Unions of kernels of supporting functionals

`NormedSpace.supportKernelUnion`

def

Unions of kernels of supporting functionals

Statement

For a subset of a seminormed space, consider the contractive continuous linear functionals that equal 1 throughout that subset. The support-kernel union consists of the vectors annihilated by at least one of these functionals. It is a union of kernels, with no assertion that the union is a vector subspace.

Definition

Let E be a seminormed space over a real-or-complex scalar field $\mathbb{K}$, and let E^* be its continuous $\mathbb{K}$-linear dual with the operator norm. For $M \subseteq E$, put

$$\Sigma_{\mathbb{K}}(M) = \{f \in E^* : \|f\| \le 1,\ f|_M = 1\},$$

where $f|_M = 1$ means $f(x) = 1$ for every $x \in M$. Define

$$\begin{aligned} I_{\mathbb{K}}(M) &= \bigcup_{f \in \Sigma_{\mathbb{K}}(M)} \ker f \\ &= \{y \in E : \exists f \in \Sigma_{\mathbb{K}}(M),\ f(y) = 0\}. \end{aligned}$$

Thus membership requires one witnessing functional. It does not require all supporting functionals to vanish. For nonempty M, the condition $f|_M = 1$ is equivalent to $f(M) = \{1\}$. The pointwise formulation also covers $M = \varnothing$.

The cited elementary identities make the empty cases explicit:

$$\begin{aligned} \Sigma_{\mathbb{K}}(M) = \varnothing &\Longrightarrow I_{\mathbb{K}}(M) = \varnothing, \\ 0 \in I_{\mathbb{K}}(M) &\Longleftrightarrow \Sigma_{\mathbb{K}}(M) \neq \varnothing, \\ I_{\mathbb{K}}(\varnothing) &= E. \end{aligned}$$

The last equality follows because the zero functional belongs to $\Sigma_{\mathbb{K}}(\varnothing)$ and has kernel E.

Assumptions

The space E may have a seminorm: definiteness and completeness are not required. The subset M is arbitrary. It need not be nonempty, convex or contained in the unit ball, and the definition does not assume that a supporting functional exists.

Conclusion

The output is the subset $I_{\mathbb{K}}(M)$ of E. For each $x \in M$, a supporting functional has $f(x) = 1$, so it cannot also annihilate x. Consequently

$$M \cap I_{\mathbb{K}}(M) = \varnothing, \qquad M \neq \varnothing \Longrightarrow I_{\mathbb{K}}(M) \neq E.$$

This properness result, proved in the cited source, is what later permits the hull–kernel construction for maximal faces.

Main citations

- The supporting family and its nonempty image characterization: `NormedSpace.supportFunctionals`, lines 42–43; the equivalence for nonempty subsets is at lines 59–78.
- Exact union and union-of-kernels formula: `NormedSpace.supportKernelUnion`, lines 108–122.
- Disjointness, properness and the empty cases: lines 130–161.
- The real-or-complex scalar interface: Mathlib `RCLike`, at the pinned mathlib commit.
- arXiv:2504.05551v1, Definition 4.1: the support-kernel union for maximal faces of a nontrivial Banach space. The present declaration allows arbitrary subsets of seminormed spaces.

Lean source signature (exact)

The exact definition uses the scalar and space declarations reproduced separately below.

```
def supportKernelUnion (F : Set E) : Set E :=
  {x | ∃ f ∈ supportFunctionals 𝕜 F, f x = 0}
```

In the source	Mathematical meaning
Exact declaration `𝕜, [RCLike 𝕜]`	The scalar field $\mathbb{K}$ is real or complex.
Exact declaration `E, [SeminormedAddCommGroup E], [NormedSpace 𝕜 E]`	E is a seminormed vector space over $\mathbb{K}$; completeness and a nonzero vector are not assumed.
Exact declaration `F : Set E`	The arbitrary subset $M \subseteq E$, possibly empty.

In the source	Mathematical meaning
Exact declaration `supportFunctionals 𝕜 F`	$\Sigma_{\mathbb{K}}(M) = \{f \in E^* : \|f\| \le 1,\ f(x) = 1 \text{ for every } x \in M\}$, where E^* is the continuous linear dual (`StrongDual 𝕜 E`).
Exact declaration `{x | ∃ f ∈ supportFunctionals 𝕜 F, f x = 0}`	The full defining subset $I_{\mathbb{K}}(M) = \{x \in E : \exists f \in \Sigma_{\mathbb{K}}(M),\ f(x) = 0\}$. For each vector x, one functional must satisfy both supporting conditions and annihilate x.
Exact declaration `: Set E`	The output is a subset of E, the union of these kernels. It is not bundled as a linear subspace, and the supporting family need not be nonempty.

Source context (separate exact excerpt)

The surrounding declarations below are from lines 26–27; their mathematical meaning is included in the guide above.

```
variable (𝕜 : Type v) [RCLike 𝕜]
variable {E : Type u} [SeminormedAddCommGroup E] [NormedSpace 𝕜 E]
```

Exact declaration for the source guide

```
def supportKernelUnion (F : Set E) : Set E :=
  {x | ∃ f ∈ supportFunctionals 𝕜 F, f x = 0}

@[simp]
```

RCLike · fixed Mathlib definition

Fixed source, lines 60–84 · Raw UTF-8 source

```
class RCLike (K : semiOutParam Type*) extends DenselyNormedField K, StarRing K,
    NormedAlgebra ℝ K, CompleteSpace K where
  /-- The real part as an additive monoid homomorphism -/
  re : K →+ ℝ
  /-- The imaginary part as an additive monoid homomorphism -/
  im : K →+ ℝ
  /-- Imaginary unit in `K`. Meant to be set to `0` for `K = ℝ`. -/
  I : K
  I_re_ax : re I = 0
  I_mul_I_ax : I = 0 ∨ I * I = -1
  re_add_im_ax : ∀ z : K, 𝓚 (re z) + 𝓚 (im z) * I = z
  ofReal_re_ax : ∀ r : ℝ, re (𝓚 r) = r
  ofReal_im_ax : ∀ r : ℝ, im (𝓚 r) = 0
  mul_re_ax : ∀ z w : K, re (z * w) = re z * re w - im z * im w
  mul_im_ax : ∀ z w : K, im (z * w) = re z * im w + im z * re w
  conj_re_ax : ∀ z : K, re (conj z) = re z
  conj_im_ax : ∀ z : K, im (conj z) = -im z
  conj_I_ax : conj I = -I
  norm_sq_eq_def_ax : ∀ z : K, ‖z‖ ^ 2 = re z * re z + im z * im z
  mul_im_I_ax : ∀ z : K, im z * im I = im z
  /-- only an instance in the `ComplexOrder` scope -/
  [toPartialOrder : PartialOrder K]
  le_iff_re_im {z w : K} : z ≤ w ↔ re z ≤ re w ∧ im z = im w
  -- note we cannot put this in the `extends` clause
  [toDecidableEq : DecidableEq K]
```