Structural Induction on Trees

Structural Induction on Trees

- Structural induction is not limited to lists; it applies to any tree structure
- The general induction principle is the following: *To prove a property P(t) for all trees t of a certain type, show that P(l) holds for all leaves l of a tree and for each type of internal node t, with subtrees* $s_1, \dots s_n$ *show that* $P(s_1) \wedge \dots \wedge P(s_n)$ *implies P(t)*

Example: IntSets

```
abstract class IntSet {
  def incl(x: Int): IntSet
  def contains(x: Int): Boolean
}

object Empty extends IntSet {
   def contains(x: Int): Boolean = false
   def incl(x: Int): IntSet = NonEmpty(x, Empty, Empty)
}

case class NonEmpty(elem: Int, left: IntSet, right: IntSet) extends IntSet {
   def contains(x: Int): Boolean =
      if (x < elem) left contains x
      else if (x > elem) right contains x
      else true
   def incl(x: Int): IntSet =
       if (x < elem) NonEmpty(elem, left incl x, right)
       else if (x > elem) NonEmpty(elem, left, right incl x)
       else this
}
```

The Laws of IntSets

- One way to define and show the correctness of an implementation consists of proving the laws that it respects
- In the case of IntSet, we have the following three laws, for any set s and elements x and y:
  - ∗ ***Empty contains x == false***
  - ∗ ***(s incl x) contains x == true***
  - ∗ ***(s incl x) contains y == s contains y***, if x != y

Proving the Laws of IntSet

Proposition 1: *Empty contains x == false*
Proof: *According to the definition of contains in Empty*

Proposition 2: *(s incl x) contains x == true*
Proof: *by structural induction on s*

- *Base case: Empty*
  (Empty incl x) contains x == NonEmpty(x, Empty, Empty) contains x == true
- *Induction step: NonEmpty(x, l, r)*
  (NonEmpty(x, l, r) incl x) contains x == NonEmpty(x, l, r) contains x == true
- *Induction step: NonEmpty(y, l, r) where y < x*
  (NonEmpty(y, l, r) incl x) contains x == Nonempty(y, l, r incl x) contains x ==
  (r incl x) contains x == true
- *Induction step: NonEmpty(y, l, r) where y > x is analogous to the case where y < x*

Proposition 3: if x != y then *(xs incl y) contains x == xs contains x*
Proof: *by structural induction on s. Assume that y < x (the dual case for x < y is analogous)*

- *Base case: Empty*
  (Empty incl y) contains x == NonEmpty(y, Empty, Empty) contains x ==
  Empty contains x
- *Induction step: NonEmpty(x, l, r)*
  (NonEmpty(x, l, r) incl y) contains x == NonEmpty(x, l incl y, r) contains x == true
  == NonEmpty(x, l, r) contains x
- *Induction step: NonEmpty(y, l, r)*
  (NonEmpty(y, l, r) incl y) contains x == NonEmpty(y, l, r) contains x
- *Induction step: NonEmpty(z, l, r) where z < y < x*
  (NonEmpty(z, l, r) incl y) contains x == NonEmpty(z, l, r incl y) contains x ==
  (r incl y) contains x == r contains x == NonEmpty(z, l, r) contains x
- *Induction step: NonEmpty(z, l, r) where y < z < x*
  (NonEmpty(z, l, r) incl y) contains x == NonEmpty(z, l incl y, r) contains x ==
  r contains x == NonEmpty(z, l, r) contains x
- *Induction step: NonEmpty(z, l, r) where y < x < z*
  (NonEmpty(z, l, r) incl y) contains x == NonEmpty(z, l incl y, r) contains x ==
  (l incl y) contains x == l contains x == NonEmpty(z, l, r) contains x