```
ReverseOp: trait
  % An operator on lists commonly used
  % to demonstrate theorem provers.
  assumes List
  introduces reverse: C → C
  asserts ∀ e: E, l, l1, l2: C
    reverse(empty) == empty;
    reverse(e ⊣ l) == reverse(l) ⊢ e
  implies ∀ e: E, l, l1, l2: C
      reverse(reverse(l)) == l;
      l ≠ empty ⇒ head(reverse(l)) = last(l);
      l ≠ empty
        ⇒ tail(reverse(l)) = reverse(init(l));
      len(reverse(l)) == len(l);
      reverse(l1 ‖ l2) == reverse(l2) ‖ reverse(l1)
    converts reverse
```