

```

lemma hnr_frame:
  assumes
    " $\Gamma_P \Rightarrow_A \Gamma * F$ "
    " $\text{hnr } \Gamma \text{ fi } \Gamma' \text{ f}$ "
  shows
    " $\text{hnr } \Gamma_P \text{ fi } (\lambda r \text{ ri. } \Gamma' \text{ r ri} * F) \text{ f}$ "

attribute_setup framed =
  <Scan.succeed (Thm.rule_attribute [] (fn _ => fn thm => @{thm hnr_frame} OF [asm_rl, t
  <Add frame to hnr rule>

lemma frame_prepare:
  assumes
    " $\text{emp} * P * \text{emp} \Rightarrow_A \text{emp} * Q * F$ "
  shows
    " $P \Rightarrow_A Q * F$ "

lemma split_id_assn: "id_assn p pi = id_assn (fst p) (fst pi) * id_assn (snd p) (snd pi)"

method frame_norm_assoc =
  (simp only: mult.left_assoc[where 'a=assn] split_id_assn fst_conv snd_conv)?

method frame_prepare = rule frame_prepare, frame_norm_assoc

lemma frame_no_match:
  assumes
    " $P_{s1} * (P * P_{s2}) \Rightarrow_A Q_s * Q * F$ "
  shows
    " $P_{s1} * P * P_{s2} \Rightarrow_A Q_s * Q * F$ "

lemma frame_match_pure:
  assumes
    " $P_{s1} * \uparrow(P) * P_{s2} \Rightarrow_A Q_s * F$ "
  shows
    " $P_{s1} * \uparrow(P) * P_{s2} \Rightarrow_A Q_s * \uparrow(P) * F$ "

lemma frame_match:
  assumes
    " $P \Rightarrow_A Q$ "
    " $P_{s1} * P_{s2} \Rightarrow_A Q_s * F$ "
  shows
    " $P_{s1} * P * P_{s2} \Rightarrow_A Q_s * Q * F$ "

lemma frame_match_emp:
  assumes
    " $P_s \Rightarrow_A Q_s * F$ "
  shows
    " $P_s \Rightarrow_A Q_s * \text{emp} * F$ "

lemma frame_done: " $F * \text{emp} \Rightarrow_A \text{emp} * F$ "

method frame_try_match methods match_atom = then_else
  <rule frame_match_pure | rule frame_match, (match_atom; fail) | rule frame_match_emp>
  <frame_norm_assoc>
  <rule frame_no_match, frame_try_match match_atom>

method frame_done = simp only: assn_one_left mult_1_right[where 'a=assn], rule ent_refl

method hnr_frame_inference methods match_atom =
  frame_prepare, (frame_try_match match_atom)+, frame_done

```

end