A Sample Proof on Programs
COMP 280, Spring 2000

Consider the following representation for sets and the following two programs which implement subset and intersection on sets (no duplicates).

A set is either
- *empty*
- (*cons* E S) where E is an element and S is a set

```
;; subset : set set → boolean
;; returns true iff the first set is a subset of the second set
(define (subset check-set in-set)
  (cond [(empty? check-set) true]
        [else (and (member (first check-set) in-set)
                   (subset (rest check-set) in-set))]))

;; intersect : set set → set
;; returns a set containing all elements that are in both sets
(define (intersect set1 set2)
  (cond [(empty? set1) empty]
        [else (if (member (first set1) set2)
                  (cons (first set1) (intersect (rest set1) set2))
                  (intersect (rest set1) set2))]))
```

Prove for all sets *S1* and *S2* that

```
(if (subset S1 S2)
    (intersect S1 S2)   ≡   S1
    S1)
```

Proof:
Base: Let *S1* be *empty*. We must prove

```
(if (subset empty S2)
    (intersect empty S2)   ≡   empty
    empty)
```

```
    (if (subset empty S2)
        (intersect empty S2)
        empty)
=
    (if true
        (intersect empty S2)
        empty)
=
    (intersect empty S2)
=
    empty
```

Inductive: Let *S1* be some set A and assume that

$$\begin{array}{l}(\textbf{if}\ (\mathit{subset}\ A\ S2)\\ \quad(\mathit{intersect}\ A\ S2)\\ \quad A)\end{array} \quad\equiv\quad A$$

We must prove that the desired result holds for the set $(\mathit{cons}\ E\ A)$, *i.e.*:

$$\begin{array}{l}(\textbf{if}\ (\mathit{subset}\ (\mathit{cons}\ E\ A)\ S2)\\ \quad(\mathit{intersect}\ (\mathit{cons}\ E\ A)\ S2)\\ \quad(\mathit{cons}\ E\ A))\end{array} \quad\equiv\quad (\mathit{cons}\ E\ A)$$

(**if** (*subset* (*cons* *E* *A*) *S2*)
  (*intersect* (*cons* *E* *A*) *S2*)
  (*cons* *E* *A*))
=
(**if** (**and** (*member* (*first* (*cons* *E* *A*)) *S2*)
    (*subset* (*rest* (*cons* *E* *A*)) *S2*))
  (*intersect* (*cons* *E* *A*) *S2*)
  (*cons* *E* *A*))

Case 1: Assume that (*member* *E* *S2*) is true

(**if** (**and** (*member* (*first* (*cons* *E* *A*)) *S2*)
    (*subset* (*rest* (*cons* *E* *A*)) *S2*))
  (*intersect* (*cons* *E* *A*) *S2*)
  (*cons* *E* *A*))
=
(**if** (*subset* (*rest* (*cons* *E* *A*)) *S2*)
  (*intersect* (*cons* *E* *A*) *S2*)
  (*cons* *E* *A*))
=
(**if** (*subset* *A* *S2*)
  (**if** (*member* (*first* (*cons* *E* *A*)) *S2*)
    (*cons* (*first* (*cons* *E* *A*)) (*intersect* (*rest* (*cons* *E* *A*)) *S2*))
    (*intersect* (*rest* (*cons* *E* *A*) *S2*))
  (*cons* *E* *A*))
=
(**if** (*subset* *A* *S2*)
  (*cons* (*first* (*cons* *E* *A*)) (*intersect* (*rest* (*cons* *E* *A*)) *S2*))
  (*cons* *E* *A*))
=
(**if** (*subset* *A* *S2*)
  (*cons* *E* (*intersect* *A* *S2*))
  (*cons* *E* *A*))
=
(*cons* *E* (**if** (*subset* *A* *S2*)
    (*intersect* *A* *S2*)
    A))
=
(*cons* *E* *A*)

Case 2 : Assume that (*member E in-set*) is false

(**if** (**and** (*member* (*first* (*cons E A*)) *S2*)
(*subset* (*rest* (*cons E A*)) *S2*))
(*intersect* (*cons E A*) *S2*)
(*cons E A*))

=

(**if** (**and** *false*
(*subset* (*rest* (*cons E A*)) *S2*))
(*intersect* (*cons E A*) *S2*)
(*cons E A*))

=

(**if** *false*
(*intersect* (*cons E A*) *S2*)
(*cons E A*))

=

(*cons E A*)