

## In-Class Problems — Week 7, Wed

### An example Substitution Model evaluation

```
submodel-eval>> (letrec ((iter-fact
  (lambda (n)
    (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (iter-fact 2))

==(0, lets)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (iter-fact 2))
==(1, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) 2))
==(2, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  ((lambda () (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
==(3, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (if (<= n 0)
    1
    (* n (iter-fact (- n 1))))
==(4, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (if (<= 2 0)
    1
    (* n (iter-fact (- n 1))))
==(5, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (if ()
    1
    (* n (iter-fact (- n 1))))
==(6, if)==>
```

```

(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (* n (iter-fact (- n 1))))
==(7, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (* 2 (iter-fact (- n 1))))
==(8, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (* 2 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) (- n 1))))
==(9, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (* 2 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) (- 2 1))))
==(10, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))
  (* 2 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) 1)))
==(11, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 ((lambda () (if (<= n_0 0) 1 (* n_0 (iter-fact (- n_0 1)))))))
==(12, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (if (<= n_0 0) 1 (* n_0 (iter-fact (- n_0 1))))))
==(13, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (if (<= 1 0) 1 (* n_0 (iter-fact (- n_0 1))))))
==(14, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (if () 1 (* n_0 (iter-fact (- n_0 1))))))
==(15, if)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (* n_0 (iter-fact (- n_0 1))))))
==(16, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2))

```

```

      (n_0 1))
    (* 2 (* 1 (iter-fact (- n_0 1))))))
==(17, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (* 1 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) (- n_0 1))))))
==(18, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (* 1 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) (- 1 1))))))
==(19, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1))
  (* 2 (* 1 ((lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1)))) 0))))
==(20, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 (* 1 ((lambda () (if (<= n_1 0) 1 (* n_1 (iter-fact (- n_1 1))))))))))
==(21, lambda)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 (* 1 (if (<= n_1 0) 1 (* n_1 (iter-fact (- n_1 1)))))))
==(22, instantiate)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 (* 1 (if (<= 0 0) 1 (* n_1 (iter-fact (- n_1 1)))))))
==(23, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 (* 1 (if #t 1 (* n_1 (iter-fact (- n_1 1)))))))
==(24, if)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 (* 1 1)))

```

```

==(25, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  (* 2 1))
==(26, builtin)==>
(letrec ((iter-fact (lambda (n) (if (<= n 0) 1 (* n (iter-fact (- n 1))))))
  (n 2)
  (n_0 1)
  (n_1 0))
  2)
Final value after 27 steps:
2

submodel-eval>>

```

**Problem.** For each numbered step in the sample Substitution Model evaluations below, indicate the control parse and rewriting rule applied.

```

submodel-eval>>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
              1
              (* n (fact (- n 1))))))))
      ((y fact-def) n))))
  (factorial 1))
==(1)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
              1
              (* n (fact (- n 1))))))))
      ((y fact-def) n))))
  ((lambda (n)
    (letrec ((y

```



```

      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
    ((y fact-def) n))
== (4) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))
    == (5) ==>
    (letrec ((factorial
      (lambda (n)
        (letrec ((y
          (lambda (f)
            ((lambda (x)
              (f (lambda (z) ((x x) z))))
              (lambda (x)
                (f (lambda (z) ((x x) z)))))))
            (fact-def
              (lambda (fact)
                (lambda (n)
                  (if (<= n 0)
                      1
                      (* n (fact (- n 1)))))))
            ((y fact-def) n))))
          (n 1)
          (y
            (lambda (f)
              ((lambda (x)
                (f (lambda (z) ((x x) z))))
                (lambda (x)
                  (f (lambda (z) ((x x) z)))))))
              (fact-def
                (lambda (fact)
                  (lambda (n)
                    (if (<= n 0)
                        1
                        (* n (fact (- n 1)))))))
              ((lambda (f)
                (

