Notes 2.1-2.9

Pair

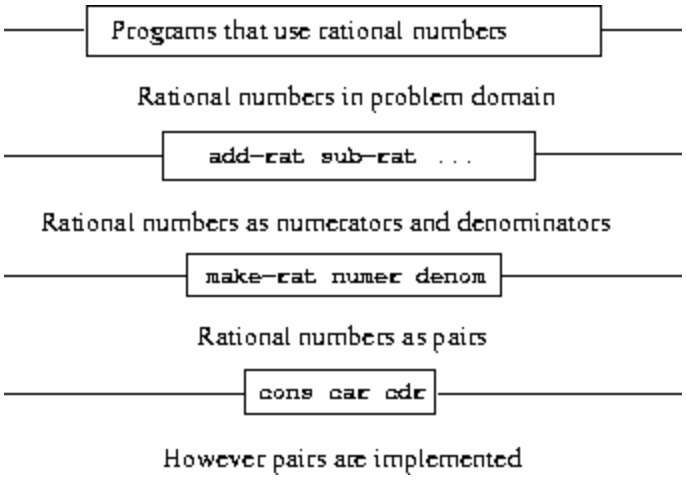
- Compound structure
- This procedure takes two arguments and returns a compound data object that contains the two arguments as parts
- implemented by the procedures cons, car, and cdr

Complete Abstraction

- Constructor
  - use components to make a compound data
- Selector (accessors)
  - go from data abstraction to component of data, give a point, get x or y
- Example → 2.1
  - Constructor: make-rat
  - Selector: numer, denom

Abstraction Barriers

- At each level, the barrier separates the programs (above) that use the data abstraction from the programs (below) that implement the data abstraction.



- in this case, the abstraction is number/denon
- instead of four numbers, we only think about two
- make-rat (constructor), numer/denom (selector)
- when using procedures (e.g. make-rat), the result is a list which is why we call print in order to make it readable

Summary

- The interface between these two parts of our system will be a set of procedures, called *selectors* and *constructors*, that implement the abstract data in terms of the concrete representation
- The hard part of data abstraction is determining the input of each function
    - For example, length must take rectangle
    - If it takes two points, it is not data abstraction