepting 1.