```
List (E, C): trait
  % Add singleton and concatenation
  includes Deque
  introduces
    {__}: E → C
    __ || __: C, C → C
  asserts ∀ e: E, ls, ls1, ls2: C
    {e} == empty ⊢ e;
    ls || empty == ls;
    ls1 || (ls2 ⊢ e) == (ls1 || ls2) ⊢ e
  implies
    C generated by empty, {__}, ||
    converts head, last, tail, init, len, {__}, ||
      exempting head(empty), last(empty),
        tail(empty), init(empty)
```