Assignment 4 Q 4

```
LIFO Stack Stk = Φ ; /* DF consumption Stack */
HashTable T = Φ ; /* for visited states */
HashTable T2 = Φ ; /* for discovered states */
/* Returns true iff Φ holds in all the reachable states */
bool DFS (NFSS, S, SafetyProperty Φ){
let S = (S, I, A, next );
/* is there an initial state which is an error state? */
 foreach s in I {
     if (! IfNotVisitedCheckPushStack (s, Φ))
        /* s is an error state and S does not satisfy Φ */
        return false ;
 }
 while ( Stk ≠ Φ) { /* main DFS */
    s = pop (Stk);
    if (! IsInDiscoverHashTable (s, T2){
              HashInsert (T2, s)
    foreach (s-next, a) in next (s) {/* s is
                 HashInsert (T2, s-next)
                 ~~s = s-next;~~
    foreach  item  in T2 {
            if (! IfNotVisitedCheckPushStack(item, Φ))
                              return false;
            HashRemove (T2, item);
 } /* while */
 return true; /* error not found, S satisfies Φ */
}/* DFS */
/* Returns false if s is an error states (i.e. does not
            satisfy Φ ), true otherwise */
bool IfNotVisitedCheckPushStack (s , SafetyProperty Φ){
   if (s is not in T) {
        if (!Φ(s))) return False;
        HashInsert (T, s)
        Push (stk, s);
    }
    return true;
  } /* If NotVisitedCheckPushStack() */

bool IsInDiscoverHashTable(s, Hash Table T2){
            if(Hash Search (T2, s))
                return true;
            return false;
  }.
```