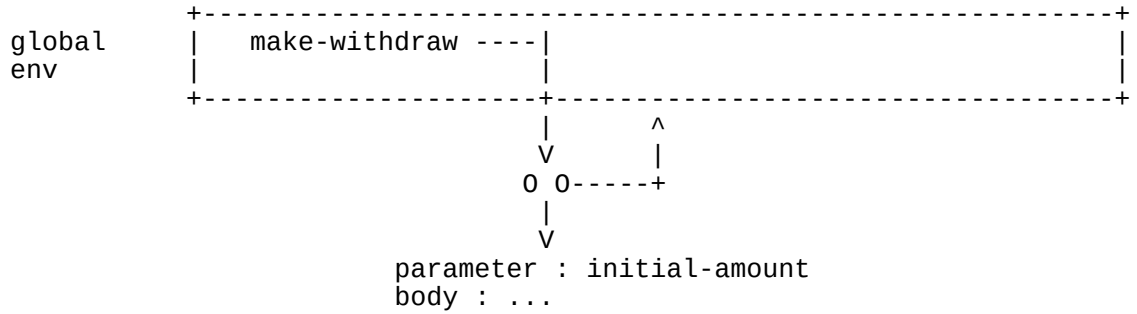
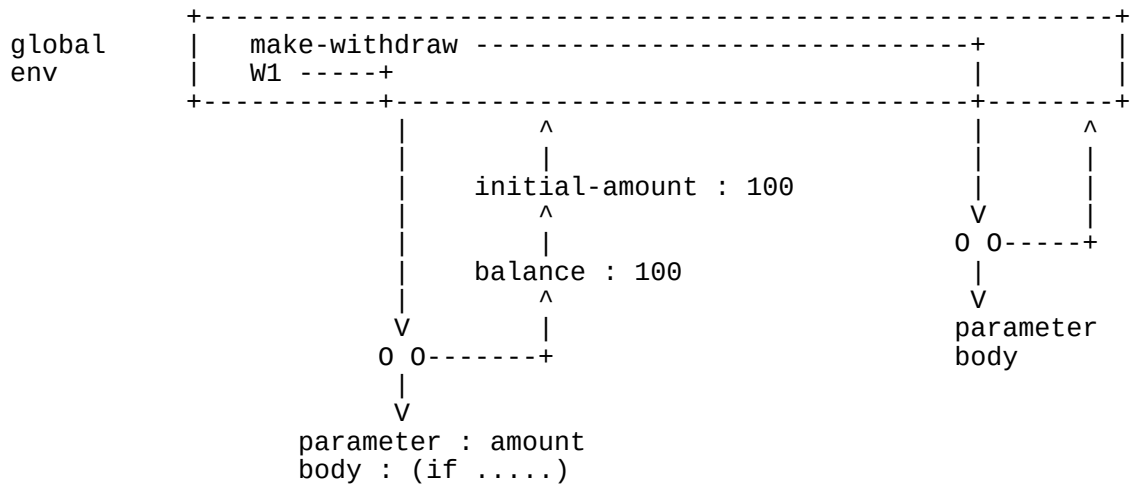
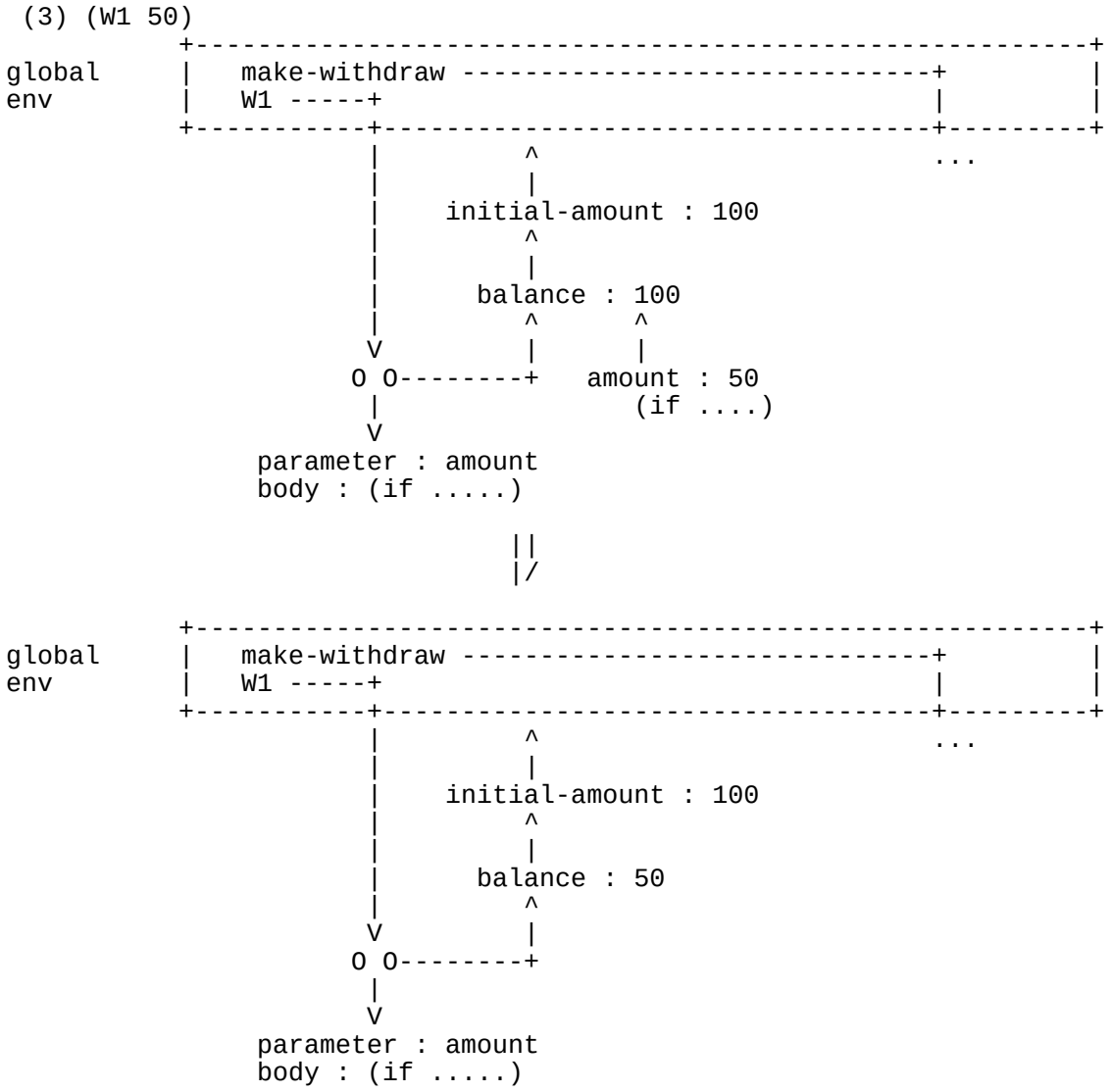
```
; 2018 Fall : Homework solution No. 6

================================
Problem 1: prev, make-prev
--------------------------------

   a.   (define prev
          (let ((now 'first-call))
            (lambda (value)
              (begin (let ((pre now))
                (set! now value) pre)))))

   b.   (define (make-prev now)
          (lambda (value)
            (begin (let ((pre now))
              (set! now value) pre))))



================================
Problem 2:    ex 3.10
--------------------------------

(0). let expr. -> lambda
        (define (make-withdraw initial-amount)
                (lambda (balance)
                        (lambda (amount)
                                .....
                )
                initial-amount)




(1) (define (make-withdraw initial-amount)
           ......
           )
```



(2) (define W1 (make-define 100))
.....
)




(3) (W1 50)
global env
make-withdraw
W1
initial-amount : 100
balance : 100
amount : 50
(if ....)
parameter : amount
body : (if .....)
...
global env
make-withdraw
W1
initial-amount : 100
balance : 50
parameter : amount
body : (if .....)
...

(4) (define W2 (make-withdraw 100))