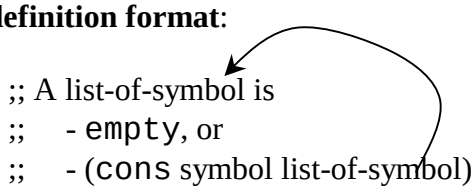
Data Definition Guidelines : Lists of Atomic Data

Example: a list of symbols

Data definition format:

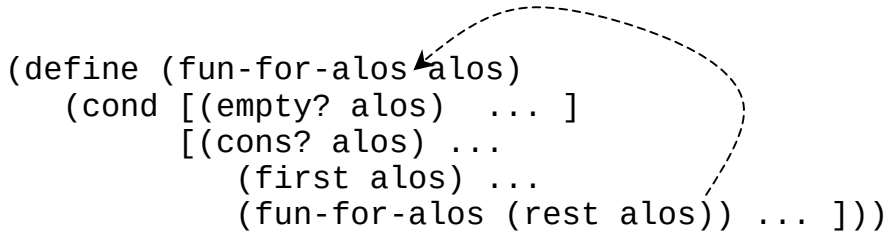


```
;; A list-of-symbol is
;;   - empty, or
;;   - (cons symbol list-of-symbol)
```

This definition has a similar format to mixed data definition. The main difference is that lists are recursive: they refer to themselves in the `cons` case. We annotate the data definition with the arrow to capture this self-reference.

Template format: the template does three things:

- decides which case of data we have
- pulls out pieces in the cons case (the case that has pieces)
- uses a recursive call to mimic the arrow in text



```
(define (fun-for-alos alos)
   (cond [(empty? alos)  ... ]
         [(cons? alos) ...
            (first alos) ...
            (fun-for-alos (rest alos)) ... ]))
```

Note how similar this is to the template for mixed data: all we've really added is the arrow/recursive call, which is the only thing we added to the data definition. The dashed arrow in the template matches the arrow in the data defintion. **The number of arrows in the data definition and template must always match**!

.