1. (6 pts) Write regular expressions for the following languages over the alphabet \( \Sigma = \{a, b\} \):

(a) (2 pts) The set of all strings ending with \( aab \).
\[
(a|b)^*aab
\]

(b) (2 pts) The set of all strings with three consecutive \( a \)'s.
\[
(a|b)^*aaa(a|b)^*
\]

(c) (2 pts) The set of all strings which do not contain the substring \( bab \).
\[
a^*(b^+aa^+)^*b^*a^*
\]

2. (6 pts) Draw DFA’s for each of the languages from question 1.

(a) (2 pts)

(b) (2 pts)

(c) (2 pts)
3. (4 pts) Consider the following deterministic finite automaton (DFA) over the alphabet \( \Sigma = \{0, 1\} \) (Omitted).

(2 pts) Give a one-sentence description of the language recognized by the DFA.

- The set of strings where any maximal substring of consecutive 1s is of length 1 or 3.

(2 pts) Write a regular expression for this language.

- \( 0^* (1|111) 0^+ (1|111)? \)

4. (3 pts) Let \( L \) be the language over \( \Sigma = \{0, 1\} \) such that every string in \( L \) contains a pair of 0’s that are separated by a string whose length is \( 3i + 1 \), for some \( i \geq 0 \). Draw a non-deterministic finite automaton (NFA) for \( L \). Give an upper bound on the number of states in the corresponding DFA.

The corresponding DFA has less than \( 2^6 = 64 \) states.
5. (4 pts) For each of the following specifications written in Flex, give a regular expression describing the language of possible outputs. Assume that all inputs are strings consisting of characters a, b, A, and B.

(a) (2 pts) Specification 1:

```
[bB][bB] { printf ("w"); }
[aA] { printf ("x"); }
[aA][aA] { printf ("y"); }
[bB] { printf ("z"); }
```

- An even length substring of as or As generates ys, while an odd length substring of as will have one x at the end (because of the maximal munch rule). Thus, strings of as generate the language $y^*x^?$. Similarly, strings of bs or Bs generate $w^*z^?$. In addition, substrings of [aA] and substrings of [bB] appear as a interleaving sequence. So the full language is:

$$y^*x^?((z|w^+z^?)(x|y^+x^?))^*(w^*z^?)$$

Many students write $((w^*z^?)(y^*x^?)^*$ but this regular expression accepts zz which should be rejected.

(b) (2 pts) Specification 2:

```
([bB]+)([aA]*) { printf("x"); }
[aA] { printf("y"); }
[bB] { printf("z"); }
```

- The last rule never gets used as it is subsumed by the first. If the string starts with as then it has to be matched using ys. Thus, this specification can print any string in the language $y^*x^*$.