

```

theorem funext {f1 f2 : ∀ (x : α), β x} (h : ∀ x, f1 x = f2 x) : f1 = f2 := by
  show extfunApp (Quotient.mk' f1) = extfunApp (Quotient.mk' f2)
  apply congrArg
  apply Quotient.sound
  exact h

def horn_filling_condition (X : SSet) (n i : Nat): Prop :=
  ∀ f : Δ[n, i] → X, ∃ g : Δ[n] → X,
  f = SSet.hornInclusion n i » g

/-- A simplicial set is called an ∞-category if it has the extension property
for all inner horn inclusions 'Δ[n, i] → Δ[n]', n ≥ 2, 0 < i < n. -/
def InfCategory := {X : SSet //
  ∀ (n i : Nat), n ≥ 2 ∧ 0 < i ∧ i < n → horn_filling_condition X n i}

```