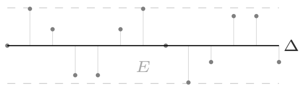
COARSE SPACES

commentary on the formalization

01.1 /

THE COCONTROLLED FILTER



controlled sets ⊆ tubes around Δ

coarse space axioms

A coarse space is a generalization of a metric space that retains only the large-scale structure. A metric assigns distances to pairs of points, so the abstraction axiomatizes which sets of pairs represent "bounded distance" directly: these are the controlled sets in a coarse space. Controlled sets are required to form a set-theoretic ideal: they are closed under subsets and finite unions. The remaining axioms lift the metric axioms to this setting: the diagonal is controlled (reflexivity), controlled sets are closed under swapping coordinates (symmetry), and controlled sets compose (the triangle inequality). Where a uniformity refines these axioms inward to capture small-scale structure, a coarse space applies them outward.

Note that since we wish to allow disconnected spaces, or in other words, extended metrics, we do not require controlled sets to form a bornology, and opt for the weaker ideal axioms to obtain some notion of "bounded distance".

Since the cocontrolled sets i.e complements of controlled sets, form a filter, and Mathlib has a mature filter API, we state the axioms in these dual terms. In particular, they directly mirror the uniformity axioms.

cocontrolled	`Filter (α × α)`
refl	Diagonal is controlled. Stated as ≤𝓟 rather than membership to stay within the filter API.
symm	Controlled sets are symmetric.
comp	Controlled sets are closed under composition. The weaker condition, namely closure under self-composition suffices to derive arbitrary composition (See. `IsControlled.comp`). Taking complements noting that $(R \circ S)^c = R \setminus S^c$, gives the filter statement.

```
ForMathlib.Data.Rel

/-- The residual `R \ S` where
    `x ~[R \ S] z` iff every
    `R`-successor of `x` is an
    `S`-predecessor of `z`. -/
def res (R : SetRel α β)
    (S : SetRel β γ) : SetRel α γ :=
  {p | ∀ y, p.1 ~[R] y → y ~[S] p.2}

theorem compl_comp : (R ○ S)ᶜ = R \ Sᶜ := ...
```

```
CoarseSpace.Defs

class CoarseSpace (α : Type u) where
  /-- The filter of cocontrolled sets
      in a coarse space. -/
  protected cocontrolled :
    Filter (α × α)
  /-- The complement of the diagonal
      is cocontrolled. -/
  protected refl :
    cocontrolled ≤ 𝓟 (SetRel.id)ᶜ
  /-- If `s ∈ cocontrolled`, then
      `Prod.swap ⁻¹' s ∈ cocontrolled`. -/
  protected symm :
    Tendsto Prod.swap
      cocontrolled cocontrolled
  /-- Composition: if `sᶜ` and `tᶜ` are
      cocontrolled, so is `(sᶜ ○ tᶜ)ᶜ`.
      Stated dually using
      the residual. -/
  protected comp :
    cocontrolled ≤
      cocontrolled.lift' (fun s ↦ sᶜ \ s)

def IsCocontrolled (s : SetRel α α) : Prop :=
  s ∈ CoarseSpace.cocontrolled

def IsControlled (s : SetRel α α) : Prop :=
   IsCocontrolled sᶜ
```

Notation

$\mathcal{C}$ cocontrolled filter

$\mathcal{C}$`[c]`

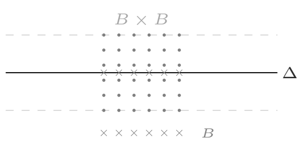
`IsCocontrolled[c]`

`IsControlled[c]`

`IsCoarselyBounded[c]`

Bracketed variants for non-default coarse structures

Scoped under `Coarse`.



coarsely bounded ⇔ $B \times B$ is controlled

Apart from controlled sets, the other fundamental notion is that of bounded sets. A set s : `Set α` is *bounded* if $s \times^s s$ is controlled. We define this as `IsCoarselyBounded` to avoid confusion with `Bornology.IsBounded`, this is open to change if we wish to namespace.

```
def IsCoarselyBounded (s : Set α) : Prop :=
  IsControlled (s ×ˢ s)
```

controlled sets

recovering ideal-based axioms

empty	`IsControlled ∅` `@[simp]`	`isControlled_empty`
subset	`IsControlled t → s ⊆ t → IsControlled s`	`IsControlled.subset`
union	`IsControlled s → IsControlled t → IsControlled (s ∪ t)`	`IsControlled.union`

