

Providing restriction Spaces with an ultrametric Structure

Benoît Ballenghien Benjamin Puyobro Burkhardt Wolff

October 13, 2025

Abstract

We investigate the relationship between restriction spaces and classical metric structures by instantiating the former as ultrametric spaces. This is classically captured by defining the distance as

$$\text{dist } x \ y = \inf_{x \downarrow n = y \downarrow n} \left(\frac{1}{2} \right)^n$$

but we actually generalize this perspective by introducing a hierarchy of increasingly refined type classes to systematically relate ultrametric and restriction-based notions. This layered approach enables a precise comparison of structural and topological properties. In the end, our main result establishes that completeness in the sense of restriction spaces coincides with standard metric completeness, thus bridging the gap between `Restriction_Spaces` and Banach’s fixed-point theorem established in `HOL-Analysis`.

Contents

1	Definitions on Functions of Metric Space	1
1.1	Definitions	1
1.1.1	Lipschitz Map	1
1.1.2	Non-expanding Map	3
1.1.3	Contraction Map	4
1.2	Properties	5
1.3	Banach’s fixed-point Theorems	7
2	Locales factorizing the proof Work	7
2.1	Preliminaries on strictly decreasing Sequences	7
2.2	The Construction with Locales	8
3	Ultrametric Structure of restriction Spaces	11
3.1	The Construction with Classes	12
3.2	Equivalence between Lipschitz Map and Restriction shift Map	19

4	Functions	20
4.1	Restriction Space	20
4.2	Completeness	22
4.3	Kind of Extensionality	23
5	Product	23
5.1	Isomorphic Product Construction	23
5.1.1	Definition and First Properties	23
5.2	Syntactic Sugar	25
5.3	Product	25
5.4	Completeness	27
5.4.1	Preliminaries	27
5.4.2	Complete Restriction Space	28
6	Main entry Point	30

1 Definitions on Functions of Metric Space

In this theory, we define the notion of lipschitz map, non-expanding map and contraction map. We also establish correspondences.

1.1 Definitions

1.1.1 Lipschitz Map

This notion is a generalization of contraction map and non-expanding map.

definition *lipschitz-with-on* :: $\langle 'a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, \text{real}, 'a \text{ set} \rangle \Rightarrow \text{bool}$

where $\langle \text{lipschitz-with-on } f \ \alpha \ A \equiv 0 \leq \alpha \wedge (\forall x \in A. \forall y \in A. \text{dist } (f \ x) \ (f \ y) \leq \alpha * \text{dist } x \ y) \rangle$

abbreviation *lipschitz-with* :: $\langle 'a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, \text{real} \rangle \Rightarrow \text{bool}$

where $\langle \text{lipschitz-with } f \ \alpha \equiv \text{lipschitz-with-on } f \ \alpha \ \text{UNIV} \rangle$

lemma *lipschitz-with-onI* :

$\langle \llbracket 0 \leq \alpha; \bigwedge x \ y. \llbracket x \in A; y \in A; x \neq y; f \ x \neq f \ y \rrbracket \implies \text{dist } (f \ x) \ (f \ y) \leq \alpha * \text{dist } x \ y \rrbracket \implies \text{lipschitz-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-withI* :

$\langle \llbracket 0 \leq \alpha; \bigwedge x \ y. x \neq y \implies f \ x \neq f \ y \implies \text{dist } (f \ x) \ (f \ y) \leq \alpha * \text{dist } x \ y \rrbracket \implies \text{lipschitz-with } f \ \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-with-onD1* : $\langle \text{lipschitz-with-on } f \ \alpha \ A \implies 0 \leq \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-withD1* : $\langle \text{lipschitz-with } f \ \alpha \implies 0 \leq \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-with-onD2* :
 $\langle \text{lipschitz-with-on } f \ \alpha \ A \implies x \in A \implies y \in A \implies \text{dist } (f \ x) \ (f \ y) \leq \alpha * \text{dist } x \ y \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-withD2* :
 $\langle \text{lipschitz-with } f \ \alpha \implies \text{dist } (f \ x) \ (f \ y) \leq \alpha * \text{dist } x \ y \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-with-imp-lipschitz-with-on* : $\langle \text{lipschitz-with } f \ \alpha \implies \text{lipschitz-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-with-on-imp-lipschitz-with-on-ge* : $\langle \text{lipschitz-with-on } f \ \beta \ A \rangle$
if $\langle \alpha \leq \beta \rangle$ **and** $\langle \text{lipschitz-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

theorem *lipschitz-with-on-comp-lipschitz-with-on* :
 $\langle \text{lipschitz-with-on } (\lambda x. g \ (f \ x)) \ (\beta * \alpha) \ A \rangle$
if $\langle f \text{ ' } A \subseteq B \rangle$ $\langle \text{lipschitz-with-on } g \ \beta \ B \rangle$ $\langle \text{lipschitz-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

corollary *lipschitz-with-comp-lipschitz-with* :
 $\langle \llbracket \text{lipschitz-with } g \ \beta; \text{lipschitz-with } f \ \alpha \rrbracket \implies \text{lipschitz-with } (\lambda x. g \ (f \ x)) \ (\beta * \alpha) \rangle$
 $\langle \text{proof} \rangle$

1.1.2 Non-expanding Map

definition *non-expanding-on* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, 'a \ \text{set}] \Rightarrow \text{bool} \rangle$
where $\langle \text{non-expanding-on } f \ A \equiv \text{lipschitz-with-on } f \ 1 \ A \rangle$

abbreviation *non-expanding* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}] \Rightarrow \text{bool} \rangle$
where $\langle \text{non-expanding } f \equiv \text{non-expanding-on } f \ \text{UNIV} \rangle$

lemma *non-expanding-onI* :
 $\langle \llbracket \bigwedge x \ y. \llbracket x \in A; y \in A; x \neq y; f \ x \neq f \ y \rrbracket \implies \text{dist } (f \ x) \ (f \ y) \leq \text{dist } x \ y \rrbracket \rangle$

$x\ y\| \implies$
 $\text{non-expanding-on } f\ A\ \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expandingI* :
 $\langle \llbracket \bigwedge x\ y. x \neq y \implies f\ x \neq f\ y \implies \text{dist } (f\ x) (f\ y) \leq \text{dist } x\ y \rrbracket \implies$
 $\text{non-expanding } f \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expanding-onD* :
 $\langle \text{non-expanding-on } f\ A \implies x \in A \implies y \in A \implies \text{dist } (f\ x) (f\ y) \leq$
 $\text{dist } x\ y \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expandingD* : $\langle \text{non-expanding } f \implies \text{dist } (f\ x) (f\ y) \leq \text{dist } x\ y \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expanding-imp-non-expanding-on*: $\langle \text{non-expanding } f \implies$
 $\text{non-expanding-on } f\ A \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expanding-on-comp-non-expanding-on* :
 $\langle \llbracket f\ 'A \subseteq B; \text{non-expanding-on } g\ B; \text{non-expanding-on } f\ A \rrbracket \implies$
 $\text{non-expanding-on } (\lambda x. g\ (f\ x))\ A \rangle$
 $\langle \text{proof} \rangle$

corollary *non-expanding-comp-non-expanding* :
 $\langle \llbracket \text{non-expanding } g; \text{non-expanding } f \rrbracket \implies \text{non-expanding } (\lambda x. g\ (f\ x)) \rangle$
 $\langle \text{proof} \rangle$

1.1.3 Contraction Map

definition *contraction-with-on* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, \text{real}, 'a\ \text{set}] \Rightarrow \text{bool} \rangle$

where $\langle \text{contraction-with-on } f\ \alpha\ A \equiv \alpha < 1 \wedge \text{lipschitz-with-on } f\ \alpha\ A \rangle$

abbreviation *contraction-with* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, \text{real}] \Rightarrow \text{bool} \rangle$

where $\langle \text{contraction-with } f\ \alpha \equiv \text{contraction-with-on } f\ \alpha\ \text{UNIV} \rangle$

definition *contraction-on* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}, 'a\ \text{set}] \Rightarrow \text{bool} \rangle$

where $\langle \text{contraction-on } f\ A \equiv \exists \alpha. \text{contraction-with-on } f\ \alpha\ A \rangle$

abbreviation *contraction* :: $\langle ['a :: \text{metric-space} \Rightarrow 'b :: \text{metric-space}]$

$\Rightarrow \text{bool}$
where $\langle \text{contraction } f \equiv \text{contraction-on } f \text{ UNIV} \rangle$

lemma *contraction-with-onI* :
 $\langle \llbracket 0 \leq \alpha; \alpha < 1; \bigwedge x y. \llbracket x \in A; y \in A; x \neq y; f x \neq f y \rrbracket \implies \text{dist } (f x) (f y) \leq \alpha * \text{dist } x y \rrbracket \implies \text{contraction-with-on } f \alpha A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-withI* :
 $\langle \llbracket 0 \leq \alpha; \alpha < 1; \bigwedge x y. x \neq y \implies f x \neq f y \implies \text{dist } (f x) (f y) \leq \alpha * \text{dist } x y \rrbracket \implies \text{contraction-with } f \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-onD1* : $\langle \text{contraction-with-on } f \alpha A \implies 0 \leq \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-withD1* : $\langle \text{contraction-with } f \alpha \implies 0 \leq \alpha \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-onD2* : $\langle \text{contraction-with-on } f \alpha A \implies \alpha < 1 \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-withD2* : $\langle \text{contraction-with } f \alpha \implies \alpha < 1 \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-onD3* :
 $\langle \text{contraction-with-on } f \alpha A \implies x \in A \implies y \in A \implies \text{dist } (f x) (f y) \leq \alpha * \text{dist } x y \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-withD3* : $\langle \text{contraction-with } f \alpha \implies \text{dist } (f x) (f y) \leq \alpha * \text{dist } x y \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-imp-contraction-with-on*:
 $\langle \text{contraction-with } f \alpha \implies \text{contraction-with-on } f \alpha A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-imp-contraction-on*: $\langle \text{contraction } f \implies \text{contraction-on } f A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-on-imp-contraction-on* :

