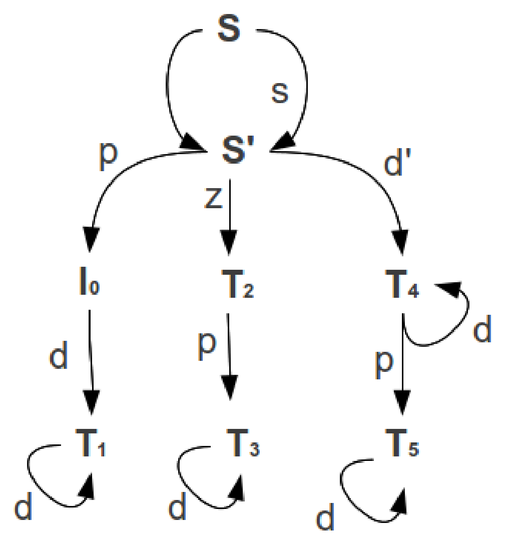
Programming languages and paradigms II. Task 1.

Implement a finite state machine which accepts floating numbers. The definition of the automata is the following:



Examples:

1	OK 1.0
-3.14	OK 3.14
0	OK 0
+0.1	OK 0.1
.1	OK 0.1
1	OK 1.0
.	FAIL
+ x1	FAIL
-	FAIL
3.14e-2	FAIL

Non-terminal symbols:

S: start symbol of signed float
S’: start symbol of unsigned float
T_i: accepted states (terminals)
I_i: non-accepted states (non-terminals)

Terminal symbols:

s: {+,-}
p: {.}
z: {0}
d: {1,..,9}

Production rules:

S -> S’ | sS’
S’ -> pI_0 | zT_2 | dT_4
I_0 -> dT_1
T_1 -> dT_1 | e
T_2 -> pT_3 | e
T_3 -> dT_3 | e
T_4 -> dT_4 | pT_5 | e
T_5 -> dT_5 | e

Program should read standard input line by line until EOF. Each line contains one word and the program prints OK and the value, or FAIL to the standard output. Implement the program in two languages: your favourite language (you are expert) and an other you just want to get familiar. Do not use ANY parser library, or function like Integer.parseInt() in Java.