CS 1622 – Homework 1

1.) Write the following regular expressions:

a.) Binary numbers that are multiples of eight

(0|1)*000

b.) Binary numbers that are an integer power of 2.

0*10*

c.) Valid C/Java integer constants that can be negative or positive, in decimal, octal, or hexadecimal.

The simple but “wrong” way:

```
([+-]?([1-9][0-9]+)|(0[0-7]+))|(0[xX][0-9a-fA-F]+)
```

The “Valid” part requires us to only match numbers in the proper range. It’s impossible for C, since int literals are different depending on the architecture. But for Java and C if we assumed 32-bit, we might do:

```
0[xX][0-9a-fA-F]{,8}
```

d.) A string literal without escape sequences

```
“[^\”]*”
```

Your regex engine probably would require escaping some of those in the regex itself.

e.) A block comment without nesting (/* to */)

```
Star = \*
CommentStart = /{Star}
CommentEnd = {Star}+/
NotAStar = [^*]
Newline = [\n]
NotAStarOrSlash = [^*/]
```

Ignoring the whitespace in the diagram below, this is what our regex will look like:

```
{CommentStart}
(
      {NotAStar}
      | {Newline}
      |
      (
            {Star}+
            (
                  {NotAStarOrSlash}
                  | {Newline}
            )
      )
)*
{CommentEnd}
```

Yields:

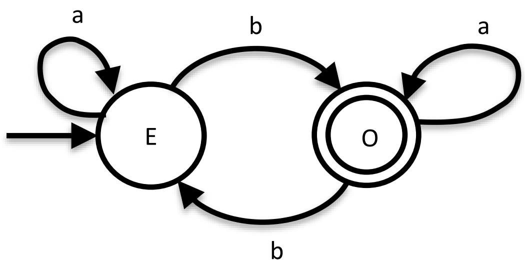
```
/\*([^*]|[\n]|(\*+([^*/]|[\n])))*\*+/
```

Or JFlex will let you do:

```
“/\* ~ \*/”
```

f.) A string of a’s and b’s with an odd number of b’s.

It’s probably easiest to start with a DFA:



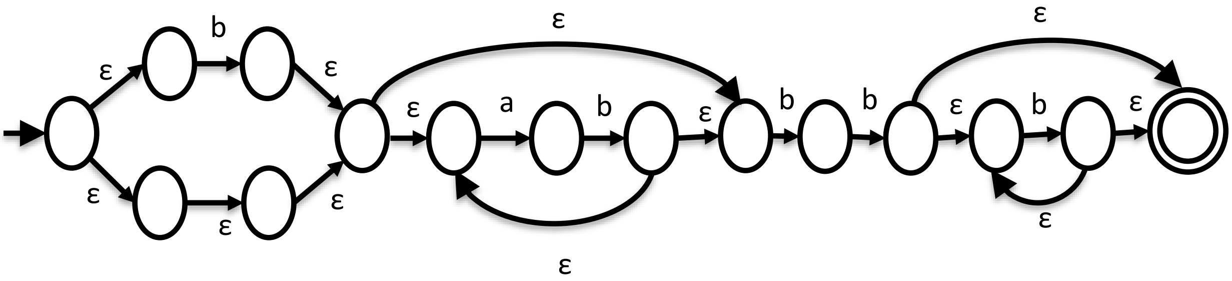
From this we can see repetition of a can be anywhere, and that we need one b and then zero or more pairs of b’s to accept. To convert that into a RE:

a*ba*(a*ba*ba*)*

2.) Using the Thompson's algorithm construction from lecture, convert the following regular expression to an NFA (alphabet is {a,b}):

`b?(ab)*bb+`

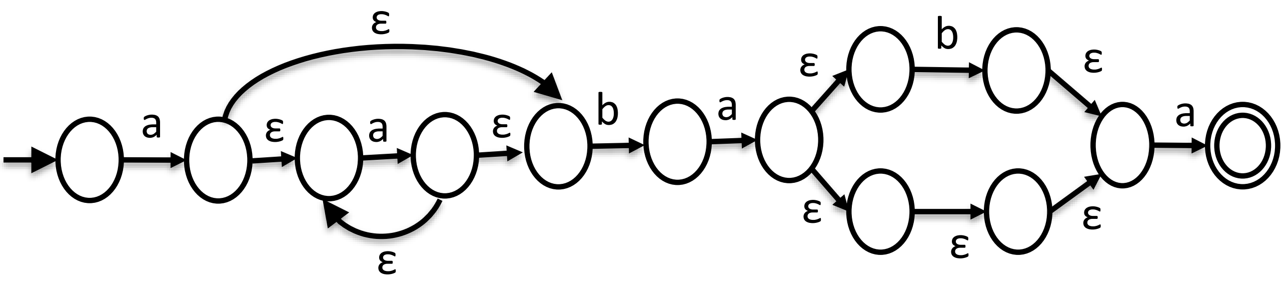
`As a RE of fundamental operations: (b|ε)(ab)*bbb*`



3.) Using the Thompson's algorithm construction from lecture, convert the following regular expression to an NFA (alphabet is {a,b}):

`a+bab?a`

`As a RE of fundamental operations: aa*ba(b|ε)a`


ε
a
ε
a
ε
ε
b
a
ε
b
ε
ε
ε
ε
a