Inductive List

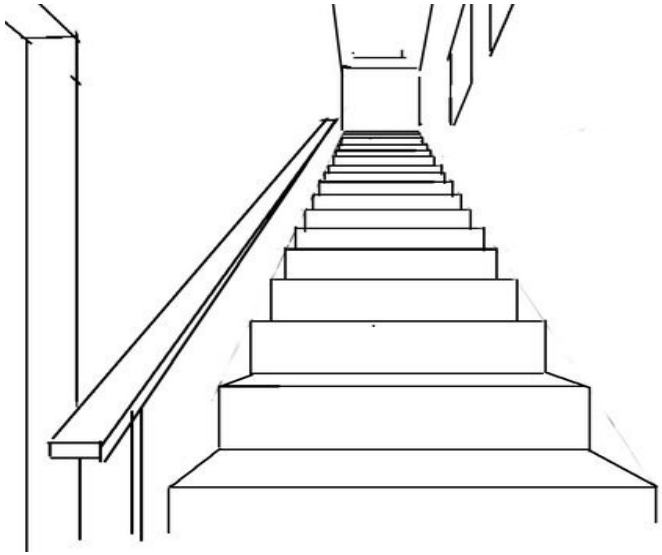
What is a list?

- Empty, or
- Non-empty, consisting of a head element and the reminder of the list.

Exercise: Predict the results of the following expressions:

1. `(car '(1 2 3))`
2. `(cdr '(1 2 3))`
3. `(car (cdr (cdr '(1 2 3))))`
4. `(null? (cdr (cdr (cdr '(1 2 3)))))`

Recursive Design



- A function to achieve a goal over an object.
- Decompose an object into a single "head" and the rest (same object with smaller size).
- Call the function over a smaller sized object (the recursive assumption).
- "Sum up" all the results until the smaller sized object cannot be decomposed anymore.

Pattern Matching with Lists

- Match the data structure, in this case our list, use patter match to get rid of let bindings.
- case 1: base case, empty $\rightarrow$ action.
- case 2: head and tail $\rightarrow$ action.

Maybe more cases?

```
(define (fibo n)
  (match (n)
    [0 1]
    [1 1]
    [_ (+ (fibo (- n 1)) (fibo (- n 2)))])
    )
```

Inductive Reasoning

From recursive function to inductive reasoning

- The object is inductively constructed, it's infinite.
- Why a proposition over this object in general is correct? (It's impossible to go over one by one.)
- Starting the proof with the decomposing of the object (e.g. List).

One of proof techniques in formal math.

base case + inductive hypothesis (same claim but smaller size)
$\rightarrow$ For all cases.

Q & A