lifted metric axioms

id	`IsControlled SetRel.id` `@[simp]`	`isControlled_id`
inv	`IsControlled s → IsControlled s.inv`	`IsControlled.inv`
comp	`IsControlled s → IsControlled t → IsControlled (s ○ t)` Since s, t are controlled, $s^c \cap t^c \in \mathcal{C}$. The comp axiom gives $(s \cup t) \setminus (s^c \cap t^c) \in \mathcal{C}$. By `compl_comp` $(s \cup t) \setminus (s \cup t)^c = ((s \cup t) \circ (s \cup t))^c$. Then $s \circ t \subseteq (s \cup t) \circ (s \cup t)$ by subset on each factor.	`IsControlled.comp`

basic results

iff	`IsControlled s ↔ IsCocontrolled sᶜ`	`isControlled_iff`
iterate	`IsControlled s → IsControlled t → IsControlled ((s ○ ·)^[n] t)`	`IsControlled.iterate_comp`

coarsely bounded sets

iff	`IsCoarselyBounded s ↔ IsControlled (s ×ˢ s)`	`isCoarselyBounded_iff`
singleton	`IsCoarselyBounded {x}` `@[simp]`	`isCoarselyBounded_singleton`
empty	`IsCoarselyBounded ∅` `@[simp]`	`isCoarselyBounded_empty`

subset	`IsCoarselyBounded s → t ⊆ s → IsCoarselyBounded t` `IsCoarselyBounded.subset`

union′	`IsCoarselyBounded s → IsCoarselyBounded t → (s ∩ t).Nonempty → IsCoarselyBounded (s ∪ t)`
	Non-disjoint unions only, since in a disconnected space bounded components need not have bounded union. Two points can be infinitely far apart.
	`IsCoarselyBounded.union'`

ball	`IsControlled R → IsCoarselyBounded (R.ball x₀)`
	`ForMathlib.Data.Rel`
	`def ball : Set α := {a | a ~[R] b}`
	The ball about b w.r.t. a relation is the set of a related to b.
	See `mathlib4#33077`.
	`IsControlled.isCoarselyBounded_ball`

01.3 /

CONSTRUCTORS AND INSTANCES

constructors

We provide two constructors, one to make a coarse space out of a filter basis satisfying coarse space axioms and another to make a coarse space from a set of relations satisfying the ideal version of the coarse space axioms.

`CoarseSpace.Defs`

```
def CoarseSpace.mkOfBasis
    {α : Type u} (B : FilterBasis (α × α))
    (refl :
      ∀ r ∈ B, r ⊆ (SetRel.id)ᶜ)
    (symm :
      ∀ r ∈ B, ∃ t ∈ B, t ⊆ Prod.swap ⁻¹' r)
    (comp :
      ∀ r ∈ B, ∃ t ∈ B, t ⊆ rᶜ \ r)
    : CoarseSpace α

def CoarseSpace.ofControlled
    {α : Type*} (C : Set (SetRel α α))
    (subset_mem :
      ∀ s₁ ∈ C, ∀ s₂ ⊆ s₁, s₂ ∈ C)
    (union_mem :
      ∀ s₁ ∈ C, ∀ s₂ ∈ C, s₁ ∪ s₂ ∈ C)
    (refl_mem : SetRel.id ∈ C)
    (symm_mem :
      ∀ s ∈ C, Prod.swap ⁻¹' s ∈ C)
    (comp_mem : ∀ s ∈ C, s ○ s ∈ C)
    : CoarseSpace α
```

`ofControlled` basic API

cocontrolled iff	`IsCocontrolled s ↔ sᶜ ∈ C` `isCocontrolled_ofControlled_iff`
controlled iff	`IsControlled s ↔ s ∈ C` `isControlled_ofControlled_iff`

instances

Currently, we only register one type of instance: the coarse space structure associated to an extended metric space. We define it viz `ofControlled` and we set the controlled sets to be exactly subsets of tubes in the product space.

