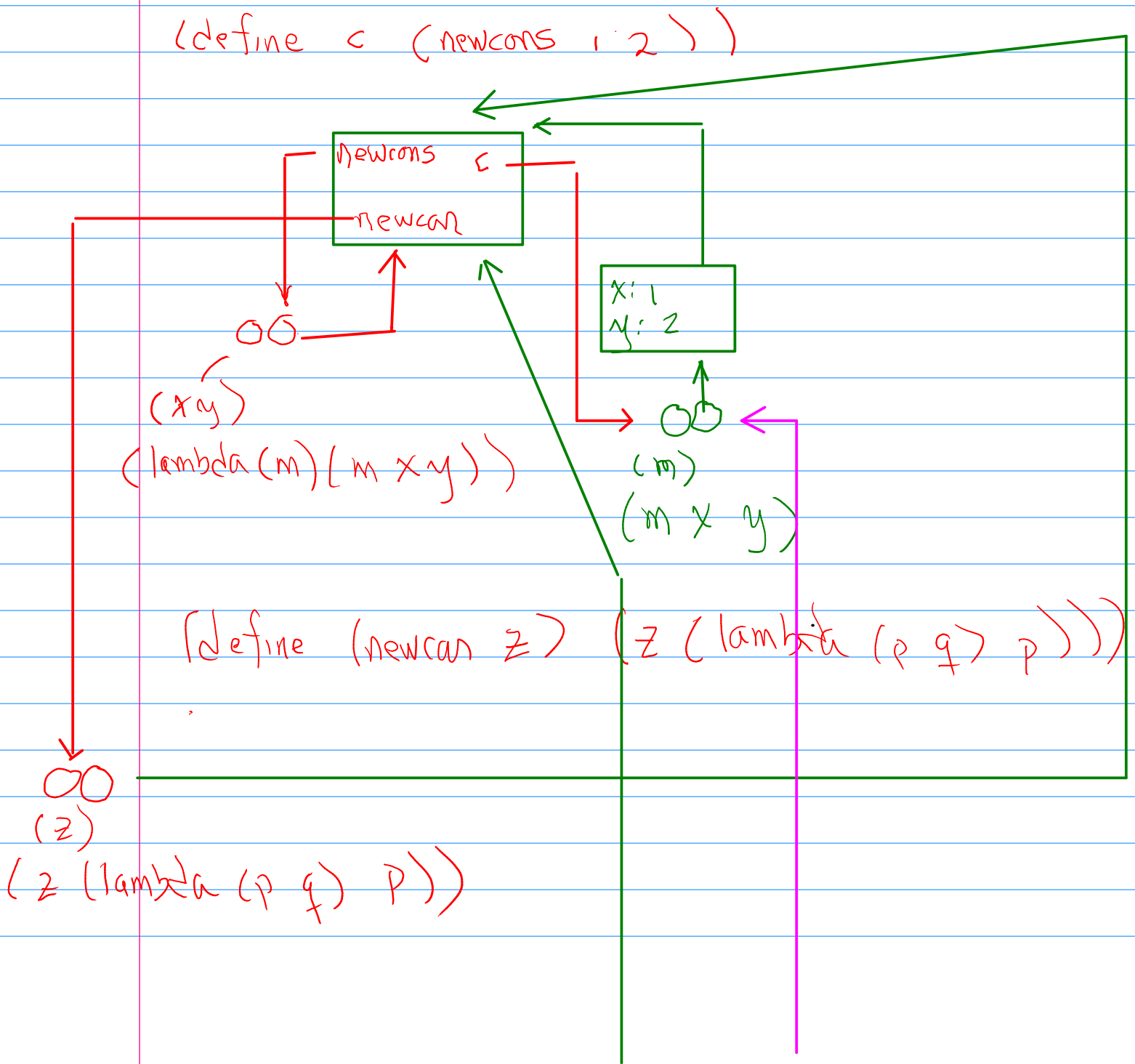
; functional implementation of cons, car and cdr

; Exercise 2.4

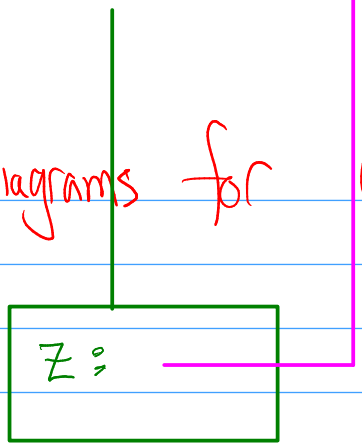
```
(define (newcons x y)
  (lambda (m) (m x y)))

(define (newcar z)
  (z (lambda (p q) p)))

(define (newcdr z)
  (z (lambda (p q) q)))
```

To understand how these work, draw the diagrams!



Can we draw the diagrams for (newcar c)?

child frame for newcar



(z (lambda (p q) p))

So - looking at the value of z - z is a function of 1 (function) argument m, and it applies that function to x and y - x=1, y=2. So the result is 1.

(Please finish the diagram so that the evaluation of (newcar c) is completely displayed)

Strongly suggested: figure out how you can use these functions at the top level of TLS. That is, find an expression in TLS which (i) provides bindings for newcons, newcar, and newcdr, and then (ii) computes some values using these -- for example, (newcar (newcons 1 2))

This would allow us to remove the pair primitives from TLS, bringing us closer to a pure lambda calculus implementation.