Bicompletions of distance matrices

To Samson Abramsky on the occasion of his 60th birthday

Dusko Pavlovic
Email: dusko.pavlovic@rhul.ac.uk

Royal Holloway, University of London, and University of Twente

Abstract. In the practice of information extraction, the input data are usually arranged into *pattern matrices*, and analyzed by the methods of linear algebra and statistics, such as principal component analysis. In some applications, the tacit assumptions of these methods lead to wrong results. The usual reason is that the matrix composition of linear algebra presents information as flowing in waves, whereas it sometimes flows in particles, which seek the shortest paths. This wave-particle duality in computation and information processing has been originally observed by Abramsky. In this paper we pursue a particle view of information, formalized in *distance spaces*, which generalize metric spaces, but are slightly less general than Lawvere's *generalized metric spaces*. In this framework, the task of extracting the 'principal components' from a given matrix of data boils down to a *bicompletion*, in the sense of enriched category theory. We describe the bicompletion construction for distance matrices. The practical goal that motivates this research is to develop a method to estimate the hardness of attack constructions in security.

1 Introduction

Dedication. When Samson Abramsky offered me the position of 'Human Capital Mobility Research Fellow' in his group at Imperial College back in 1993, I was an ex-programmer with postdoctoral experience in category theory. It was a questionable investment. Category theoretical models of computation were, of course, already in use in theoretical computer science; but the emphasis was on the word 'theoretical'. A couple of years later, I left academia to build software using categorical models. While it is clear and well understood that Samson's work and results consolidated and enriched categorical methods of theoretical computer science, their applications in the practice of computation may not be as well known. In the long run, I believe, the impact of the methods and of the approach that we learned from Samson will become increasingly clear, as the abstract structures that we use, including the fully abstract ones, are becoming more concrete, more practical, and more often indispensable.

In the present paper, I venture into an extended exercise in enriched category theory, directly motivated by concrete problems of security [17, 16] and of data analysis [18]. Although the story is not directly related to Samson's own work, I hope that it is appropriate for the occasion, since he is the originator of the general spirit of categorical variations on computational themes, even if I can never hope to approach his balance and style.

Motivation: Distances between algorithms

Suppose that you are given an algorithm a, and you need to construct another algorithm b, such that some predicate $P(a, b)$ is satisfied. Or more concretely, suppose that a is a software system, and b should be an attack on a, contradicting a's security claim by realizing a property $P(a, b)$. Since reverse engineering is easy [2, 5], we can assume that the code of a is readily available, and your task is thus to code the attack b. Note that a is in principle an algorithmic pattern, that can be implemented in many ways, and may have many versions and instances. So your attack b should also be an algorithmic pattern, related to a by some polymorphic transformation. The derivation of b from a should thus be polymorphic, i.e. a uniform construction: it should be a program p that inputs a description of a and outputs a corresponding description $p(a) = b$. How hard is it to find p? An approach to answering such questions is suggested in algorithmic information theory [25, 13]. The notion of *Kolmogorov complexity* is that the distance from an algorithm a to an algorithm b can be measured by the length of the shortest programs that construct b from a, i.e.

$$d\,(a, b) = \bigwedge_{p(a)=b} |p| \tag{1}$$

where $|p|$ denotes the length of the program p. It is easy to see that the above formula yields the triangle law $d\,(a, b) + d\,(b, c) \stackrel{+}{\geq} d\,(a, c)$, where the superscript '+' means that the uniform order relation $\geq$ is taken up to a constant, which is in this case the length of the program composition operation, needed to get a program to construct c from a by composing a program that constructs c from b with a program that constructs b from a. Algorithmic information theory always works with such order relations [13, 4]. The equation $d\,(a, a) \stackrel{+}{=} 0$ holds in the same sense, up to the constant length of the shortest identity program, that just inputs and outputs identical data. This distance of algorithms, in the style of Kolmogorov complexity, was proposed in [16] as a tool to measure how hard it is to construct an attack on a given system. The point was that a system could be effectively secure even when some attacks on it exist, provided that these attacks are provably hard to construct. The goal of the present note is to spell out some general results about distance that turn out to be needed for this particular application.

But why do we need general results about distances to answer the concrete question about the hardness of constructing attack programs from system programs? The reason is that the task of finding an attack algorithm not too far from a system algorithm naturally leads to the task of construcing a *completion* of the space around the system algorithm. The attacker sees the system, and may be familiar with some other algorithms in its neighborhood; but it is not known whether an attack exists, and how far it is. The task of discovering the attack is the task of completing the space around the system. And the construction of a completion is easier in general, than in some concrete cases.

How does a real attacker search for an algorithm p to derive an attack b from the system a? He is not trying to guess the construction in isolation, but in the context of his algorithmic knowledge. This knowledge has at least two components. On one hand, there is some algorithmic knowledge A about the software systems $a_0, a_1, a_2 \ldots$, and a

distance measure $A \times A \xrightarrow{d_A} [0,\infty]$ between them, which express how they are related with each other. On the other hand, there is some algorithmic knowledge B about the attacks $b_0, b_1, b_2 \ldots$, and their distances $B \times B \xrightarrow{d_B} [0,\infty]$. Last but not least, there is some knowledge which attacks are related to which systems. This knowledge is expressed as a distance matrix $A \times B \xrightarrow{\Phi} [0,\infty]$, where shorter distances suggest easier attacks. In order to determine whether there are any attacks in the proximity of a given system a, our task is to conjoin the distance space A of systems with the distance space B of attacks consistently with the distance matrix $A \times B \xrightarrow{\Phi} [0,\infty]$ where the observed connections between the systems and attacks are recorded. In this conjoined space, we need to find the unknown attacks close to the target system. We find them by completing the space of the known attacks. But since the completion is in general an infinite object, we first study it abstractly, to determine how to construct just the parts of interest.

Related work. The completions that we study are based on Lawvere's view of metric spaces as enriched categories [10]. Lawvere's generalized metric spaces were extensively used in denotational semantics of programming languages [22, 3, 9], and recently in ecology [12], following a renewed mathematical interest in the enriched category approach [11]. In my own work, closely related results arose in the framework of information extraction and concept analysis [18]. That work was, however, not based on distance spaces as categories enriched in the additive monoid $[0,\infty]$, but on *proximity spaces*, or *proxets*, as categories enriched in the multiplicative monoid $[0,1]$. Proxets are a more natural framework for concept analysis, because they generalize posets, as categories enriched over the multiplicative monoid $\{0,1\}$, and the existing theory and intuitions are largely based on posets. Distance spaces, on the other hand, appear to be a more convenient framework for relating algorithms.

Outline of the paper. In Sec. 2 we define distance spaces and describe some examples. In Sec. 3 we spell out the notions of limit in distance spaces, the basic completion constructions, and the adjunctions as they arise from the limit preserving morphisms. In Sec. 4, we introduce distance matrices, and describe their decomposition. In Sec. 5 we put the previously presented components together to construct the bicompletions of distance matrices. Sec. 6 provides a summary of the obtained results and a discussion of future work.

2 Distance spaces

2.1 Definition and background

Definition 2.1. *A* distance space *is a set A with a* metric $d_A : A \times A \to [0,\infty]$ *which is*

- *reflexive:* $d(x,x) = 0$,
- *transitive:* $d(x,y) + d(y,z) \geq d(x,z)$, *and*
- *antisymmetric:* $d(x,y) = 0 = d(y,x) \implies x = y$

A contraction *between the distance spaces A and B is a function $f : A \to B$ such that for all $x, y \in A$ holds $d_A(x,y) \geq d_B(fx, fy)$. The category of distance spaces and contractions is denoted* Dist.

