GSS rigidity when either algebra has an elementary model

`MathlibAnnex.Analysis.CStarAlgebra.nonempty_starAlgEquiv_iff_nonempty_geometricStructure_homeomorph_of_exists_compactOperatorEquiv`

theorem

Obtains algebraic rigidity from an elementary compact-operator model on either side.

Statement

Let A, B be complex C*-algebras. For a complex C*-algebra D, write $S(D)$ for its geometric structure space. Its points are the unions of kernels of norm-one bounded complex-linear functionals supporting a maximal proper face of the closed unit ball of D; faces use real convexity. Write c_D for its hull–kernel closure:

$$c_D(E) = \left\{ I \in S(D) : \bigcap_{J \in E} J \subseteq I \right\}, \qquad E \subseteq S(D).$$

These are the geometric structure definition and its closure, with the norm-one description justified by the norm-one support characterization.

Let $\mathcal{K}(H)$ denote the algebra of compact operators on a complex Hilbert space H. Assume that at least one of A, B is $*$-isomorphic to $\mathcal{K}(H)$ for some such H. Then A and B are $*$-isomorphic if and only if $S(A)$ and $S(B)$ are homeomorphic as closure spaces. Here a closure-space homeomorphism is a bijection $e : S(A) \to S(B)$ satisfying

$$e(c_A(E)) = c_B(e(E)), \qquad E \subseteq S(A).$$

The equality is required for every subset, not just finite subsets.

Assumptions

The compact-operator model is required on at least one side. It is an isomorphism onto the whole algebra $\mathcal{K}(H)$, not merely a representation with compact values. The Hilbert space is complex and complete; it and the algebras may be zero. No unitality, separability, countability, or finite-dimensionality hypothesis is imposed.

Conclusion

The equivalence concerns existence of a $*$-algebra isomorphism and existence of a closure-space homeomorphism. It does not require the isomorphism to induce a prescribed homeomorphism, or assert uniqueness.

Main citations

The either-side elementary-model theorem uses rigidity for a specified compact-operator equivalence. The given-action compact rigidity theorem applies to the representation on that same model space, including the zero case.

Lean source signature (exact)

Exact source and declaration. Only the literal signature is displayed here; the linked source retains the proof.

```
theorem
nonempty_starAlgEquiv_iff_nonempty_geometricStructure_homeomorph_of_exists_compactOperatorEquiv
    (hModel :
      (∃ (H : Type w) (normed : NormedAddCommGroup H),
        letI : NormedAddCommGroup H := normed
        ∃ inner : InnerProductSpace ℂ H,
          letI : InnerProductSpace ℂ H := inner
          ∃ complete : CompleteSpace H,
            letI : CompleteSpace H := complete
            Nonempty (A ≃⋆ₐ[ℂ] ContinuousLinearMap.compactOperatorStarSubalgebra ℂ H)) ∨
      (∃ (K : Type z) (normed : NormedAddCommGroup K),
        letI : NormedAddCommGroup K := normed
        ∃ inner : InnerProductSpace ℂ K,
          letI : InnerProductSpace ℂ K := inner
          ∃ complete : CompleteSpace K,
            letI : CompleteSpace K := complete
            Nonempty (B ≃⋆ₐ[ℂ] ContinuousLinearMap.compactOperatorStarSubalgebra ℂ K))) :
    Nonempty (A ≃⋆ₐ[ℂ] B) ↔
      Nonempty (ClosureSpace.Homeomorph (NormedSpace.geometricStructureClosure ℂ A)
        (NormedSpace.geometricStructureClosure ℂ B))
```

In the source	Mathematical meaning

Two C*-algebras
`variable {A : Type u} {B : Type v}`
`variable [NonUnitalCStarAlgebra A]`
`[NonUnitalCStarAlgebra B]`
	The inputs A, B are arbitrary complex C*-algebras, including zero. Their size levels are independent. The source type allows unital algebras but does not require a unit.

Normed model on the A side
`(∃ (H : Type w) (normed : NormedAddCommGroup H),`
`letI : NormedAddCommGroup H := normed`
	This alternative requires a normed additive group, called `H` in the source and H in the Statement. The first `letI` installs this chosen normed structure on the same space for the inner-product type that follows.

Complex inner product on the A model
`∃ inner : InnerProductSpace ℂ H,`
`letI : InnerProductSpace ℂ H := inner`
	The same model must have a compatible complex inner product. This `letI` installs that inner product, so subsequent Hilbert-space operator types use this structure, not a separately chosen one.

