292 – Homework Assignment 2

Fall 2021

Due: Tuesday, November 2nd

Do not discuss the problems with anyone other than the instructor.

1.

(a) Consider the following function:

```
int test1(int x, int y) {
  int z = 0;
  if (x > y )
    z = x;
  else
    z = y;

  if (z > 0)
    return 1;
  else
    return 0;
}
```

Show how classic symbolic execution (without concrete values) starting from the symbolic state $x = m, y = n$ would compute the path conditions for this function. Show the symbolic state and path conditions.

(b) Consider the following function:

```
int test2(int x, int y) {
  int z;
  if (x > 100) {
    if (x + y = 20)
      z = 1;
  } else
    z = 0;

  if (z = 1) {
    if (y > 0)
      z = 2;
    else if (-y > 200)
      z = 3;
  }

  return z;
}
```

Show how concolic execution starting from the concrete state $x = 5, y = 8$ would compute the path conditions for this function. Show the concrete state, symbolic state and path conditions.

2. Consider the following functions:

```
int test3(int x, int y) {
  int result = 0;
  if (x < 15 && y < 15)
    result = 1;
  else {
    if (y == 10 && x == 5)
      result = 2;
  }
  if (y > 20)
    result = 3;
  else {
    if (x == 40)
      result = 4;
  }
  return result;
}

int test4(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 two functions (using classic symbolic execution) assuming each input value is symbolic (show the symbolic state and the path conditions). Show the resulting path conditions 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 condition and determine the probability of each execution path.

3. Consider the following boolean logic formulas:

$(a \vee b) \wedge (b \vee c) \wedge c$

$(\neg c \vee a) \wedge b \wedge (\neg c \vee \neg a \vee \neg b) \wedge (c \vee a)$

$(a \vee \neg b) \wedge (b \vee \neg a) \wedge (b \vee \neg c) \wedge (\neg b \vee c)$

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

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

(a) $(x \leq 40) \wedge (x > 20 \vee x = 30) \wedge (x \leq 40 \vee x > 20)$

(b) $(x \leq 60 \vee y \leq 30 \vee x = y) \wedge (y > 30 \vee x = 80) \wedge (y > 30 \vee x = y)$

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 we have a model counting constraint solver for conjunctions of linear arithmetic constraints, and using the approach we discussed in Lecture 9, what is the minimum number of calls needed to the model counting constraint solver in calculating the number of models for the formula (a) and formula (b).

5.

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

`https://compsys-tools.ens-lyon.fr/z3/`