$\langle \text{contraction-with-on } f \ \alpha \ A \implies \text{contraction-on } f \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-imp-contraction*: $\langle \text{contraction-with } f \ \alpha \implies \text{contraction } f \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-onE*:
 $\langle \llbracket \text{contraction-on } f \ A; \bigwedge \alpha. \text{contraction-with-on } f \ \alpha \ A \implies \text{thesis} \rrbracket \implies \text{thesis} \rangle$
 $\langle \text{proof} \rangle$

lemma *contractionE*:
 $\langle \llbracket \text{contraction } f; \bigwedge \alpha. \text{contraction-with } f \ \alpha \implies \text{thesis} \rrbracket \implies \text{thesis} \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-on-imp-contraction-with-on-ge* :
 $\langle \llbracket \alpha \leq \beta; \beta < 1; \text{contraction-with-on } f \ \alpha \ A \rrbracket \implies \text{contraction-with-on } f \ \beta \ A \rangle$
 $\langle \text{proof} \rangle$

1.2 Properties

lemma *contraction-with-on-imp-lipschitz-with-on[simp]* :
 $\langle \text{contraction-with-on } f \ \alpha \ A \implies \text{lipschitz-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expanding-on-imp-lipschitz-with-one-on[simp]* :
 $\langle \text{non-expanding-on } f \ A \implies \text{lipschitz-with-on } f \ 1 \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-on-imp-non-expanding-on[simp]* :
 $\langle \text{contraction-on } f \ A \implies \text{non-expanding-on } f \ A \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-on-comp-contraction-with-on* :
 $\langle \text{contraction-with-on } (\lambda x. g \ (f \ x)) \ (\beta * \alpha) \ A \rangle$
if $\langle f \text{ ‘ } A \subseteq B \rangle \langle \text{contraction-with-on } g \ \beta \ B \rangle \langle \text{contraction-with-on } f \ \alpha \ A \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-with-comp-contraction-with* :
 $\langle \llbracket \text{contraction-with } g \ \beta; \text{contraction-with } f \ \alpha \rrbracket \implies \text{contraction-with } (\lambda x. g \ (f \ x)) \ (\beta * \alpha) \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-on-comp-contraction-on* :
 $\langle \llbracket f \text{ ' } A \subseteq B; \text{contraction-on } g \text{ } B; \text{contraction-on } f \text{ } A \rrbracket \implies \text{contraction-on } (\lambda x. g \text{ } (f \text{ } x)) \text{ } A \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-comp-contraction* :
 $\langle \llbracket \text{contraction } g; \text{contraction } f \rrbracket \implies \text{contraction } (\lambda x. g \text{ } (f \text{ } x)) \rangle$
 $\langle \text{proof} \rangle$

lemma *contraction-with-on-comp-non-expanding-on* :
 $\langle \text{contraction-with-on } (\lambda x. g \text{ } (f \text{ } x)) \text{ } \beta \text{ } A \rangle$
if $\langle f \text{ ' } A \subseteq B \rangle \langle \text{contraction-with-on } g \text{ } \beta \text{ } B \rangle \langle \text{non-expanding-on } f \text{ } A \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-with-comp-non-expanding* :
 $\langle \llbracket \text{contraction-with } g \text{ } \beta; \text{non-expanding } f \rrbracket \implies \text{contraction-with } (\lambda x. g \text{ } (f \text{ } x)) \text{ } \beta \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-on-comp-non-expanding-on* :
 $\langle \llbracket f \text{ ' } A \subseteq B; \text{contraction-on } g \text{ } B; \text{non-expanding-on } f \text{ } A \rrbracket \implies \text{contraction-on } (\lambda x. g \text{ } (f \text{ } x)) \text{ } A \rangle$
 $\langle \text{proof} \rangle$

corollary *contraction-comp-non-expanding* :
 $\langle \llbracket \text{contraction } g; \text{non-expanding } f \rrbracket \implies \text{contraction } (\lambda x. g \text{ } (f \text{ } x)) \rangle$
 $\langle \text{proof} \rangle$

lemma *non-expanding-on-comp-contraction-with-on* :
 $\langle \text{contraction-with-on } (\lambda x. g \text{ } (f \text{ } x)) \text{ } \alpha \text{ } A \rangle$
if $\langle f \text{ ' } A \subseteq B \rangle \langle \text{non-expanding-on } g \text{ } B \rangle \langle \text{contraction-with-on } f \text{ } \alpha \text{ } A \rangle$
 $\langle \text{proof} \rangle$

corollary *non-expanding-comp-contraction-with* :
 $\langle \llbracket \text{non-expanding } g; \text{contraction-with } f \text{ } \alpha \rrbracket \implies \text{contraction-with } (\lambda x. g \text{ } (f \text{ } x)) \text{ } \alpha \rangle$
 $\langle \text{proof} \rangle$

corollary *non-expanding-on-comp-contraction-on* :
 $\langle \llbracket f \text{ ' } A \subseteq B; \text{non-expanding-on } g \text{ } B; \text{contraction-on } f \text{ } A \rrbracket \implies \text{contraction-on } (\lambda x. g \text{ } (f \text{ } x)) \text{ } A \rangle$
 $\langle \text{proof} \rangle$

corollary *non-expanding-comp-contraction* :
 $\langle \llbracket \text{non-expanding } g; \text{contraction } f \rrbracket \implies \text{contraction } (\lambda x. g \text{ } (f \text{ } x)) \rangle$
 $\langle \text{proof} \rangle$

1.3 Banach's fixed-point Theorems

We rewrite the Banach's fixed-point theorems with our new definition.

theorem *Banach-fix-type* : $\langle \text{contraction } f \implies \exists !x. f\ x = x \rangle$
for $f :: \langle 'a :: \text{complete-space} \Rightarrow 'a \rangle$
 $\langle \text{proof} \rangle$

theorem *Banach-fix*:
 $\langle \text{contraction-on } f\ s \implies \exists !x. x \in s \wedge f\ x = x \rangle$ **if** $\langle \text{complete } s \rangle \langle s \neq \{\} \rangle \langle f\ 's \subseteq s \rangle$
 $\langle \text{proof} \rangle$

2 Locales factorizing the proof Work

2.1 Preliminaries on strictly decreasing Sequences

abbreviation *strict-decseq* :: $\langle (\text{nat} \Rightarrow 'a :: \text{order}) \Rightarrow \text{bool} \rangle$
where $\langle \text{strict-decseq} \equiv \text{monotone } (<) (\lambda x\ y. y < x) \rangle$

lemma *strict-decseq-def* : $\langle \text{strict-decseq } X \longleftrightarrow (\forall m\ n. m < n \longrightarrow X\ n < X\ m) \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseqI* : $\langle \text{strict-decseq } X \rangle$ **if** $\langle \bigwedge n. X\ (\text{Suc } n) < X\ n \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseqD* : $\langle \text{strict-decseq } X \implies m < n \implies X\ n < X\ m \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseq-def-bis* : $\langle \text{strict-decseq } X \longleftrightarrow (\forall m\ n. X\ n < X\ m \longleftrightarrow m < n) \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseq-def-ter* : $\langle \text{strict-decseq } X \longleftrightarrow (\forall m\ n. X\ n \leq X\ m \longleftrightarrow m \leq n) \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseq-imp-decseq* : $\langle \text{strict-decseq } \sigma \implies \text{decseq } \sigma \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseq-SucI* : $\langle (\bigwedge n. X\ (\text{Suc } n) < X\ n) \implies \text{strict-decseq } X \rangle$
 $\langle \text{proof} \rangle$

lemma *strict-decseq-SucD* : $\langle \text{strict-decseq } A \implies A\ (\text{Suc } i) < A\ i \rangle$
 $\langle \text{proof} \rangle$

Classically, a restriction space is given the structure of a metric space by defining $\text{dist } x \ y \equiv \text{Inf } \{(1 / 2) ^ n \mid n. x \downarrow n = y \downarrow n\}$. This obviously also works if we replace $1 / 2$ by any real δ such that $0 < \delta$ and $\delta < 1$. But more generally, this still works if we set $\text{dist } x \ y \equiv \text{Inf } \{\sigma \ n \mid n. x \downarrow n = y \downarrow n\}$ where σ is a sequence of *real* verifying $\forall n. 0 < \sigma \ n$ and $\sigma \longrightarrow 0$. As you would expect, the more structure you have, the more powerful theorems you get. We explore all these variants in the theory below.

