

```

1 (require spd/tags)
2
3 (@htdd ListOfString)
4 ;; ListOfString is one of:
5 ;; - empty
6 ;; - (cons String ListOfString)
7 ;; interp. a list of strings
8
9 (define LOS1 empty)
10 (define LOS2 (cons "yellow" (cons "dog" (cons "pie" empty))))
11
12 (@dd-template-rules one-of
13     atomic-distinct
14     compound
15     self-ref)
16
17 (define (fn-for-los los)
18   (cond [(empty? los) (...)]
19         [else
20          (... (first los)
21              (fn-for-los (rest los)))]))
22
23
24 (@htdf concatenate>5)
25 (@signature ListOfString -> String)
26 ;; produce the concatenation of all the strings with length >5 in the list
27 (check-expect (concatenate>5 empty) "")
28 (check-expect (concatenate>5 (cons "apple" empty)) "") B
29 (check-expect (concatenate>5 (cons "orange" empty)) "orange")
30 (check-expect (concatenate>5 (cons "hi" (cons "yellow" empty))) "yellow")
31 (check-expect (concatenate>5 (cons "yellow" (cons "hi" empty))) "yellow")
32 (check-expect (concatenate>5 (cons "elephant"
33     (cons "dog"
34     (cons "horse"
35     (cons "raccoon" empty)))))
36     "elephantraccoon")
37
38 (@template-origin ListOfString) C
39
40 (define (concatenate>5 los) D
41   (cond [(empty? los) ""]
42         [else
43          (if (len>5? (first los))
44              (string-append (first los) (concatenate>5 (rest los)))
45              (concatenate>5 (rest los)))]))
46
47
48 (@htdf len>5?)
49 (@signature String -> Boolean)
50 ;; produce true if length of s > 5, false otherwise
51 (check-expect (len>5? "") false) E
52 (check-expect (len>5? "that") false)
53 (check-expect (len>5? "those") false)
54 (check-expect (len>5? "theirs") true)
55
56 (@template-origin String)
57
58 (define (len>5? s)
59   (> (string-length s) 5)) F

```