Homework Assignment #5

Due Time/Date

11:59PM Tuesday, June 6, 2023. Late submission will be penalized by 20% for each working day overdue.

How to Submit

Please use a word processor or scan hand-written answers to produce a single PDF file. Name your file according to this pattern: "r117250xx-hw5". Upload the PDF file to the NTU COOL course site for Software Development Methods 2023. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

This assignment contains several exercise problems for you to practice writing formal statements in first-order logic. We assume the binding powers of the logical connectives decrease in this order: $\neg$, $\{\forall, \exists\}$, $\{\wedge, \vee\}$, $\rightarrow$, $\leftrightarrow$ (so that you may avoid using some parentheses).

1. (20 points) Consider the set of natural numbers (including 0) with addition $(\mathbb{N}, \{+\})$ and the set of integers with addition $(\mathbb{Z}, \{+\})$. Give a first-order sentence that is true in one but false in the other. (Note: two structures are said to be *elementarily equivalent* if they satisfy the same set of first-order sentences. So, the sentence you would give shows that $(\mathbb{N}, \{+\})$ and $(\mathbb{Z}, \{+\})$ are not elementarily equivalent.)

2. (40 %) The following C function `originalEuclid` implements the original Euclidean algorithm. Please give a suitable function contract, namely the pre and post-conditions, for `originalEuclid`, using either ACSL or the conventional logic notation. Let us assume that, for this problem, the type `int` is the same as the set $\mathbb{Z}$ of integers. And, the formulae you will write are intended for the semantic structure $\mathcal{Z} = (\mathbb{Z}, \{+, -, \times, 0, 1, <\})$, i.e., the set of integers with the usual arithmetic functions ($+$, $-$, and $\times$), constants (0 and 1), and predicates ($<$ and $\leq$); "$=$" is implicitly assumed to be a binary predicate. You certainly would bring up the notion of "a divides b", perhaps in the form of a predicate $\mathit{divides}(a, b)$, or alternatively $a \mid b$, but this is not directly available in the assumed language and you would need to spell out the defining formula..

```
int originalEuclid (int m, int n)
{ int x,y,tmp;

  x = m;
  y = n;
  while (x!=y) {
    if (x < y) {
      tmp = x;
      x = y;
      y = tmp;
    }
```

```
    x = x - y;
  }
  return x;
}
```

3. (40 %) Please examine the following C function `findxy` carefully and give a suitable function contract, namely the pre and post-conditions, for `findxy`, using either ACSL or the conventional logic notation. You may omit the condition concerning proper memory allocation for the input array and focus on what the function does. Again, let us assume that, for this problem, the type `int` is the same as the set $\mathbb{Z}$ of integers. However, the formulae you will write are intended for a simpler semantic structure $\mathcal{Z} = (\mathbb{Z}, \{+, -, 0, 1, <\})$; "=" is implicitly assumed to be a binary predicate.

```
int findxy (int* a, int n, int x, int y)
{ int m, i, j;

  m = -1;
  for (i=0; i<n; i++)
    if (a[i] == x) {
      m = i;
      break;
    }
  for (j=m+1; j<n; j++)
    if (a[j] == y) {
        m = j;
        break;
    }
  return m;
}
```