2.2 The Construction with Locales

Our formalization will extend the class *metric-space*. But some proofs are redundant, especially when it comes to the product type. So first we will be working with locales, and interpret them with the classes.

locale *NonDecseqRestrictionSpace* = *PreorderRestrictionSpace* +
 — Factorization of the proof work.
fixes $M :: \langle 'a \text{ set} \rangle$ **and** $\text{restriction-}\sigma :: \langle \text{nat} \Rightarrow \text{real} \rangle (\sigma_{\downarrow})$
and $\text{restriction-dist} :: \langle 'a \Rightarrow 'a \Rightarrow \text{real} \rangle (\text{dist}_{\downarrow})$
assumes $\text{restriction-}\sigma\text{-tendsto-zero} : \langle \text{restriction-}\sigma \longrightarrow 0 \rangle$
and $\text{zero-less-restriction-}\sigma \text{ [simp]} : \langle 0 < \text{restriction-}\sigma \ n \rangle$
and $\text{dist-restriction-is} :$
 $\langle \text{dist}_{\downarrow} \ x \ y = (\text{INF } n \in \text{restriction-related-set } x \ y. \text{restriction-}\sigma \ n) \rangle$
begin

lemma $\text{zero-le-restriction-}\sigma \text{ [simp]} : \langle 0 \leq \sigma_{\downarrow} \ n \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{restriction-}\sigma\text{-neq-zero [simp]} : \langle \sigma_{\downarrow} \ n \neq 0 \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{bounded-range-restriction-}\sigma : \langle \text{bounded } (\text{range } \sigma_{\downarrow}) \rangle$
 $\langle \text{proof} \rangle$

abbreviation $\text{restriction-}\sigma\text{-related-set} :: \langle 'a \Rightarrow 'a \Rightarrow \text{real set} \rangle$
where $\langle \text{restriction-}\sigma\text{-related-set } x \ y \equiv \sigma_{\downarrow} \text{ ` restriction-related-set } x \ y \rangle$

abbreviation $\text{restriction-}\sigma\text{-not-related-set} :: \langle 'a \Rightarrow 'a \Rightarrow \text{real set} \rangle$
where $\langle \text{restriction-}\sigma\text{-not-related-set } x \ y \equiv \sigma_{\downarrow} \text{ ` restriction-not-related-set } x \ y \rangle$

lemma *nonempty-restriction- σ -related-set* :
 $\langle \text{restriction-}\sigma\text{-related-set } x \ y \neq \{\} \rangle \langle \text{proof} \rangle$

lemma *restriction- σ -related-set-Un-restriction- σ -not-related-set* :
 $\langle \text{restriction-}\sigma\text{-related-set } x \ y \cup \text{restriction-}\sigma\text{-not-related-set } x \ y = \text{range } \sigma_{\downarrow} \rangle$
 $\langle \text{proof} \rangle$

lemma $\langle \text{bdd-above } (\text{restriction-}\sigma\text{-related-set } x \ y) \rangle$
 $\langle \text{proof} \rangle$

lemma $\langle \text{bdd-above } (\text{restriction-}\sigma\text{-not-related-set } x \ y) \rangle$
 $\langle \text{proof} \rangle$

lemma *bounded-restriction- σ -related-set*: $\langle \text{bounded } (\text{restriction-}\sigma\text{-related-set } x \ y) \rangle$
 $\langle \text{proof} \rangle$

lemma *bounded-restriction- σ -not-related-set*: $\langle \text{bounded } (\text{restriction-}\sigma\text{-not-related-set } x \ y) \rangle$
 $\langle \text{proof} \rangle$

corollary *restriction-space-Inf-properties*:
 $\langle a \in \text{restriction-}\sigma\text{-related-set } x \ y \implies \text{dist}_{\downarrow} x \ y \leq a \rangle$
 $\langle \llbracket \bigwedge a. a \in \text{restriction-}\sigma\text{-related-set } x \ y \implies b \leq a \rrbracket \implies b \leq \text{dist}_{\downarrow} x \ y \rangle$
 $\langle \text{proof} \rangle$

lemma *restriction- σ -related-set-alt* :
 $\langle \text{restriction-}\sigma\text{-related-set } x \ y = \{\sigma_{\downarrow} n \mid n. n \in \text{restriction-related-set } x \ y\} \rangle$
 $\langle \text{proof} \rangle$

lemma *exists-less-restriction- σ* : $\langle \exists n. m < n \wedge \sigma_{\downarrow} n < \sigma_{\downarrow} m \rangle$
 $\langle \text{proof} \rangle$

lemma $\langle \text{dist}_{\downarrow} x \ y = \text{Inf } (\text{restriction-}\sigma\text{-related-set } x \ y) \rangle$
 $\langle \text{proof} \rangle$

lemma *not-related-imp-dist-restriction-is-some-restriction- σ* :
 $\langle \exists n. \text{dist}_{\downarrow} x \ y = \sigma_{\downarrow} n \wedge (\forall m \leq n. x \downarrow m \lesssim y \downarrow m) \rangle \text{ if } \langle \neg x \lesssim y \rangle$
 $\langle \text{proof} \rangle$

lemma *not-related-imp-dist-restriction-le-some-restriction- σ* :
 $\langle \neg x \lesssim y \implies \exists n. \text{dist}_{\downarrow} x \ y \leq \sigma_{\downarrow} n \wedge (\neg x \downarrow \text{Suc } n \lesssim y \downarrow \text{Suc } n) \wedge (\forall m \leq n. x \downarrow m \lesssim y \downarrow m) \rangle$

$\langle \text{proof} \rangle$

lemma *restriction-dist-eq-0-iff-related* : $\langle \text{dist}_{\downarrow} x y = 0 \longleftrightarrow x \lesssim y \rangle$
 $\langle \text{proof} \rangle$

end

locale *DecseqRestrictionSpace* = *NonDecseqRestrictionSpace* +
assumes *decseq-restriction-σ* : $\langle \text{decseq } \sigma_{\downarrow} \rangle$
begin

lemma *dist-restriction-is-bis* :
 $\langle \text{dist}_{\downarrow} x y = (\text{if } x \lesssim y \text{ then } 0 \text{ else } \sigma_{\downarrow} (\text{Sup } (\text{restriction-related-set } x y))) \rangle$
if $\langle x \in M \rangle$ **and** $\langle y \in M \rangle$
 $\langle \text{proof} \rangle$

lemma *not-eq-imp-dist-restriction-is-restriction-σ-Sup-restriction-eq* :
 $\langle \text{dist}_{\downarrow} x y = \sigma_{\downarrow} (\text{Sup } (\text{restriction-related-set } x y)) \rangle$
 $\langle \forall m \leq \text{Sup } (\text{restriction-related-set } x y). x \downarrow m \lesssim y \downarrow m \rangle$
 $\langle \forall m > \text{Sup } (\text{restriction-related-set } x y). \neg x \downarrow m \lesssim y \downarrow m \rangle$
if $\langle \neg x \lesssim y \rangle$ **and** $\langle x \in M \rangle$ **and** $\langle y \in M \rangle$
 $\langle \text{proof} \rangle$

theorem *restriction-dist-tendsto-zero-independent-of-restriction-σ* :
— Very powerful theorem: the convergence of the distance to 0 is actually independent from the restriction sequence chosen.
— This is the theorem for which we had to work with locales first.

assumes $\langle \text{DecseqRestrictionSpace } (\downarrow) (\lesssim) \text{ restriction-}\sigma' \text{ restriction-dist}' \rangle$
and $\langle \Sigma \in M \rangle$ **and** $\langle \text{range } \sigma \subseteq M \rangle$
shows $\langle (\lambda n. \text{dist}_{\downarrow} (\sigma n) \Sigma) \longrightarrow 0 \longleftrightarrow (\lambda n. \text{restriction-dist}' (\sigma n) \Sigma) \longrightarrow 0 \rangle$
 $\langle \text{proof} \rangle$

end

3 Ultrametric Structure of restriction Spaces

This has only be proven with the sort constraint, not inside the context of the class *metric-space* ...

context *metric-space* **begin**

lemma *LIMSEQ-def* : $\langle X \longrightarrow L \longleftrightarrow (\forall r > 0. \exists no. \forall n \geq no. \text{dist } (X\ n) \ L < r) \rangle$
 $\langle \text{proof} \rangle$

lemma *LIMSEQ-iff-nz*: $\langle X \longrightarrow L \longleftrightarrow (\forall r > 0. \exists no > 0. \forall n \geq no. \text{dist } (X\ n) \ L < r) \rangle$
 $\langle \text{proof} \rangle$

lemma *metric-LIMSEQ-I*: $\langle (\bigwedge r. 0 < r \implies \exists no. \forall n \geq no. \text{dist } (X\ n) \ L < r) \implies X \longrightarrow L \rangle$
 $\langle \text{proof} \rangle$

lemma *metric-LIMSEQ-D*: $\langle X \longrightarrow L \implies 0 < r \implies \exists no. \forall n \geq no. \text{dist } (X\ n) \ L < r \rangle$
 $\langle \text{proof} \rangle$

lemma *LIMSEQ-dist-iff*:
 $\langle f \longrightarrow l \longleftrightarrow (\lambda x. \text{dist } (f\ x) \ l) \longrightarrow 0 \rangle$
 $\langle \text{proof} \rangle$

lemma *Cauchy-converges-subseq*:

fixes $u :: \text{nat} \Rightarrow 'a$
assumes *Cauchy* u
strict-mono r
 $(u \circ r) \longrightarrow l$
shows $u \longrightarrow l$
 $\langle \text{proof} \rangle$

end

3.1 The Construction with Classes

class *restriction-σ* = *restriction-space* +
fixes *restriction-σ* :: $\langle 'a \text{ itself} \Rightarrow \text{nat} \Rightarrow \text{real} \rangle (\langle \sigma_{\downarrow} \rangle)$