```



```

      (lambda (x)
        (f (lambda (z) ((x x) z))))))
(fact-def
 (lambda (fact)
  (lambda (n)
   (if (<= n 0)
       1
       (* n (fact (- n 1))))))
 (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(((lambda ()
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 n))
==(8)==>
(letrec ((factorial
 (lambda (n)
  (letrec ((y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 ((y fact-def) n))))
 (n 1)
 (y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 ((lambda (x) (f (lambda (z) ((x x) z)))) (lambda (x) (f (lambda (z) ((x x) z))))))
 n))
==(9)==>
(letrec ((factorial
 (lambda (n)
  (letrec ((y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 ((y fact-def) n))))
 (n 1)
 (y
 (lambda (f)
  ((lambda (x)

```





```

(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(x (lambda (x) (f (lambda (z) ((x x) z))))))
(((lambda (fact)
  (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1))))))
  (lambda (z)
    ((x x) z)))
 n))
==(12)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1))))))))
      ((y fact-def) n))))
  (n 1)
  (y
   (lambda (f)
     ((lambda (x)
       (f (lambda (z) ((x x) z))))
      (lambda (x)
       (f (lambda (z) ((x x) z)))))))
    (fact-def
     (lambda (fact)
       (lambda (n)
         (if (<= n 0)
             1
             (* n (fact (- n 1)))))))
     (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
     (x (lambda (x) (f (lambda (z) ((x x) z))))))
     (fact (lambda (z) ((x x) z))))
    ((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))) n))
  ))
==(13)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1))))))))
      ((y fact-def) n))))
  (n 1)
  (y
   (lambda (f)
     ((lambda (x)
       (f (lambda (z) ((x x) z))))
      (lambda (x)
       (f (lambda (z) ((x x) z)))))))
    (fact-def
     (lambda (fact)
       (lambda (n)
         (if (<= n 0)
             1
             (* n (fact (- n 1)))))))
     (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
     (x (lambda (x) (f (lambda (z) ((x x) z))))))
     (fact (lambda (z) ((x x) z))))
    ((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))) n))
  ))

```



```

      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
         (f (lambda (z) ((x x) z)))))))
(fact-def
 (lambda (fact)
  (lambda (n)
   (if (<= n 0)
       1
       (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
(x (lambda (x) (f (lambda (z) ((x x) z))))))
(fact (lambda (z) ((x x) z)))
(n_0 1))
((lambda () (if (<= n_0 0) 1 (* n_0 (fact (- n_0 1))))))
==(16)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
       (fact-def
        (lambda (fact)
         (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
        ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
      (fact-def
       (lambda (fact)
        (lambda (n)
         (if (<= n 0)
             1
             (* n (fact (- n 1)))))))
       (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
        (x (lambda (x) (f (lambda (z) ((x x) z))))))
        (fact (lambda (z) ((x x) z)))
        (n_0 1))
        (if (<= n_0 0)
            1
            (* n_0 (fact (- n_0 1))))))
    (* n_0 (fact (- n_0 1))))))
==(17)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
       (fact-def
        (lambda (fact)
         (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
        ((y fact-def) n))))
    (y
     (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
      (fact-def
       (lambda (fact)
        (lambda (n)
         (if (<= n 0)
             1
             (* n (fact (- n 1)))))))
       (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
        (x (lambda (x) (f (lambda (z) ((x x) z))))))
        (fact (lambda (z) ((x x) z)))
        (n_0 1))
        (if (<= n_0 0)
            1
            (* n_0 (fact (- n_0 1))))))
    (* n_0 (fact (- n_0 1))))))

```











```

        (lambda (x)
          (f (lambda (z) ((x x) z))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n)))
(n 1)
(y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
 (x (lambda (x) (f (lambda (z) ((x x) z))))))
 (fact (lambda (z) ((x x) z)))
 (n_0 1)
 (z 0))
(* 1 ((x x) z))
==(26)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
          ((y fact-def) n))))
      (n 1)
      (y
       (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
          (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
          (x (lambda (x) (f (lambda (z) ((x x) z))))))
          (fact (lambda (z) ((x x) z)))
          (n_0 1)
          (z 0))
        (* 1 (((lambda (x) (f (lambda (z) ((x x) z)))) x) z)))
      ))))
==(27)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y

