Assignment 2: Safety
15-316 Software Foundations of Security and Privacy

1. **Arbitrary assignment (20 points).** Sometimes when modeling a computation, we need to avoid making assumptions about what exactly might transpire at runtime. For example, suppose that we wish to write a program that interacts with a network connection, and we need to ensure that after reading some bytes from the network, the program continues to behave safely. Because there is no way of knowing which bytes will arrive in advance, we must assume that they could be arbitrary when we do our safety analysis. Nondeterminism is the appropriate way to handle cases like this.

   Part 1 (5 points). Extend the language discussed in lecture by defining the semantics of a nondeterministic assignment command, $x := *$. Informally, this command should modify the state by assigning to x an arbitrary Integer value.

   Solution.

   Part 2 (5 points). Design an axiom that allows you to reason about box modalities around nondeterministic assignment:

$$[x := *]p(x) \leftrightarrow \dots$$

   The right side of this equivalence should not contain a box or diamond modality, but only first-order formulas. You do not need to prove that your axiom is sound, but explain informally why you believe that it is.

   Solution.

Part 3 (10 points). Suppose that two programs α and β are identical in every way, except that all of the assignments in β are nondeterministic, and all of the assignments in α are deterministic. For example, the following programs would match this description:

$$\begin{aligned} \alpha \equiv \quad & x := 1; \texttt{if}(y < 0)\, z := y\, \texttt{else}\, z := -y \\ \beta \equiv \quad & x := *; \texttt{if}(y < 0)\, z := *\, \texttt{else}\, z := * \end{aligned}$$

If β satisfies a given safety property Φ, then will α necessarily satisfy it as well? Likewise, if α satisfies Φ, then must β? For both questions, if you believe that both will satisfy Φ, then use your semantics from Part 1 to argue the case. Otherwise, provide a counterexample set of α, β, and Φ where only one of α, β satisfy Φ.

Hint: Depending on your answer to this question, either $[\alpha]P \rightarrow [\beta]P$, $[\beta]P \rightarrow [\alpha]P$, or both, for any formula P. Try using your axiom from Part 2 on some examples to see which of these might possibly be valid.

Solution.

Verification conditions (15 points). Recall that a verification condition for a program path $\alpha_1; \ldots; \alpha_n$ is a formula of arithmetic whose validity implies the validity of the DL formula $[\alpha_1; \ldots; \alpha_n]P$, for some P. Consider the following program that we will denote α:

```
i := 0;
while(i < k) {
  if(i < 0) { i := k; }
  if(0 <= Mem(i)) { x := x + Mem(i); }
  i := i + 1;
}
```

In this question, assume that we always use the [unfold] axiom to reason about loops.

Part 1 (5 points). Assume that k is a constant that we fix in advance. How many verification conditions are required to check $[\alpha]0 \leqslant x$, as a function of k? Explain the rationale behind your answer.

Solution.

Part 2 (10 points). List the verification conditions for $k = 1$ and the postcondition $0 \leqslant x$ from the previous problem. Note that it is not necessary to conduct a sequent calculus proof for this problem, and you will receive full credit for listing out the paths and their corresponding verification conditions. It is also not necessary to say whether the VCs are valid or not.

Solution.