Conditionals and Value Definitions

Conditional Expressions

- To express choosing between two alternatives, Scala has a conditional expression: ***if-then-else***
- It resembles an if-else in Java, but is used for expressions, not statements

Example:

def abs (x: Int) = if x >= 0 then x else -x
in this case ***x >= 0*** *is a predicate of type Boolean*
in Java there is a way to express the if as an expression: ***x >= 0 ? x : -x***

Boolean Expressions

- Boolean expressions b can be composed of:
  - constants: **true, false**
  - negation: **!b**
  - conjunction: **b && b**
  - disjunction: **b || b**
  - usual comparison operations: **e <= e, e >= e, e < e, e > e, e == e, e != e**
- Here are reduction rules for Boolean expressions:
  - !true → false
  - !false →true
  - true && e → e
  - false && e → false
  - true || e → true
  - false || e → e
- Note that && and || do not always need their right operand to be evaluated. These expressions use "short-circuit evaluation"

Exercise: *Rewrite rules for if-then-else*

- *if true then* e_1 *else* e_2 → e_1
- *if false then* e_1 *else* e_2 → e_2

Value Definitions

- Definitions can be referred by value or by name
- The ***def*** form is by-name, its right-hand side is evaluated on each use
- The ***val*** form is by-value and it is evaluated at the point of the definition itself; afterwards the name refers to the value
- The difference between ***val*** and ***def*** becomes apparent when the right-hand side does not terminate

Exercise: *Write functions* ***and*** *and* ***or*** *such that for all argument expressions x and y:*

and (x, y) == x && y, or (x, y) == x || y,

but do not use || and && in your implementation. What are good operands to test that the equalities hold?

def and (x: Boolean, y: => Boolean): Boolean = if (x) y else false

the second parameter is passed by name because ***and (false, loop)*** *must return* ***false****.*

def or (x: Boolean, y: => Boolean): Boolean = if (x) true else y

the second parameter is passed by name because ***or (true, loop)*** *must return* ***true****.*