`CoarseSpace.Metric.Basic`

```
instance (α : Type*) [PseudoEMetricSpace α]
: CoarseSpace α :=
  CoarseSpace.ofControlled
    { s | ∃ r : ℝ≥0, ∀ p ∈ s, edist p.1 p.2 ≤ r }
    ...
```

metric characterization

controlled iff	`IsControlled s ↔ ∃ r : ℝ≥0, ∀ p ∈ s, edist p.1 p.2 ≤ r`
	`@[simp]` Extended pseudo metric
	`Metric.isControlled_iff_bounded_edist`

controlled iff	`IsControlled s ↔ ∃ r : ℝ, ∀ p ∈ s, dist p.1 p.2 ≤ r`
	`@[simp]` Pseudo metric
	`Metric.isControlled_iff_bounded_dist`

relation to bornologically bounded sets in pseudo metric spaces

bounded	`IsCoarselyBounded s → Bornology.IsBounded s`	`IsCoarselyBounded.isBounded`
coarsely bounded	`Bornology.IsBounded s → IsCoarselyBounded s`	`Bornology.IsBounded.isCoarselyBounded`
iff	`IsCoarselyBounded s ↔ Bornology.IsBounded s` `@[simp]`. The two notions coincide in a pseudo-metric space.	`isCoarselyBounded_iff_isBounded`

02.1 /

COARSE MAPS

close maps

In the standard paradigm of coarse metric geometry (e.g in geometric group theory), one works with quasi-isometries. Recall that if $f : X \to Y$ is a quasi-isometry, one can form the "inverse" quasi-isometry $g : Y \to X$ which inverts f only up to bounded error i.e $d(g(f(x)), x) \leq C$ and $d(f(g(y)), y) \leq C$ for some fixed C. Closeness qualitatively abstracts this notion: we talk about inverses up to closeness.

```
CoarseSpace.Defs

def IsClose [CoarseSpace  β] (f g : α → β) : Prop :=
  IsControlled ◁ .map f g '' SetRel.id
```

Notation

`=ᶜ`   `IsClose`

Scoped under `open Coarse`.

`f =ᶜ g`: bounded deviation

One can, in fact, talk about closeness of arbitrary functions into coarse spaces: two functions are close if they deviate by a bounded amount. Stated more precisely $\{(f(x), g(x)) \mid x : \alpha\}$ is a controlled set. Closeness of functions forms an equivalence relation and is the default notion of equivalence of maps in the coarse setting.

equivalence relation

refl	`f =ᶜ f` `@[refl]`	`IsClose.refl`
symm	`f =ᶜ g → g =ᶜ f` `@[symm]`	`IsClose.symm`
trans	`f =ᶜ g → g =ᶜ h → f =ᶜ h` `@[trans]`	`IsClose.trans`

basic API

comp	`f =ᶜ g → f ∘ h =ᶜ g ∘ h`	`IsClose.comp`
lcomp	`f =ᶜ g → IsControlledMap h → h ∘ f =ᶜ h ∘ g` See controlled maps.	`IsClose.lcomp`

coarse maps

Coarse maps are the morphisms in the category of coarse spaces. They combine two conditions. It is a controlled map: the map doesn't send nearby things far apart, and it is a coarsely proper map: the map doesn't collapse unbounded regions to bounded ones. Without properness, a constant map would be controlled e.g a single point is bounded, but all large-scale structure is destroyed.

CoarseSpace.Defs

```
@[fun_prop]
structure Coarse (f : α → β) : Prop where
  controlled : IsControlledMap f
  proper : IsCoarselyProperMap f
```

basic API

id

```
Coarse id
@[fun_prop]
```

Coarse.id

comp

```
Coarse g → Coarse f → Coarse (g ∘ f)
@[fun_prop]
```

Coarse.comp

02.1 /

COARSE MAPS

See **[Roe03]**

bornologous

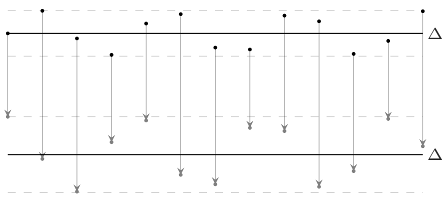
functions

controlled maps

Controlled maps are those maps which preserve controlled sets. In particular, they directly abstract the notion of coarsely Lipschitz maps. We define it filter-theoretically following the pattern established by `Bornology.Hom`'s `LocallyBoundedMap` (except we keep our version as a `def` not as a `structure`) and we derive the standard definition from the filter based one.

