---

1. Consider the following formula φ^{UF} in equality logic with uninterpreted functions (EUF):

$$\varphi^{UF} := x_1 = x_2 \vee (F(x_1) \neq F(x_2) \wedge false)$$

   (a) (2 points) Derive an *equisatisfiable* formula φ^A in equality logic using Ackermann's reduction and construct the equality graph for it.

   (b) (2 points) Derive an *equisatisfiable* formula φ^B in equality logic using Bryant's reduction and construct the equality graph for it.

   (c) (1 point) Now perform range allocation for φ^B. State the range you obtained (you need not show the steps of your derivation). Is your range adequate for φ^A?

   In the equality graphs, use dashed and solid edges to denote equality and disequality edges respectively. Note that the reductions studied in the class give us an equality logic formula that is *valid* iff the corresponding EUF formula is valid. You are required to obtain equisatisfiable formulae here.

2. Consider the following pseudo-code derived from merge sort:

```
1  assume(right >= left+1);
2  int mid = (left+right)/2;
3  for (int i = left; i <= right; i++)
4     temp[i] = array[i];
5  int i1 = left; int i2 = mid+1;
6  int curr = left;
7  while(i1 <= mid && i2 <= right) {
8  if ( ... )
9     array[curr++] = temp[i1++];
10 else
11    array[curr++] = temp[i2++];
12 ...
13 }
```

   We want to check whether the accesses to array `temp` on line 9 refer to array locations that are initialized in line 4. Otherwise, the program will have an "array out-of-bounds access" error.

   (a) (2 points) Derive one or more integer linear arithmetic formulae (as necessary) whose unsatisfiability implies that the above requirement holds.

   (b) (3 points) Check for satisfiability of the formulae obtained in the previous step using Fourier-Motzkin variable elimination. Show all steps clearly. Is there any access to an array location that was not initialized earlier? If yes, describe the access.