```
theory Hnr
  imports Hnr_Rules Hnr_Fallback Hnr_Frame Hnr_Recursion
begin

lemma let_to_bind: "(let x = v in f x) = (do { x ← Some v; f x })"

method hnr_rule methods frame_match_atom uses rule_set =
  (rule rule_set[framed] hnr_pass[framed], hnr_frame_inference frame_match_atom)
  | rule hnr_rule hnr_const

method hnr_step methods frame_match_atom keep_atom uses rule_set normalization_rules =
     simp only: let_to_bind split_def
   | hnr_rule frame_match_atom rule_set: rule_set
   | keep_drop keep_atom
   | normalize rules: normalization_rules
   | merge rules: normalization_rules
   | partial_function_mono
   | hnr_fallback
   | hnr_solve_recursive_call frame_match_atom

method hnr methods frame_match_atom keep_atom uses rule_set normalization_rules =
  (hnr_step frame_match_atom keep_atom
      rule_set: rule_set normalization_rules: normalization_rules)+

end
```