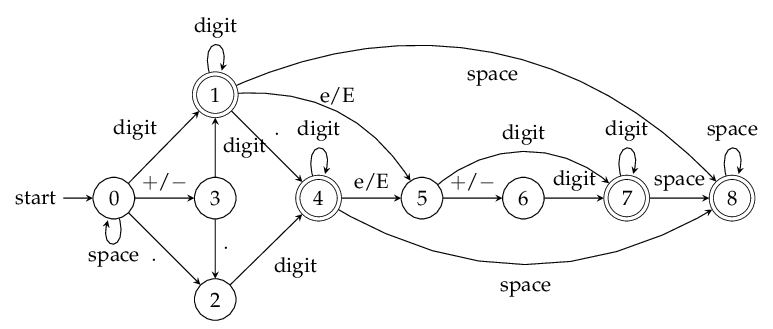
Valid Number

Validate if a given string is numeric.

Some examples:  
"0" => true  
" 0.1 " => true  
"abc" => false  
"1 a" => false  
"2e10" => true

**Note:** It is intended for the problem statement to be ambiguous. You should gather all requirements up front before implementing one.

.

enum InputType {

INVALID(0), // 0

SPACE(1), // 1

SIGN(2), // 2

DIGIT(3), // 3

DOT(4), // 4

EXPONENT(5), // 5

NUM\_INPUTS(6); // 6

int value;

private InputType(int value) {

this.value = value;

}

};

**public boolean isNumber\_X(String s) {**

**int transitionTable[][] = {**

**{-1, 0, 3, 1, 2, -1} // next states for state 0**

**, {-1, 8, -1, 1, 4, 5} // next states for state 1**

**, {-1, -1, -1, 4, -1, -1} // next states for state 2**

**, {-1, -1, -1, 1, 2, -1} // next states for state 3**

**, {-1, 8, -1, 4, -1, 5} // next states for state 4**

**, {-1, -1, 6, 7, -1, -1} // next states for state 5**

**, {-1, -1, -1, 7, -1, -1} // next states for state 6**

**, {-1, 8, -1, 7, -1, -1} // next states for state 7**

**, {-1, 8, -1, -1, -1, -1} // next states for state 8**

**};**

**int state = 0, i = 0;**

**char ch;**

**while (i < s.length()) {**

**InputType input = InputType.INVALID;**

**ch = s.charAt(i);**

**if (ch == ' ')**

**input = InputType.SPACE; //1**

**else if (ch == '+' || ch == '-')**

**input = InputType.SIGN; //2**

**else if (isValidDigit(ch))**

**input = InputType.DIGIT; //3**

**else if (ch == '.')**

**input = InputType.DOT; //4**

**else if (ch == 'e' || ch == 'E')**

**input = InputType.EXPONENT; //5**

**// Get next state from current state and input symbol**

**state = transitionTable[state][input.value];**

**if (state == -1) return false;**

**++i;**

**}**

**// If the current state belongs to one of the accepting (final) states, then the number is valid**

**return state == 1 || state == 4 || state == 7 || state == 8;**

**}**