Background. In topology, distance spaces have been studied since the 1930s under the name *quasi-metric spaces* [23, 8]. The prefix 'quasi' refers to the fact that the metric symmetry law $d(x, y) = d(y, x)$ is not necessarily satisfied. When the antisymmetry law is not satisfied either, then the topologists speak of *pseudo-quasi-metric spaces* [24]. Lawvere [10] observed that pseudo-quasi-metric spaces, which he called *generalized metric spaces*, could be viewed as enriched categories [7]. They are enriched over the additive monoid $[0, \infty]$, viewed as a monoidal category with a uniqe arrow $x \to y$ if and only if $x \geq y$. The distance $d(x, y) \in [0, \infty]$ is thus viewed as the 'hom-set' in the enriched sense. Lawvere's main result was the characterization of the Cauchy completion of a metric space as an enriched category construction. This view of distances and contractions turned out to provide an alternative to domains for denotational semantics [22], and their categorical completions were elaborated in [3, 9]. Distance spaces as defined in 2.1 are a special case of generalized metric spaces, since they are required to satisfy the antisymmetry law. This is mainly a matter of convenience, as the following lemma shows.

Lemma 2.2. *A map $d_A : A \times A \to [0, \infty]$ which is reflexive and transitive in the sense of Def. 2.1 is also antisymmetric if and only if it satisfies either of the following equivalent conditions*

- $(\forall z.\, d(z, x) = d(z, y)) \Rightarrow x = y$
- $(\forall z.\, d(x, z) = d(y, z)) \Rightarrow x = y$

Proof. In the presence of transitivity and reflexivity, $d(x, y) = 0$ holds if and only if $\forall z.\, d(z, x) \geq d(z, y)$, or equivalently if and only if $\forall z.\, d(x, z) \leq d(y, z)$. The result follows. □

Corollary 2.3. *Distance spaces are just the* skeletal *generalized metric spaces.*

2.2 Examples

The first example of a distance space is, of course, the interval $[0, \infty]$ itself, with the metric

$$d_{[0,\infty]}(x, y) \;=\; x \multimap y = \begin{cases} y - x & \text{if } x < y \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The $\multimap$ notation is convenient because the operation $d_{[0,\infty]} = \multimap\colon [0, \infty] \times [0, \infty] \to [0, \infty]$ makes $[0, \infty]$ into a closed category

$$x + y \;\geq\; z \iff x \geq y \multimap z \tag{3}$$

Any metric space is obviously an example of a distance space. But in distance spaces, the distance $d(a, b)$ from a to b does not have to be the same as the distance $d(b, a)$ from b to a. E.g., a may be on a hill, and b in the valley, and traveling one way may be easier than traveling the other way. For our purposes described in the Introduction, this distinction is quite important, since a program constructing an attack b from a system

code a does not have to be related in any obvious way to the program performing the construction the other way.

For a non-metric family of distance spaces, take any poset $(S, \sqsubseteq_S)$ and define a distance space $(\mathsf{W}S, d_{\mathsf{W}S})$ by setting $d_{\mathsf{W}S}(x, y) = 0$ if $x \sqsubseteq_S y$, otherwise ∞. The other way around, any distance space A induces two posets, ΥA and ΛA, with the same underlying set and

$$x \sqsubseteq_{\Upsilon A} y \iff d_A(x, y) = 0 \qquad\qquad x \sqsubseteq_{\Lambda A} y \iff d_A(x, y) < \infty$$

The constructions W, Υ and Λ form the adjunctions $\Lambda \dashv \mathsf{W} \dashv \Upsilon : \mathsf{Dist} \to \mathsf{Pos}$. Since $\mathsf{W} : \mathsf{Pos} \hookrightarrow \mathsf{Dist}$ is an embedding, Pos is thus a reflective and correflective subcategory of Dist.

Distance spaces are thus a common generalization of posets and metric spaces. For an example not arising from posets of metric spaces, take any family of sets $\mathcal{X} \subseteq \wp X$, and define

$$d(x, y) = |y \setminus x| \tag{4}$$

The distance of x and y is thus the number of elements of y that are not in x. If X is a set of terms, say in a dictionary, and $\mathcal{X}$ is a set of documents, each viewed as a set of terms, then the distance between two documents is the number of terms that occur in one document and not in the other. In natural language processing, documents are usually presented as multisets (bags) of terms, and the distance is defined in terms of multiset subtraction, which generalizes the set difference used in (4). In any case, it is clear that the asymmetry of the notion of distance is as essential for such applications as it is for the one described in the Introduction.

2.3 Basic constructions

Given two distance space A and B, we define:

- *dual* A^o: take the same underlying set and define the dual metric to be $d_{A^o}(x, y) = d_A(y, x)$;
- *product* $A \times B$: take the cartesian product of the underlying sets and set the product metric to be $d_{A\times B}(x, u, y, v) = d_A(x, y) \vee d_B(u, v)$
- the *power* B^A: take the set of contractions $\mathsf{Dist}(A, B)$ to be the underlying set and set the metric to be $d_{B^A}(f, g) = \bigvee_{x\in A} d_B(fx, gx)$.

These constructions induce the natural correspondences

$$\mathsf{Dist}(A, B) \times \mathsf{Dist}(A, C) \cong \mathsf{Dist}(A, B \times C) \quad \text{and} \quad \mathsf{Dist}(A \times B, C) \cong \mathsf{Dist}(A, C^B)$$

Terminology. Contractions $f : A \to B$ are called *covariant*, whereas contractions $f : A^o \to B$ are *contravatiant*.

3 Sequences and their limits

3.1 Left and right sequences

Intuitively, to complete a metric space means to add enough points so that every suitably convergent sequence has a limit. But usually many different sequences have the same limit. The main problem of the standard theory of completions is to recognize such sequences. The categorical approach overcomes this problem by considering *canonical* sequences. Instead of the sequences $s, t : \mathbb{N} \to A$ such that $\lim_{i\to\infty} s_i = \psi = \lim_{i\to\infty} t_i$, we consider a canonical sequence $\psi : A \to [0,\infty]$ where ψx intuitively denotes the distance from ψ to x.

Definition 3.1. *In a distance space A, a* (canonical) sequence *is defined to be a contraction into* $[0,\infty]$. *More precisely, we define that*

- *a* left *sequence is a covariant contraction* $\overleftarrow{\lambda} : A \to [0,\infty]$
  - *we write its value at* $x \in A$ *as* $\overleftarrow{\lambda} x$
- *a* right *sequence is a contravariant contraction* $\overrightarrow{\varrho} : A^o \to [0,\infty]$
  - *we write its value at* $x \in A$ *as* $x\overrightarrow{\varrho}$.

Each of the sets of sequences

$$\overleftarrow{A} = \left([0,\infty]^A\right)^o \qquad \text{and} \qquad \overrightarrow{A} = [0,\infty]^{(A^o)}$$

forms a distance space, with the metrics

$$d_{\overleftarrow{A}}\left(\overleftarrow{\lambda}, \overleftarrow{\theta}\right) = \bigvee_{x\in A} \overleftarrow{\theta} x \multimap \overleftarrow{\lambda} x \qquad \text{and} \qquad d_{\overrightarrow{A}}\left(\overrightarrow{\varrho}, \overrightarrow{\mu}\right) = \bigvee_{x\in A} x\overrightarrow{\varrho} \multimap x\overrightarrow{\mu}$$

Remarks. The conditions $d_A(x,y) \geq \overleftarrow{\lambda} x \multimap \overleftarrow{\lambda} y$ and $d_A(x,y) \geq y\overrightarrow{\varrho} \multimap x\overrightarrow{\varrho}$, which say that $\overleftarrow{\lambda}$ and $\overrightarrow{\varrho}$ are left and right contraction respectively, are by (3) respectively equivalent to

$$\overleftarrow{\lambda} x + d(x,y) \geq \overleftarrow{\lambda} y \qquad\qquad d(x,y) + y\overrightarrow{\varrho} \geq x\overrightarrow{\varrho}$$