$\langle ML \rangle$

```

class non-decseq-restriction-space =
  uniformity-dist + open-uniformity + restriction-σ +
  — We do not assume the restriction sequence to be decseq yet.
  assumes restriction-σ-tendsto-zero' :  $\langle \sigma \downarrow \text{TYPE}('a) \longrightarrow 0 \rangle$ 
    and zero-less-restriction-σ' [simp] :  $\langle 0 < \sigma \downarrow \text{TYPE}('a) \ n \rangle$ 

    and dist-restriction-is' :  $\langle \text{dist } x \ y = (\text{INF } n \in \{n. \ x \downarrow n = y \downarrow n\}. \sigma \downarrow \text{TYPE}('a) \ n) \rangle$ 
begin

sublocale NonDecseqRestrictionSpace  $\langle (\downarrow) \rangle \langle (=) \rangle \langle \text{UNIV} \rangle \langle \sigma \downarrow \text{TYPE}('a) \rangle$ 
  dist
   $\langle \text{proof} \rangle$ 

end

 $\langle \text{ML} \rangle$ 

We hide duplicated facts  $\sigma \downarrow \text{TYPE}('a) \longrightarrow 0$ 
 $0 < \sigma \downarrow \text{TYPE}('a) \ ?n$ 
 $\text{dist } ?x \ ?y = \text{Inf } (\text{restriction-}\sigma\text{-related-set } ?x \ ?y).$ 
hide-fact restriction-σ-tendsto-zero' zero-less-restriction-σ' dist-restriction-is'

context non-decseq-restriction-space begin

subclass ultrametric-space
 $\langle \text{proof} \rangle$ 

end

context non-decseq-restriction-space begin

lemma restriction-tendsto-self:  $\langle (\lambda n. \ x \downarrow n) \longrightarrow x \rangle$ 
 $\langle \text{proof} \rangle$ 

end

class decseq-restriction-space = non-decseq-restriction-space +
  assumes decseq-restriction-σ' :  $\langle \text{decseq } (\sigma \downarrow \text{TYPE}('a)) \rangle$ 
begin

sublocale DecseqRestrictionSpace  $\langle (\downarrow) \rangle \langle (=) \rangle \langle \text{UNIV} :: 'a \text{ set} \rangle$ 
   $\langle \text{restriction-}\sigma \ \text{TYPE}('a) \rangle \text{ dist}$ 

```

$\langle \text{proof} \rangle$
lemmas $\text{dist-restriction-is-bis-simplified} = \text{dist-restriction-is-bis}[\text{simplified}]$
and $\text{not-eq-imp-dist-restriction-is-restriction-}\sigma\text{-Sup-restriction-eq-simplified}$
 $=$
 $\text{not-eq-imp-dist-restriction-is-restriction-}\sigma\text{-Sup-restriction-eq}[\text{simplified}]$

end

We hide duplicated fact *antimono-on UNIV* $(\sigma \downarrow \text{TYPE}(\text{'a}))$.

hide-fact $\text{decseq-restriction-}\sigma'$

class $\text{strict-decseq-restriction-space} = \text{non-decseq-restriction-space} +$
assumes $\text{strict-decseq-restriction-}\sigma : \langle \text{strict-decseq } (\sigma \downarrow \text{TYPE}(\text{'a})) \rangle$
begin

subclass $\text{decseq-restriction-space}$
 $\langle \text{proof} \rangle$

end

Generic Properties

lemma (**in** metric-space) $\text{dist-sequences-tendsto-zero-imp-tendsto-iff} :$
 $\langle (\lambda n. \text{dist } (\sigma \ n) (\psi \ n)) \longrightarrow 0 \implies \sigma \longrightarrow \Sigma \longleftrightarrow \psi \longrightarrow \Sigma \rangle$
 $\langle \text{proof} \rangle$

lemma (**in** $\text{non-decseq-restriction-space}$) $\text{restricted-sequence-tendsto-iff}$
 $:$
 $\langle (\lambda n. \sigma \ n \downarrow n) \longrightarrow \Sigma \longleftrightarrow \sigma \longrightarrow \Sigma \rangle$
 $\langle \text{proof} \rangle$

lemma (**in** $\text{non-decseq-restriction-space}$) $\text{Cauchy-restriction-chain} :$
 $\langle \text{Cauchy } \sigma \rangle$ **if** $\langle \text{chain}_{\downarrow} \sigma \rangle$
 $\langle \text{proof} \rangle$

lemma (**in** $\text{non-decseq-restriction-space}$) $\text{restriction-tendsto-imp-tendsto}$
 $:$
 $\langle \sigma \longrightarrow \Sigma \rangle$ **if** $\langle \sigma \dashrightarrow \Sigma \rangle$
 $\langle \text{proof} \rangle$

In Decseq Restriction Space

context $\text{decseq-restriction-space}$ **begin**

lemma *le-dist-to-restriction-eqE* :
obtains *k* **where** $\langle n \leq k \rangle \langle \bigwedge x y :: 'a. \text{dist } x y \leq \sigma_{\downarrow} \text{ TYPE}('a) k \implies x \downarrow n = y \downarrow n \rangle$
 $\langle \text{proof} \rangle$

theorem *tendsto-iff-restriction-tendsto* : $\langle \sigma \longrightarrow \Sigma \longleftrightarrow \sigma -\downarrow\rightarrow \Sigma \rangle$
 $\langle \text{proof} \rangle$

corollary *convergent-iff-restriction-convergent* : $\langle \text{convergent } \sigma \longleftrightarrow \text{convergent}_{\downarrow} \sigma \rangle$
 $\langle \text{proof} \rangle$

theorem *complete-iff-restriction-complete* :
 $\langle (\forall \sigma. \text{Cauchy } \sigma \longrightarrow \text{convergent } \sigma) \longleftrightarrow (\forall \sigma. \text{chain}_{\downarrow} \sigma \longrightarrow \text{convergent}_{\downarrow} \sigma) \rangle$
— The following result shows that we have not lost anything with our definitions of convergence, completeness, etc. specific to restriction spaces.
 $\langle \text{proof} \rangle$

end

The following classes will be useful later.

class *complete-decseq-restriction-space* = *decseq-restriction-space* +
assumes *restriction-chain-imp-restriction-convergent'* : $\langle \text{chain}_{\downarrow} \sigma \implies \text{convergent}_{\downarrow} \sigma \rangle$
begin

subclass *complete-restriction-space*
 $\langle \text{proof} \rangle$

subclass *complete-ultrametric-space*
 $\langle \text{proof} \rangle$

end

We hide duplicated fact $\text{chain}_{\downarrow} ?\sigma \implies \text{convergent}_{\downarrow} ?\sigma$.

hide-fact *restriction-chain-imp-restriction-convergent'*

class *complete-strict-decseq-restriction-space* = *strict-decseq-restriction-space* +
assumes *restriction-chain-imp-restriction-convergent'* : $\langle \text{chain}_{\downarrow} \sigma \implies \text{convergent}_{\downarrow} \sigma \rangle$

begin

subclass *complete-decseq-restriction-space*
 $\langle \text{proof} \rangle$

end

We hide duplicated fact $\text{chain}_{\downarrow} ?\sigma \implies \text{convergent}_{\downarrow} ?\sigma$.

hide-fact *restriction-chain-imp-restriction-convergent'*

class *restriction- δ* = *restriction- σ* +
fixes *restriction- δ* :: $\langle 'a \text{ itself} \Rightarrow \text{real} \rangle (\delta_{\downarrow})$
assumes *zero-less-restriction- δ* [*simp*] : $\langle 0 < \delta_{\downarrow} \text{ TYPE}('a) \rangle$
and *restriction- δ -less-one* [*simp*] : $\langle \delta_{\downarrow} \text{ TYPE}('a) < 1 \rangle$
begin
lemma *zero-le-restriction- δ* [*simp*] : $\langle 0 \leq \delta_{\downarrow} \text{ TYPE}('a) \rangle$
and *restriction- δ -le-one* [*simp*] : $\langle \delta_{\downarrow} \text{ TYPE}('a) \leq 1 \rangle$
and *zero-le-pow-restriction- δ* [*simp*] : $\langle 0 \leq \delta_{\downarrow} \text{ TYPE}('a) \wedge n \rangle$
 $\langle \text{proof} \rangle$

lemma *pow-restriction- δ -le-one* [*simp*] : $\langle \delta_{\downarrow} \text{ TYPE}('a) \wedge n \leq 1 \rangle$
 $\langle \text{proof} \rangle$

lemma *pow-restriction- δ -less-one* [*simp*] : $\langle n \neq 0 \implies \delta_{\downarrow} \text{ TYPE}('a) \wedge n < 1 \rangle$
 $\langle \text{proof} \rangle$

end

$\langle \text{ML} \rangle$

class *at-least-geometric-restriction-space* =
uniformity-dist + *open-uniformity* + *restriction- δ* +
assumes *zero-less-restriction- σ'* : $\langle 0 < \sigma_{\downarrow} \text{ TYPE}('a) \ n \rangle$
and *restriction- σ -le* :
 $\langle \sigma_{\downarrow} \text{ TYPE}('a) (\text{Suc } n) \leq \delta_{\downarrow} \text{ TYPE}('a) * \sigma_{\downarrow} \text{ TYPE}('a) \ n \rangle$
and *dist-restriction-is'* :
 $\langle \text{dist } x \ y = (\text{INF } n \in \{n. x \downarrow n = y \downarrow n\}. \sigma_{\downarrow} \text{ TYPE}('a) \ n) \rangle$

