Booleans, Conditionals, and Symbolic Values
Last Lecture

- Testing: Lots of ways to do it!
  - Static analysis (including type checking)
    maybe unintuitive, but very powerful

- Contracts: For this class, if the user
  “defaults”, they’re on their own…

- Function abstraction

- Name abstraction
Today’s Goals

- Booleans (named after Bool)
- Symbolic values
  - Back-quoted literals/values
    - Can be passed around, and
    - Can be tested for equality
  - Strings (quoted values), Images, Movies, etc…
Booleans

- Booleans values ∈ \{true, false\}
- Booleans are NOT equal to any other values (like 0, 1, “true”, etc)

Examples

- \((= (+ 1 1) 2) → \text{true}\)
- \((\text{not false}) → \text{true}\)
- \((\text{or true (not true)}) → \text{true}\)
- \((\text{and true (or true (not true))}) → \text{true}\)
Conditionals

- The cond statement let us do more:

  (define (day n)
   (cond
    [(= n 1) (...lots of work 7/(n-2) ...“Monday”)]
    [(= n 2) (...lots of work ...“Tuesday”)] ...
    [(= n 7) “Sunday”]
    [else “What?”]])

- (day 2) → “Tuesday”, (day 14) → “What?”
- Careful about what is evaluated!
Symbolic Information

- “It’s all zeros and ones”
  - Convenient for the machine, but not me!
- We’ve seen: numbers, booleans, strings
- There are a few others. For example:
  - Symbols: ‘rabbit, ’Ringo, ‘submarine
    - (symbol=? ’rabbit ’bunny) → false
  - Strings: “Tommy Lee”, “MIB II”
    - (= ’rabbit “rabbit”) → error
- Mike’s point: (define rabbit ’rabbit)