3.2 Limits

Definition 3.2. *An element u of a distance space A is an* upper bound *of a right sequence $\overrightarrow{\varrho}$ in A if for all $x \in A$ holds*

$$x\overrightarrow{\varrho} \geq d_A(x,u) \tag{5}$$

An element ℓ of a distance space A is a lower bound *of a left sequence $\overleftarrow{\lambda}$ in A if for all $y \in A$ holds*

$$\overleftarrow{\lambda} y \geq d_A(\ell, y) \tag{6}$$

Proposition 3.3. *An element $u \in A$ is an upper bound $\overrightarrow{\varrho}$ and $\ell \in A$ is a lower bound of $\overleftarrow{\lambda}$ if and only if the following conditions hold for all $x, y \in A$*

$$d_A(u, y) \geq \bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d_A(x, y) \tag{7}$$

$$d_A(x, \ell) \geq \bigvee_{y \in A} \overleftarrow{\lambda} y \multimap d_A(x, y) \tag{8}$$

Proof. Condition (3) implies that (9) and (10) are respectively equivalent with

$$x\overrightarrow{\varrho} + d_A(u, y) \geq d_A(x, y) \tag{9}$$

$$d_A(x, \ell) + \overleftarrow{\lambda} y \geq d_A(x, y) \tag{10}$$

The claim follows by instantiating y to u in (7) and x to ℓ in (8). □

Definition 3.4. *The* supremum $\coprod \overrightarrow{\varrho}$ *of the right sequence* $\overrightarrow{\varrho}$ *and the* infimum $\prod \overleftarrow{\lambda}$ *of the left sequence* $\overleftarrow{\lambda}$ *are the elements of A that satisfy for every $x, y \in A$*

$$d_A\left(\coprod \overrightarrow{\varrho}, y\right) = \bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d_A(x, y) \tag{11}$$

$$d_A\left(x, \prod \overleftarrow{\lambda}\right) = \bigvee_{y \in A} \overleftarrow{\lambda} y \multimap d_A(x, y) \tag{12}$$

Suprema and infima constitute the limits *of a distance space.*

The distance space A is right (resp. left) complete *if every right (resp. left) sequence has a limit. The suprema and the infima thus yield the operations*

$$\coprod : \overrightarrow{A} \to A \qquad \text{and} \qquad \prod : \overleftarrow{A} \to A$$

One apparent shortcoming of treating sequences categorically, i.e. saturating them to canonical sequences, is that it is not obvious how to define continuity, i.e. how to distinguish the contractions which preserve suprema or infima. Clearly, a left continuous contraction $f : A \to B$ should map the infimum of a left sequence $\overleftarrow{\lambda}$ in A into the infimum of the f-image of $\overleftarrow{\lambda}$ in B. But what is the f-image of $\overleftarrow{\lambda} : A \to [0, \infty]$ in B? This question calls for a slight generalization of the concept of sequence, and limit.

3.3 Weighted limits

Limits are a special case of *weighted limits*, which are studied in general enriched categories [7, Ch. 3]. We just sketch theory of weighted limits in distance spaces.

Definition 3.5. *For distance spaces A and K we define*

- left diagrams *as pairs of contractions* $\left\langle k : K \to A, \overleftarrow{\lambda} : K \to [0, \infty] \right\rangle$
- right diagrams *as pairs of contractions* $\left\langle k : K \to A, \overrightarrow{\varrho} : K^o \to [0, \infty] \right\rangle$

Terminology and notation. The component $k : K \to A$ of a diagram is called its *shape*. Using the angular brackets to denote the functions into cartesian products, we also write

- $\left\langle k, \overleftarrow{\lambda} \right\rangle : K \to A \times [0,\infty]$ for $\left\langle k : K \to A, \overleftarrow{\lambda} : K \to [0,\infty] \right\rangle$
- $\left\langle k, \overrightarrow{\varrho}^{o} \right\rangle : K \to A \times [0,\infty]^o$ for $\left\langle k : K \to A, \overrightarrow{\varrho} : K^o \to [0,\infty] \right\rangle$

Definition 3.6. *The* weighted supremum $\coprod_{\overrightarrow{\varrho}} k$ *of the right diagram* $\langle k, \overrightarrow{\varrho}^{o} \rangle : K \to A \times [0,\infty]^o$ *and the* weighted infimum $\prod_{\overleftarrow{\lambda}} k$ *of the left diagram* $\langle k, \overleftarrow{\lambda} \rangle : K \to A \times [0,\infty]$ *are the elements of* A *that satisfy for every* $x, y \in A$

$$d_A\left(\coprod_{\overrightarrow{\varrho}} k, y\right) = \bigvee_{x \in K} x\overrightarrow{\varrho} \multimap d_A\left(kx, y\right) \tag{13}$$

$$d_A\left(x, \prod_{\overleftarrow{\lambda}} k\right) = \bigvee_{y \in K} \overleftarrow{\lambda} y \multimap d_A\left(x, ky\right) \tag{14}$$

Remarks. Limits arise as a special case of weighted limits, by viewing sequences as diagrams of shape $k = \mathrm{id} : A \to A$. A contraction $f : A \to B$ thus maps, say, a left sequence $\langle \mathrm{id}, \overleftarrow{\lambda} \rangle : A \to A \times [0,\infty]$ to the diagram $\langle f, \overleftarrow{\lambda} \rangle : A \to B \times [0,\infty]$ in B. More generally, it maps a left sequence $\langle k, \overleftarrow{\lambda} \rangle : K \to A \times [0,\infty]$ to the diagram $\langle f \circ k, \overleftarrow{\lambda} \rangle : K \to B \times [0,\infty]$ in B. It is thus clear and easy to state what it means that a contraction preserves a weighted limit.

Definition 3.7. *A contraction* $f : A \to B$ *preserves*

- *weighted suprema if* $f\left(\coprod_{\overrightarrow{\varrho}} k\right) = \coprod_{\overrightarrow{\varrho}} (f \circ k)$*, and*
- *weighted infima if* $f\left(\prod_{\overleftarrow{\lambda}} k\right) = \prod_{\overleftarrow{\lambda}} (f \circ k)$.

On the other hand, although convenient to work with, weighted limits of diagrams in distance spaces also boil down to the limits of suitable sequences. We just state this fact, since it simplifies the construction of the completions; but leave the proof for another paper, since the proof construction is not essential for the goal of the present paper.

Proposition 3.8. *A distance space has*

- *the weighted suprema of all right diagrams if and only if it has the suprema of all right sequences;*
- *the weighted infima of all left diagrams if and only if it has the infima of all left sequences.*

3.4 Completions

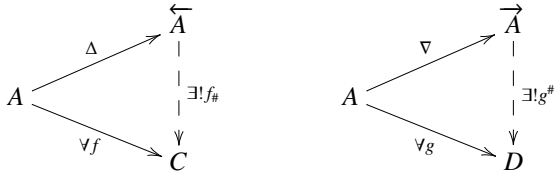
Every element a of a distance space A induces two *representable* sequences

$$\begin{aligned} \Delta a \;:\; A &\to [0,\infty] & \qquad \nabla a \;:\; A^o &\to [0,\infty] \\ x &\mapsto d_A\left(a, x\right) & x &\mapsto d_A\left(x, a\right) \end{aligned}$$

These induced contractions $\Delta : A \to \overleftarrow{A}$ and $\nabla : A \to \overrightarrow{A}$ correspond to the *Yoneda-Cayley embeddings* [15, Sec. III.2]. They make $\overleftarrow{A}$ into the lower completion, and $\overrightarrow{A}$ into the upper completion of the distance space A.

Proposition 3.9. *$\overleftarrow{A}$ is left complete and $\overrightarrow{A}$ is right complete. Each of them is universal among distance spaces with the corresponding completeness properties, in the sense that*

- *any monotone $f : A \to C$ into a complete distance space C induces a unique $\prod$-preserving morphism $f_\# : \overleftarrow{A} \to C$ such that $f = f_\# \circ \Delta$;*
- *any monotone $g : A \to D$ into a cocomplete distance space D induces a unique $\coprod$-preserving morphism $g^\# : \overrightarrow{A} \to D$ such that $g = g^\# \circ \nabla$.*



These constructions for have been thoroughly analyzed in [3, 9]. Here we just state the basic facts that justify our notations, and substantiate the further developments.