CoarseSpace.Defs

```
def IsControlledMap (f : α → β) : Prop :=
  (𝓒 : Filter (β × β)).comap (.map f f) ≤ 𝓒
```



basic API

iff

```
IsControlledMap f ↔ ∀ s, IsControlled s →
IsControlled (Prod.map f f '' s)
```

isControlledMap_iff

id

```
IsControlledMap id
@[fun_prop]
```

isControlledMap_id

comp

```
IsControlledMap g → IsControlledMap f → IsControlledMap (g ∘ f)
@[fun_prop]
```

IsControlledMap.comp

bounded image

```
IsControlledMap f → IsCoarselyBounded s →
IsCoarselyBounded (f '' s)
```

`IsControlledMap.isCoarselyBounded_image`

of close

```
f =ᶜ g → IsControlledMap g → IsControlledMap f
```

Transfer along closeness. The image of s under $f \times f$ is sandwiched: $(f \times f)''s \subseteq \{(f,g)\} \circ (g \times g)''s \circ \{(g,f)\}$. All three are controlled, so the composition is controlled. Uses `mem_comp_comp` in `ForMathlib.Data.Rel`.

`IsClose.isControlledMap`

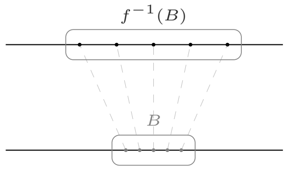
relation to closeness

lcomp

```
f =ᶜ g → IsControlledMap h → h ∘ f =ᶜ h ∘ g
```

h is controlled, so it maps controlled sets forward. The closeness set $\{(f,g)\}$ is controlled by hypothesis. Its image under $h \times h$ is $\{(h \circ f, h \circ g)\}$, which is exactly the closeness set for $h \circ f$ and $h \circ g$.

`IsClose.lcomp`



preimages of bounded sets are bounded

coarsely proper maps

`CoarseSpace.Defs`

```
def IsCoarselyProperMap (f : α → β) : Prop :=
  ∀ s : Set β, IsCoarselyBounded s →
  IsCoarselyBounded (f⁻¹' s)
```

Coarsely proper maps are defined as one would expect: pre-images of coarsely bounded sets are coarsely bounded.

basic API

id

```
IsCoarselyProperMap id
```

`isCoarselyProperMap_id`

comp

```
IsCoarselyProperMap g → IsCoarselyProperMap f →
IsCoarselyProperMap (g ∘ f)
```

`IsCoarselyProperMap.comp`

relation to closeness

of close comp id

```
IsControlledMap f → f ∘ g =ᶜ id → IsCoarselyProperMap g
```

If f is a controlled left inverse of g up to closeness, then g is coarsely proper. Given B bounded, $g^{-1}(B) \subseteq \{(\mathrm{id}, fg)\} \circ f''(B) \circ \{(fg, \mathrm{id})\}$. The outer sets are controlled (closeness of fg to id) and the middle is bounded (controlled maps preserve boundedness).

`IsControlledMap.isCoarselyProperMap_of_isClose_comp_id`

02.2 /

COARSE EQUIVALENCES

Notation

≃ᶜ coarse equivalence

coarse equivalences

Coarse equivalences abstract the standard notion of quasi-isometry used in geometric group theory. Recall that quasi isometries satisfy a coarse bilipschitz property, and those are abstracted to controlled maps f, g that are inverses up to closeness. In the standard presentation of coarse equivalences, the maps f and g are required to be coarse, but one can obtain coarse properness via `isCoarselyProperMap_of_isClose_comp_id` in `CoarseSpace.Basic`, hence requiring only controlled maps suffice.

CoarseSpace.CoarseEquiv.Defs

```
/-- Coarse equivalence between `α` and `β`. -/
structure CoarseEquiv
  (α β : Type*)
  [CoarseSpace α]
  [CoarseSpace β] where
  toFun : α → β
  invFun : β → α
  controlled_toFun : IsControlledMap toFun :=
    by fun_prop
  controlled_invFun : IsControlledMap invFun :=
    by fun_prop
  isClose_id_right : toFun ∘ invFun =ᶜ id
  isClose_id_left : invFun ∘ toFun =ᶜ id
```

coercion and extensionality

coercion `CoeFun (α ≃ᶜ β) (fun _ ↦ α → β)`

Write `e x` for `e.toFun x`.

coe forward `(e.toFun : α → β) = e`

`@[simp]`

`CoarseEquiv.coe_toFun`

coe inverse `(e.invFun : β → α) = e.symm`

`@[simp]`

`CoarseEquiv.coe_invFun`

ext `(∀ x, e₁ x = e₂ x) → (∀ y, e₁.invFun y = e₂.invFun y) → e₁ = e₂`

`@[ext]`. Extensionality on both directions.

