Sample Solution for Homework 2

Problem 1 AMP, p.41, Exercise 11: Flaky Lock (12 Points)

- The protocol satisfies mutual exclusion. For a proof by contradiction, suppose it did not. By inspecting the protocol code, we conclude that the following happens-before relationships must hold:

  $$\text{write}_A(\texttt{turn} = A) \rightarrow \text{write}_A(\texttt{turn} = B) \rightarrow CS_A$$

  and

  $$\text{write}_B(\texttt{turn} = B) \rightarrow \text{write}_A(\texttt{turn} = A) \rightarrow CS_B$$

  Assume without loss of generality that thread A was the last thread to write to `turn` before entering the critical section. Then

  $$\text{write}_B(\texttt{turn} = B) \rightarrow \text{write}_A(\texttt{turn} = A)$$

  which contradicts the fact that A completed its outer waiting loop.

- As we show below, the algorithm is not deadlock-free. Hence, it is also not starvation-free.

- The following sequence of events leads to a deadlock:

  $$\text{write}_A(\texttt{turn} = A) \rightarrow \text{read}_A(\texttt{busy} = \texttt{false}) \rightarrow \text{write}_B(\texttt{turn} = B)$$

Problem 2 AMP, p.41, Exercise 14: ℓ-Exclusion (13 Points)

We can turn the filter algorithm into an algorithm that solves the ℓ-exclusion problem by reducing the number of levels by $\ell - 1$ (assuming $n > \ell$).

```
class LFilter implements Lock {
  int[] level;
  int[] victim;

  public LFilter(int n, int l) {
    level = new int[max(n-l+1,0)];
    victim = new int[max(n-l+1,0)];
    for (int i = 0; i < n-l+1; i++) {
      level[i] = 0;
    }
  }

  public void lock() {
    int me = ThreadID.get();
    for (int i = 1; i < n-l+1; i++) { // attempt level i
      level[me] = i;
      victim[i] = me;
      // spin while conflicts exist
      int above = l+1;
      while (above > l && victim[i] == me) {
        above = 0;
        for (int k = 0; k < n; k++) {
          if (level[k] >= i) above++;
        }
      }
    }
  }

  public void unlock() {
    int me = ThreadID.get();
    level[me] = 0;
  }
}
```