```

```

        (lambda (f)
          ((lambda (x)
             (f (lambda (z) ((x x) z))))
           (lambda (x)
              (f (lambda (z) ((x x) z)))))))
      (fact-def
       (lambda (fact)
         (lambda (n)
           (if (<= n 0)
               1
               (* n (fact (- n 1)))))))
      ((y fact-def) n)))
(n 1)
(y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
 (x (lambda (x) (f (lambda (z) ((x x) z))))))
 (fact (lambda (z) ((x x) z)))
 (n_0 1)
 (z 0))
(*
 1
 ((lambda (x)
  (f (lambda (z) ((x x) z))))
  (lambda (x)
   (f (lambda (z) ((x x) z))))
  z)))
==(28)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
         (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
         ((y fact-def) n))))
      (n 1)
      (y
       (lambda (f)
        ((lambda (x)
         (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
        (fact-def
         (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
        (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))

```











```

(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(x (lambda (x) (f (lambda (z) ((x x) z)))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(z 0)
(x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
(fact_0 (lambda (z) ((x_0 x_0) z)))
(n_1 0))
(* 1 (if #t 1 (* n_1 (fact_0 (- n_1 1)))))
==(38)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
  (n 1)
  (y
   (lambda (f)
     ((lambda (x)
       (f (lambda (z) ((x x) z))))
      (lambda (x)
       (f (lambda (z) ((x x) z)))))))
   (fact-def
    (lambda (fact)
      (lambda (n)
        (if (<= n 0)
            1
            (* n (fact (- n 1)))))))
    (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
    (x (lambda (x) (f (lambda (z) ((x x) z)))))
    (fact (lambda (z) ((x x) z)))
    (n_0 1)
    (z 0)
    (x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
    (fact_0 (lambda (z) ((x_0 x_0) z)))
    (n_1 0))
  (* 1 1))
==(39)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
  (n 1)
  (y
   (lambda (f)
     ((lambda (x)
       (f (lambda (z) ((x x) z))))
      (lambda (x)
       (f (lambda (z) ((x x) z)))))))
   (fact-def
    (lambda (fact)
      (lambda (n)
        (if (<= n 0)
            1
            (* n (fact (- n 1)))))))
    (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
    (x (lambda (x) (f (lambda (z) ((x x) z)))))
    (fact (lambda (z) ((x x) z)))
    (n_0 1)
    (z 0)
    (x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
    (fact_0 (lambda (z) ((x_0 x_0) z)))
    (n_1 0))
  (* 1 1))

```



```

(* n (fact (- n 1))))))
  ((y fact-def) n)))
(n 1)
(y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
      (lambda (x)
        (f (lambda (z) ((x x) z)))))))
(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1))))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
  (x (lambda (x) (f (lambda (z) ((x x) z))))
  (fact (lambda (z) ((x x) z)))
  (n_0 1)
  (z 0)
  (x_0 (lambda (x) (f (lambda (z) ((x x) z))))
  (fact_0 (lambda (z) ((x_0 x_0) z)))
  (n_1 0)
1)
Final value after 40 steps:
1
submodel-eval>>

```

**SOLUTION**

```

submodel-eval>>
(letrec ((factorial
  (lambda (n)
    (letrec ((y (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z))))))))
      (fact-def (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1))))))))
      ((y fact-def) n))))
  (factorial 1))

==(0, lets)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
         (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1))))))))
        ((y fact-def) n))))
  (factorial 1))

==(1, instantiate)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
         (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1))))))))
        ((y fact-def) n))))
  ((lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
         (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1))))))))
        ((y fact-def) n))))
    1))