$\langle \text{ML} \rangle$

context *at-least-geometric-restriction-space* **begin**

lemma *restriction- σ -le-restriction- σ -times-pow-restriction- δ* :
 $\langle \sigma_{\downarrow} \text{ TYPE}('a) (n + k) \leq \sigma_{\downarrow} \text{ TYPE}('a) \ n * \delta_{\downarrow} \text{ TYPE}('a) \wedge k \rangle$

$\langle proof \rangle$

lemma *restriction- σ -le-pow-restriction- δ* :
 $\langle \sigma \downarrow TYPE('a) \ n \leq \sigma \downarrow TYPE('a) \ 0 * \delta \downarrow TYPE('a) \ \wedge \ n \rangle$
 $\langle proof \rangle$

subclass *strict-decseq-restriction-space*
 $\langle proof \rangle$

lemma $\langle 0 < \delta \downarrow TYPE('a) \ \wedge \ n \rangle \langle proof \rangle$

end

We hide duplicated facts $0 < \sigma \downarrow TYPE('?a) \ ?n$
 $dist \ ?x \ ?y = Inf \ (restriction-\sigma-related-set \ ?x \ ?y).$

hide-fact *zero-less-restriction- σ' dist-restriction-is'*

class *complete-at-least-geometric-restriction-space* = *at-least-geometric-restriction-space*
 $+$
assumes *restriction-chain-imp-restriction-convergent'* : $\langle chain \downarrow \ \sigma$
 $\implies convergent \downarrow \ \sigma \rangle$
begin

subclass *complete-strict-decseq-restriction-space*
 $\langle proof \rangle$

end

We hide duplicated fact $chain \downarrow \ ?\sigma \implies convergent \downarrow \ ?\sigma.$

hide-fact *restriction-chain-imp-restriction-convergent'*

$\langle ML \rangle$

class *geometric-restriction-space* = *uniformity-dist* + *open-uniformity*
 $+$ *restriction- δ* +
assumes *restriction- σ -is* : $\langle \sigma \downarrow TYPE('a) \ n = \delta \downarrow TYPE('a) \ \wedge \ n \rangle$
and *dist-restriction-is'* : $\langle dist \ x \ y = (INF \ n \in \{n. \ x \downarrow n = y \downarrow n\}.$
 $\sigma \downarrow TYPE('a) \ n) \rangle$

begin

This is what “restriction space” usually mean in the literature.
 The previous classes are generalizations of this concept (even this
 one is a generalization, since we usually have $\delta_{\downarrow} \text{TYPE}('a) = 1$
 $/ 2$).

subclass *at-least-geometric-restriction-space*
 $\langle \text{proof} \rangle$

lemma $\langle 0 < \delta_{\downarrow} \text{TYPE}('a) \wedge n \rangle \langle \text{proof} \rangle$

end

$\langle ML \rangle$

We hide duplicated fact $\text{dist } ?x \ ?y = \text{Inf } (\text{restriction-}\sigma\text{-related-set } ?x \ ?y)$.

hide-fact *dist-restriction-is'*

class *complete-geometric-restriction-space* = *geometric-restriction-space*
 +
assumes *restriction-chain-imp-restriction-convergent'* : $\langle \text{chain}_{\downarrow} \sigma \implies \text{convergent}_{\downarrow} \sigma \rangle$
begin

subclass *complete-at-least-geometric-restriction-space*
 $\langle \text{proof} \rangle$

end

We hide duplicated fact $\text{chain}_{\downarrow} ?\sigma \implies \text{convergent}_{\downarrow} ?\sigma$.

hide-fact *restriction-chain-imp-restriction-convergent'*

theorem *geometric-restriction-space-completeI* : $\langle \text{convergent } \sigma \rangle$
if $\langle \bigwedge \sigma :: \text{nat} \Rightarrow 'a. \text{restriction-chain } \sigma \implies \exists \Sigma. \forall n. \Sigma \downarrow n = \sigma \ n \rangle$
and $\langle \text{Cauchy } \sigma \rangle$ **for** $\sigma :: \langle \text{nat} \Rightarrow 'a :: \text{geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma (**in** *non-decseq-restriction-space*) *restriction-cball-subset-cball*
 :
 $\langle \sigma_{\downarrow} \text{TYPE}('a) \ n \leq r \implies \mathcal{B}_{\downarrow}(\Sigma, n) \subseteq \{x. \text{dist } \Sigma \ x \leq r\} \rangle$
 $\langle \text{proof} \rangle$

corollary *restriction-cball-subset-cball-bis* :
 $\langle \sigma_{\downarrow} \text{TYPE}('a) \ n \leq r \implies \mathcal{B}_{\downarrow}(\Sigma, n) \subseteq \text{cball } \Sigma \ r \rangle$
for $\Sigma :: \langle 'a :: \text{non-decseq-restriction-space} \rangle$

$\langle \text{proof} \rangle$

lemma (in *non-decseq-restriction-space*) *restriction-ball-subset-ball* :
 $\langle \sigma_{\downarrow} \text{TYPE}('a) \ n < r \implies \text{restriction-ball } \Sigma \ n \subseteq \{x. \text{dist } \Sigma \ x < r\} \rangle$
 $\langle \text{proof} \rangle$

corollary *restriction-ball-subset-ball-bis* :
 $\langle \sigma_{\downarrow} \text{TYPE}('a) \ n < r \implies \text{restriction-ball } \Sigma \ n \subseteq \text{ball } \Sigma \ r \rangle$
for $\Sigma :: \langle 'a :: \text{non-decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma (in *strict-decseq-restriction-space*)
restriction-cball-is-cball : $\langle \mathcal{B}_{\downarrow}(\Sigma, n) = \{x. \text{dist } \Sigma \ x \leq \sigma_{\downarrow} \text{TYPE}('a) \ n\} \rangle$
 $\langle \text{proof} \rangle$

lemma *restriction-cball-is-cball-bis* : $\langle \mathcal{B}_{\downarrow}(\Sigma, n) = \text{cball } \Sigma \ (\sigma_{\downarrow} \text{TYPE}('a) \ n) \rangle$
for $\Sigma :: \langle 'a :: \text{strict-decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma (in *strict-decseq-restriction-space*)
restriction-ball-is-ball : $\langle \text{restriction-ball } \Sigma \ n = \{x. \text{dist } \Sigma \ x < \sigma_{\downarrow} \text{TYPE}('a) \ n\} \rangle$
 $\langle \text{proof} \rangle$

lemma *restriction-ball-is-ball-bis* : $\langle \text{restriction-ball } \Sigma \ n = \text{ball } \Sigma \ (\sigma_{\downarrow} \text{TYPE}('a) \ n) \rangle$
for $\Sigma :: \langle 'a :: \text{strict-decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma *isCont-iff-restriction-cont-at* : $\langle \text{isCont } f \ \Sigma \longleftrightarrow \text{restriction-cont-at } f \ \Sigma \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma (in *strict-decseq-restriction-space*)
open-iff-restriction-open : $\langle \text{open } U \longleftrightarrow \text{open}_{\downarrow} U \rangle$
 $\langle \text{proof} \rangle$

lemma (in *strict-decseq-restriction-space*)
closed-iff-restriction-closed : $\langle \text{closed } U \longleftrightarrow \text{closed}_{\downarrow} U \rangle$
 $\langle \text{proof} \rangle$

lemma *continuous-on-iff-restriction-cont-on* :
 $\langle \text{open } U \implies \text{continuous-on } U f \longleftrightarrow \text{restriction-cont-on } f U \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

3.2 Equivalence between Lipschitz Map and Restriction shift Map

For a function $f :: 'a \Rightarrow 'b$, it is equivalent to have *restriction-shift-on* $f k A$ and *lipschitz-with-on* $f (\delta_{\downarrow} \text{TYPE}('b))^k A$ when $'a$ is of sort *geometric-restriction-space* and $\sigma_{\downarrow} \text{TYPE}('b) = \sigma_{\downarrow} \text{TYPE}('a)$ ($'b$ is then necessarily also of sort *geometric-restriction-space*).