`CoarseEquiv.ext`

groupoid structure

refl `CoarseEquiv.refl α : α ≃ᶜ α`

`@[simps! -fullyApplied toFun invFun]`

`CoarseEquiv.refl`

symm `e.symm : β ≃ᶜ α`

`@[symm]`

`CoarseEquiv.symm`

trans `e₁.trans e₂ : α ≃ᶜ γ`

`@[trans]`

`CoarseEquiv.trans`

simp lemmas

double inverse	`e.symm.symm = e` `@[simp]`	`CoarseEquiv.symm_symm`
apply comp	`(e₁.trans e₂) x = e₂ (e₁ x)` `@[simp]`	`CoarseEquiv.trans_apply`
inverse comp	`(e₁.trans e₂).symm = e₂.symm.trans e₁.symm` `@[simp]`	`CoarseEquiv.symm_trans`

coarseness of components

coarse	`Coarse e.toFun` `@[fun_prop]`. Properness derived via `isCoarselyProperMap_of_isClose_comp_id` from the inverse.	`CoarseEquiv.coarse`
coarse inverse	`Coarse e.symm.toFun` `@[fun_prop]`	`CoarseEquiv.coarse_symm`
controlled	`IsControlledMap e.toFun` `@[fun_prop]`. Unwrapped from the structure field for `fun_prop`.	`CoarseEquiv.controlled_toFun'`
controlled inverse	`IsControlledMap e.invFun` `@[fun_prop]`	`CoarseEquiv.controlled_invFun'`

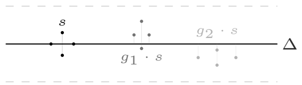
constructor

of equiv	`(e : α ≃ β) → IsControlledMap e → IsControlledMap e.symm → α ≃ᶜ β` Lift an `Equiv` to a coarse equivalence. Closeness is trivial since true inverses are close to id by refl.	`CoarseEquiv.ofEquiv`

03.1 /

UNIFORMLY CONTROLLED ACTIONS

See **[BDM08]** actions by uniformly bornologous functions



$\bigcup_g (g, g) \cdot s$ controlled

generalized isometric actions

In order to state the Švarc-Milnor lemma in the coarse space setting, we need to abstract the notion of a group acting by isometries to work without a metric. We do this by using the notion of uniformly controlled actions.

Recall that when a group G acts by isometries we have that $d(x, y) \leq r$ iff $d(g \cdot x, g \cdot y) \leq r$ for all $g : G$. In the coarse setting, we can state this as translating a controlled set by a group element preserves controlled-ness in a way that works for all group elements.

Concretely, we say G acts on a coarse space α in a uniformly controlled fashion if $\cup_{g:G}(g, g) \cdot s$ is controlled whenever s is controlled.

Since we work with the cocontrolled filter, the axiom is stated dually via intersection and preimages rather than images: preimages commute with complements, keeping the dual formulation clean. We express this using `lift'` as a single filter inequality and recover the standard notion as a theorem. Phrasing the axiom via preimages rather than images means the definition extends to arbitrary monoid actions without requiring invertibility; in the group setting the two formulations coincide.

`CoarseSpace.Algebra.UniformlyControlledSMul`

```
/-- An action of `Γ` on a coarse
space `α` is *uniformly controlled*
if the cocontrolled filter is below by
 its lift along the intersection
 of all translates... -/
class UniformlyControlledSMul
  (Γ : Type*) (α : Type*)
  [CoarseSpace α] [SMul Γ α] : Prop where
  uniformly_controlled_smul :
  𝓒 ≤  (𝓒 : Filter (α × α)).lift'
        (fun s ↦ ⋂ γ : Γ, .map (γ • ·) (γ • ·) ⁻¹' s)
```

Additive version is `UniformlyControlledVAdd`.
Exports: `uniformly_controlled_smul`

basic API

controlled preimage

`IsControlled s → IsControlled (⋃ γ, (γ • ·, γ • ·) ⁻¹' s)`

`@[to_additive]`. Direct consequence of the filter inequality.

`isControlled_iUnion_preimage_smul`

controlled preimage iff

`(∀ s, IsControlled s → IsControlled (⋃ γ, (γ • ·, γ • ·) ⁻¹' s)) ↔ UniformlyControlledSMul Γ α`

`@[to_additive]`. Equivalent statement in terms of controlled sets.