Proposition 3.10. *("The Yoneda Lemma") For every $\overrightarrow{\varphi} \in \overrightarrow{A}$ and $\overleftarrow{\lambda} \in \overleftarrow{A}$ and holds*

$$a\overrightarrow{\varphi} = \bigvee_{x \in A} x\,(\nabla a) \multimap x\overrightarrow{\varphi} \;=\; d_{\overrightarrow{A}}\left(\nabla a, \overrightarrow{\varphi}\right)$$
$$\overleftarrow{\lambda} b = \bigvee_{x \in A} (\Delta b)\, x \multimap \overleftarrow{\lambda} x \;=\; d_{\overleftarrow{A}}\left(\overleftarrow{\lambda}, \Delta b\right)$$

Instantiating in the preceding proposition $\overleftarrow{\lambda}$ to Δa and $\overrightarrow{\varphi}$ to ∇b yields

Corollary 3.11. *The embeddings $\Delta : A \to \overleftarrow{A}$ and $\nabla : A \to \overrightarrow{A}$ are isometries*

$$d_A\,(a, b) \;=\; d_{\overrightarrow{A}}\,(\nabla a, \nabla b) \;=\; d_{\overleftarrow{A}}\,(\Delta a, \Delta b)$$

3.5 Adjunctions

Notation. In any distance space A, if is often convenient to abbreviate $d_A\,(x, y) = 0$ to $x \underset{A}{\rightsquigarrow} y$. For $f, g : A \to B$, it is easy to see that $f \underset{B^A}{\rightsquigarrow} g$ if and only if $fx \underset{B}{\rightsquigarrow} gx$ for all $x \in A$.

Proposition 3.12. *For any contraction $f : A \to B$ holds*

$$(a) \iff (b) \iff (c) \quad \textit{and} \quad (d) \iff (e) \iff (f)$$

where

(a) $f\left(\coprod \overrightarrow{\varphi}\right) = \coprod_f \left(\overrightarrow{\varphi}\right)$
(b) $\exists f_* : B \to A\ \forall x \in A\ \forall y \in B.\ d_B(fx, y) = d_A(x, f_* y)$
(c) $\exists f_* : B \to A.\ \mathrm{id}_A \rightsquigarrow f_* f \wedge f f_* \rightsquigarrow \mathrm{id}_B$
(d) $f\left(\prod \overleftarrow{\lambda}\right) = \prod_f \left(\overleftarrow{\lambda}\right)$
(e) $\exists f^* : B \to A\ \forall x \in A\ \forall y \in B.\ d_B(f^* y, x) = d_A(y, fx)$
(f) $\exists f^* : B \to A.\ f^* f \rightsquigarrow \mathrm{id}_A \wedge \mathrm{id}_B \rightsquigarrow f f^*$

Each of the morphisms f^* and f_* is uniquely determined by f, whenever they exist.

Definition 3.13. *A* right adjoint *is a contraction satisfying (a-c) of Prop. 3.12; a* left adjoint *satisfies (d-f). A* (distance) adjunction *between the distance spaces A and B is a pair of contractions $f^* : A \rightleftarrows B : f_*$ related as in (b-c) and (e-f).*

Equations (11) and (12) immediately yield the following fact.

Proposition 3.14. *Limits are adjoints to the Yoneda-Cayley embeddings:*

$$d_A\left(\coprod \overrightarrow{\varphi}, y\right) = d_{\overrightarrow{A}}\left(\overrightarrow{\varphi}, \nabla y\right) \quad \text{and} \quad d_A\left(x, \prod \overleftarrow{\lambda}\right) = d_{\overleftarrow{A}}\left(\Delta x, \overleftarrow{\lambda}\right)$$

Putting Propositions 3.12 and 3.14 together yields yet another familiar fact.

Proposition 3.15. *The sup-completion $\nabla : A \to \overrightarrow{A}$ preserves any infima that exist in A. The inf-completion $\Delta : A \to \overleftarrow{A}$ suprema that exist exist in A.*

3.6 Projectors and nuclei

Proposition 3.16. *For any adjunction $f^* : A \rightleftarrows B : f_*$ holds*

$$(a) \iff (b) \quad \text{and} \quad (c) \iff (d)$$

where

(a) $\forall xy \in B.\ d_A(f_* x, f_* y) = d_B(x, y)$
(b) $f^* f_* = \mathrm{id}_B$
(c) $\forall xy \in A.\ d_B(f^* x, f^* y) = d_A(x, y)$
(d) $f_* f^* = \mathrm{id}_A$

Definition 3.17. *A map g from a distance space A to a distance space B is an* embedding *if it preserves the distance, i.e. satisfies $d_A(x, y) = d_B(gx, gy)$ for all $x, y \in A$. An adjoint of an embedding is called a* projection*.*

An adjunction $p^* : A \rightleftarrows B : e_*$ of a left projection and right adjoint, as in Prop. 3.16(a-b), is called a reflection*. An adjunction $e^* : A \rightleftarrows B : p_*$ of a left embedding and right projection, as in Prop. 3.16(c-d), is called a* coreflection*.*

Definition 3.18. *A* nucleus *of the adjunction $f^* : A \rightleftarrows B : f_*$ consists of a distance space $\wr f \int$ together with*

– *embeddings $A \overset{e_*}{\hookleftarrow} \wr f \int \overset{e^*}{\hookrightarrow} B$*

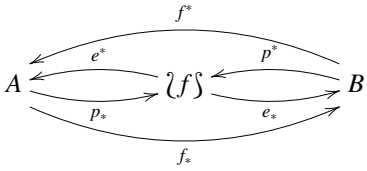
– *projections* $A \overset{p^*}{\twoheadrightarrow} \wr f \int \overset{p_*}{\twoheadleftarrow} B$

such that $f^* = e^* p^*$ *and* $f_* = e_* p_*$.

Proposition 3.19. *Any adjunction factors through its nucleus by reflection followed by a coreflection. The nucleus of the adjunction* $f^* : A \rightleftarrows B : f_*$ *is in the form*

$$\wr f \int = \{\langle x, y\rangle \in A \times B \mid f^* x = y \wedge x = f_* y\} \tag{15}$$

and the factoring is

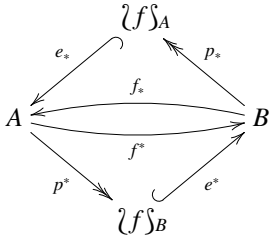


Any right adjoint factors through the nucleus by a right projection followed by a right embedding, and any left adjoint factors through the nucleus by a left projection followed by a left embedding. This factorization is unique up to isomorphism.

Proof. For any adjunction $f^* : A \rightleftarrows B : f_*$, form the distance spaces

$$\wr f \int_A = \{x \in A \mid f_* f^* x = x\} \qquad\qquad \wr f \int_B = \{y \in B \mid f^* f_* y = y\}$$

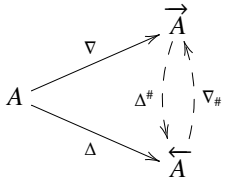
are easily seen to be isomorphic with the nucleus. The factorisation is thus



□

3.7 Cones and cuts

The *cone extensions* are the contractions $\Delta^\#$ and $\nabla_\#$



$$a\left(\Delta^{\#}\overrightarrow{\varrho}\right) = \bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d(x, a) \qquad\qquad \left(\nabla_{\#}\overleftarrow{\lambda}\right) a = \bigvee_{x \in A} \overleftarrow{\lambda} x \vdash d(a, x)$$

induced by the universal properties of the Yoneda embeddings ∇ and Δ, as per Prop. 3.9. Since $\Delta^{\#}$ thus preserves suprema, and $\nabla_{\#}$ preserves infima, Prop. 3.12 implies that each of them is an adjoint, and it is not hard to see that they are adjoint to each other, i.e. $\Delta^{\#} : \overrightarrow{A} \rightleftarrows \overleftarrow{A} : \nabla_{\#}$.

