```
theory Hnr_Base
  imports Base
begin

definition hnr where
  "hnr Γ c Γ' a = (
    case a of None ⇒ True |
              Some a ⇒ <Γ> c <λr. Γ' a r>_t
  )"

lemma hnr_none [simp]: "hnr Γ c Γ' None"

named_theorems hnr_rule

lemma hnr_hoare: "(∀x. a = Some x ⟶ <Γ> c <λr. Γ' x r>_t) ⟷ (hnr Γ c Γ' a)"

lemmas hnrI = hnr_hoare[THEN iffD1, rule_format]
lemmas hnrD = hnr_hoare[THEN iffD2, rule_format]

definition id_rel where "id_rel a c ≡ c = a"

abbreviation id_assn where "id_assn a c ≡ ↑(id_rel a c)"

abbreviation array_assn where "array_assn xs xsi ≡ xsi ↦_a xs"

lemma hnr_const: "hnr Γ (return x) (λr ri. Γ * id_assn r ri) (Some x)"

lemma hnr_pass_general: "hnr (Γ x xi) (return xi) Γ (Some x)"

lemma hnr_pass: "hnr (id_assn x xi) (return xi) id_assn (Some x)"

end
```