Lecture 4: Compound Procedures

15CSE402 :: SICP

R Sreekumar

August 3, 2020

Recap

We have seen in the last class:

- Basic elements
    - Numbers and arithmetic operations[1]
    - Nesting of combinations
    - Definitions as means of Abstractions

[1] Note that, numbers as primitive data types an operations are primitive procedures.

Procedure Definitions

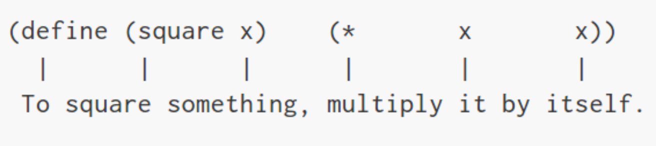
- Powerful abstraction technique
- Used to give name for compound expression

Example

"To square something we multiply it by itself."

```
(define (square x) (* x x))
```

Code



```
(define (square x)    (*        x         x))
  |        |        |        |        |         |
 To square something, multiply it by itself.
```

Compound Procedures

- Giving a name for an operation
  - squaring is an operation representing “multiplying something by itself ”.
  - The thing to be multiplied is given a local name x (method parameter).

Some of the uses

Code

```
> (square 21)
441
> (square (+ 2 5))
49
> (square (square 3))
81
```

General Form

Syntax

(define (<name> <formal parameters>) <body>)

Abstractions as building blocks

Code

```
> (define (sum-of-squares x y)
    (+ (square x) (square y)))

> (sum-of-squares 3 4)
25
```

Substitution Model

Steps

```
(sum-of_squares (+ 5 1) (* 5 2)
(+ (square 6) (square 10))
(+ 36 100)
136
```

Applicative Order

Steps

```
(sum-of-squares (+ 5 1) (* 5 2))

(+ (square (+ 5 1))
   (square (* 5 2)))

(+ (* (+ 5 1) (+ 5 1))
   (* (* 5 2) (* 5 2)))
```