SoCV HW5

R04943179 謝佳霖

(i)   Implementation

```
itpUbmc(monitor){
    I <- buildInitStates();
    addBoundedVerifyData(I, 0);
    addBoundedVerifyData(monitor, 0);
    for each newly created clause c
        markOnsetClause(c);
    assumRelease();
    assumProperty(I, false, 0);
    assumProperty(monitor, false, 0);
    if(SAT) {                      // Cex found in timeframe 0
        setFired(0);
        return;
    }
    assertProperty(monitor, true, 0);
    for each Latch l in ntk {
        addBoundedVerifyData(l, 1);
        mapVar2Net(getVerifyData(l, 1), l);
    }
    for each newly created clause c
        markOnsetClause(c);
    for( i = 1 to MaxDepth ) {
        addBoundedVerifyData(monitor, i);
        for each newly create clause c
            markOffsetClause(c);
        assumeRelease();
        assumeProperty(I, false, 0);
        assumeProperty(monitor, false, i);
        if(SAT) {                  // Cex found in timeframe i
            setFired(i);
            return;
        }
```

```
        currState <- I;
            while(1) {
                itp <- getItp();
                orState <- itp | currState;
                fixed <- orState & ~currState;
                addBoundedVerifyData(fixed, 0);
                for each newly created clause c
                    markOnsetClause(c);
                assumeRelease();
                assumeProperty(fixed, false, 0);
                if(UNSAT) {                    // fixed point reached in timeframe i
                    setProved(i);
                    return;
                }
                assumeRelease();
                assumeProperty(itp, false, 0);
                assumeProperty(monitor, false, i);
                if(SAT) break;                 // Spurious Cex found, go to next timeframe
                currState <- orState;
            }
            assertProperty(monitor, i)
            for each newly create clause c
                markOffsetClause(c);
        }
    }
```

(ii)   Verification results

a: 0.04s, all safe

b: 5.16s, p1/p2 safe, p3 violates at time 42

c: 0s, z1 violates at time 5, z2 violates at time 6, z3 safe

sat.dofile: 138s

unsat.dofile: 130.9s

vending_abs: 0.25s, all safe

vending_fixed: 跑不出來

(iii)   Comparison with ref program

a: 0.05s

b: 4.72s

c: 0s

sat.dofile: 180s

unsat.dofile: 92.5s

vending_abs: 0.44s

vending_fixed: 跑不出來

We can see that both my and ref programs are capable of almost every cases, and the runtime between these programs are similar, mine are faster in some cases, and slower in others.