```
theory Hnr_Array
  imports Hnr Array_Safe
begin

named_theorems hnr_rule_arr

definition New_Arr :: "'a list ⇒ 'a list" where
  "New_Arr xs = xs"

lemma hnr_of_list [hnr_rule_arr]:
  "hnr
    emp
    (Array.of_list xs)
    array_assn
    (Some (New_Arr xs))"

lemma hnr_lookup [hnr_rule_arr]:
   "hnr
     (xsi ↦a xs * id_assn i ii)
     (Array_Safe.lookup xsi ii)
     (λr ri. array_assn xs xsi * id_assn r ri)
     (Some (xs ! i))"

lemma hnr_update [hnr_rule_arr]:
   "hnr
      (array_assn xs xsi * id_assn i ii * id_assn v vi)
      (Array_Safe.update ii vi xsi)
      array_assn
      (Some (xs [i:= v]))"

lemma hnr_length [hnr_rule_arr]:
   "hnr
      (array_assn xs xsi)
      (Array.len xsi)
      (λr ri. array_assn xs xsi * id_assn r ri)
      (Some (length xs))"

lemma hnr_pass_arr [hnr_rule_arr]:
  "hnr (array_assn x xi) (return xi) array_assn (Some x)"

method ent_refl = rule ent_refl

method hnr_arr = hnr ent_refl ent_refl rule_set: hnr_rule_arr

method hnr_step_arr = hnr_step ent_refl ent_refl rule_set: hnr_rule_arr

end
```