Problem: LC67

A valid **number** can be split up into(in order):

1. a **decimal number** or an **integer**.
2. (optional) An `'e'` or `'E'`, followed by an integer.

A **decimal number** can split up into(in order):

1. (optional) A sign character, either `'+'` or `'-'`.
2. One of the following formats:
   (a) One or more digits, followed by `'.'`.
   (b) One or more digits, followed by `'.'`, followed by one more digits
   (c) A dot `'.'`, followed by one or more digits.

An **integer** can be split up into:

1. (optional) A sign character, either `'+'` or `'-'`.
2. one or more digits.

For example, all the following are valid numbers:

```
"2", "0089", "-0.1", "+3.14", "4.", "-.9", "2e10", "-90E3", "3e+7",
"+6e-1", "53.5e93", "-123.456e789"
```

while the following are not valid numbers:

```
"abc", "1a", "1e", "e3", "99e2.5", "--6", "-+3", "95a54e53"
```

Given a string `s` return `true` if `s` is a valid **number** or `false` otherwise.

Example 1:
Input: s = "0"
Output: true
Example 2:
Input: s = "e"
Output: false
Example 3:
Input: s = "."

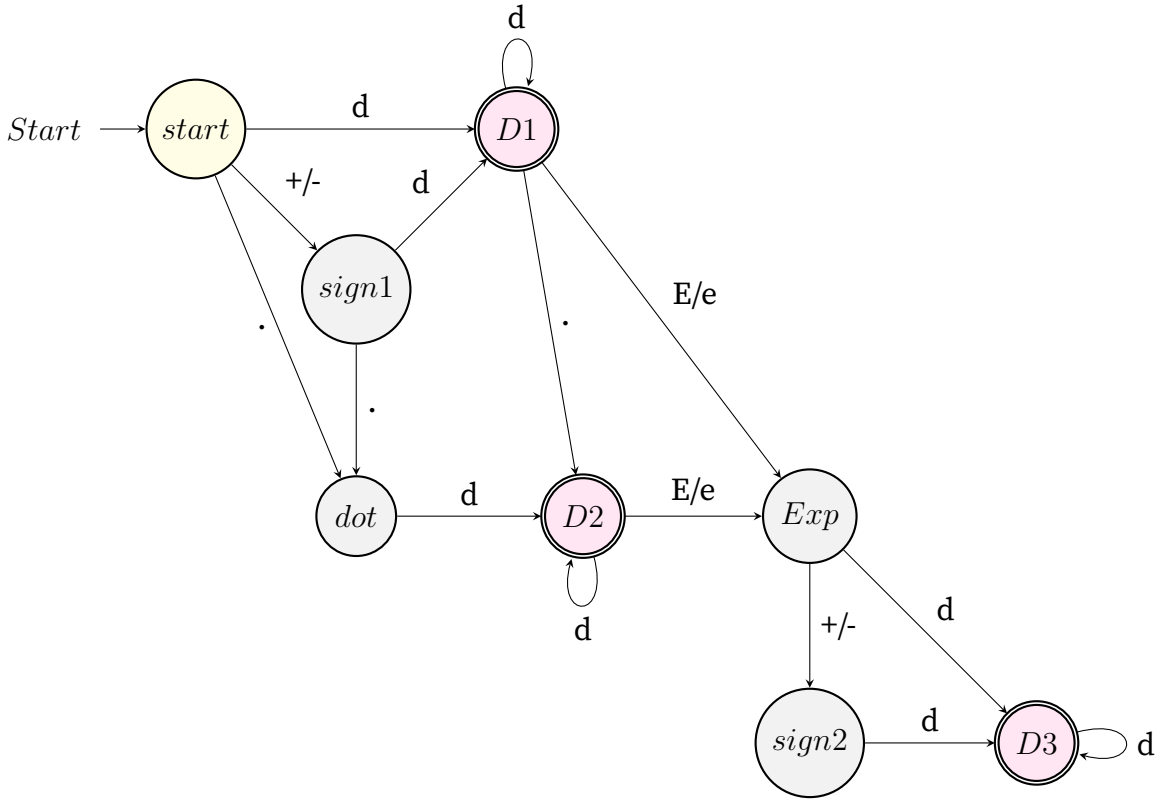
Output: false

Example 4:

Input: s = ".1"

Output: true

Here is the finite state automata based on the problem description. The yellow node is the start, and the pink nodes are terminals.



From the diagram, we can create the corresponding transitions.

```
transitions = {
    'start': {'.': 'dot', '+': 'sign1', '-': 'sign1', 'd': 'D1'},
    'sign1': {'.': 'dot', 'd': 'D1'}, # base sign
    'dot': {'d': 'D2'}, # base dot
    'D1': { 'd': 'D1', '.': 'D2', 'E': 'Exp', 'e': 'Exp'}, # integer
    'D2': {'d': 'D2', 'E': 'Exp', 'e': 'Exp'}, # float
    'Exp': {'+': 'sign2', '-': 'sign2', 'd': 'D3'},
    'sign2': {'d': 'D3'}, # exponent sign
```

```
    'D3': {'d': 'D3'} # exponent
}
```

The terminals are

```
terminals = {'D1', 'D2', 'D3'}
```

The automata program is

```
state = 'start'
for c in s:
    if c.isdigit(): c = 'd'
    tr = transitions[state]
    if c not in tr: return False
    state = tr[c]

return state in terminals
```