Proposition 3.20. *For every $\overrightarrow{\varrho} \in \overrightarrow{A}$ every $\overleftarrow{\lambda} \in \overleftarrow{A}$ holds*

$$\left(\overrightarrow{\varrho} \rightsquigarrow \nabla_{\#}\Delta^{\#}\overrightarrow{\varrho} \quad \textit{and} \quad \nabla_{\#}\Delta^{\#}\overrightarrow{\varrho} \rightsquigarrow \overrightarrow{\varrho}\right) \iff \exists \overleftarrow{\lambda}.\, \overrightarrow{\varrho} = \Delta^{\#}\overleftarrow{\lambda}$$

$$\left(\overleftarrow{\lambda} \rightsquigarrow \Delta^{\#}\nabla_{\#}\overleftarrow{\lambda} \quad \textit{and} \quad \Delta^{\#}\nabla_{\#}\overleftarrow{\lambda} \rightsquigarrow \overleftarrow{\lambda}\right) \iff \exists \overrightarrow{\varrho}.\, \overleftarrow{\lambda} = \nabla_{\#}\overrightarrow{\varrho}$$

The transpositions make the following subspaces isomorphic

$$\left(\overrightarrow{A}\right)_{\nabla_{\#}\Delta^{\#}} = \left\{\overrightarrow{\varrho} \in \overrightarrow{A} \mid \overrightarrow{\varrho} = \nabla_{\#}\Delta^{\#}\overrightarrow{\varrho}\right\}$$

$$\left(\overleftarrow{A}\right)_{\Delta^{\#}\nabla_{\#}} = \left\{\overleftarrow{\lambda} \in \overleftarrow{A} \mid \overleftarrow{\lambda} = \Delta^{\#}\nabla_{\#}\overleftarrow{\lambda}\right\}$$

Proof. Unfolding the definitions of $\nabla_{\#}$ and $\Delta^{\#}$ gives

$$a\left(\nabla_{\#}\Delta^{\#}\overrightarrow{\varrho}\right) = \bigvee_{u \in A}\left(\bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d(x,u)\right) \multimap d(a,u)$$

which shows that the first claim follows from the fact that for every $u \in A$ holds

$$\dfrac{\dfrac{\bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d(x,u) \;\geq\; a\overrightarrow{\varrho} \multimap d_A(a,u)}{a\overrightarrow{\varrho} + \left(\bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d(x,u)\right) \;\geq\; d_A(a,u)}}{a\overrightarrow{\varrho} \geq \left(\bigvee_{x \in A} x\overrightarrow{\varrho} \multimap d(x,u)\right) \multimap d_A(a,u)}$$

□

Definition 3.21. *The* cones *in a distance space A are the sequences in $\left(\overrightarrow{A}\right)_{\nabla_{\#}\Delta^{\#}}$ and $\left(\overleftarrow{A}\right)_{\Delta^{\#}\nabla_{\#}}$. A* cut *in A is a pair of cones $\gamma = \langle \overrightarrow{\gamma}, \overleftarrow{\gamma} \rangle \in \left(\overrightarrow{A}\right)_{\nabla_{\#}\Delta^{\#}} \times \left(\overleftarrow{A}\right)_{\Delta^{\#}\nabla_{\#}}$ such that $\overrightarrow{\gamma} = \nabla_{\#}\overleftarrow{\gamma}$. The set of cuts is denoted by $\overleftrightarrow{A}$.*

Lemma 3.22. *There are bijections $\left(\overrightarrow{A}\right)_{\nabla_{\#}\Delta^{\#}} \cong \overleftrightarrow{A} \cong \left(\overleftarrow{A}\right)_{\Delta^{\#}\nabla_{\#}}$, extending the isomorphism $\left(\overrightarrow{A}\right)_{\nabla_{\#}\Delta^{\#}} \cong \left(\overleftarrow{A}\right)_{\Delta^{\#}\nabla_{\#}}$ from Prop. 3.20.*

Proposition 3.23. *The set of cuts $\overleftrightarrow{A}$ with the distance defined by*

$$d_{\overleftrightarrow{A}}(\gamma,\varphi) \;=\; d_{\overrightarrow{A}}\left(\overrightarrow{\gamma},\overrightarrow{\varphi}\right) \;=\; d_{\overleftarrow{A}}\left(\overleftarrow{\gamma},\overleftarrow{\varphi}\right)$$

is a left and right complete distance space.

Notation. We often abuse notation and write

- $\overleftarrow{\varrho}$ for the associated cone $\nabla_\# \overrightarrow{\varrho}$, and
- $\overrightarrow{\lambda}$ for the associated cone $\Delta^\# \overleftarrow{\lambda}$.

Proof of Prop. 3.23. The $\overleftrightarrow{A}$-infima are constructed in $\overrightarrow{A}$, the $\overleftrightarrow{A}$-suprema in $\overleftarrow{A}$. To spell this out, consider $\overleftarrow{\lambda} : \overleftrightarrow{A} \to [0,\infty]$ and $\overrightarrow{\varrho} : \overleftrightarrow{A}^o \to [0,\infty]$. Extend them along the isomorphisms

$$\left(\overrightarrow{A}\right)_{\nabla_\#\Delta^\#} \cong \overleftrightarrow{A} \cong \left(\overleftarrow{A}\right)_{\Delta^\#\nabla_\#} \cong \overleftrightarrow{A}$$

to get $\overleftarrow{\lambda} : \left(\overrightarrow{A}\right)_{\nabla_\#\Delta^\#} \to [0,\infty]$ and $\overrightarrow{\varrho} : \left(\overleftarrow{A}\right)^o_{\Delta^\#\nabla_\#} \to [0,\infty]$. Then

$$\prod \overleftarrow{\lambda} = \overleftarrow{\lambda} \circ \nabla \in \left(\overrightarrow{A}\right)_{\nabla_\#\Delta^\#} \qquad \coprod \overrightarrow{\varrho} = \overrightarrow{\varrho} \circ \Delta \in \left(\overleftarrow{A}\right)_{\Delta^\#\nabla_\#}$$

The claim now boils down to showing that the inclusion $\left(\overrightarrow{A}\right)_{\nabla_\#\Delta^\#} \hookrightarrow \overrightarrow{A}$ preserves infima, whereas the inclusion $\left(\overleftarrow{A}\right)_{\Delta^\#\nabla_\#} \hookrightarrow \overleftarrow{A}$ preserves the suprema. But this is immediate from the next Lemma. □

Lemma 3.24. *The limits of the cut sequences*

$$\overrightarrow{\Upsilon} : \overrightarrow{A}^o \to [0,\infty] \qquad \overleftarrow{\Lambda} : \overrightarrow{A} \to [0,\infty]$$
$$\overrightarrow{K} : \overleftarrow{A}^o \to [0,\infty] \qquad \overleftarrow{\Psi} : \overleftarrow{A} \to [0,\infty]$$

can be computed as follows

$$a\left(\coprod \overrightarrow{\Upsilon}\right) = \bigwedge_{\overrightarrow{\xi} \in \overrightarrow{A}} a\overrightarrow{\xi} + \overrightarrow{\xi}\overrightarrow{\Upsilon} \qquad \left(\prod \overleftarrow{\Lambda}\right) a = \overleftarrow{\Lambda}(\nabla a)$$
$$a\left(\coprod \overrightarrow{K}\right) = (\Delta a)\overrightarrow{K} \qquad \left(\prod \overleftarrow{\Psi}\right) a = \bigwedge_{\overleftarrow{\zeta} \in \overleftarrow{A}} \overleftarrow{\Psi}\overleftarrow{\zeta} + \overleftarrow{\zeta} a$$

Corollary 3.25. *A distance space A has all suprema if and only if it has all infima.*

Dedekind-MacNeille completion is a special case. If A is a poset, viewed as the distance space $\mathsf{W}A$, then $\overleftrightarrow{\mathsf{W}}A$ is the Dedekind-MacNeille completion of A. The above construction extends the Dedekind-MacNeille completion of posets [14] to distance spaces, in the sense that it satisfies in the same universal property, spelled out in [1].

