Geometric Group Theory: Exercises

Prof. Dr. C. Löh/M. Uschold — Sheet 7, June 7, 2022

Quick check A (compositions of quasi-isometric embeddings). Show that compositions of quasi-isometric embeddings are quasi-isometric embeddings.

Quick check B (quasi-isometry types of finitely generated groups).

1. Is $\mathbb{Z}/2$ quasi-isometric to D_{2022} ?
2. Is $\mathbb{Z}/2$ quasi-isometric to D_∞ ?

Quick check C (free products and bilipschitz equivalence). Let G_1, G_2, H_1, H_2 be finitely generated groups, let G_1 be bilipschitz equivalent to H_1, and let G_2 be bilipschitz equivalent to H_2. Does this imply that $G_1 * G_2$ is bilipschitz equivalent to $H_1 * H_2$? Prove your claim!

Exercise 1 (free products and quasi-isometry; 4 credits). Let G_1, G_2, H_1, H_2 be finitely generated groups and let $G_1 \sim_{\mathrm{QI}} H_1$ and $G_2 \sim_{\mathrm{QI}} H_2$. Does this imply that $G_1 * G_2 \sim_{\mathrm{QI}} H_1 * H_2$? Prove your claim!

Exercise 2 (bijective quasi-isometries; 8 credits).

1. Let G and H be finitely generated groups and let $f \colon G \longrightarrow H$ be a bijective quasi-isometry. Prove that f is a bilipschitz equivalence.
2. Does the analogous statement also hold for general metric spaces instead of finitely generated groups? Prove your claim!

Exercise 3 (finite distance between maps; 4 credits). Let X and Y be metric spaces and let $f, g \colon X \longrightarrow Y$ be quasi-isometric embeddings. Show that the following are equivalent:

1. The maps f and g have finite distance to each other.
2. There exists a quasi-isometric embedding $h \colon X \times [0,1] \longrightarrow Y$ with
$$h(\,\cdot\,, 0) = f \quad \text{and} \quad h(\,\cdot\,, 1) = g.$$
Here, $X \times [0,1]$ carries the maximum metric of the given metric on X and the standard metric on $[0,1]$.

Bonus problem (isometries are quasi-isometries; 4 credits). Formalise the following definitions/statements/proofs in Lean:

1. definition: isometries of metric spaces;
2. lemma: every isometry is a quasi-isometry;
3. proof of this lemma.

Hints. Please submit a .lean source file. You can use the template quasiisometry_exercise.lean and try out your code in the Lean web interface: https://leanprover.github.io/live/latest/ For the proof: Split the proof into small steps and first write up a detailed(!) pen-and-paper proof. Abstraction helps to avoid doing the same thing multiple times.

Submission before June 14, 2022, 8:30, via GRIPS (in English or German)

The Quick checks are not to be submitted and will not be graded; they will be solved and discussed in the exercise class on June 13, 2022.

```
/-
Copyright (c) 2022 Clara L"oh. All rights reserved.
Released under Apache 2.0 license as described in the file LICENSE.txt.
Author: Clara L"oh.
-/

import tactic                       -- standard proof tactics
import topology.metric_space.basic -- basics on metric spaces

open classical  -- we work in classical logic

/-
We define quasi-isometries as quasi-isometric embeddings
that admit a quasi-inverse quasi-isometric embedding.

Similarly, we define isometries between metric spaces
and show that isometries are quasi-isometries.
-/

/-
# Quasi-isometric embeddings and quasi-isometries
-/

-- quasi-isometric embeddings
def is_QIE_lower
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
    (c b : ℝ)
:= ∀ x x' : X, dist (f x) (f x') ≥ 1/c * dist x x' - b

def is_QIE_upper
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
    (c b : ℝ)
:= ∀ x x' : X, dist (f x) (f x') ≤ c * dist x x' + b

def is_QIE'
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
    (c b : ℝ)
:= is_QIE_upper f c b
 ∧ is_QIE_lower f c b

def is_QIE
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
:= ∃ c : ℝ, ∃ b : ℝ,
   c > 0
 ∧ b > 0
 ∧ is_QIE' f c b

-- finite distance
def has_fin_dist'
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f g : X → Y)
    (c : ℝ)
:= ∀ x : X, dist (f x) (g x) ≤ c

def has_fin_dist
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f g : X → Y)
:= ∃ c : ℝ,
   c > 0
 ∧ has_fin_dist' f g c

def are_quasi_inverse
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
    (g : Y → X)
:= has_fin_dist (g ∘ f) id
 ∧ has_fin_dist (f ∘ g) id

-- quasi-isometry
def is_QI
    {X Y : Type*} [metric_space X] [metric_space Y]
    (f : X → Y)
:= is_QIE f
 ∧ ∃ g : Y → X, is_QIE g
             ∧ are_quasi_inverse f g

/-
# Isometric embeddings and isometries
-/

-- definition of isometries
def is_isometry
     {X Y : Type*} [metric_space X] [metric_space Y]
     (f : X → Y)
:= sorry
-- Exercise: complete this definition;
-- keeping the definition structurally close to the QI case
-- will make the proof of the theorem below easier

-- Exercise: it could be helpful to prove individual steps
-- of the main proof in separate lemmas

-- Every isometry is a quasi-isometry
theorem isometries_are_quasiisometries
-- Exercise: complete the statement of the theorem
:=
begin
  -- Exercise: complete the proof of the theorem
end
```