`isControlled_iUnion_preimage_smul_iff`

controlled image

`IsCoarselyBounded t → 1 ∈ t → (∀ (a,b) ∈ s, f(a)⁻¹ * f(b) ∈ t) → IsControlled (.map f f '' s)`

`@[to_additive]`. If s-related pairs have differences landing in a bounded neighborhood of the identity t, then $f \times f$ maps s into a controlled set. Since t is bounded, $t \times^s t$ is controlled, so $\bigcup_g (g, g)^{-1}(t \times^s t)$ is controlled by `isControlled_iUnion_preimage_smul`. We show $(f \times f)''s$ is a subset: given $(a, b) \in s$, left-translate $(f(a), f(b))$ by $f(a)^{-1}$ to get $(1, f(a)^{-1}f(b))$. This is in $t \times^s t$ by the hypotheses.

`isControlled_image_of_inv_mul_mem`

instances

isometric

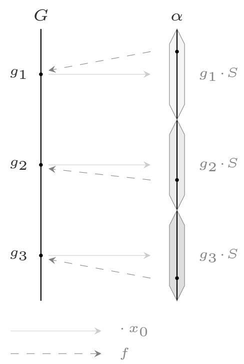
`[IsIsometricSMul G α] → UniformlyControlledSMul G α`

`@[to_additive]`. Isometric actions on (extended pseudo) metric spaces are uniformly controlled: translating a tube doesn't change distances. Anonymous instance.

Švarc–Milnor lemma

The Švarc–Milnor lemma is the fundamental observation in geometric group theory: a group acting properly discontinuously and cocompactly by isometries on a proper metric space is quasi-isometric to it, once the group is equipped with the word metric. In particular, the orbit map $g \mapsto gx_0$, where x_0 is any chosen base point, is a quasi-isometry. The group's large-scale geometry is entirely determined by any space it acts on nicely.

Despite the conceptual simplicity of the standard proof, it requires formally developing the machinery of quasi-isometries, word metrics, Cayley graphs, and so on; the bookkeeping quickly becomes unwieldy to formalize. Conveniently, the coarse version requires much less infrastructure, and the standard metric statement can be recovered as a corollary (left for a future PR). We have already generalized isometric actions; all that remains is to generalize cocompactness.



cocompact actions

The standard alternative characterization of cocompactness says there is a closed ball S such that $G \cdot S = X$. We view S as a coarsely bounded set in a coarse space α and assume the chosen base point satisfies $x_0 \in S$. To avoid choice, we assume a function $f : \alpha \to G$ is given explicitly such that $x \in f(x) \cdot S$ for all $x : \alpha$. In the proof, f acts as the quasi-inverse to the orbit map. In the metric specialization, we use the standard characterization `[CompactSpace (MulAction.orbitRel.Quotient G α)]`.

coarse Švarc–Milnor

Let G be a group endowed with a coarse space structure, and suppose G acts on itself in a uniformly controlled fashion (generalizing possession of a left-invariant metric). Let $x_0 : \alpha$, and suppose further that G acts on a coarse space α in a uniformly controlled fashion and that the orbit map $g \mapsto g \cdot x_0$ is a coarse map. Then G is coarsely equivalent to α.

closeness proofs

Requires: `IsCoarselyBounded S`   `x₀ ∈ S`   `∀ x, x ∈ f x • S`

---

right

`[UniformlyControlledSMul G α]`

`((· • x₀) ∘ f) =ᶜ id`

`@[to_additive]` The closeness set $\{(f(a)\cdot x_0,\ a) \mid a : \alpha\}$ must be shown controlled. We show it is a subset of $\bigcup_g (g,g)^{-1}(S \times^s S)$. Let $(f(a)\cdot x_0, a)$ be in the closeness set, then $f(a)^{-1} \cdot (f(a) \cdot x_0) = x_0 \in S$, the second gives $f(a)^{-1} \cdot a \in S$ by the covering hypothesis. So every pair in the closeness set belongs to $\bigcup_g (g,g)^{-1}(S \times^s S)$, which is controlled since S is bounded so $S \times^s S$ is controlled and the action is uniformly controlled.

