Lecture 3: Basics

15CSE402 :: SICP

R Sreekumar

August 3, 2020

Introduction

We have seen:

- Introduction to the Course
- Introduction to the language (we are going to use)[1]

[1] If you didn't understand the language, need not worry, we will be repeating the same through out the course.

A Powerful Programming Language

Quote

A powerful programming language is more than just a means for instructing a computer to perform tasks. The language also serves as a framework within which we organize our ideas about processes.

Three Mechanisms

- primitive expressions, which represent simplest entities the language is concerned with,
- means of combination, by which compound elements are built from simpler ones, and
- means of abstraction, by which compound elements can be named and manipulated as units.

Basic Elements

- Data The "stuff" we want to manipulate
- Procedures Descriptions of the rules that manipulate data

Note: Later you will discover that there is no distinction between *data* and *procedure*.

Primitive Expressions

We already know from the previous class:

```
> 486
486
```

Compound Expressions

Combining expressions with arithmetic operators.

```
> (+ 137 349)
486
> (- 1000 334)
666
> (* 5 99)
495
> (/ 10 5)
2
> (+ 2.7 10)
12.7
```

Compound Expressions

- Expressions delimiting a list of expressions by parentheses
- Known as combinations.
- The left most element is *operator* or *procedure*.
- The rest of the elements are *operands* or *parameters*.
- The convention is *prefix notation*.

Evaluation Rules

To evaluate a combination:

- Evaluate the sub expressions of the combination
- Apply the procedure to the arguments (operands)
  - for us, the procedure is arithmetic operators
- the evaluated values of sub expressions become the arguments (operands) of higher expression.

Recursion

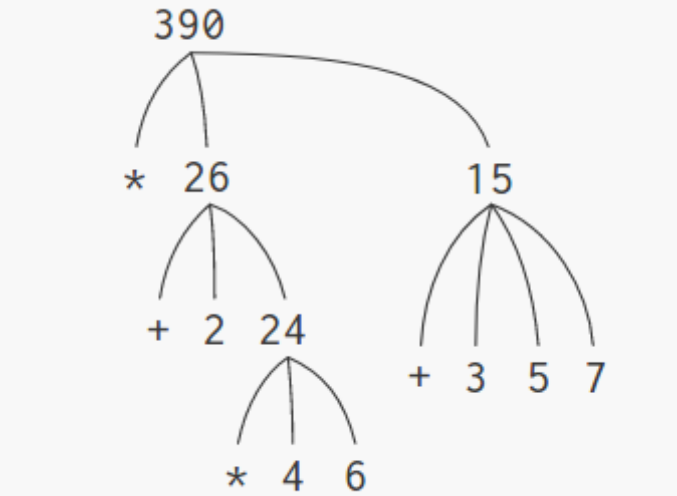
The evaluation Process is Natural Recursion

Evaluation

Code

```
(* (+ 2
      (* 4 6))
   (+ 3 5 7))
```

Tree



Naming variables

```
> (define pi 3.14159)
> (define radius 10)

> (* pi (* radius radius))
314.159

> (define circumference (* 2 pi radius))

> circumference
62.8318
```

Compound Procedures