In the source	Mathematical meaning

Complete A model and isomorphism
`∃ complete : CompleteSpace H,`
`  letI : CompleteSpace H := complete`
`  Nonempty (A ≃⋆ₐ[ℂ] ContinuousLinearMap.compactOperatorStarSubalgebra ℂ H)) ∨`
	The model must be complete; the last `letI` installs this completeness for its operator algebra. `Nonempty` requires a $*$-isomorphism $A \to \mathcal{K}(H)$ onto the entire operator algebra. The two model branches are alternatives; their spaces are quantified separately and may be zero.

Normed model on the B side
`(∃ (K : Type z) (normed : NormedAddCommGroup K),`
`  letI : NormedAddCommGroup K := normed`
	This alternative requires a normed additive group, called `K` in the source and H in the Statement. The first `letI` installs this chosen normed structure on the same space for the inner-product type that follows.

Complex inner product on the B model
`∃ inner : InnerProductSpace ℂ K,`
`  letI : InnerProductSpace ℂ K := inner`
	The same model must have a compatible complex inner product. This `letI` installs that inner product, so subsequent Hilbert-space operator types use this structure, not a separately chosen one.

Complete B model and isomorphism
`∃ complete : CompleteSpace K,`
`  letI : CompleteSpace K := complete`
`  Nonempty (B ≃⋆ₐ[ℂ] ContinuousLinearMap.compactOperatorStarSubalgebra ℂ K))) :`
	The model must be complete; the last `letI` installs this completeness for its operator algebra. `Nonempty` requires a $*$-isomorphism $B \to \mathcal{K}(H)$ onto the entire operator algebra. The two model branches are alternatives; their spaces are quantified separately and may be zero.

Existence of a star-isomorphism `Nonempty (A ≃⋆ₐ[ℂ] B) ↔`	`Nonempty` says that the type of $*$-algebra isomorphisms $A \to B$ has an element. The outer equivalence relates this existence to the existence in the next row, not to a choice of a prescribed map.

Existence of a closure-space homeomorphism
`Nonempty (ClosureSpace.Homeomorph (NormedSpace.geometricStructureClosure ℂ A)`
`  (NormedSpace.geometricStructureClosure ℂ B)) :=`
`by`
	The complete arguments are c_A on $S(A)$ and c_B on $S(B)$. `Nonempty` requires one homeomorphism $e : S(A) \to S(B)$ between these closure spaces.

The same map and inverse
`structure Homeomorph (c : ClosureSpace α) (d : ClosureSpace β) extends α ≃ β where`
`  continuous_toFun : c.Continuous d toFun`
`  continuous_invFun : d.Continuous c invFun`
	The inherited equivalence supplies the bijection e and its inverse e^{-1}. The first field requires, for every $E \subseteq S(A)$,
	$$e(c_A(E)) \subseteq c_B(e(E)).$$
	The second requires, for every $F \subseteq S(B)$,
	$$e^{-1}(c_B(F)) \subseteq c_A(e^{-1}(F)).$$
	Both concern the same e and together express the closure equality in the Statement.

In the source	Mathematical meaning

Continuity for arbitrary subsets
`def Continuous (c : ClosureSpace α) (d : ClosureSpace β) (f : α → β) : Prop :=`
`  ∀ s : Set α, f '' c.closure s ⊆ d.closure (f '' s)`
	For a map $r : (X, c) \to (Y, d)$, continuity means
	$$r(c(E)) \subseteq d(r(E)) \quad (E \subseteq X).$$
	In the preceding row this definition is applied first to $r = e$ and then to $r = e^{-1}$, with the domain and target closures interchanged.

Exact surrounding source context (separate excerpts)

Exact surrounding source:

```
universe u v w z

variable {A : Type u} {B : Type v}
variable [NonUnitalCStarAlgebra A] [NonUnitalCStarAlgebra B]
```

The model spaces in the two alternatives have the fixed independent source levels `w` and `z`; the algebra levels are `u` and `v`. The source names for those model spaces are local to their respective existential branches. No additional ambient Hilbert space is an input to this equivalence.

Related exact source for the reading guide

Existence of a closure-space homeomorphism · Fixed source

```
      Nonempty (ClosureSpace.Homeomorph (NormedSpace.geometricStructureClosure ℂ A)
        (NormedSpace.geometricStructureClosure ℂ B)) := by
```

Continuity for arbitrary subsets · Fixed source

```
def Continuous (c : ClosureSpace α) (d : ClosureSpace β) (f : α → β) : Prop :=
  ∀ s : Set α, f '' c.closure s ⊆ d.closure (f '' s)
```

The same map and inverse · Fixed source

```
structure Homeomorph (c : ClosureSpace α) (d : ClosureSpace β) extends α ≃ β where
  continuous_toFun : c.Continuous d toFun
  continuous_invFun : d.Continuous c invFun
```