```
(define-struct drop (x y))
;; Drop is (make-drop Integer Integer)
;; a raindrop at (x,y) in screen coordinates
(define D1 (make-drop 34 25))

(define (fn-for-drop d)
  (... (drop-x d)
       (drop-y d)))
;; Template rules used
;; - compound: 2 fields


;; ListOfDrop is one of:
;; - empty
;; - (cons Drop ListOfDrop)
;; interp. a list of raindrops
(define LOD1 empty)
(define LOD2 (cons (make-drop 2 66)
                   (cons (make-drop 23 99) empty)))

(define (fn-for-lod lod)
  (cond [(empty? lod) (...)]
        [else
         (... (fn-for-drop (first lod))    ; Drop
              (fn-for-lod (rest lod)))]))  ; ListOfDrop

;; Template rules used:
;; - one of: 2 cases
;; - atomic distinct: empty
;; - compound: (cons Drop ListOfDrop)
;; - reference: (first lod) is Drop
;; - self-reference: (rest lod) is ListOfDrop
```

R2

NH2

SR 1

NR 1