4 Distance matrices

4.1 Definitions

Definition 4.1. *A* distance matrix Φ *from distance space A to distance space B is a sequence $\Phi : A^o \times B \to [0,\infty]$. We denote it by $\Phi : A \nrightarrow B$, and the value of Φ at $x \in A$*

and $y \in B$ is written $x\Phi y$. The matrix composition of $\Phi : A \nrightarrow B$ and $\Psi : B \nrightarrow C$ is defined

$$x(\Phi\,;\Psi)z = \bigwedge_{y\in B} x\Phi y + y\Psi z$$

With this composition and the identities $\mathrm{Id}_A : A \nrightarrow A$ where $x(\mathrm{Id}_A)x' = d_A(x,x')$, distance spaces and distance space matrices form the category Matr.

Remark. Note that the defining condition $d_A(u,x) + d_B(y,v) \geq d(x\Phi y, u\Phi v)$, which says that Φ is a contraction $A^o \times B \to [0,\infty]$, can be equivalently written

$$d_A(u,x) + x\Phi y + d_B(y,v) \;\geq\; u\Phi v \tag{16}$$

Definition 4.2. *Transposing the indices yields the* transposed *matrix:*

$$\frac{\Phi \;:\; A \nrightarrow B \;:\; x\Phi y}{\Phi^o \;:\; B^o \nrightarrow A^o \;:\; y\Phi^o x}$$

The dual *$\Phi^\ddagger : B \nrightarrow A$ of a matrix $\Phi : A \nrightarrow B$ has the entries*

$$\frac{\Phi \;:\; A \nrightarrow B \;:\; x\Phi y}{\Phi^\ddagger \;:\; B \nrightarrow A \;:\; y\Phi^\ddagger x = \bigvee\limits_{\substack{u\in A\\ v\in B}} u\Phi v \multimap (d_A(u,x) + d_B(y,v))}$$

A matrix $\Phi : A \nrightarrow B$ where $\Phi^{\ddagger\ddagger} = \Phi$ is called a suspension.

Remarks. The transposition is obviously an involutive operation, i.e. $\Phi^{oo} = \Phi$. It is easy to derive from Prop. 3.20 that $d_\Phi(x,y) \geq d_{\Phi^{\ddagger\ddagger}}(x,y)$ holds for all $x \in A$ and $y \in B$, and that $\Phi = \Phi^{\ddagger\ddagger}$ holds if and only if there is some $\Psi : B \nrightarrow A$ such that $\Phi = \Psi^\ddagger$. Since $\Phi \rightsquigarrow \Psi \Rightarrow \Psi^\ddagger \rightsquigarrow \Phi^\ddagger$, it follows that $\Phi \rightsquigarrow \Phi^{\ddagger\ddagger}$ implies $\Phi^\ddagger = \Phi^{\ddagger\ddagger\ddagger}$.

Proposition 4.3. *$\Phi : A \nrightarrow B$ and $\Phi^\ddagger : B \nrightarrow A$ satisfy $\Phi\,;\Phi^\ddagger \rightsquigarrow \mathrm{Id}_A$ and $\Phi^\ddagger\,;\Phi \rightsquigarrow \mathrm{Id}_B$.*

Proof. The condition $\Phi\,;\Phi^\ddagger \rightsquigarrow \mathrm{Id}_A$ is proven as follows:

$$\frac{\dfrac{\bigvee\limits_{\substack{u\in A\\ v\in B}} u\Phi v \multimap (d_A(u,x') + d_B(y,v)) \;\geq\; x\Phi y \multimap d_A(x,x')}{x\Phi y + \left(\bigwedge\limits_{\substack{u\in A\\ v\in B}} u\Phi v \multimap (d_A(u,x') + d_B(y,v))\right) \;\geq\; d_A(x,x')}}{x\Phi y + y\Phi^\ddagger x' \;\geq\; d_A(x,x')}$$

The second condition is proven analogously. □

Definition 4.4. *A matrix $\Phi : A \nrightarrow B$ is* embedding *if $\Phi ; \Phi^\ddagger = \mathrm{Id}_A$; and a* projection *if $\Phi^\ddagger ; \Phi = \mathrm{Id}_B$.*

Definition 4.5. *A* decomposition *of a matrix $\Phi : A \nrightarrow B$ consists of a distance space D, with*

- *projection matrix $P : A \nrightarrow D$, i.e. $d_D(d, d') = \bigwedge_{x \in A} dP^\ddagger x + xPd'$,*
- *embedding matrix $E : D \nrightarrow B$, i.e. $d_D(d, d') = \bigwedge_{y \in B} dEy + yE^\ddagger d'$,*

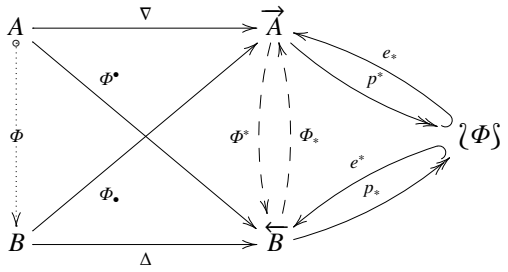
such that $\Phi = P ; E$, i.e. $x\Phi y = \bigwedge_{d \in D} xPd + dEy$.

Matrices as adjunctions. A matrix $\Phi : A \nrightarrow B$ can be equivalently presented as either of the two contractions $\Phi_\bullet$ and $\Phi^\bullet$, which extend to Φ_* and Φ^* using Prop. 3.9

$$\frac{A^o \times B \xrightarrow{\Phi} [0,\infty]}{\dfrac{A \xrightarrow{\Phi_\bullet} \overleftarrow{B} \qquad B \xrightarrow{\Phi^\bullet} \overrightarrow{A}}{\overrightarrow{A} \xrightarrow{\Phi^*} \overleftarrow{B} \qquad \overleftarrow{B} \xrightarrow{\Phi_*} \overrightarrow{A}}}$$

$$\left(\Phi^* \overrightarrow{\varphi}\right) b = \bigvee_{x \in A} x\overrightarrow{\varphi} \multimap x\Phi b \qquad \left(\Phi_* \overleftarrow{\lambda}\right) a = \bigvee_{y \in B} \overleftarrow{\lambda} y \multimap a\Phi y \tag{17}$$

Both extensions, and their nucleus, are summarized in the following diagram



(18)

The adjunction $\Phi^* : \overrightarrow{A} \rightleftarrows \overleftarrow{B} : \Phi_*$ means that

$$d_{\overleftarrow{B}}\left(\Phi^* \overrightarrow{\varphi}, \overleftarrow{\lambda}\right) = \bigvee_{y \in B} \overleftarrow{\lambda} y \multimap (\Phi^* \overrightarrow{\varphi}) y = \bigvee_{x \in A} x\overrightarrow{\varphi} \multimap x(\Phi_* \overleftarrow{\lambda}) = d_{\overrightarrow{A}}\left(\overrightarrow{\varphi}, \Phi_* \overleftarrow{\lambda}\right)$$

holds. The other way around, it can be shown that any adjunction between $\overrightarrow{A}$ and $\overleftarrow{B}$ is completely determined by the induced matrix from A to B.

