
CS 331 — List Zippers

Mattox Beckman

Introduction

Zippers: the benefits of doubly linked lists without the pain! This activity will give you practice writing a few list zipper functions.

Code

```
(defrecord ListZip [before after])
(defn make-list-zip [x]
  (ListZip. '() x))
(defn current [z] (first (:after z)))
(defn forward [z]
  (ListZip. (cons (-> z :after first) (:before z))
            (rest (:after z))))
(defn backward [x]
  (ListZip. (rest (:after z))
            (cons (-> z :before first) (:after z))))
```

Questions

1. Using zippers, write a search function that takes a list and an element and returns a list with seven items: the three elements before the one you found, the one you found, and the three elements after the one you found.
2. Using zippers, write the function `(delete-all f xx)` that takes a function `f` and zipper `xx` and returns a new zipper in which all $x \in xx$ such that $(f\ x)$ is true have been deleted. So `(delete-all odd xx)` will remove all the odd elements from the zipper.