`private smul_comp_isClose_id`

---

left

`[UniformlyControlledSMul G G]`   `IsCoarselyProperMap (· • x₀)`

`(f ∘ (· • x₀)) =ᶜ id`

`@[to_additive]` Our goal is to show $\{(f(g \cdot x_0), g) \mid g : G\}$ is controlled. Let $T = (\cdot\, x_0)^{-1}(S)$, this is bounded in G by properness of the orbit map, so T is the G-side analogue of S. Furthermore $1 \in T$ since $1 \cdot x_0 = x_0 \in S$. We will show that the closeness set is contained in $\bigcup_\gamma (\gamma,\gamma)^{-1}(T \times^s T)$. Let $(f(g \cdot x_0), g)$ be in the closeness set, then $f(g \cdot x_0)^{-1} \cdot f(g \cdot x_0) = 1 \in T$ and furthermore for $f(g \cdot x_0)^{-1} \cdot g \in T$, we require $(f(g \cdot x_0)^{-1} \cdot g) \cdot x_0 \in S$, equivalently, $g \cdot x_0 \in f(g \cdot x_0) \cdot S$, which we obtain by the covering hypothesis.

`private comp_smul_isClose_id`

controlled map

```
[UniformlyControlledSMul G G]   [UniformlyControlledSMul G α]
IsCoarselyProperMap (· • x₀)
```

```
((· • x₀) ∘ f) =ᶜ id → IsControlledMap f
```

@[to_additive] We use the equivalent characterization of controlled maps isControlledMap_iff. Let U be controlled in α. Now we will try to apply isControlled_image_of_inv_mul_mem: it suffices to find a bounded neighbourhood of 1 in G such that $f(a)^{-1}f(b)$ lands in it for every $(a, b) \in U$. To this end, let $C = \{(f(a) \cdot x_0, a)\}$ be the closeness set from the hypothesis. Since $f(a) \cdot x_0 \sim_C a \sim_U b \sim_{C^{-1}} f(b) \cdot x_0$, the tile centers are within controlled distance $T = C \circ U \circ C^{-1}$. Translating both by $f(a)^{-1}$: the first goes to x_0, the second to $(f(a)^{-1}f(b)) \cdot x_0$. By uniform controlledness this pair lives in $V = \bigcup_g (g, g)^{-1}T$, still controlled. Since V is controlled, the ball $B = (V \cup \mathrm{id})^{-1}$.ball x_0 is bounded (union with id ensures $x_0 \in B$), and $(f(a)^{-1}f(b)) \cdot x_0 \in B$ since it is V-related to x_0. Pulling back through the orbit map, $f(a)^{-1}f(b)$ lies in $(\cdot\, x_0)^{-1}(B)$, a bounded neighbourhood of 1 in G.

`private isControlledMap_of_smul_comp_isClose_id`

coarse equivalence

```
[UniformlyControlledSMul G G]   [UniformlyControlledSMul G α]
IsCoarselyBounded S     x₀ ∈ S     ∀ x, x ∈ f x • S
Coarse (· • x₀)
```

```
G ≃ᶜ α
```

@[to_additive] Assembles the coarse equivalence. Forward map is the orbit map $g \mapsto g \cdot x_0$, inverse is f. Controlledness of the orbit map is a hypothesis; controlledness of f is derived from the right closeness proof via isControlledMap_of_smul_comp_isClose_id. Both closeness fields are supplied by smul_comp_isClose_id and comp_smul_isClose_id.

`mulOrbitCoarseEquiv`

03.3 /

METRIC SPECIALIZATION

coarse metric case

We specialize to the coarse metric setting as follows: Suppose G is a group, endowed with a left-invariant metric. Suppose it is a proper metric space and possesses the discrete topology. Suppose further that G acts, by isometries, on a proper metric space α properly discontinuously and cocompactly, then G with its metric coarse space structure is coarsely equivalent to α with its metric coarse space structure. In particular, for any chosen base point x_0, the orbit map $g \mapsto g \cdot x_0$ is a coarse equivalence.

cocompactness

Standalone result: See ForMathlib.Topology.MetricSpace.IsometricSMul

Requires: `[Group G] [PseudoMetricSpace X] [IsIsometricSMul G X]` `[CompactSpace (orbitRel.Quotient G X)]`

bounded range

```
BddAbove (range (fun x ↦ infDist x (orbit G x₀)))
```

@[to_additive]. Since infDist to the orbit is constant on orbits (by isometry), it descends to a continuous function on the quotient. By compactness of the quotient, this function attains its supremum.

