1. Consider the following functions:

```c
int equal(int x, int y) {
    int result;
    if (x == y)
        result = 1;
    else
        result = 0;
    return result;
}
```

```c
int max(int x, int y, int z) {
    int max = x;
    if (y > max)
        max = y;
    if (z > max)
        max = z;
    return max;
}
```

```c
int min(int x, int y) {
    if (x < y) {
        x = x + y;
        y = x - y;
        x = x - y;
    }
    if (y < x)
        return y;
    else
        return x;
}
```

(a) Construct the symbolic execution tree for these three functions assuming each input value is symbolic (show the symbolic state and the path constraints). Show the resulting path constraints for each execution path.
(b) Assume that each input value is uniformly distributed between 1 and 100. Count the number of solutions for each path constraint and determine the probability of each execution path.

2. Consider the following boolean logic formulas:

\[(a \lor b) \land (b \lor c) \land c\]
\[(-c \lor a) \land b \land (-c \lor -a \lor -b) \land (c \lor a)\]
\[(a \lor -b) \land (b \lor -a) \land (b \lor -c) \land (-b \lor c)\]

(a) Use DPLL algorithm to check their satisfiability and to count the number of solutions/models for each formula.

(b) Construct the reduced ordered BDD for each formula for the variable order \(a < b < c\). Count the number of solutions/models for each formula using the paths of the corresponding BDD.

3. Consider the following linear arithmetic formulas on integer variables:

\[(x \leq 50) \land (x > 20 \lor x = 40) \land (x \leq 50 \lor x > 20)\]
\[(x \leq 50 \lor y \leq 20 \lor x = y) \land (y > 20 \lor x = 60) \land (y > 20 \lor x = y)\]

Show the Boolean skeleton of these formulas and find all the satisfying solutions for the Boolean skeletons. Assuming that each integer variable takes a value between 1 and 100, calculate the number of solutions/models for these formulas. Assuming that you have a model counting constraint solver for conjunctions of linear arithmetic constraints, explain how to minimize the calls to such a model counting constraint solver in calculating the number of models.

4. Consider the following constraints:

\[(x \in a(a|b)^*) \land (\text{len}(x) \leq 3)\]
\[(x \in (a|b)^*b) \land (\text{len}(x) \geq 3)\]

Construct DFAs accepting solutions to these constraints and show how the model counting function can be computed 1) by using matrix exponentiation and 2) by defining a recurrence relation (show the matrix and the recurrence relation). Compute the counting function for string lengths 0, 1, 2, 3, 4, 5.

5.

For the second formula in each of the problems 2, 3 and 4 above, write the formulas in the SML-LIB format and use Z3 to check their satisfiability and to generate a model for each formula. Turn in the SMT-LIB specification for the formula and the Z3 output. Online version of Z3 is available here:

https://rise4fun.com/Z3