Proposition 4.6. *The matrices $\Phi \in \mathsf{Matr}(A, B)$ are in a bijective correspondence with the adjunctions $\Phi^* : \overrightarrow{A} \rightleftarrows \overleftarrow{B} : \Phi_*$.*

Lemma 4.7. $d_{\overleftarrow{B}}(\Phi^* \nabla x, \Delta y) = x\Phi y = d_{\overrightarrow{A}}(\nabla x, \Phi_* \Delta y)$

4.2 Decomposition through nucleus

Proposition 4.8. *For every* $\overrightarrow{\alpha} \in \overrightarrow{A}$ *every* $\overleftarrow{\beta} \in \overleftarrow{B}$ *holds*

$$\begin{array}{llll} \overrightarrow{\alpha} \rightsquigarrow \Phi_* \Phi^* \overrightarrow{\alpha} & \text{and} & \Phi_* \Phi^* \overrightarrow{\alpha} \rightsquigarrow \overrightarrow{\alpha} \iff \exists \overleftarrow{\beta} \in \overleftarrow{B}.\, \overrightarrow{\alpha} = \Phi^* \overleftarrow{\beta} \\ \overleftarrow{\beta} \rightsquigarrow \Phi^* \Phi_* \overleftarrow{\beta} & \text{and} & \Phi^* \Phi_* \overleftarrow{\beta} \rightsquigarrow \overleftarrow{\beta} \iff \exists \overrightarrow{\alpha} \in \overrightarrow{A}.\, \overleftarrow{\beta} = \Phi_* \overrightarrow{\alpha} \end{array}$$

The adjunction $\Phi^* : A \rightleftarrows B : \Phi_*$ *induces the isomorphisms between the following distance spaces*

$$\begin{aligned} \wr \Phi \wr_A &= \left\{ \overrightarrow{\alpha} \in \overrightarrow{A} \mid \overrightarrow{\alpha} = \Phi_* \Phi^* \overrightarrow{\alpha} \right\} \\ \wr \Phi \wr_B &= \left\{ \overleftarrow{\beta} \in \overleftarrow{B} \mid \overleftarrow{\beta} = \Phi^* \Phi_* \overleftarrow{\beta} \right\} \\ \wr \Phi \wr &= \left\{ \gamma = \langle \overrightarrow{\gamma}, \overleftarrow{\gamma} \rangle \in \overrightarrow{A} \times \overleftarrow{B} \mid \overrightarrow{\gamma} = \Phi_* \overleftarrow{\gamma} \wedge \Phi^* \overrightarrow{\gamma} = \overleftarrow{\gamma} \right\} \end{aligned}$$

with the metric

$$d_{\wr \Phi \wr}(\gamma, \varphi) \;=\; d_{\overrightarrow{A}}\left(\overrightarrow{\gamma}, \overrightarrow{\varphi}\right) \;=\; d_{\overleftarrow{B}}\left(\overleftarrow{\gamma}, \overleftarrow{\varphi}\right)$$

Definition 4.9. $\wr \Phi \wr$ *is called the* nucleus *of the matrix* Φ*. Its elements are the* Φ-cuts*.*

Theorem 4.10. *The nucleus* $\wr \Phi \wr$ *of the adjunction* $\Phi^* : \overrightarrow{A} \rightleftarrows \overleftarrow{B} : \Phi_*$ *induces the decomposition of the matrix* $\Phi : A \nrightarrow B$ *into*

- *the projection* $P^* : A \nrightarrow \wr \Phi \wr$ *with* $xP^* \langle \overrightarrow{\alpha}, \overleftarrow{\beta} \rangle = x\overrightarrow{\alpha}$*, and*
- *the embedding* $E^* : \wr \Phi \wr \nrightarrow B$ *with* $\langle \overrightarrow{\alpha}, \overleftarrow{\beta} \rangle E^* y = \overleftarrow{\beta} y$

where $\langle \overrightarrow{\alpha}, \overleftarrow{\beta} \rangle \in \wr \Phi \wr$ *is an arbitrary* Φ*-cut, i.e.* $\overrightarrow{\alpha} = \Phi_* \overleftarrow{\beta}$ *and* $\Phi^* \overrightarrow{\alpha} = \overleftarrow{\beta}$*.*

Proof (sketch). We prove that $\Phi = P^*; E^*$ as follows:

$$\begin{aligned} x(P^*; E^*)y &= \bigwedge_{\overrightarrow{\alpha}} xP^* \left\langle \overrightarrow{\alpha}, \Phi^* \overrightarrow{\alpha} \right\rangle + \left\langle \overrightarrow{\alpha}, \Phi^* \overrightarrow{\alpha} \right\rangle Ey \\ &= \bigwedge_{\overrightarrow{\alpha}} x\overrightarrow{\alpha} + \left(\Phi^* \overrightarrow{\alpha}\right) y \\ &\leq x \nabla x + (\Phi^* \nabla x)\, y \\ &= d_A(x, x) + d_{\overleftarrow{B}}(\Phi^* \nabla x, \Delta y) \\ &= x \Phi y \end{aligned}$$

using Lemma 4.7 at the last step. The facts that P^* is a projection and E^* is an embedding matrix are proved using the following lemma, which says that $\wr \Phi \wr$ is $\coprod$-generated by A and $\prod$-generated by B. □

Lemma 4.11. *The $\wr\Phi\wr$-infima are computed in $\overrightarrow{A}$, whereas its suprema are computed in $\overleftarrow{B}$. To state this precisely, consider $\overleftarrow{\lambda} : \wr\Phi\wr \to [0,\infty]$ and $\overrightarrow{\varrho} : \wr\Phi\wr^o \to [0,\infty]$. Extend them along the isomorphisms $\wr\Phi\wr_A \cong \wr\Phi\wr \cong \wr\Phi\wr_B$ to get $\overleftarrow{\lambda} : \wr\Phi\wr_A \to [0,\infty]$ and $\overrightarrow{\varrho} : \wr\Phi\wr_B{}^o \to [0,\infty]$. Then*

$$\prod \overleftarrow{\lambda} \;=\; \overleftarrow{\lambda} \circ \nabla \;\in\; \wr\Phi\wr_A \qquad\qquad \coprod \overrightarrow{\varrho} \;=\; \overrightarrow{\varrho} \circ \Delta \;\in\; \wr\Phi\wr_B$$

are constructed in $\overrightarrow{A}$ and $\overleftarrow{B}$, because $\wr\Phi\wr_A \hookrightarrow \overrightarrow{A}$ preserves the infima, whereas $\wr\Phi\wr_B \hookrightarrow \overleftarrow{B}$ preserves the suprema.

Corollary 4.12. *The monotone maps $A \overset{\nabla}{\to} \overrightarrow{A} \overset{p^*}{\twoheadrightarrow} \wr\Phi\wr \overset{p_*}{\twoheadleftarrow} \overleftarrow{B} \overset{\Delta}{\leftarrow} B$*

- *preserve any infima that exist in A, and any suprema that exist in B,*
- *generate $\wr\Phi\wr$ by the suprema from A and by the infima from B, in the sense that for any $\langle\overrightarrow{\alpha}, \overleftarrow{\beta}\rangle \in \wr\Phi\wr$ holds*

$$\coprod_{\overrightarrow{\alpha}} \nabla \;=\; \langle\overrightarrow{\alpha}, \overleftarrow{\beta}\rangle \;=\; \prod_{\overleftarrow{\beta}} \Delta$$

5 Bicompletion

Any distance space morphism $f : A \to B$ induces two matrices, $\Omega f : A \nrightarrow B$ and $\mho f : B \nrightarrow A$ with

$$x\Omega f y = d_B(fx, y) \qquad\qquad y\mho f x = d_B(y, fx)$$

Lemma 5.1. *For every matrix $\Omega f : A \nrightarrow B$ induced by a distance space morphism $f : A \to B$ holds $\Omega f^{\ddagger} = \mho f$.*

Proof. Since $y\mho f x = d_B(y, fx)$ by definition, the claim boils down to $y(\Omega f)^o x = d_B(y, fx)$, which can be proved as follows

$$\begin{aligned} y(\Omega f)^o x &= \bigvee_{\substack{u\in A\\ v\in B}} d_B(fu, v) \multimap (d_B(y, v) + d_A(u, x)) \\ &\geq d_B(fx, fx) \multimap (d_B(y, fx) + d_A(x, x)) \;=\; d_B(y, fx) \end{aligned}$$

□

5.1 Nucleus as a completion

Lemma 5.2. *If the distance space B is complete, then for any matrix $\Phi : A \nrightarrow B$ there is a distance space morphism $f : A \to B$ such that $\Phi = \Omega f$.*

Corollary 5.3. *If both A and B are complete, then any matrix $\Phi : A \nrightarrow B$ corresponds to an adjunction $\Phi^* : A \rightleftarrows B : \Phi_*$ such that $\Phi = \Omega\Phi^* = \mho\Phi_*$.*

Definition 5.4. *A* distance matrix homomorphism $h : \Phi \to \Gamma$ *where* $\Phi : A \nrightarrow B$ *and* $\Gamma : C \nrightarrow D$*, is a pair of contractions* $h = \langle h_0 : A \to C, h_1 : B \to D\rangle$ *such that*

- $\Omega h_0 ; \Gamma = \Phi ; \Omega h_1$,
- h_0 *preserves any suprema that may exist in* A,
- h_1 *preserves any infima that may exist in* B.