Weaker versions of this result can be established with weaker assumptions on the sort, this is what we do below.

lemma *restriction-shift-nonneg-on-imp-lipschitz-with-on* :
 $\langle \text{lipschitz-with-on } f (\text{restriction-}\delta \text{ TYPE}('b) \wedge k) A \rangle \text{ if } \langle \text{restriction-shift-on } f (\text{int } k) A \rangle$
and *le-restriction- σ* : $\langle \bigwedge n. \text{restriction-}\sigma \text{ TYPE}('b) n \leq \text{restriction-}\sigma \text{ TYPE}('a) n \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{at-least-geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

corollary *restriction-shift-nonneg-imp-lipschitz-with* :
 $\langle \llbracket \text{restriction-shift } f (\text{int } k); \bigwedge n. \text{restriction-}\sigma \text{ TYPE}('b) n \leq \text{restriction-}\sigma \text{ TYPE}('a) n \rrbracket$
 $\implies \text{lipschitz-with } f (\text{restriction-}\delta \text{ TYPE}('b) \wedge k) \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{at-least-geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma *lipschitz-with-on-imp-restriction-shift-neg-on* :
 $\langle \text{restriction-shift-on } f (- \text{int } k) A \rangle \text{ if } \langle \text{lipschitz-with-on } f (\text{restriction-}\delta \text{ TYPE}('b) \text{ powi } - \text{int } k) A \rangle$
and *le-restriction- σ* : $\langle \bigwedge n. \text{restriction-}\sigma \text{ TYPE}('a) n \leq \text{restriction-}\sigma \text{ TYPE}('b) n \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{at-least-geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

corollary *lipschitz-with-imp-restriction-shift-neg* :

$\langle \llbracket \text{lipschitz-with } f \text{ (restriction-}\delta \text{ TYPE('b) powi - int k);}$
 $\bigwedge n. \text{restriction-}\sigma \text{ TYPE('a) } n \leq \text{restriction-}\sigma \text{ TYPE('b) } n \rrbracket$
 $\implies \text{restriction-shift } f \text{ (- int k)} \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{at-least-geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

We obtained that *restriction-shift* implies *lipschitz-with* when $0 \leq k$ and that *lipschitz-with* implies *restriction-shift* when $k \leq 0$.
 To cover the remaining cases, we give move from *at-least-geometric-restriction-space* to *geometric-restriction-space*.

theorem *lipschitz-with-on-iff-restriction-shift-on* :
 $\langle \text{lipschitz-with-on } f \text{ (restriction-}\delta \text{ TYPE('b) powi k) } A \longleftrightarrow \text{restriction-shift-on } f \text{ k } A \rangle$
if *same-restriction- σ* : $\langle \text{restriction-}\sigma \text{ TYPE('b) = restriction-}\sigma \text{ TYPE('a)} \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

corollary *lipschitz-with-iff-restriction-shift* :
 $\langle \text{restriction-}\sigma \text{ TYPE('b) = restriction-}\sigma \text{ TYPE('a)} \implies$
 $\text{lipschitz-with } f \text{ (restriction-}\delta \text{ TYPE('b) powi k)} \longleftrightarrow \text{restriction-shift}$
 $f \text{ k} \rangle$
for $f :: \langle 'a :: \text{decseq-restriction-space} \Rightarrow 'b :: \text{geometric-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

4 Functions

4.1 Restriction Space

instantiation $\langle \text{fun} \rangle :: (\text{type}, \text{restriction-}\sigma) \text{restriction-}\sigma$
begin

definition *restriction- σ -fun* : $\langle ('a \Rightarrow 'b) \text{ itself} \Rightarrow \text{nat} \Rightarrow \text{real} \rangle$
where $\langle \text{restriction-}\sigma\text{-fun} - \equiv \text{restriction-}\sigma \text{ TYPE('b)} \rangle$

instance $\langle \text{proof} \rangle$

end

instantiation $\langle \text{fun} \rangle :: (\text{type}, \text{non-decseq-restriction-space}) \text{non-decseq-restriction-space}$
begin

definition *dist-fun* : $\langle ['a \Rightarrow 'b, 'a \Rightarrow 'b] \Rightarrow \text{real} \rangle$
where $\langle \text{dist-fun } f g \equiv \text{INF } n \in \text{restriction-related-set } f g. \text{restriction-}\sigma$
 $\text{TYPE('a} \Rightarrow 'b) \text{ } n \rangle$

definition *uniformity-fun* : $\langle (('a \Rightarrow 'b) \times ('a \Rightarrow 'b)) \text{ filter} \rangle$
where $\langle \text{uniformity-fun} \equiv \text{INF } e \in \{0 < ..\}. \text{principal } \{(x, y). \text{dist } x y < e\} \rangle$

definition *open-fun* :: $\langle 'a \Rightarrow 'b \rangle \text{ set} \Rightarrow \text{bool}$
where $\langle \text{open-fun } U \equiv \forall x \in U. \text{ eventually } (\lambda(x', y). x' = x \longrightarrow y \in U) \text{ uniformity} \rangle$

instance $\langle \text{proof} \rangle$

end

instance $\langle \text{fun} \rangle :: (\text{type}, \text{decseq-restriction-space}) \text{ decseq-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\langle \text{fun} \rangle :: (\text{type}, \text{strict-decseq-restriction-space}) \text{ strict-decseq-restriction-space}$
 $\langle \text{proof} \rangle$

instantiation $\langle \text{fun} \rangle :: (\text{type}, \text{restriction-}\delta) \text{ restriction-}\delta$
begin

definition *restriction- δ -fun* :: $\langle 'a \Rightarrow 'b \rangle \text{ itself} \Rightarrow \text{real}$
where $\langle \text{restriction-}\delta\text{-fun} \equiv \text{restriction-}\delta \text{ TYPE('b)} \rangle$

instance $\langle \text{proof} \rangle$

end

instance $\langle \text{fun} \rangle :: (\text{type}, \text{at-least-geometric-restriction-space}) \text{ at-least-geometric-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\langle \text{fun} \rangle :: (\text{type}, \text{geometric-restriction-space}) \text{ geometric-restriction-space}$
 $\langle \text{proof} \rangle$

lemma *dist-image-le-dist-fun* : $\langle \text{dist } (f \ x) \ (g \ x) \leq \text{dist } f \ g \rangle$
for $f \ g :: \langle 'a \Rightarrow 'b :: \text{non-decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

lemma *Sup-dist-image-le-dist-fun* : $\langle (\text{SUP } x. \text{dist } (f \ x) \ (g \ x)) \leq \text{dist } f \ g \rangle$
for $f \ g :: \langle 'a \Rightarrow 'b :: \text{non-decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

context **fixes** $f \ g :: \langle 'a \Rightarrow 'b :: \text{decseq-restriction-space} \rangle$ **begin**

lemma *reached-dist-fun* : $\langle \exists x. \text{dist } f \ g = \text{dist } (f \ x) \ (g \ x) \rangle$

⟨proof⟩

lemma *dist-fun-eq-Sup-dist-image* : ⟨ $\text{dist } f \ g = (\text{SUP } x. \text{dist } (f \ x) \ (g \ x))$ ⟩
 ⟨proof⟩

lemma *fun-restriction-space-Sup-properties* :
 ⟨ $\text{dist } (f \ x) \ (g \ x) \leq \text{dist } f \ g$ ⟩
 ⟨ $(\bigwedge x. \text{dist } (f \ x) \ (g \ x) \leq b) \implies \text{dist } f \ g \leq b$ ⟩
 ⟨proof⟩

end

4.2 Completeness

Actually we can obtain even better: when the instance *'b* of *decseq-restriction-space* is also an instance of *complete-space*, the type *'a* $\Rightarrow$ *'b* is an instance of *complete-space*.

This is because when *'b* is an instance of *decseq-restriction-space* (and not only *non-decseq-restriction-space*) the distance between two functions is reached (see $\exists x. \text{dist } ?f \ ?g = \text{dist } (?f \ x) \ (?g \ x)$).

The only remaining thing is to prove that completeness is preserved on higher-order.

instance ⟨*fun*⟩ :: (type, complete-decseq-restriction-space) complete-decseq-restriction-space
 ⟨proof⟩

instance ⟨*fun*⟩ :: (type, complete-strict-decseq-restriction-space) complete-strict-decseq-restriction-space
 ⟨proof⟩

instance ⟨*fun*⟩ :: (type, complete-at-least-geometric-restriction-space) complete-at-least-geometric-restriction-space
 ⟨proof⟩

instance ⟨*fun*⟩ :: (type, complete-geometric-restriction-space) complete-geometric-restriction-space
 ⟨proof⟩

4.3 Kind of Extensionality

context fixes $f :: \langle [a :: \text{metric-space}, b :: \text{type}] \Rightarrow c :: \text{decseq-restriction-space} \rangle$ **begin**

lemma *lipschitz-with-simplification*:
 ⟨ $\text{lipschitz-with } f \ \alpha \longleftrightarrow (\forall y. \text{lipschitz-with } (\lambda x. f \ x \ y) \ \alpha)$ ⟩

⟨proof⟩

lemma *non-expanding-simplification* :
 ⟨non-expanding $f \longleftrightarrow (\forall y. \text{non-expanding } (\lambda x. f\ x\ y))$ ⟩
 ⟨proof⟩

lemma *contraction-with-simplification*:
 ⟨contraction-with $f\ \alpha \longleftrightarrow (\forall y. \text{contraction-with } (\lambda x. f\ x\ y)\ \alpha)$ ⟩
 ⟨proof⟩

end

5 Product

The product type $'a \times 'b$ of to metric spaces is already instantiated as a metric space by setting $\text{dist } x\ y = \text{sqrt } ((\text{dist } (\text{fst } x)\ (\text{fst } y))^2 + (\text{dist } (\text{snd } x)\ (\text{snd } y))^2)$. Unfortunately, this definition is not compatible with the distance required by the *non-decseq-restriction-space*.. We first have to define a new product type with a trivial **typedef**.

