Existence of a faithful compact representation

`MathlibAnnex.Analysis.CStarAlgebra.NonUnitalCStarRepresentation.exists_isCompactlyRepresented_iff`

theorem

Characterizes existence of a faithful compact action by compact irreducible images and discrete primitive spectrum.

Statement

Let A be a complex C*-algebra. For a complex Hilbert space L, let $\mathcal{K}(L)$ be the algebra of compact operators on L. Write $\mathrm{Prim}(A)$ for the primitive ideal spectrum: its points are the kernels of pure-state GNS representations. Its hull–kernel topology has closure

$$c_{\mathrm{Prim}(A)}(E) = \left\{ P \in \mathrm{Prim}(A) : \bigcap_{Q \in E} Q \subseteq P \right\}, \qquad E \subseteq \mathrm{Prim}(A).$$

The following conditions are equivalent:

- There is a complex Hilbert space H and a faithful $*$-representation $\pi : A \to \mathcal{B}(H)$ such that $\pi(A) \subseteq \mathcal{K}(H)$.
- Every irreducible $*$-representation $\rho : A \to \mathcal{B}(K)$ on a complex Hilbert space satisfies $\rho(A) = \mathcal{K}(K)$, and $\mathrm{Prim}(A)$ is discrete: every singleton is open.

Assumptions

Hilbert spaces are complex and complete. Representations are $*$-homomorphisms, without a requirement of unitality; irreducibility includes a nonzero action. Zero algebras, zero representation spaces, and empty primitive spectra are allowed. No separability or countability assumption is made.

Conclusion

The two conditions characterize the existence of a faithful representation by compact operators. The cited reduced-atomic characterization gives such a representation when they hold. This does not require an arbitrary faithful representation to have compact values, or require its image to be all compact operators on its Hilbert space.

Main citations

The faithful compact-representation criterion is the stated equivalence. The fixed reduced-atomic characterization separates compactness of all irreducible images from discreteness of the existing spectrum. The existential representation comparison gives the same canonical reduced-atomic

witness; transport from an action on an independent Hilbert space does not shrink the original space.

Lean source signature (exact)

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

```
theorem exists_isCompactlyRepresented_iff :
    (∃ (K : Type u) (normed : NormedAddCommGroup K),
      letI : NormedAddCommGroup K := normed
      ∃ inner : InnerProductSpace ℂ K,
        letI : InnerProductSpace ℂ K := inner
        ∃ complete : CompleteSpace K,
          letI : CompleteSpace K := complete
          ∃ pi : NonUnitalCStarRepresentation A K, pi.IsCompactlyRepresented) ↔
    (∀ (K : Type u) [NormedAddCommGroup K] [InnerProductSpace ℂ K]
      [CompleteSpace K] (rho : NonUnitalCStarRepresentation A K),
      rho.IsIrreducible →
        (NonUnitalStarAlgHom.range rho) = ContinuousLinearMap.compactOperatorStarSubalgebra ℂ
K) ∧
    DiscreteTopology (PrimitiveSpectrum A)
```

The principal equivalence fixes the universe of A and quantifies its Hilbert spaces in that same universe. The independently specified Hilbert space of the separately cited transport theorem is not an all-universe quantifier in this equivalence.

In the source	Mathematical meaning
Algebra and positivity `attribute [local instance] _root_.CStarAlgebra.spectralOrder _root_.CStarAlgebra.spectralOrderedRing`	The C*-algebra A uses its spectral order to define positive functionals and pure states. This is the order underlying the primitive spectrum in the theorem.
Ambient algebra `variable {A : Type u} [NonUnitalCStarAlgebra A]`	The input is an arbitrary complex C*-algebra A, including the zero algebra. The type does not require a unit, and does not require the algebra to lack a unit.
Representation space and norm `(∃ (K : Type u) (normed : NormedAddCommGroup K), letI : NormedAddCommGroup K := normed`	There exists a space, called `K` here and H in the first condition, with a normed additive group structure. The `letI` makes that same norm available when specifying its inner product.
Inner product and completeness `∃ inner : InnerProductSpace ℂ K, letI : InnerProductSpace ℂ K := inner ∃ complete : CompleteSpace K, letI : CompleteSpace K := complete`	The existential quantifiers require a compatible complex inner product and completeness on the same H. The first `letI` installs the chosen inner product; the second installs completeness. Together they make H a complex Hilbert space on which the next representation is defined.

