

1

1.1

The DFA will recognize all strings that contain two 1's. As this will make it reach the end state. If anything else is given inbetween or after the two ones, a string such as: 00100010101010. It will simply loop to the same node and eventually reach the end state. Therefore any string with less than two 1's will not be recognized.

1.2

$(\backslash s^*)[+-]?(\backslash s^*)(\backslash d+(\backslash \backslash d+)?)(\backslash s^*)([+-](\backslash s^*)(\backslash d+(\backslash \backslash d+)?)?i)|(\backslash s^*)[+-]?(\backslash s^*)(\backslash d+(\backslash \backslash d+)?)$

As the assignment was somewhat unclear on the format of the numbers, i decided to include whitespace. I realized in the last minutes that numbers only with the realpart should also be accepted, I therefore added:

$(\backslash s^*)[+-]?(\backslash s^*)(\backslash d+(\backslash \backslash d+)?)$

to the previous iteration:

$(\backslash s^*)[+-]?(\backslash s^*)(\backslash d+(\backslash \backslash d+)?)(\backslash s^*)([+-](\backslash s^*)(\backslash d+(\backslash \backslash d+)?)?i)$

The newest addition simply says that if the input doesnt match a number of the form: $2+2i$ it should also check if it matches something like: 2.

It will also recognize: $0+i$, $+0$, -0 , 0 , $-0+i$, $+0-i$, $0-123123i$, etc.

1.3

Since a regular expression can be recognized by a DFA it is a regular language. A regular expression cannot recognize an arbitrary number of nested elements, such as parenthesis. It can however recognize a predetermined number of nested elements, just as a DFA can. There are implementations, with different engines, of regular expressions that can implement nested elements, they are fundamentally different.

2

2.1

