Kernel Analysis - OperatorKernelO6

File: `OperatorKernelO6/OperatorKernelO6\Kernel.lean`
Author: Moses
Generated: 2025-08-05 01:07:40
File Size: 1393 characters

Overview

Core trace definitions and reduction rules

Source Code

```
namespace OperatorKernelO6

inductive Trace : Type
| void : Trace
| delta : Trace → Trace
| integrate : Trace → Trace
| merge : Trace → Trace → Trace
| recΔ : Trace → Trace → Trace → Trace
| eqW : Trace → Trace → Trace

open Trace

inductive Step : Trace → Trace → Prop
| R_int_delta : ∀ t, Step (integrate (delta t)) void
| R_merge_void_left : ∀ t, Step (merge void t) t
| R_merge_void_right : ∀ t, Step (merge t void) t
| R_merge_cancel : ∀ t, Step (merge t t) t
| R_rec_zero : ∀ b s, Step (recΔ b s void) b
| R_rec_succ : ∀ b s n, Step (recΔ b s (delta n)) (merge s (recΔ b s n))
| R_eq_refl : ∀ a, Step (eqW a a) void
| R_eq_diff : ∀ {a b}, a ≠ b → Step (eqW a b) (integrate (merge a b))


inductive StepStar : Trace → Trace → Prop
| refl : ∀ t, StepStar t t
| tail : ∀ {a b c}, Step a b → StepStar b c → StepStar a c

def NormalForm (t : Trace) : Prop := ¬ ∃ u, Step t u

theorem stepstar_trans {a b c : Trace} (h1 : StepStar a b) (h2 : StepStar b c) : StepStar a c := by
  induction h1 with
  | refl => exact h2
  | tail hab _ ih => exact StepStar.tail hab (ih h2)

theorem stepstar_of_step {a b : Trace} (h : Step a b) : StepStar a b :=
  StepStar.tail h (StepStar.refl b)

theorem nf_no_stepstar_forward {a b : Trace} (hnf : NormalForm a) (h : StepStar a b) : a = b :=
  match h with
  | StepStar.refl _ => rfl
  | StepStar.tail hs _ => False.elim (hnf ⟨_, hs⟩)


end OperatorKernelO6
```