5.1 Isomorphic Product Construction

5.1.1 Definition and First Properties

typedef ($'a, 'b$) prod_{max} (⟨ $(- \times_{max} / -)$ ⟩ [21, 20] 20) = ⟨UNIV :: ($'a \times 'b$) set⟩
morphisms *from-prod_{max} to-prod_{max}* ⟨proof⟩

declare *from-prod_{max}-inject* [simp]
from-prod_{max}-inverse [simp]

lemmas *to-prod_{max}-inject-simplified* [simp] = *to-prod_{max}-inject* [simplified]
and *to-prod_{max}-inverse-simplified*[simp] = *to-prod_{max}-inverse*[simplified]

lemmas *to-prod_{max}-induct-simplified* = *to-prod_{max}-induct*[simplified]
and *to-prod_{max}-cases-simplified* = *to-prod_{max}-cases* [simplified]
and *from-prod_{max}-induct-simplified* = *from-prod_{max}-induct*[simplified]
and *from-prod_{max}-cases-simplified* = *from-prod_{max}-cases* [simplified]

setup-lifting *type-definition-prod_{max}*

lift-definition $Pair_{max} :: \langle 'a \Rightarrow 'b \Rightarrow 'a \times_{max} 'b \rangle$ **is** $Pair \langle proof \rangle$

free-constructors $case-prod_{max}$ **for** $Pair_{max}$ fst_{max} snd_{max}
 $\langle proof \rangle$

lemma $fst_{max}-def : \langle fst_{max} \equiv map-fun from-prod_{max} id fst \rangle$
 $\langle proof \rangle$

lemma $fst_{max}-rep-eq : \langle fst_{max} x = fst (from-prod_{max} x) \rangle$
 $\langle proof \rangle$

lemma $fst_{max}-abs-eq [simp] : \langle fst_{max} (to-prod_{max} y) = fst y \rangle$
 $\langle proof \rangle$

lemma $fst_{max}-transfer [transfer-rule] : \langle rel-fun (pcr-prod_{max} (=) (=))$
 $(=) fst fst_{max} \rangle$
 $\langle proof \rangle$

lemma $snd_{max}-def : \langle snd_{max} \equiv map-fun from-prod_{max} id snd \rangle$
 $\langle proof \rangle$

lemma $snd_{max}-rep-eq : \langle snd_{max} x = snd (from-prod_{max} x) \rangle$
 $\langle proof \rangle$

lemma $snd_{max}-abs-eq [simp] : \langle snd_{max} (to-prod_{max} y) = snd y \rangle$
 $\langle proof \rangle$

lemma $snd_{max}-transfer [transfer-rule] : \langle rel-fun (pcr-prod_{max} (=)$
 $(=)) (=) snd snd_{max} \rangle$
 $\langle proof \rangle$

lemma $case-prod_{max}-def : \langle case-prod_{max} \equiv map-fun id (map-fun$
 $from-prod_{max} id) case-prod \rangle$
 $\langle proof \rangle$

lemma $case-prod_{max}-rep-eq : \langle case-prod_{max} f p = (case from-prod_{max}$
 $p \text{ of } (x, y) \Rightarrow f x y) \rangle$
 $\langle proof \rangle$

lemma $case-prod_{max}-abs-eq [simp] : \langle case-prod_{max} f (to-prod_{max} q)$
 $= (case q \text{ of } (x, y) \Rightarrow f x y) \rangle$
 $\langle proof \rangle$

lemma $case-prod_{max}-transfer [transfer-rule] : \langle rel-fun (=) (rel-fun$
 $(pcr-prod_{max} (=) (=)) (=) case-prod case-prod_{max} \rangle$
 $\langle proof \rangle$

5.2 Syntactic Sugar

The following syntactic sugar is of course recovered from the theory *HOL.Product-Type*.

nonterminal *tuple-args_{max}* and *patterns_{max}*

syntax

$-tuple_{max} :: 'a \Rightarrow tuple_args_{max} \Rightarrow 'a \times_{max} 'b \quad (\langle 1\langle -, / - \rangle \rangle)$
 $-tuple_arg_{max} :: 'a \Rightarrow tuple_args_{max} \quad (\langle - \rangle)$
 $-tuple_args_{max} :: 'a \Rightarrow tuple_args_{max} \Rightarrow tuple_args_{max} \quad (\langle -, / - \rangle)$
 $-pattern_{max} :: pttrn \Rightarrow patterns_{max} \Rightarrow pttrn \quad (\langle \langle -, / - \rangle \rangle)$
 $:: pttrn \Rightarrow patterns_{max} \quad (\langle - \rangle)$
 $-patterns_{max} :: pttrn \Rightarrow patterns_{max} \Rightarrow patterns_{max} \quad (\langle -, / - \rangle)$

translations

$\langle x, y \rangle \equiv CONST\ Pair_{max}\ x\ y$
 $-pattern_{max}\ x\ y \equiv CONST\ Pair_{max}\ x\ y$
 $-patterns_{max}\ x\ y \equiv CONST\ Pair_{max}\ x\ y$
 $-tuple_{max}\ x\ (-tuple_args_{max}\ y\ z) \equiv -tuple_{max}\ x\ (-tuple_arg_{max}\ (-tuple_{max}\ y\ z))$
 $\lambda \langle x, y, zs \rangle. b \equiv CONST\ case_prod_{max}\ (\lambda x\ \langle y, zs \rangle. b)$
 $\lambda \langle x, y \rangle. b \equiv CONST\ case_prod_{max}\ (\lambda x\ y. b)$
 $-abs\ (CONST\ Pair_{max}\ x\ y)\ t \rightarrow \lambda \langle x, y \rangle. t$
 — This rule accommodates tuples in *case C ...* $\langle x, y \rangle \dots \Rightarrow \dots$

The $\langle x, y \rangle$ is parsed as *Pair_{max} x y* because it is *logic*, not *pttrn*.

With this syntactic sugar, one can write *case a of* $\langle b, c, d, e \rangle \Rightarrow \langle c, d \rangle, \lambda \langle y, u \rangle. a, \lambda \langle a, b \rangle. \langle a, b, c, d, e \rangle, \lambda \langle a, b, c \rangle. a, \dots$ as for the type $'a \times 'b$.

lemmas *to-prod_{max}-tuple* [simp] = *Pair_{max}.abs-eq*[symmetric]
and *from-prod_{max}-tuple_{max}* [simp] = *Pair_{max}.rep-eq*

5.3 Product

We first redo the work of *Restriction-Spaces.Restricton-Spaces-Prod*.

instantiation *prod_{max}* :: (*restriction, restriction*) *restriction*
begin

lift-definition *restriction-prod_{max}* :: $\langle 'a \times_{max} 'b \Rightarrow nat \Rightarrow 'a \times_{max} 'b \rangle$ is $\langle (\downarrow) \rangle \langle proof \rangle$

lemma *restriction-prod_{max}-def'* : $\langle p \downarrow n = \langle fst_{max}\ p \downarrow n, snd_{max}\ p \downarrow n \rangle \rangle$
 $\langle proof \rangle$

instance $\langle proof \rangle$

end

instance $prod_{max} :: (restriction\text{-}space, restriction\text{-}space) restriction\text{-}space$
 $\langle proof \rangle$

instantiation $prod_{max} :: (restriction\text{-}\sigma, restriction\text{-}\sigma) restriction\text{-}\sigma$
begin

definition $restriction\text{-}\sigma\text{-}prod_{max} :: \langle ('a \times_{max} 'b) itself \Rightarrow nat \Rightarrow real \rangle$
where $\langle restriction\text{-}\sigma\text{-}prod_{max} - n \equiv$
 $max (restriction\text{-}\sigma \ TYPE('a) \ n) (restriction\text{-}\sigma \ TYPE('b) \ n) \rangle$

instance $\langle proof \rangle$
end

instantiation $prod_{max} :: (non\text{-}decseq\text{-}restriction\text{-}space, non\text{-}decseq\text{-}restriction\text{-}space)$
 $non\text{-}decseq\text{-}restriction\text{-}space$
begin

definition $dist\text{-}prod_{max} :: \langle ['a \times_{max} 'b, 'a \times_{max} 'b] \Rightarrow real \rangle$
where $\langle dist\text{-}prod_{max} \ f \ g \equiv INF \ n \in restriction\text{-}related\text{-}set \ f \ g. \ re\text{-}$
 $striction\text{-}\sigma \ TYPE('a \times_{max} 'b) \ n \rangle$

definition $uniformity\text{-}prod_{max} :: \langle (('a \times_{max} 'b) \times 'a \times_{max} 'b) filter \rangle$
where $\langle uniformity\text{-}prod_{max} \equiv INF \ e \in \{0 < ..\}. \ principal \ \{(x, y). \ dist$
 $x \ y < e\} \rangle$