```







```

      (f (lambda (z) ((x x) z))))))
  n))
==(8, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
  (n 1)
  (y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
     (lambda (x)
      (f (lambda (z) ((x x) z)))))))
  (fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
  (((lambda (x) (f (lambda (z) ((x x) z)))) (lambda (x) (f (lambda (z) ((x x) z))))
  n))
==(9, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
  (n 1)
  (y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
     (lambda (x)
      (f (lambda (z) ((x x) z)))))))
  (fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
  (x (lambda (x) (f (lambda (z) ((x x) z))))))
  (((lambda () (f (lambda (z) ((x x) z)))) n))
==(10, lambda)==>

```

```

(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1))))))))
        ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
       ((lambda (x)
         (f (lambda (z) ((x x) z))))
         (lambda (x)
           (f (lambda (z) ((x x) z)))))))
       (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
        (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
        (x (lambda (x) (f (lambda (z) ((x x) z)))))))
      ((f (lambda (z) ((x x) z))) n))
    ==(11, instantiate)==>
    (letrec ((factorial
      (lambda (n)
        (letrec ((y
          (lambda (f)
            ((lambda (x)
              (f (lambda (z) ((x x) z))))
              (lambda (x)
                (f (lambda (z) ((x x) z)))))))
            (fact-def
              (lambda (fact)
                (lambda (n)
                  (if (<= n 0)
                      1
                      (* n (fact (- n 1)))))))
            ((y fact-def) n))))
          (n 1)
          (y
           (lambda (f)
             ((lambda (x)
               (f (lambda (z) ((x x) z))))
               (lambda (x)
                 (f (lambda (z) ((x x) z)))))))
             (fact-def
              (lambda (fact)
                (lambda (n)
                  (if (<= n 0)
                      1
                      (* n (fact (- n 1)))))))
              (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
              (x (lambda (x) (f (lambda (z) ((x x) z)))))))
            ((lambda (fact)
              (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1)))))))
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))

```

```

(lambda (z)
  ((x x) z)))
n))
==(12, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
  (y
    (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact (lambda (z) ((x x) z))))
    ((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))) n))
  ))
==(13, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
  (y
    (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact (lambda (z) ((x x) z))))
    ((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))) n))
  ))

```



```

      (fact (lambda (z) ((x x) z))))
    ((lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))) n))
== (14, instantiate)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))
        (x (lambda (x) (f (lambda (z) ((x x) z))))
          (fact (lambda (z) ((x x) z))))
        ((lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))) 1))
      == (15, lambda)==>
      (letrec ((factorial
        (lambda (n)
          (letrec ((y
            (lambda (f)
              ((lambda (x)
                (f (lambda (z) ((x x) z))))
              (lambda (x)
                (f (lambda (z) ((x x) z)))))))
            (fact-def
              (lambda (fact)
                (lambda (n)
                  (if (<= n 0)
                    1
                    (* n (fact (- n 1)))))))
            ((y fact-def) n))))
            (n 1)
            (y
              (lambda (f)
                ((lambda (x)
                  (f (lambda (z) ((x x) z))))
                (lambda (x)
                  (f (lambda (z) ((x x) z)))))))
              (fact-def
                (lambda (fact)
                  (lambda (n)
                    (if (<= n 0)
                      1
                      (* n (fact (- n 1)))))))
              (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))
                (x (lambda (x) (f (lambda (z) ((x x) z))))
                  (fact (lambda (z) ((x x) z))))
                (fact (lambda (z) ((x x) z))))

```





```

(lambda (fact)
  (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))
          (fact (lambda (z) ((x x) z)))
              (n_0 1))
      (* n_0 (fact (- n_0 1))))
  == (20, instantiate) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))
          ((y fact-def) n))))
      (n 1)
      (y
       (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
       (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))
          (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))
              (x (lambda (x) (f (lambda (z) ((x x) z))))
                  (fact (lambda (z) ((x x) z)))
                      (n_0 1))
              (* 1 (fact (- n_0 1))))
          == (21, instantiate) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))
          ((y fact-def) n))))
      (n 1)
      (y
       (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))

