

CS 135 – L11: Lists of Lists

Luke Lu • 2025-11-11

Website: student.cs.uwaterloo.ca/~cs135

Points & Distance

```
(define (mk-point x y) (list x y))
(define (get-x p) (first p))
(define (get-y p) (second p))
(define (d p0 p1)
  (sqrt (+ (sqr (- (get-x p1) (get-x p0)))
            (sqr (- (get-y p1) (get-y p0))))))
```

Polygon Perimeter (wrap once)

```
(define (add-segments/wrap p0 lop)
  (cond [(empty? lop) 0]
        [(empty? (rest lop)) (d p0 (first lop))]
        [else (+ (d (first lop) (second lop))
                  (add-segments/wrap p0 (rest lop))))])

(define (perimeter poly) (add-segments/wrap (first poly) poly))
```

Tables

```
(define (index n lst)
  (cond [(empty? lst) empty]
        [(zero? n) (first lst)]
        [else (index (sub1 n) (rest lst))]))

(define (index-table n m table)
  (index m (index n table)))
```

Subsets (non-empty)

```
(define (expand sym subsets)
  (cond [(empty? subsets) (list (list sym))]
        [else (cons (cons sym (first subsets))
                    (cons (first subsets)
                          (expand sym (rest subsets))))])))

(define (subsets set)
  (cond [(empty? set) empty]
        [else (expand (first set) (subsets (rest set))))]))
```

Practice

- Convert perimeter to a version that uses an explicit accumulator remembering the first point.
- Write rows-of that builds k identical rows, then map an index to fill them.