```
theory Diff_Arr_Constructors
  imports Diff_Arr_Rel Master_Assn
begin

(* TODO: Use Locale instead of context *)
context
begin

type_synonym 'a diff_arr = "'a cell ref"

qualified definition from_array ::
  "('a::heap) array ⇒ 'a diff_arr Heap"
where
  "from_array a = do {
    ref (Array a)
  }"

qualified definition from_list ::
  "('a::heap) list ⇒ 'a diff_arr Heap"
where
  "from_list xs = do {
    a ← Array.of_list xs;
    from_array a
  }"

lemma from_array' [sep_heap_rules]:
  "<a ↦a xs>
      from_array a
   <λr. let t = [(r, Array' xs)]
        in master_assn t * ↑(t ⊢ xs ~ r)>"

lemma from_list' [sep_heap_rules]:
  "<emp>
    from_list xs
   <λr. let t = [(r, Array' xs)]
        in  master_assn t * ↑(t ⊢ xs ~ r)>"

lemma from_array [sep_heap_rules]:
  "<a ↦a xs> from_array a <λr. ∃At. master_assn t * ↑(t ⊢ xs ~ r)>"

lemma from_list [sep_heap_rules]:
  "<emp> from_list xs <λr. ∃At. master_assn t * ↑(t ⊢ xs ~ r)>"

end

end
```