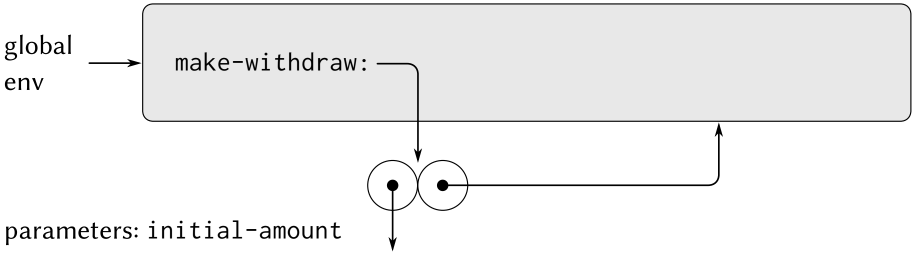
Exercise 3.10

The let-syntax is replaced with the equivalent lambda-construct:

```
(define (make-withdraw initial-amount)
  ((lambda (balance)
     (lambda (amount)
       (if (>= balance amount)
           (begin (set! balance
                        (- balance amount))
                  balance)
           "Insufficient funds")))
   initial-amount))
```

After executing the above definition in the global environment:



parameters: `initial-amount`

body:
```
      (lambda (amount)
       (if (>= initial-amount amount)
           (begin (set! initial-amount (- initial-amount amount))
                  initial-amount)
           "Insufficient funds"))
```

After this, the other diagrams will be exactly like figures 3.7 - 3.10 in the book, except balance is replaced with initial-amount. So, objects defined with let-syntax behave the same.

This is a shortcut. In reality, after (W1 50) is executed, another frame under "initial-amount: 100" is created: "balance: initial-amount". Briefly before set! command the balance is 100. After set!, the new frame reads: "balance: 50".