CS 2800 Homework 8

Assumptions

Throughout this assigment, you may assume the following definitions:

```
;; app : tlp x tlp -> tlp
;; Append two lists
(defun app (x y)
  (if (endp x)
    y
    (cons (car x) (app (cdr x) y))))

;; rev : tlp -> tlp
;; Reverse a list
(defun rev (x)
  (if (endp x)
      nil
    (app (rev (cdr x)) (list (car x)))))

;; in : All x tlp -> Bool
;; Is e an element of l?
(defun in (e l)
  (if (endp l)
      nil
    (or (equal e (car l))
        (in e (cdr l)))))

;; rem-el : All x tlp -> tlp
;; rem-elete all occurrences of e from l.
(defun rem-el (e l)
  (if (endp l)
      nil
    (if (equal e (car l))
        (rem-el e (cdr l))
      (cons (car l) (rem-el e (cdr l))))))

;; =<: tlp x tlp -> Bool
;; Is X a subset of Y?
(defun =< (X Y)
  (if (endp X)
    t
```

```
    (and (in (car X) Y)
         (=< (cdr X) Y))))

;; diff : tlp x tlp -> tlp
;; remove all elements of y from x.
(defun diff (x y)
  (if (endp y)
      x
    (rem-el (car y) (diff x (cdr y)))))

;; repl : All x All x tlp -> tlp
;; Replace all occurrences of x in l with y.
(defun repl (x y l)
  (if (endp l)
    nil
    (if (= x (car l))
      (cons y (repl x y (cdr l)))
      (cons (car l) (repl x y (cdr l))))))

;; fact : nat -> nat
;; Compute factorial
(defun fact (n)
  (if (zp n)
      1
    (* n (fact (- n 1)))))

;; fact*-acc : nat x nat -> nat
;; Helper function for fact*
(defun fact*-acc (x acc)
  (if (zp x)
      acc
    (fact*-acc (- x 1) (* x acc))))

;; fact* : nat -> nat
;; Compute factorial (tail recursive)
(defun fact* (x)
  (fact*-acc x 1))

;; booleanp : All -> Bool
(defun booleanp (x)
  (if (equal x t)
    t
```

```
(equal x nil)))
```

Problems

Prove the following theorems. If you use induction, clearly indicate what functions were used to generate the induction schemes.

1.
```
(booleanp (=< X Y))
```

2.
```
(true-listp (repl x y l))
```

3.
```
(natp (fact n))
```

4.
```
(equal (in e (rev X))
       (in e X))
```

5.
```
(not (in e (rem-el e X)))
```

6.
```
(implies (in e X)
         (in d (repl e d X)))
```