1.

Base case: For any set A, '( ) $\in$ LIST(A)

So we can get '( ) $\in$ LIST(SEXP)

Because ATOM = BOOL $\cup$ NUM $\cup$ SYM $\cup$ {'( )}, '( ) $\in$ ATOM

Because ATOM $\in$ SEXP, '( ) $\in$ SEXP as well.

we know for base case '( ), LIST(SEXP) $\in$ SEXP

Induction step:

Induction hypothesis: For any a list xs, xs $\in$ LIST(SEXP), xs $\in$ SEXP, LIST(SEXP) $\in$ SEXP

we assume x is a element from BOOL or NUM or SYM or {'( )}

Because $x \in$ ATOM, ATOM $\in$ SEXP, we can get $x \in$ SEXP

From induction hypothesis, we know xs $\in$ SEXP as well,

So Based on the general S-expression: SEXP = ATOM $\cup$ {(cons v_1 v_2) | $v_1 \in$ SEXP $\wedge$ $v_2 \in$ SEXP}, (cons x xs) $\in$ SEXP because we have (x $\in$ SEXP) $\wedge$ (xs $\in$ SEXP)

Because x $\in$ ATOM, ATOM $\in$ SEXP, we can get x $\in$ SEXP,

From induction hypothesis, we know xs $\in$ LIST(SEXP),

so Based on the equation: LIST(A) = {'( )} $\cup$ {(cons a as) | a $\in$ A $\wedge$ as $\in$ LIST(A)}, we can get (cons x xs) $\in$ LIST(SEXP) because we have (x $\in$ SEXP $\wedge$ xs $\in$ LIST(SEXP)).

Finally, we prove LIST(SEXP) $\in$ SEXP

35.

Base case: xs = '()

```
(length (reverse xs))
={substitute actual parameters in definition of reverse}
(length (revapp xs '()))
={substitute actual parameters in definition of revapp}
(length (if (null? xs)
            '()
            (revapp (cdr xs) ((cons (car xs) ys))))
= {substitute defition of xs}
(length (if (null? '())
            '()
            (revapp (cdr xs) (cons (car xs) ys))))
= {null? -empty law}
(length (if #t
            '()
            (revapp (cdr xs) (cons (car xs) ys))))
= {if-#t law}
(length '())
∴ (length (reverse '()))) = (length xs)
```

Induction step:

Induction hypothesis (length (reverse zs)) = (length zs)

we assume xs is not nill, xs = (cons z zs)

```
(length (reverse xs)) = (length xs)
= {substitute actual parameters in definition of reverse}
(length (revapp xs '()))
= {substitute actual parameters in the definition of revapp}
(length (if (null! xs)
            '()
            (revapp (cdr xs) (cons (car xs) '()))))
= {by assumption that xs is not nil, xs = ((cons z zs)}
```

```
(length (if (null? (cons z zs))
            '()
            (revapp (cdr (cons z zs)) (cons (car (cons z zs)) '()))))
= {null?-cons law}
(length (if #f
            '()
            (revapp (cdr (cons z zs)) (cons (car (cons z zs)) '()))))
= {if-#f law}
(length (revapp (cdr (cons z zs)) (cons (car (cons z zs)) '()))))
= {cdr-cons law and car-cons law}
(length (revapp zs (cons z '())))
= {(revapp xs ys) = (append (reverse xs) ys) law}
(length (append (reverse zs) (cons z '())))
= {(length (append xs ys)) = (+ (length xs) (length ys)) law}
(+ (length (reverse zs)) (length (cons z '())))
= {induction hypothesis}
(+ (length zs) (length (cons z '())))
= {(length (cons x xs)) = (+1 (length xs)) law}
(+ (length zs) (+1 length '()))
= {length-null? law}
(+ (length zs) (+1 0))
= {zero is the addtive identity}
(+ (length zs) 1)
= {(length (cons x xs)) = (+1 (length xs)) law}
= (length (cons z zs))
= {assume xs = (cons z zs)
= (length xs)
```

Finally, we prove (length (reverse xs)) = (length xs)

2(a)

$$\langle CONS, \xi, \phi, \rho \rangle \Downarrow \langle PRIMITIVE(cons), \xi, \phi, \rho \rangle$$

$$\frac{x \in dom\,\rho \quad \rho(x) = x}{\langle VAR(x), \xi, \phi, \rho \rangle \Downarrow \langle x, \xi, \phi, \rho \rangle} \qquad \frac{xs \in dom\,\rho, \quad \rho(xs) = xs}{\langle VAR(xs), \xi, \phi, \rho \rangle \Downarrow \langle xs, \xi, \phi, \rho \rangle}$$

$$\frac{}{\langle APPLY(CONS, VAR(x), VAR(XS)), \xi, \phi, \rho \rangle \Downarrow \langle CONS\langle X, XS \rangle, \xi, \phi, \rho \rangle}$$

$$\frac{\langle CDR, \xi, \phi, \rho \rangle \Downarrow \langle PRIMITIVE(cdr), \xi, \phi, \rho \rangle}{\langle APPLY(CDR, (CONS, (VAR(X)) (VAR(XS))))), \xi, \phi, \rho \rangle \Downarrow \langle XS, \xi, \phi, \rho \rangle}$$

$\therefore$ so we have cdr (cons x xs) = XS

1(b)

we assume : x = '(b c)
e₁ = set x '(a)
e₂ = x

(cdr (cons e₁ e₂))
= (cdr (cons (set x '(a)) e₂))
First evaluating e₁, x become '(a) so e₂ = x = '(a)
= (cdr (cons '(a) '(a)))
= '(a)
But we know e₂ = '(b c)

$\therefore$ (cdr (cons e₁ e₂)) ≠ e₂

so we disprove the following conjecture.