Programming Paradigms

Lab 2. Recursion on lists

Outline

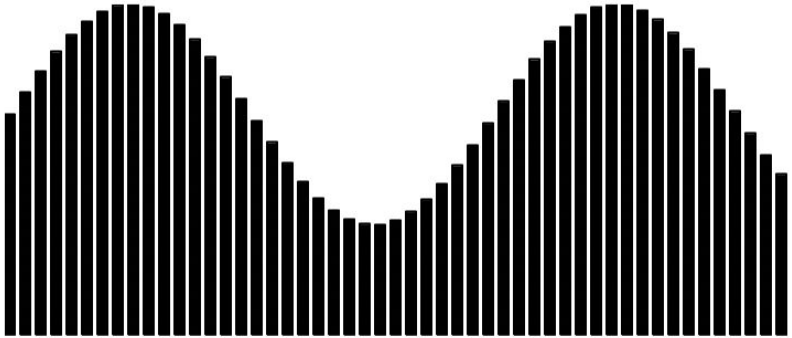
- Lists recap
- Exercise: plotting data series
- Exercise: Conway’s Game of Life

Warm-up exercise: plot data series as a histogram

Exercise 2.1.

Implement function `plot-bars` that renders a histogram given a list of values.

```
(define (ex-sin x) (abs (+ 100 (* 50 (sin x)))))

(plot-bars (map ex-sin (range 0 10 0.2)))
```

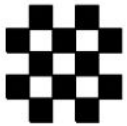


Conway’s Game of Life

1. Render universe (represented as a function)
    a. Render one cell
    b. Render one row
    c. Render 2D universe
2. Implement stepping function
    a. Find all neighbours of a cell
    b. Count alive neighbours
    c. Implement stepping function
3. Verify your implementation

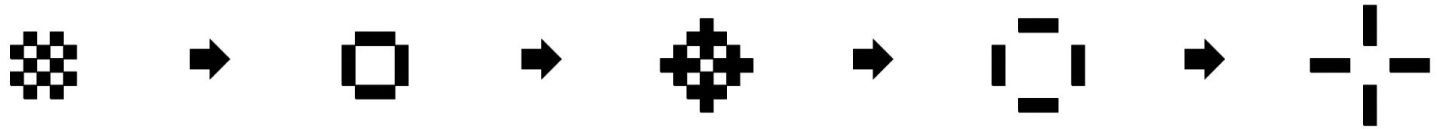
Conway's Game of Life: Example

```
(define conway-example
  (lambda (i j)
    (cond
      [(and (< (abs i) 3)
            (< (abs j) 3)
            (odd? (+ i j)))
       alive]
      [else dead])))

(render-conway conway-example)
```



Conway's Game of Life: Example

```
(define conway-example
  (lambda (i j)
    (cond
      [(and (< (abs i) 3)
            (< (abs j) 3)
            (odd? (+ i j)))
       alive]
      [else dead])))

(render-conway-steps conway-example 5)
```



Homework

1. Read **SICP 1.2 Procedures and the Processes They Generate**
2. Solve **exercises 1.11, 1.14, 1.16, 1.26** from **SICP**
3. Implement function in Racket that renders a Koch snowflake of given rank:

```
> (koch-snowflake 4)
```