Homework 1

2.1) Write regular expressions to capture the following.

(a) Strings in C. These are delimited by double quotes ("), and may not contain newline characters. They may contain double-quote or back-slash characters if and only if those characters are "escaped" by a preceding backslash. You may find it helpful to introduce shorthand notation to represent any character that is not a member of a small specified set.

A regular expression for a string in C can be :

```
" ( [^"\\\n] | \\[^\n] )* "
```

Match any character that is NOT a double-quote (") backslash (\), or newline (\n)

OR

The backslash (\) must be followed by a non newline character (\n)

0 or more times

(b) Comments in Pascal. These are delimited by (* and *) or by { and }. They are not permitted to nest.

A regular expression for comments of the form { } in Pascal can be:

```
\{ [^{}]* \}
```

{ literal

Accept anything that is not another { } for 0 or more times

} literal

A regular expression for comments of the form (* *) in Pascal can be:

```
\( \* ([^*]|\*[^)])* \* \)
```

(literal

* literal

Match any number of characters that are not * OR we do have an * but it most not be followed by a)

* literal

) literal

(f) Financial quantities in American notation. These have a leading dollar sign ($), an optional string of asterisks (*—used on checks to discourage fraud), a string of decimal digits, and an optional fractional part consisting of a decimal point (.) and two decimal digits. The string of digits to the left of the decimal point may consist of a single zero (0). Otherwise it must not start with a zero. If there are more than three digits to the left of the decimal point, groups of three (counting from the right) must be separated by commas (,). Example: $**2,345.67. (Feel free to use "productions" to define abbreviations, so long as the language remains regular.)

A regular expression for financials quantities in American notation can be:

(d is shorthand for integer)

```
\$ \** (0|[1-9]\d{0,2}(,\d{3})*) (\.\d{2})?
```

$ literal

0 or more of the * literal

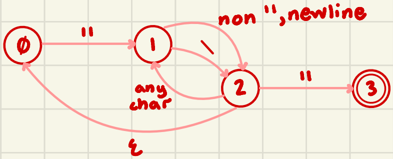
Left-side of decimal portion:

- 0 OR first digit is an integer and up to only two more digits (integers)
- Then we can have 0 or more number of comma + 3-digit groups that can follow e.g. 123,456

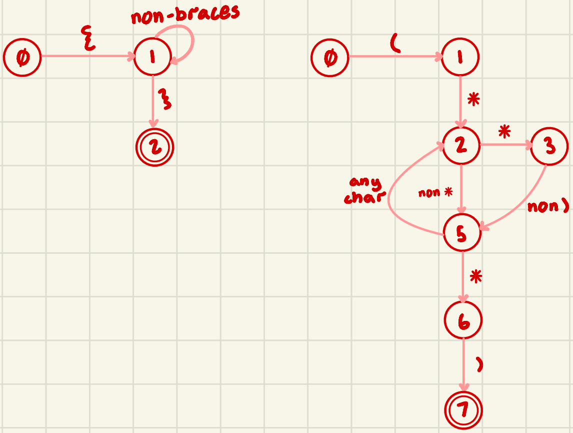
0 or 1 occurrence of a period followed by exactly two digits (for cents) e.g. $0.99 or $100

2.2) Show (as "circles-and-arrows" diagrams) the finite automata for Exercise 2.1.

(a)

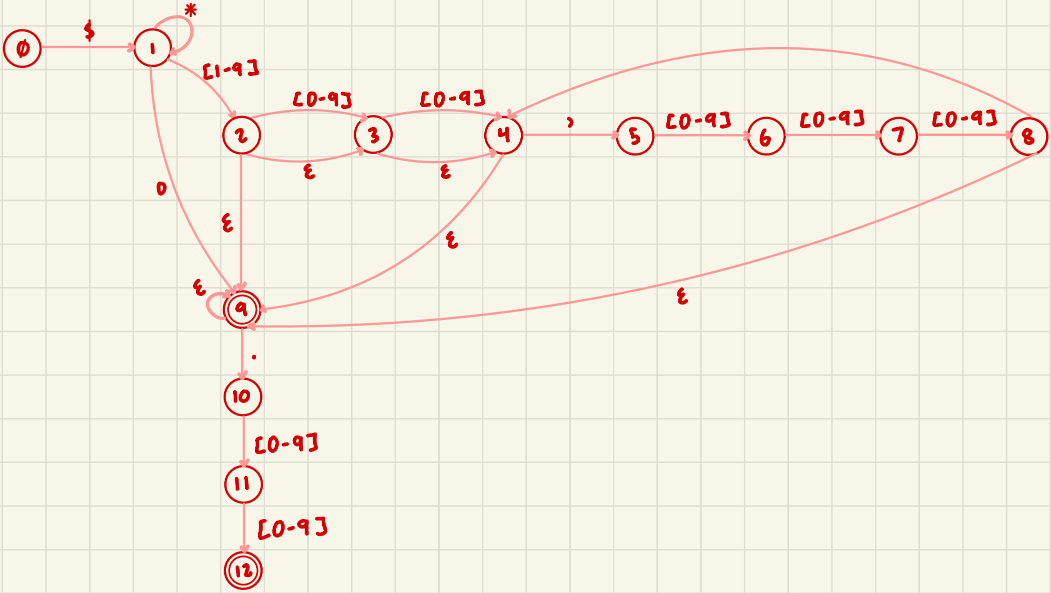


(b)



looping stuff confusing me....

(c)



NOTE: A double circle indicates a final accepting state