Yinglai Wang

30. (length (reverse xs)) = (length xs)

Proof: Basic step: Assume xs is null.

(length (if (null? '()) '()
(append (reverse (cdr '())) (list1 (car '())))
)))

= {null-empty law}

(length (if (#t '()) (append (reverse (cdr '())) (list (car '()))))))

= {if-#t law}

(length '()) == (length xs) when xs is null.

Inductive step: Assume xs = (cons y ys)

(length (reverse (cons y ys)))

= (length (if (null? (cons y ys)) '() (append (reverse ys) (list1 y)))))

= {if-#f law}

(length (append (reverse ys) (list1 z)))

= {length-append law}

(+ (length (reverse ys)) (length (list1 z)))

= {length-list1}

(+ 1 (length (reverse ys)))

= {induction hypothesis}

(+ 1 (length ys)) = {cons-length law}

(length (cons y ys)) == (length xs)

37. (a)

$$\frac{\langle e, \rho, \sigma \rangle \Downarrow \langle v, \rho, \sigma' \rangle}{\langle VAL(x,e), \rho, \sigma \rangle \Downarrow \langle v, \rho'\{x \to l(x')\}, \sigma'\{l(x') \to v\} \rangle}$$

(b)

```
(val x 5)
(define addone (y) (+ x y))
 (val x 10)
 (addone 1)
```

If addone returns 11, then (val x 10) used SET to overwrite x.

If it returns 6, then (val x 10) create a new binding.

(c) I prefer the one that overwrite the old value.

It is confusing that have multiple bindings of x than a single binding that is updated.

The new semantics also cause memory leak, old location that x mapped to will lost when new binding happens.

It would be easier to use a new variable.

3.(a) Prove (cdr (cons x xs)) = xs, set (cons x xs) as A.

$$\frac{\langle e, \rho, \sigma_0 \rangle \Downarrow \langle \text{PRIMITIVE}(\text{cons}), \sigma_4 \rangle \quad \langle x, \rho, \sigma_4 \rangle \Downarrow \langle x, \sigma_5\{l_1 \to x\} \rangle \quad \langle xs, \rho, \sigma_5 \rangle \Downarrow \langle xs, \sigma_6\{l_2 \to xs\} \rangle \quad l_1 \notin \text{dom}\,\sigma_2 \quad l_2 \notin \text{dom}\,\sigma_2 \quad l_1 \neq l_2}{\langle \text{APPLY}(\text{cons } x\ xs), \rho, \sigma_0 \rangle \Downarrow \langle \text{CONS}(x, xs), \sigma_2\{l_1 \to x, l_2 \to xs\} \rangle}$$

$$\frac{\langle (\text{cdr } A), \rho, \sigma \rangle \Downarrow \langle \text{PRIMITIVE}(\text{cdr}), \sigma_1 \rangle \quad \langle A, \rho, \sigma_1 \rangle \Downarrow \langle \text{CONS}(x, xs), \sigma_1 \rangle}{\langle \text{APPLY}(\text{cdr } (\text{cons } x\ xs)), \rho, \sigma_0 \rangle \Downarrow \langle \sigma_1(l_2), \sigma_1\{l_1 \to x, l_2 \to xs\} \rangle}$$

$$\boxed{\begin{gathered}\sigma_0 = \{\} \\ \sigma_1 = \{l_1 \to x, l_2 \to xs\}\end{gathered}}$$

(b) Prove $(cdr\ (cons\ e1\ e2)) = e2$, set $(cons\ e1\ e2) = A$

V1 is the value when e1 terminates.

V2 is the value when e2 terminates.

$$\dfrac{\begin{array}{c} \langle e, \rho, \{\} \rangle \Downarrow \langle PRIMITIVE(cons), \{\} \rangle \\ \langle e1, \rho, \{\} \rangle \Downarrow \langle V1, \{l_1 \to V1\} \rangle \\ \langle e2, \rho, \{l_1 \to V1\} \rangle \Downarrow \langle V_2, \{l_1 \to V_1, l_2 \to V_2\} \rangle \\ l_1 \notin \sigma_1 \quad l_2 \notin \sigma_1 \quad l_1 \neq l_2 \end{array}}{\langle (cons\ e1\ e2), \rho, \{\} \rangle \Downarrow \langle CONS(l_1, l_2), \sigma_1 \{l_1 \to V1, l_2 \to V2\} \rangle}$$

$$\dfrac{\langle (cdr\ A), \rho, \{\} \rangle \Downarrow \langle PRIMITIVE(cdr), \{l_1 \to V1, l_2 \to V2\} \rangle \quad \langle A, \rho, \{l_1 \to V1, l_2 \to V2\} \rangle \Downarrow \langle CONS(l_1, l_2), \{l_1 \to V1, l_2 \to V2\} \rangle}{\langle (cdr(cons\ e1\ e2)), \rho, \sigma_0 \rangle \Downarrow \langle V2, \{l_1 \to V1, l_2 \to V2\} \rangle}$$