```
theory Hnr_Rules
  imports Hnr_Base Keep_Drop Norm Merge
begin

lemma hnr_case_sum [hnr_rule]:
  assumes
    "⋀s' si'. hnr (Γ * id_assn s si * id_assn s' si') (cli si') (Γa s' si') (cl s')"
    "⋀l' li' ri r. Keep_Drop (Γa l' li' r ri) (Γa' r ri) (Dropa l' li' r ri)"
    "⋀r ri. Norm (Γa' r ri) (Γa'' r ri)"

    "⋀s' si'. hnr (Γ * id_assn s si * id_assn s' si') (cri si') (Γb s' si') (cr s')"
    "⋀r' ri' ri r. Keep_Drop (Γb r' ri' r ri) (Γb' r ri) (Dropb r' ri' r ri)"
    "⋀r ri. Norm (Γb' r ri) (Γb'' r ri)"

    "⋀r ri. Merge (Γa'' r ri) (Γb'' r ri) (Γc r ri)"
  shows
    "hnr
      (Γ * id_assn s si)
      (case si of Inl l ⇒ cli l | Inr r ⇒ cri r)
       Γc
      (case s of Inl l ⇒ cl l | Inr r ⇒ cr r)"

lemma hnr_case_nat[hnr_rule]:
  assumes
    "hnr (Γ * id_assn n ni) ci0 Γa c0"

    "⋀n' ni'. hnr (Γ * id_assn n ni * id_assn n' ni') (ci ni') (Γb n' ni') (c n')"
    "⋀n ni ri r. Keep_Drop (Γb n ni r ri) (Γb' r ri) (Drop n ni r ri)"
    "⋀r ri. Norm (Γb' r ri) (Γb'' r ri)"

    "⋀r ri. Merge (Γa r ri) (Γb'' r ri) (Γc r ri)"
  shows
    "hnr
      (Γ * id_assn n ni)
      (case ni of 0 ⇒ ci0 | Suc n' ⇒ ci n')
      Γc
      (case n of 0 ⇒ c0 | Suc n' ⇒ c n')"

lemma hnr_case_list [hnr_rule]:
  assumes
    "hnr (Γ * id_assn xs xsi) ci0 Γa c0"

    "⋀x' xi' xs' xsi'.
      hnr
        (Γ * id_assn xs xsi * id_assn x' xi' * id_assn xs' xsi')
        (ci xi' xsi')
        (Γb x' xi' xs' xsi')
        (c x' xs')"
    "⋀x xi xs xsi ri r. Keep_Drop (Γb x xi xs xsi r ri) (Γb' r ri) (Drop x xi xs xsi r ri
    "⋀r ri. Norm (Γb' r ri) (Γb'' r ri)"

    "⋀r ri. Merge (Γa r ri) (Γb'' r ri) (Γc r ri)"
  shows
    "hnr
      (Γ * id_assn xs xsi)
      (case xsi of [] ⇒ ci0 | x#xs ⇒ ci x xs)
      Γc
      (case xs of [] ⇒ c0 | x#xs ⇒ c x xs)"

end
```