```

```

(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(x (lambda (x) (f (lambda (z) ((x x) z)))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(* 1 ((lambda (z) ((x x) z)) (- n_0 1)))
==(22, instantiate)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
     (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z)))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (* 1 ((lambda (z) ((x x) z)) (- 1 1))))
    ))
  (letrec ((factorial
    (lambda (n)
      (letrec ((y
        (lambda (f)
          ((lambda (x)
            (f (lambda (z) ((x x) z))))
           (lambda (x)
            (f (lambda (z) ((x x) z)))))))
          (fact-def
            (lambda (fact)
              (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1)))))))
            ((y fact-def) n))))
        (n 1)
        (y
         (lambda (f)
          ((lambda (x)
            (f (lambda (z) ((x x) z))))
           (lambda (x)
            (f (lambda (z) ((x x) z)))))))
          (fact-def
            (lambda (fact)
              (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1)))))))
            (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
            (x (lambda (x) (f (lambda (z) ((x x) z)))))
            (fact (lambda (z) ((x x) z)))
            (n_0 1)
            (* 1 ((lambda (z) ((x x) z)) (- 1 1))))
          ))
        ))
    ))
  ))

```

```

      (f (lambda (z) ((x x) z))))))
(fact-def
 (lambda (fact)
  (lambda (n)
   (if (<= n 0)
       1
       (* n (fact (- n 1))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
(x (lambda (x) (f (lambda (z) ((x x) z))))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(* 1 ((lambda (z) ((x x) z)) 0)))
==(24, lambda)==>
(letrec ((factorial
 (lambda (n)
  (letrec ((y
            (lambda (f)
              ((lambda (x)
                 (f (lambda (z) ((x x) z))))
                (lambda (x)
                 (f (lambda (z) ((x x) z))))))
             (fact-def
              (lambda (fact)
               (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1))))))
              ((y fact-def) n))))
 (n 1)
 (y
  (lambda (f)
   ((lambda (x)
      (f (lambda (z) ((x x) z))))
    (lambda (x)
     (f (lambda (z) ((x x) z))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1))))))
 (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
 (x (lambda (x) (f (lambda (z) ((x x) z))))))
 (fact (lambda (z) ((x x) z)))
 (n_0 1)
 (z 0))
 (* 1 ((lambda () ((x x) z))))))
==(25, lambda)==>
(letrec ((factorial
 (lambda (n)
  (letrec ((y
            (lambda (f)
              ((lambda (x)
                 (f (lambda (z) ((x x) z))))
                (lambda (x)
                 (f (lambda (z) ((x x) z))))))
             (fact-def
              (lambda (fact)
               (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1))))))
              ((y fact-def) n))))
 (n 1)
 (y
  (lambda (f)
   ((lambda (x)

```



```

(y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
      (lambda (x)
        (f (lambda (z) ((x x) z))))))))
(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1))))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
  (x (lambda (x) (f (lambda (z) ((x x) z))))
  (fact (lambda (z) ((x x) z)))
  (n_0 1)
  (z 0))
(*
  1
  ((lambda (x)
    (f (lambda (z) ((x x) z))))
    (lambda (x)
      (f (lambda (z) ((x x) z))))
    z)))
==(28, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1))))))))
          ((y fact-def) n))))
  (n 1)
  (y
    (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z))))))))
    (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
    (* 1 ((lambda () (f (lambda (z) ((x_0 x_0) z)))) z)))
==(29, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1))))))))
          ((y fact-def) n))))
  (n 1)
  (y
    (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z))))))))
    (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
    (* 1 ((lambda () (f (lambda (z) ((x_0 x_0) z)))) z)))
==(29, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1))))))))
          ((y fact-def) n))))
  (n 1)
  (y
    (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
        (lambda (x)
          (f (lambda (z) ((x x) z))))))))
    (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
    (* 1 ((lambda () (f (lambda (z) ((x_0 x_0) z)))) z)))
==(29, lambda)==>

