Software Verification

- One of the major problems of software engineering is verifying that a program is correct.

- One approach is to test the program on many different inputs.

- However, subtle bugs may remain undiscovered, only to appear at random, inconvenient, or dangerous moments.

- Another approach is to <u>prove</u> that a program is correct for <u>all</u> inputs. This is especially useful for safety-critical software (eg. Air-traffic control systems).

Proving Properties of Programs

Example 1

```
(define (append X Y)
  (if (null? X) Y
      (cons (car X)
            (append (cdr X) L))))
```

```
(define (length X)
  (if (null? X) 0
      (+ 1 (length (cdr X)))))
```

Prove the following:

Theorem: (length (append X Y))

= (length X) + (length Y)

for <u>all</u> lists X, Y.

Proof Outline

- Use mathematical induction on the length of X.

- ie, First, prove that the theorem is true for lists of length 0. Basis

  Then, prove that if the theorem is true for lists of length N, then it is also true for lists of length N+1. Inductive Step.

This implies that the theorem is true for lists of any length (ie, for any list).

Structural Induction

- Actually, we will use a variation of induction that emphasizes the structure of lists not their length.

- ie, First, prove that the theorem is true for X = '() (ie, when X has length 0) — Basis

  Then, prove that if the theorem is true for X = L, then it is also true for X = (cons E L) — Inductive Step

  (note: if L has length N, then (cons E L) has length N+1)

Preliminaries

- Before using induction (or any other technique) to prove a complex property of a program, write down the basic properties that can be trivially verified by inspecting the program code.

- The inductive proof should only use these properties of the code.

- ie, the code itself plays no other role in a proof of correctness, and can be henceforth ignored.

- <u>Note</u>: If the basic properties are wrong then the entire proof is wrong. <u>So be sure to get them right!!</u>

Basic Properties of Append

```
(define (append X Y)
    (if (null? X) Y
        (cons (car X)
              (append (cdr X) Y))))
```

Using X = '()

(append '() Y) = Y ①

Using X = (cons E L)

(append (cons E L) Y) =
(cons E (append L Y)) ②

Note: If X = (cons E L)

then (car X) = E, (cdr X) = L.

Basic Properties of Length

```
(define (length X)
        (if (null? X) 0
            (+ 1 (length (cdr X)))))
```

Using X = '()

(length '()) = 0 ③

Using X = (cons E L)

(length (cons E L)) = 1 + (length L) ④

Note: If X = (cons E L)

then (cdr X) = L

Summary of Basic Properties

① (append '() Y) = Y

② (append (cons E L) Y) =
(cons E (append L Y))

③ (length '()) = 0

④ (length (cons E L)) = 1 + (length L)

Using these basic properties, we shall prove (by induction) a more complex property:

<u>Theorem</u>: (length (append X Y)) =
(length X) + (length Y)

Proof

(by induction on the structure of X)

Basis: when X = '()

(length (append X Y))

= (length (append '() Y))

= (length Y) by ①

= 0 + (length Y)

= (length '()) + (length Y) by ③

= (length X) + (length Y)

∴ The theorem is true when X = '().

Inductive Step:

Suppose the theorem holds for $X = L$

ie, suppose that

$$\left.\begin{array}{l} \texttt{(length (append L Y))} \\ = \texttt{(length L)} + \texttt{(length Y)} \end{array}\right\} \text{Inductive Hypothesis}$$

Now, prove that the theorem holds for $X = \texttt{(cons E L)}$.

Proof of Inductive Step

If X = (cons E L) then

(length (append X Y))

= (length (append (cons E L) Y))

= (length (cons E (append L Y))) by 2

= 1 + (length (append L Y)) by 4

= 1 + (length L) + (length Y)

by inductive hypothesis

= (length (cons E L)) + (length Y) by 4

= (length X) + (length Y)

QED

Summary of Proof of Theorem

For any list, Y,

Basis: The thm. holds for X = '()

Inductive Step:

If the thm. holds for X = L,
then it holds for X = (cons E L).

∴ By the principle of structural induction,
the thm holds for any lists X, Y.

Theorem: (length (append X Y))
= (length X) + (length Y)

Proving Properties of Programs

Example 2

```
(define (member A X)
    (cond ((null? X) #f)
          ((equal? A (car X)) #t)
          (else (member A (cdr X)))))
```

Prove the following:

Theorem: If (member A X)

Then (member A (append X Y))

For *any* A, and *any* lists X, Y.

Outline of Proof

- As before, we will use structural induction on X.

- ie, First, prove that the theorem is true for X = `()] Basis

  Then, prove that <u>if</u> the theorem is true for X=L, <u>then</u> it is true for X= (cons E L)] Inductive Step

- However, this time the proof will be more complex, because the program can terminate its recursion in <u>two</u> ways.

Basic Properties of Member

```
(define (member A X)
  (cond ((null? X) #f)
        ((equal? A (car X)) #t)
        (else (member A (cdr X)))))
```

---

using X = `()

(member A `()) = #f ⑤

---

using X = (cons A L)

(car X) = A

∴ (member A (cons A L)) = #t ⑥

Basic Properties of Member

```
(define (member A X)
  (cond ((null? X) #f)
        ((equal? A (car X)) #t)
        (else (member A (cdr X)))))
```

---

using $X = $ (cons E L) where $E \neq A$

(car X) = E

(cdr X) = L

$\therefore$ (member A (cons E L))

= (member A L)

7

Summary of Basic Properties

member:

⑤ (member A '()) = #f

⑥ (member A (cons A L)) = #t

⑦ If $E \neq A$ then
(member A (cons E L))
= (member A L)

append:

① (append '() Y) = Y

② (append (cons E L) Y) =
(cons E (append L Y))

Proof of Theorem

Theorem: If (member A X)
then (member A (append X Y))

Proof: (by induction on X)

Basis: If X = \`() then

(member A X)
= (member A \`())
= #f by ⑤

∴ The premise of the thm is false.
∴ The thm. is trivially true (for X = \`()).

Inductive Step:

Suppose the theorem is true for X=L.

ie, IF (member A L)
then (member A (append L Y)) $\Big\}$ Inductive Hypothesis

Now, prove the theorem is true
for X = (cons E L).

Proof of Inductive Step

Let $X = $ (cons E L)

There are two cases.

Case 1: $E = A$

$\therefore$ $X = $ (cons A L)

$\therefore$ (member A (append X Y))

= (member A (append (cons A L) Y))

= (member A (cons A (append L Y))) by ②

= #t by ⑥

$\therefore$ The conclusion of the thm. is true.

$\therefore$ The thm. itself is trivially true,

for $X = $ (cons A L).

Case 2: X = (cons E L) where $E \neq A$.

If (member A X)

then (member A (cons E L))

by ⑦

∴ (member A L)

∴ (member A (append L Y))

by induction hypothesis

∴ (member A (cons E (append L Y))) by ⑦

∴ (member A (append (cons E L) Y)) by ②

∴ (member A (append X Y))

∴ The thm. is true when

X = (cons E L) and $E \neq A$.

QED

Summary of Proof of Theorem

For *any* A, and *any* list Y,

Basis: The thm. holds for X = '()

Inductive Step:

If the thm. holds for X = L,
then it holds for X = (cons E L).

∴ By the principle of structural induction, the thm. holds for *any* A, and *any* lists X, Y.

Theorem: If (member A X)
Then (member A (append X Y))