

Equality

File: C:\Users\Moses\math_ops\OperatorKernel06\OperatorKernel06\Meta\Equality.lean
Type: lean
Generated: 2025-08-05 03:41:08
Size: 1999 characters

Overview

Equality definitions and theorems

Source Code

```

import OperatorKernel06.Kernel
import OperatorKernel06.Meta.Meta

open OperatorKernel06.Trace

namespace OperatorKernel06.Meta

-- Equality predicate using eqW
def eq_trace (a b : Trace) : Trace := eqW a b

-- Equality reflection: if eqW a b reduces to void, then a and b are equal
theorem eq_refl_reduces (a : Trace) : StepStar (eq_trace a a) void := by
  unfold eq_trace
  apply stepstar_of_step
  apply Step.R_eq_refl

-- Inequality witness: if  $a \neq b$ , then eqW a b reduces to integrate (merge a b)
theorem eq_diff_reduces (a b : Trace) (h : a  $\neq$  b) :
  StepStar (eq_trace a b) (integrate (merge a b)) := by
  unfold eq_trace
  apply stepstar_of_step
  apply Step.R_eq_diff h

-- -- Equality is decidable in normal forms
-- def eq_decidable (a b : Trace) (ha : NormalForm a) (hb : NormalForm b) :
--   (a = b)  $\vee$  (a  $\neq$  b) := by
--   classical
--   exact Classical.em (a = b)

-- -- Equality properties
-- theorem eq_symm (a b : Trace) :
--    $\exists$  c, StepStar (eq_trace a b) c  $\wedge$  StepStar (eq_trace b a) c := by
--   cases Classical.em (a = b) with
--   | inl h =>
--     rw [h]
--     use void
--     constructor <=> apply eq_refl_reduces
--   | inr h =>
--     use integrate (merge a b)
--     constructor
--     · apply eq_diff_reduces h
--     · rw [merge_comm] at *; apply eq_diff_reduces h.symm
--   where
--     merge_comm : merge a b = merge b a := by sorry -- Needs confluence

-- theorem eq_trans (a b c : Trace) :
--    $\exists$  d e f, StepStar (eq_trace a b) d  $\wedge$ 
--     StepStar (eq_trace b c) e  $\wedge$ 
--     StepStar (eq_trace a c) f := by
--   sorry -- Complex, requires case analysis and confluence

-- -- Equality substitution in contexts
-- def subst_context (ctx : Trace  $\rightarrow$  Trace) (a b : Trace) : Trace :=
--   integrate (merge (ctx a) (ctx b))

-- theorem eq_substitution (a b : Trace) (ctx : Trace  $\rightarrow$  Trace) :
--   StepStar (eq_trace a b) void  $\rightarrow$ 
--    $\exists$  d, StepStar (subst_context ctx a b) d := by
--   sorry -- Requires careful analysis of context structure

end OperatorKernel06.Meta

```