```



```

        (lambda (x)
          (f (lambda (z) ((x x) z))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n)))
(n 1)
(y
 (lambda (f)
  ((lambda (x)
   (f (lambda (z) ((x x) z))))
   (lambda (x)
    (f (lambda (z) ((x x) z)))))))
 (fact-def
  (lambda (fact)
   (lambda (n)
    (if (<= n 0)
        1
        (* n (fact (- n 1)))))))
 (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
 (x (lambda (x) (f (lambda (z) ((x x) z))))))
 (fact (lambda (z) ((x x) z)))
 (n_0 1)
 (z 0)
 (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
 (* 1 ((f (lambda (z) ((x_0 x_0) z))) z)))
== (30, instantiate) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
        ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
      ((lambda (x)
       (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
      (fact-def
       (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
       (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
        (x (lambda (x) (f (lambda (z) ((x x) z))))))
        (fact (lambda (z) ((x x) z)))
        (n_0 1)
        (z 0)
        (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
        (*
         1
         ((lambda (fact)

```

```

(lambda (n)
  (if (<= n 0)
      1
      (* n (fact (- n 1)))))
(lambda (z)
  ((x_0 x_0) z))
z)))
==(31, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))
          ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
       ((lambda (x)
         (f (lambda (z) ((x x) z))))
         (lambda (x)
           (f (lambda (z) ((x x) z)))))))
       (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))
        (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))
        (x (lambda (x) (f (lambda (z) ((x x) z))))
        (fact (lambda (z) ((x x) z)))
        (n_0 1)
        (z 0)
        (x_0 (lambda (x) (f (lambda (z) ((x x) z))))
        (fact_0 (lambda (z) ((x_0 x_0) z))))
        (* 1 (((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact_0 (- n 1))))) z)))
    ))
    ))
  ))
    (lambda (n)
      (letrec ((y
        (lambda (f)
          ((lambda (x)
            (f (lambda (z) ((x x) z))))
            (lambda (x)
              (f (lambda (z) ((x x) z)))))))
          (fact-def
            (lambda (fact)
              (lambda (n)
                (if (<= n 0)
                    1
                    (* n (fact (- n 1)))))
            ((y fact-def) n))))
        (n 1)
        (y
         (lambda (f)
           ((lambda (x)
             (f (lambda (z) ((x x) z))))
             (lambda (x)
               (f (lambda (z) ((x x) z)))))))
           (fact_0 (lambda (z) ((x_0 x_0) z))))
           (* 1 (((lambda () (lambda (n) (if (<= n 0) 1 (* n (fact_0 (- n 1))))) z)))
        ))
      ))
    ))
  ))

```

```

(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(x (lambda (x) (f (lambda (z) ((x x) z)))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(z 0)
(x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
(fact_0 (lambda (z) ((x_0 x_0) z)))
(* 1 ((lambda (n) (if (<= n 0) 1 (* n (fact_0 (- n 1)))) z)))
==(33, instantiate)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
     (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
     (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z)))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
      (fact_0 (lambda (z) ((x_0 x_0) z))))
     (* 1 ((lambda (n) (if (<= n 0) 1 (* n (fact_0 (- n 1)))) 0)))
    ))
  ))
==(34, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
         (lambda (x)
          (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (y
     (lambda (f)
      ((lambda (x)
        (f (lambda (z) ((x x) z))))
       (lambda (x)
        (f (lambda (z) ((x x) z)))))))
     (fact-def
      (lambda (fact)
        (lambda (n)
          (if (<= n 0)
              1
              (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z)))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
      (fact_0 (lambda (z) ((x_0 x_0) z))))
     (* 1 ((lambda (n) (if (<= n 0) 1 (* n (fact_0 (- n 1)))) 0)))
    ))
  ))

