```
ListStructureOps (A, E, C):  trait
  % Operators frequently used in
  % theorem proving demonstrations.
  assumes ListStructure
  introduces
    flatten, reverseAll: C → C
    countAtoms: C → Int
  asserts ∀ a: A, l, l1, l2: C
    flatten(empty) == empty;
    flatten(atom(a) ⊣ l) == atom(a) ⊣ flatten(l);
    flatten(list(l1) ⊣ l2) ==
      flatten(l1) || flatten(l2);
    reverseAll(empty) == empty;
    reverseAll(atom(a) ⊣ l) ==
      reverseAll(l) ⊢ atom(a);
    reverseAll(list(l1) ⊣ l2) ==
      reverseAll(l2) ⊢ list(reverseAll(l1));
    countAtoms(l) == len(flatten(l))
  implies
    ∀ l, l1, l2: C
      flatten(l1 || l2) == flatten(l1) || flatten(l2);
      flatten(flatten(l)) == flatten(l);
      reverseAll(l1 || l2) ==
        reverseAll(l2) || reverseAll(l1);
      reverseAll(flatten(l)) ==
        flatten(reverseAll(l));
      reverseAll(reverseAll(l)) == l;
      countAtoms(l1 || l2) ==
        countAtoms(l1) + countAtoms(l2);
      countAtoms(flatten(l)) == countAtoms(l);
      countAtoms(reverseAll(l)) == countAtoms(l)
    converts flatten, reverseAll, countAtoms
```