X → 10

fact → < e, (n) (letrec...~) >

(define fact
  (lambda (n)
    (letrec
      ([aux (lambda (n acc)
        (if (= n 0)
          acc
          (aux (- n 1)
            (* n acc)]...)
      (aux n 1)]])

(fact 2)

fact 2
n → 2 ⇒ 2

letrec
aux → < e, (n acc] (if...~) > ⇒ 2

execution model with environments

(aux 2 1)
n → 2 ⇒ 2
acc → 1

(aux 1 2)
n → 1 ⇒ 2
acc → 2

(aux 0 2)
n → 0 ⇒ 2
acc → 2

tail recursion:
optimize
~reuse aux environment
for each call
since not needed again