In the source	Mathematical meaning
Faithful compact representation `∃ pi : NonUnitalCStarRepresentation A K, pi.IsCompactlyRepresented) ↔`	There exists a $*$-representation $\pi : A \to \mathcal{B}(H)$ that is faithful and satisfies $\pi(A) \subseteq \mathcal{K}(H)$. This is the left side of the equivalence, not an equality of the two operator algebras.
Meaning of compact representability `def NonUnitalCStarRepresentation.IsCompactlyRepresented (pi : NonUnitalCStarRepresentation A H) : Prop := Function.Injective pi ∧ ∀ a : A, IsCompactOperator (pi a)`	The first conjunct is injectivity of the same π, that is, faithfulness. The second says that $\pi(a)$ is compact for every $a \in A$. Together they express exactly the first condition in the Statement.
All irreducible representations `(∀ (K : Type u) [NormedAddCommGroup K] [InnerProductSpace ℂ K] [CompleteSpace K] (rho : NonUnitalCStarRepresentation A K), rho.IsIrreducible → (NonUnitalStarAlgHom.range rho) = ContinuousLinearMap.compactOperatorStarSubalgebra ℂ K) ∧`	For every complex Hilbert space K and every irreducible $\rho : A \to \mathcal{B}(K)$, the operator-algebra image is $$\rho(A) = \mathcal{K}(K).$$ Here `range rho` and `compactOperatorStarSubalgebra` are subalgebras of the same $\mathcal{B}(K)$. This is stronger than merely having compact values.

Primitive points
`def primitiveKernelFamily : Set (Set A) := Set.range fun phi : NonUnitalPureState A => (kernelIdeal phi.gnsRepresentation : Set A)`
 	
`/-- Actual primitive ideal carriers, not geometric primitive-spectrum points. -/ def PrimitiveSpectrum := {P : Set A // P ∈ primitiveKernelFamily A}`
	The points $P \in \mathrm{Prim}(A)$ are kernels of pure-state GNS representations, regarded as subsets of A. The subtype records membership in precisely this family; it is not the geometric structure space $S(A)$.

Hull–kernel topology `def primitiveSpectrumClosure : ClosureSpace (PrimitiveSpectrum A) := ClosureSpace.HullKernel.space (primitiveKernelFamily A) (univ_notMem_primitiveKernelFamily A)`	The first argument is the family of underlying sets of primitive ideals of A. The second is the established fact that A itself, written `Set.univ`, is not a member of this family; it is the required input for constructing the hull–kernel closure on that family. For every $E \subseteq \mathrm{Prim}(A)$, $$c_{\mathrm{Prim}(A)}(E) = \left\{ P \in \mathrm{Prim}(A) : \bigcap_{Q \in E} Q \subseteq P \right\}.$$
Discreteness `DiscreteTopology (PrimitiveSpectrum A) :=`	The second conjunct says that $\mathrm{Prim}(A)$ has the discrete topology. It accompanies the assertion about every irreducible image; it is not a substitute for that assertion. The empty spectrum is allowed.

Exact surrounding source context (separate excerpts)

Exact surrounding source:

```
attribute [local instance] _root_.CStarAlgebra.spectralOrder
  _root_.CStarAlgebra.spectralOrderedRing
```

Exact surrounding source:

```
universe u v

variable {A : Type u} [NonUnitalCStarAlgebra A]
variable {H : Type v} [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H]
```

Related exact source for the reading guide

Meaning of compact representability · Fixed source

```
def NonUnitalCStarRepresentation.IsCompactlyRepresented
    (pi : NonUnitalCStarRepresentation A H) : Prop :=
  Function.Injective pi ∧ ∀ a : A, IsCompactOperator (pi a)
```

Discreteness · Fixed source

```
    DiscreteTopology (PrimitiveSpectrum A) :=
```

Primitive points · Fixed source

```
def primitiveKernelFamily : Set (Set A) :=
  Set.range fun phi : NonUnitalPureState A => (kernelIdeal phi.gnsRepresentation : Set A)

/-- Actual primitive ideal carriers, not geometric primitive-spectrum points. -/
def PrimitiveSpectrum := {P : Set A // P ∈ primitiveKernelFamily A}
```

Hull–kernel topology · Fixed source

```
def primitiveSpectrumClosure : ClosureSpace (PrimitiveSpectrum A) :=
  ClosureSpace.HullKernel.space (primitiveKernelFamily A) (univ_notMem_primitiveKernelFamily A)
```