Let MMat *denote the category of distance space matrices and matrix morphisms.*

Definition 5.5. *A matrix* $\Phi : A \nrightarrow B$ *is* complete *if* A *has suprema and* B *infima*[1]*, and* $\Phi : A^o \times B \to [0, \infty]$ *preserves the infima. Let* CMat *denote the category of complete matrices and matrix homomorphisms.*

Proposition 5.6. $\mathrm{Id}_{\wr\Phi\wr} : \wr\Phi\wr \nrightarrow \wr\Phi\wr$ *is the completion of* $\Phi : A \nrightarrow B$*. In other words, the functor* $\wr - \wr$: MMat $\to$ CMat *is left adjoint to the full inclusion* CMat $\hookrightarrow$ MMat*. The unit of the adjunction* $\eta = \langle \eta_0, \eta_1\rangle : \Phi \to \wr\Phi\wr$ *consists of*

$$\eta_0 : A \xrightarrow{\nabla} \overrightarrow{A} \xrightarrow{p^*} \wr\Phi\wr \quad \textit{and} \quad \eta_1 : B \xrightarrow{\Delta} \overleftarrow{B} \xrightarrow{p_*} \wr\Phi\wr$$

6 Summary and discussion

Given an arbitrary distance matrix $\Phi : A \nrightarrow B$, we have constructed the completion $\Phi \xrightarrow{\eta} \wr\Phi\wr$ such that

- $A \xrightarrow{\eta_0} \wr\Phi\wr$ is $\coprod$-generating and $\prod$-preserving,
- $B \xrightarrow{\eta_1} \wr\Phi\wr$ is $\prod$-generating and $\coprod$-preserving.

In terms of the motivating example of program transformations, and of the task of conjoining the algorithmic knowledge about systems and about attacks, every Φ-cut is thus a supremum of the system specifications in A, and an infimum of the attack specifications in B. Moreover, the suprema of Φ-cuts can be computed in $\overleftarrow{B}$, whereas the infima can be computed in $\overrightarrow{A}$. While the suprema[2] capture composite systems validating some composite properties, the infima describe composite attacks where the *in*validated properties add up.

But what has been achieved by providing this very abstract account? It turns out that the actual completions provide fairly concrete information. There is no space to illustrate this, but we sketch a high level view. The prior knowledge, represented by the distance spaces A and B is updated by the empiric data, represented by the matrix $\Phi : A \nrightarrow B$. In the completion $\wr\Phi\wr$, the empiric relations of as and bs are expressed as distances. Following [21, 13, Ch. 4], the task of *explaining* these empiric links can then be viewed as the task of finding short programs p with $p(a) = b$. After such completions, some distances previously recorded in A and B may increase, since some programs may be closer related *a posteriori* than *a priori*.

[1] By Corollary 3.25, both A and B are thus complete

[2] not unlike colimits of software specifications [20, 19]

The obvious task for future work is to refine the concrete applications of the presented construction. This is to some extent covered in the full paper, which is in preparation. The further work on quantifying the hardness of program derivations, and of program transformations, branches in many directions. Distances arise naturally in this framework, as described already in [16, Sec. 4.2]. In a different direction, it seems interesting to study the bicompletions in other categorical frameworks, in particular where the dualities fail in a significant way, as demonstrated a long time ago [6].

References

1. B. Banaschewski and G. Bruns. Categorical characterization of the MacNeille completion. *Archiv der Mathematik*, 18(4):369–377, September 1967.
2. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 1–18, London, UK, 2001. Springer-Verlag.
3. M. M. Bonsangue, F. van Breugel, and J. J. M. M. Rutten. Generalized metric spaces: completion, topology, and power domains via the yoneda embedding. *Theor. Comput. Sci.*, 193(1-2):1–51, 1998.
4. Rod Downey and Denis Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2010.
5. Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 553–562, Washington, DC, USA, 2005. IEEE Computer Society.
6. J.R. Isbell. *Subobjects, adequacy, completeness and categories of algebras*. Rozprawy matematyczne. Państwowe Wydawnictwo Naukowe, 1964.
7. Gregory Maxwell Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1982. Reprinted in Theory and Applications of Categories, No. 10 (2005) pp. 1-136.
8. J.C. Kelly. Bitopological spaces. *Proc. London Math. Soc.*, 13:71–89, 1963.
9. H. P. Künzi and M. P. Schellekens. On the yoneda completion of a quasi-metric space. *Theor. Comput. Sci.*, 278(1-2):159–194, 2002.
10. F. William Lawvere. Metric spaces, generalised logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43:135–166, 1973.
11. Tom Leinster. The magnitude of metric spaces, 2010. arxiv.org:1012.5857.
12. Tom Leinster and Christina Cobbold. Measuring diversity: the importance of species similarity. *Ecology*, 2012. to appear.
13. Ming Li and Paul M. B. Vitányi. *An introduction to Kolmogorov complexity and its applications (2. ed.)*. Graduate texts in computer science. Springer, 1997.
14. Holbrook Mann MacNeille. Extensions of partially ordered sets. *Proc. Nat. Acad. Sci.*, 22(1):45–50, 1936.
15. Saunders Mac Lane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1971. (second edition 1997).
16. Dusko Pavlovic. Gaming security by obscurity. In Carrie Gates and Cormac Hearley, editors, *Proceedings of NSPW 2011*, pages 125–140, New York, NY, USA, 2011. ACM. arxiv:1109.5542.
17. Dusko Pavlovic. Quantifying and qualifying trust: Spectral decomposition of trust networks. In Pierpaolo Degano, Sandro Etalle, and Joshua Guttman, editors, *Proceedings of FAST 2010*, volume 6561 of *Lecture Notes in Computer Science*, pages 1–17. Springer Verlag, 2011. arxiv.org:1011.5696.

18. Dusko Pavlovic. Quantitative Concept Analysis. In Florent Domenach, Dmitry I. Ignatov, and Jonas Poelmans, editors, *Proceedings of ICFCA 2012*, volume 7278 of *Lecture Notes in Artificial Intelligence*, pages 260–277. Springer Verlag, 2012. arXiv:1204.5802.
19. Dusko Pavlovic and Douglas R. Smith. Software development by refinement. In Bernhard K. Aichernig and Tom Maibaum, editors, *Formal Methods at the Crossroads*, volume 2757 of *Lecture Notes in Computer Science*. Springer Verlag, 2003.
20. Douglas R. Smith. Composition by colimit and formal software development. In Kokichi Futatsugi, Jean-Pierre Jouannaud, and José Meseguer, editors, *Essays Dedicated to Joseph A. Goguen*, volume 4060 of *Lecture Notes in Computer Science*, pages 317–332. Springer, 2006.
21. Ray J. Solomonoff. A formal theory of inductive inference. Part I., Part II. *Information and Control*, 7:1–22, 224–254, 1964.
22. Kim Ritter Wagner. Liminf convergence in omega-categories. *Theor. Comput. Sci.*, 184(1-2):61–104, 1997.
23. W.A. Wilson. On quasi-metric spaces. *Amer. J. Math.*, 52(3):675–684, 1931.
24. Yong woon Kim. Pseudo quasi metric spaces. *Proc. Japan Acad.*, 10:1009–10012, 1968.
25. A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the algorithmic concepts of information and randomness. *Russian Math. Surveys*, 25(6):83–124, 1970.