```

```

(n 1)
(y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
      (lambda (x)
        (f (lambda (z) ((x x) z))))))))
(fact-def
  (lambda (fact)
    (lambda (n)
      (if (<= n 0)
          1
          (* n (fact (- n 1)))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
(x (lambda (x) (f (lambda (z) ((x x) z)))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(z 0)
(x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
(fact_0 (lambda (z) ((x_0 x_0) z)))
(n_1 0))
(* 1 ((lambda () (if (<= n_1 0) 1 (* n_1 (fact_0 (- n_1 1)))))))
==(>35, lambda)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1))))))))
        ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z))))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
        (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1)))))))
        (x (lambda (x) (f (lambda (z) ((x x) z)))))
        (fact (lambda (z) ((x x) z)))
        (n_0 1)
        (z 0)
        (x_0 (lambda (x) (f (lambda (z) ((x x) z)))))
        (fact_0 (lambda (z) ((x_0 x_0) z)))
        (n_1 0))
        (* 1 (if (<= n_1 0) 1 (* n_1 (fact_0 (- n_1 1)))))))
==(>36, instantiate)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))

```

```

            (lambda (x)
              (f (lambda (z) ((x x) z))))))
  (fact-def
    (lambda (fact)
      (lambda (n)
        (if (<= n 0)
            1
            (* n (fact (- n 1)))))))
  ((y fact-def) n)))
(n 1)
(y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
      (lambda (x)
        (f (lambda (z) ((x x) z)))))))
  (fact-def
    (lambda (fact)
      (lambda (n)
        (if (<= n 0)
            1
            (* n (fact (- n 1)))))))
  (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
  (x (lambda (x) (f (lambda (z) ((x x) z))))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(z 0)
(x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
(fact_0 (lambda (z) ((x_0 x_0) z)))
(n_1 0))
(* 1 (if (<= 0 0) 1 (* n_1 (fact_0 (- n_1 1)))))
==(37, builtin)==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
          ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
        (fact-def
          (lambda (fact)
            (lambda (n)
              (if (<= n 0)
                  1
                  (* n (fact (- n 1)))))))
          (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
          (x (lambda (x) (f (lambda (z) ((x x) z))))))
          (fact (lambda (z) ((x x) z)))
          (n_0 1)
          (z 0)
          (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
          (fact_0 (lambda (z) ((x_0 x_0) z))))

```

```

    (n_1 0))
    (* 1 (if #t 1 (* n_1 (fact_0 (- n_1 1))))))
== (38, if) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      (f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact_0 (lambda (z) ((x_0 x_0) z)))
      (n_1 0))
    (* 1 1))
== (39, builtin) ==>
(letrec ((factorial
  (lambda (n)
    (letrec ((y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      ((y fact-def) n))))
    (n 1)
    (y
      (lambda (f)
        ((lambda (x)
          (f (lambda (z) ((x x) z))))
          (lambda (x)
            (f (lambda (z) ((x x) z)))))))
      (fact-def
        (lambda (fact)
          (lambda (n)
            (if (<= n 0)
                1
                (* n (fact (- n 1)))))))
      (x (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact (lambda (z) ((x x) z)))
      (n_0 1)
      (z 0)
      (x_0 (lambda (x) (f (lambda (z) ((x x) z))))))
      (fact_0 (lambda (z) ((x_0 x_0) z)))
      (n_1 0))
    (* 1 1))

```

```
1
  (* n (fact (- n 1))))))
(f (lambda (fact) (lambda (n) (if (<= n 0) 1 (* n (fact (- n 1))))))
(x (lambda (x) (f (lambda (z) ((x x) z))))
(fact (lambda (z) ((x x) z)))
(n_0 1)
(z 0)
(x_0 (lambda (x) (f (lambda (z) ((x x) z))))
(fact_0 (lambda (z) ((x_0 x_0) z)))
(n_1 0))
1)
Final value after 40 steps:
1
submodel-eval>>
```