definition $open\text{-}prod_{max} :: \langle ('a \times_{max} 'b) set \Rightarrow bool \rangle$
where $\langle open\text{-}prod_{max} \ U \equiv \forall x \in U. \ eventually \ (\lambda(x', y). \ x' = x \longrightarrow$
 $y \in U) \ uniformity \rangle$

instance
 $\langle proof \rangle$

end

instance $prod_{max} :: (decseq\text{-}restriction\text{-}space, decseq\text{-}restriction\text{-}space)$
 $decseq\text{-}restriction\text{-}space$
 $\langle proof \rangle$

instance $prod_{max} :: (strict\text{-}decseq\text{-}restriction\text{-}space, strict\text{-}decseq\text{-}restriction\text{-}space)$
 $strict\text{-}decseq\text{-}restriction\text{-}space$
 $\langle proof \rangle$

instantiation $prod_{max} :: (restriction\text{-}\delta, restriction\text{-}\delta) restriction\text{-}\delta$
begin

definition $\text{restriction-}\delta\text{-prod}_{max} :: \langle ('a \times_{max} 'b) \text{ itself} \Rightarrow \text{real} \rangle$
where $\langle \text{restriction-}\delta\text{-prod}_{max} - \equiv \max (\text{restriction-}\delta \text{ TYPE}('a))$
 $(\text{restriction-}\delta \text{ TYPE}('b)) \rangle$

instance $\langle \text{proof} \rangle$

end

instance $\text{prod}_{max} :: (\text{at-least-geometric-restriction-space}, \text{at-least-geometric-restriction-space})$
 $\text{at-least-geometric-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\text{prod}_{max} :: (\text{geometric-restriction-space}, \text{geometric-restriction-space})$
 $\text{geometric-restriction-space}$
 $\langle \text{proof} \rangle$

lemma $\text{max-dist-projections-le-dist-prod}_{max} :$
 $\langle \max (\text{dist} (\text{fst}_{max} p1) (\text{fst}_{max} p2)) (\text{dist} (\text{snd}_{max} p1) (\text{snd}_{max} p2))$
 $\leq \text{dist} p1 p2 \rangle$
 $\langle \text{proof} \rangle$

5.4 Completeness

5.4.1 Preliminaries

default-sort $\text{non-decseq-restriction-space}$ — Otherwise we should always specify.

lemma $\text{restriction-}\sigma\text{-prod}_{max}\text{-commute} :$
 $\langle \text{restriction-}\sigma \text{ TYPE}('b \times_{max} 'a) = \text{restriction-}\sigma \text{ TYPE}('a \times_{max}$
 $'b) \rangle$
 $\langle \text{proof} \rangle$

definition $\text{dist-left-prod}_{max} :: \langle ('a \times_{max} 'b) \text{ itself} \Rightarrow 'a \Rightarrow 'a \Rightarrow \text{real} \rangle$
where $\langle \text{dist-left-prod}_{max} - x y \equiv \text{INF } n \in \text{restriction-related-set } x$
 $y. \text{restriction-}\sigma \text{ TYPE}('a \times_{max} 'b) n \rangle$

definition $\text{dist-right-prod}_{max} :: \langle ('a \times_{max} 'b) \text{ itself} \Rightarrow 'b \Rightarrow 'b \Rightarrow \text{real} \rangle$
where $\langle \text{dist-right-prod}_{max} - x y \equiv \text{INF } n \in \text{restriction-related-set } x$
 $y. \text{restriction-}\sigma \text{ TYPE}('a \times_{max} 'b) n \rangle$

lemma $\text{dist-right-prod}_{max}\text{-is-dist-left-prod}_{max} :$
 $\langle \text{dist-right-prod}_{max} \text{ TYPE}('b \times_{max} 'a) = \text{dist-left-prod}_{max} \text{ TYPE}('a$
 $\times_{max} 'b) \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{dist-le-dist-left-prod}_{max} : \langle \text{dist } x \ y \leq \text{dist-left-prod}_{max} \ \text{TYPE}('a \times_{max} 'b) \ x \ y \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{dist-le-dist-right-prod}_{max} : \langle \text{dist } x \ y \leq \text{dist-right-prod}_{max} \ \text{TYPE}('b \times_{max} 'a) \ x \ y \rangle$
 $\langle \text{proof} \rangle$

lemma
fixes $p1 \ p2 :: \langle 'a :: \text{decseq-restriction-space} \times_{max} 'b :: \text{decseq-restriction-space} \rangle$
shows $\text{dist-prod}_{max}\text{-le-max-dist-left-prod}_{max}\text{-dist-right-prod}_{max} :$
 $\langle \text{dist } p1 \ p2 \leq \max (\text{dist-left-prod}_{max} \ \text{TYPE}('a \times_{max} 'b) \ (\text{fst}_{max} \ p1) \ (\text{fst}_{max} \ p2))$
 $(\text{dist-right-prod}_{max} \ \text{TYPE}('a \times_{max} 'b) \ (\text{snd}_{max} \ p1) \ (\text{snd}_{max} \ p2)) \rangle$
 $\langle \text{proof} \rangle$

default-sort *type* — Back to normal.

5.4.2 Complete Restriction Space

When the instances $'a$ and $'b$ of *decseq-restriction-space* are also instances of *complete-space*, the type $'a \times_{max} 'b$ is an instance of *complete-space*.

lemma $\text{restriction-chain-prod}_{max}\text{-iff} :$
 $\langle \text{restriction-chain } \sigma \longleftrightarrow \text{restriction-chain } (\lambda n. \text{fst}_{max} (\sigma \ n)) \wedge$
 $\text{restriction-chain } (\lambda n. \text{snd}_{max} (\sigma \ n)) \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{restriction-tendsto-prod}_{max}\text{-iff} :$
 $\langle \sigma \dashrightarrow \Sigma \longleftrightarrow (\lambda n. \text{fst}_{max} (\sigma \ n)) \dashrightarrow \text{fst}_{max} \Sigma \wedge (\lambda n. \text{snd}_{max} (\sigma \ n)) \dashrightarrow \text{snd}_{max} \Sigma \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{restriction-convergent-prod}_{max}\text{-iff} :$
 $\langle \text{restriction-convergent } \sigma \longleftrightarrow \text{restriction-convergent } (\lambda n. \text{fst}_{max} (\sigma \ n)) \wedge$
 $\text{restriction-convergent } (\lambda n. \text{snd}_{max} (\sigma \ n)) \rangle$
 $\langle \text{proof} \rangle$

instance $\text{prod}_{max} :: (\text{complete-decseq-restriction-space}, \text{complete-decseq-restriction-space})$
 $\text{complete-decseq-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\text{prod}_{\text{max}} :: (\text{complete-strict-decseq-restriction-space}, \text{complete-strict-decseq-restriction-space})$
 $\text{complete-strict-decseq-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\text{prod}_{\text{max}} :: (\text{complete-at-least-geometric-restriction-space}, \text{complete-at-least-geometric-restriction-space})$
 $\text{complete-at-least-geometric-restriction-space}$
 $\langle \text{proof} \rangle$

instance $\text{prod}_{\text{max}} :: (\text{complete-geometric-restriction-space}, \text{complete-geometric-restriction-space})$
 $\text{complete-geometric-restriction-space}$
 $\langle \text{proof} \rangle$

When the types $'a$ and $'b$ share the same restriction sequence, we have the following equality.

lemma $\text{same-restriction-}\sigma\text{-imp-restriction-}\sigma\text{-prod}_{\text{max}}\text{-is [simp] :$
 $\langle \text{restriction-}\sigma \text{ TYPE}('b :: \text{non-decseq-restriction-space}) =$
 $\text{restriction-}\sigma \text{ TYPE}('a :: \text{non-decseq-restriction-space}) \implies$
 $\text{restriction-}\sigma \text{ TYPE}('a \times_{\text{max}} 'b) = \text{restriction-}\sigma \text{ TYPE}('a) \rangle$
 $\langle \text{proof} \rangle$

lemma $\text{same-restriction-}\sigma\text{-imp-dist-prod}_{\text{max}}\text{-eq-max-dist-projections :$
 $\langle \text{dist } p1 \text{ } p2 = \text{max } (\text{dist } (\text{fst}_{\text{max}} \text{ } p1) (\text{fst}_{\text{max}} \text{ } p2)) (\text{dist } (\text{snd}_{\text{max}} \text{ } p1)$
 $(\text{snd}_{\text{max}} \text{ } p2)) \rangle$
if $\text{same-restriction-}\sigma \text{ [simp] : } \langle \text{restriction-}\sigma \text{ TYPE}('b) = \text{restriction-}\sigma$
 $\text{TYPE}('a) \rangle$
for $p1 \text{ } p2 :: \langle 'a :: \text{decseq-restriction-space} \times_{\text{max}} 'b :: \text{decseq-restriction-space} \rangle$
 $\langle \text{proof} \rangle$

Now, one can write things like $v \llbracket x, y \rrbracket. f \llbracket x, y \rrbracket$.

We could of course imagine more support for $'a \times_{\text{max}} 'b$ type, but as restriction spaces are intended to be used without recourse to metric spaces, we have not undertaken this task for the time being.

6 Main entry Point