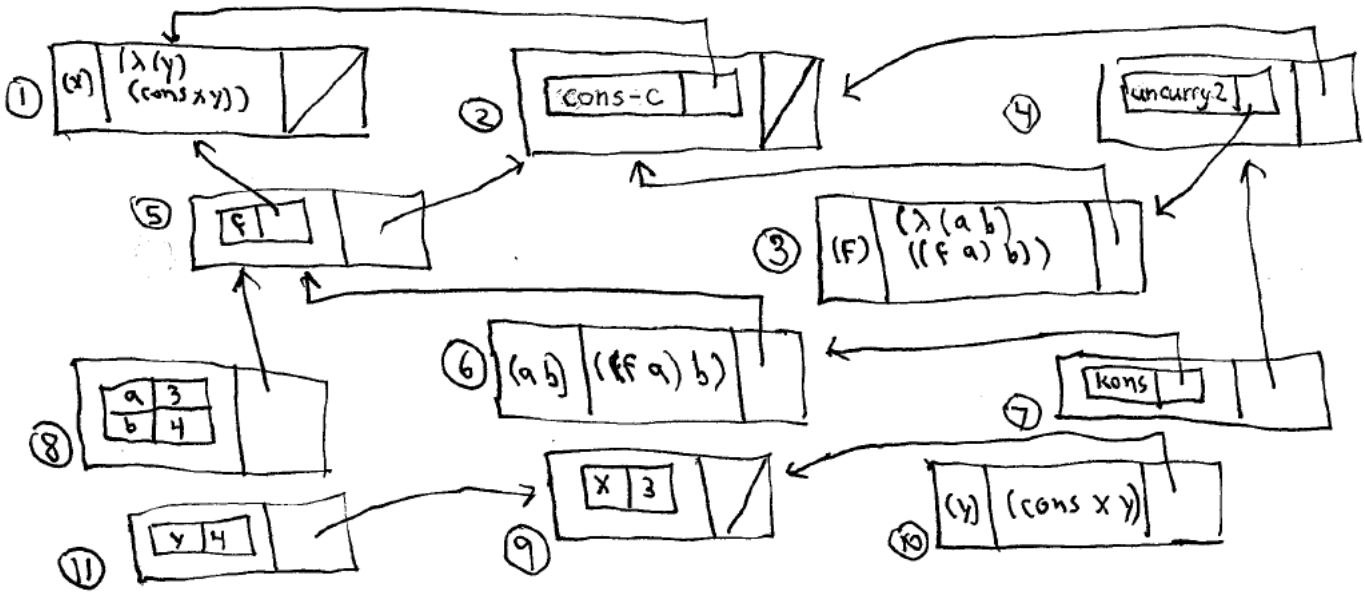
CSSE 304 E&C exam Fall 2020 solution



Notes: let* is equivalent to nested lets. ①Evaluate the first expression to be assigned to a let variable. Since this is evaluated outside the outer let, the environment pointer in the closure is null. ② Make the env for the outer let. This let is not inside any other let or lambda, so its environment pointer is null. The body of the outer let is the second nested let. We evaluate that in env ②. ③ Evaluate the second expression to be assigned to a let variable. Since this lambda-expression is evaluated in env② , the new closure's env pointer points to ②. ④ Make the env for the second let. Its env pointer is to current env , ② Then (in env ④) we evaluate .the body of the second let, which is the third let. The first thing to evaluate there is (uncurry2 cons-c). ⑤ First we make the environment for this application, binding the argument, which is ① , to ③'s parameter which is f. As always when we apply a closure, env ⑤ 's environment pointer is a copy of closure ③'s environment. Now we evaluate ③'s body with as the current env. This gives us closure ⑥ . Whenever we make a closure, its env pointer points to the enclosing env, in this case ⑤. So closure ⑥ is the value of (uncurry2 cons-c), which now gets bound to kons in env ⑦. Since the 3rd let is the body of the 2nd let, the 3rd let's env's env pointer points to the current env (the 2nd let's env, which is ④). Now all of the assignments of let variables are done, and it is time to evaluate the body of let *, which after translation from let* into nested lets, has become the 3rd let's body (kons 3 4). This is evaluated with ⑦ as the current environment. kons is in the local environment, and it is bound to closure ⑥ So we are applying to integers 3 and 4. The first step is to make environment ⑧, in which ⑥'s parameters a and b are bound to 3 and 4. ⑧'s env pointer is copied from⑥ , so it points to ⑤. Once ⑧ is established, we evaluate the body of ⑥ in env ⑧. The first thing in the body of is (f a). a is found directly in ⑧ and when we follow the pointer we find (in ⑤) that f is ① . The first step in applying ① to 3. is to create environment ⑨. ⑨'s env pointer is copied from closure ①. Now we evaluate the body of ① with⑨ as the current env. That body is a lambda expression, whose evaluation gives us closure ⑩. ⑩ 's env pointer points to the current environment , which is ⑨. ⑩ is the value returned by ((f a). That now gets applied to b. The first step in that application is to create environment ⑪. ⑪'s env pointer is a copy of ⑩'s, so it points to ⑨. Finally, we evaluate ⑩'s body , (cons x y), in env ⑪, which gives us the answer (3 . 4).

```
(let* ([cons-c (lambda (x)
                 (lambda (y)
                   (cons x y)))]
       [uncurry2 (lambda (f)
                   (lambda (a b)
                     ((f a)b)))]
       [kons (uncurry2 cons-c)])
  (kons 3 4))
```