`bddAbove_range_infDist_mulAction_orbit`

orbit proximity

```
∃ g : G, dist x (g • x₀) < 1 + sSup (range (fun x ↦ infDist x
(orbit G x₀)))
```

@[to_additive]. The infimum distance from any point to the orbit is at most the supremum, hence strictly less than $1 + R$. By definition of infDist, some orbit point witnesses this.

`dist_smul_lt_one_add_sSup_range_infDist_orbit`

covering `∃ ε, (univ : Set G) • closedBall x₀ ε = univ`

`@[to_additive]`. Combines the orbit proximity lemma with isometry to place every point inside $g \cdot \overline{B}(x_0, \varepsilon)$ for some g, giving the standard covering characterization of co-compactness in terms of a closed ball.

`exists_smul_closedBall_eq_univ`

orbit map properties

Requires: `[PseudoMetricSpace G]` `[PseudoMetricSpace α]`

controlled `[ProperSpace G]` `[IsIsometricSMul G G]` `[IsIsometricSMul G α]` `Continuous (· • x₀)`

`IsControlledMap (· • x₀ : G → α)`

`@[to_additive]`. We verify the image characterization `isControlledMap_iff`. Given a controlled set $s \subseteq G$, by `Metric.isControlled_iff_bounded_dist` it suffices to produce a uniform distance bound on the image. Let r be the distance bound on s, and let $K = (\cdot\, x_0)''(\overline{B}(1, \max(r, 0)))$. This is a continuous image of a compact ball (recall in proper spaces, closed balls are compact), so K is compact.

Let C be the maximum distance between points in K. We need to show dist$(g \cdot x_0,\ h \cdot x_0) \leq C$ for all $(g, h) \in s$. Applying isometry of the G-action on α reduces this to dist$(x_0,\ (g^{-1}h) \cdot x_0) \leq C$, which holds since both points lie in K: $x_0 \in K$ because $1 \in \overline{B}(1, \max(r, 0))$ and $1 \cdot x_0 = x_0$, and $(g^{-1}h) \cdot x_0 \in K$ because $g^{-1}h \in \overline{B}(1, \max(r, 0))$—this last membership follows from dist$(g^{-1}h,\ 1) =$ dist$(h, g) \leq r$ by the left-invariant metric on G and the bound on s.

`isControlledMap_smul`

coarsely proper `[ProperSpace α]` `[ProperlyDiscontinuousSMul G α]`

`IsCoarselyProperMap (· • x₀ : G → α)`

`@[to_additive]`. The preimage of a bounded set s under the orbit map is contained in $\{g \mid g \cdot x_0 \in s\}$. Since $s \subseteq \overline{B}(x_0, R)$ for some R, this set is finite by `ProperlyDiscontinuousSMul`: only finitely many g satisfy $g \cdot \{x_0\} \cap \overline{B}(x_0, R) \neq \emptyset$. Finite sets are bounded.

`isCoarselyProperMap_smul`

metric equivalence

coarse equivalence `[ProperSpace G]` `[DiscreteTopology G]` `[IsIsometricSMul G G]` `[ProperSpace α]` `[IsIsometricSMul G α]` `[ProperlyDiscontinuousSMul G α]` `[CompactSpace (orbitRel.Quotient G α)]`

`G ≃ᶜ α`

`@[to_additive]`. Assembles `mulOrbitCoarseEquiv` in the metric setting. We build the choice function f by obtaining the covering radius ε from `exists_smul_closedBall_eq_univ` and, for each x, choosing g such that $x \in g \cdot \overline{B}(x_0, \max(\varepsilon, 0))$. The coarsely bounded set is $\overline{B}(x_0, \max(\varepsilon, 0))$, which is coarsely bounded since closed balls are metrically bounded; x_0 lies in it since dist$(x_0, x_0) = 0$. The orbit map bundle is provided by `isControlledMap_smul` (using `continuous_of_discreteTopology` to get continuous orbit map) and `isCoarselyProperMap_smul`.

REFERENCES

[BDM08] N. Brodskiy, J. Dydak, A. Mitra. *Coarse structures and group actions.* Colloq. Math. **111** (2008), 149–158. arXiv:math/0607568

[Roe03] J. Roe. *Lectures on coarse geometry.* Univ. Lecture Ser. **31**, Amer. Math. Soc., 2003. ams.org/ulect-31