Intensional Partial Metric Spaces

Steve Matthews

Department of Computer Science
University of Warwick
Coventry, CV4 7AL, UK
Steve.Matthews@warwick.ac.uk
with
Michael Bukatin (Nokia Corporation, Boston) & Ralph Kopperman (CCNY, CUNY)

Department of Computer Science Seminar
University of Leicester
Friday June 6th 2014

WARWICK

Abstract

Partial metric spaces (1992) generalised Fréchet's metric spaces (1906) by allowing self-distance to be a non-negative real number. Originally motivated by the goal to reconcile metric space topology with the logic of computable functions and Dana Scott's innovative theory of topological domains they are increasingly too static a form of mathematics to be of use in modelling contemporary applications software (aka *Apps*) which is increasingly **pragmatic**, **interactive**, and **inconsistent** in nature. This talk faces up to the reality that if partial metric spaces are to survive in future research then they must become much more **scalable**. Wadge's *hiaton* time delay is used as a working example to study requirements for scaling partial metric spaces.

Contents

Non-zero self-distance? You cannot be serious!

"You cannot be serious!"
John McEnroe, mad on Centre Court, Wimbledon 1981

WARWICK

1 Metric spaces versus computing

1.1 Necessary zero self-distance

Definition (1)

A **metric space** (Fréchet, 1906) is a tuple $(X,\ d : X \times X \to [0, \infty))$ such that,

$$
\begin{aligned}
d(x,x) &= 0 \\
d(x,y) &= 0 \Rightarrow x = y \\
d(x,y) &= d(y,x) \\
d(x,z) &\leq d(x,y) + d(y,z)
\end{aligned}
$$

Maurice Fréchet
(1878-1973)

Example (1)

$|\cdot - \cdot| \ :\ (-\infty, +\infty)^2 \to [0, \infty)$ where $|x - y| = x - y$ if $y \leq x$, $y - x$ otherwise.

1 Metric spaces versus computing

1.1 Necessary zero self-distance

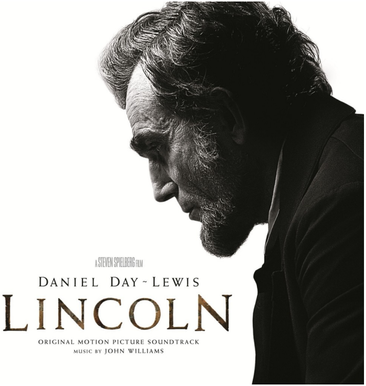
- As with most mathematics there is an unquestioned philosophical identification in the theory of metric spaces between ontology and epistemology. That is, all that exists in a metric space is presumed knowable (by examining distances), and all that is knowable (by distances) is presumed to exist.
- Hence in the time of Fréchet it was eminently sensible to axiomatize *self-distance* by $d(x, x) = 0$ and $d(x, y) = 0 \Rightarrow x = y$.
- We must be wary that knowledge of what exists can be both partially correct or inconsistent with correctness.

1 Metric spaces versus computing

1.1 Necessary zero self-distance

"Euclid’s first common notion is this: ’Things which are equal to the same thing are equal to each other.’ That’s a rule of mathematical reasoning. It’s true because it works. Has done and always will do. In his book, Euclid says this is self-evident. You see, there it is. Even in that 2,000 year old book of mechanical law, it is a self-evident truth that things which are equal to the same thing are equal to each other."

Lincoln quoting Euclid in the movie *Lincoln, (2012)*.
`www.thelincolnmovie.com`



WARWICK

1 Metric spaces versus computing

1.1 Necessary zero self-distance

Definition (2)

For each metric space (X, d) a sequence $x_0, x_1, \ldots$ is **Cauchy** if for each $\epsilon > 0$ there exists N s.t. $\forall\, n, m \geq N \,.\, d(x_n, x_m) < \epsilon$.

Definition (3)

For each metric space (X, d) a sequence x_n **converges** to a point $l \in X$ if for any $\epsilon > 0$ there exists N s.t. $d(x_n, l) < \epsilon$ for any $n \geq N$.

Definition (4)

A metric space (X, d) is **complete** if every Cauchy sequence converges.

WARWICK

Definition (5)

For each metric space (X, d) and function $f : X \to X$, f is a **contraction (mapping)** if there exists a constant $0 \leq c < 1$ s.t. $d(f(x), f(y)) \leq c \times d(x, y)$ for all $x, y \in X$.

Definition (6)

For each metric space (X, d) and function $f : X \to X$, $a \in X$ is a **fixed point** of f if $f(a) = a$.

Theorem (1 (Banach, 1922))

Each contraction over a complete metric space has a (unique) fixed point.

Stefan Banach (1892-1945)

1 Metric spaces versus computing

1.1 Necessary zero self-distance

Definition (7)

A **topological space** is a pair $(X,\ \tau \subseteq 2^X)$ s.t.,

$\{\} \in \tau$ and $X \in \tau$

$\forall A, B \in \tau \,.\, A \cap B \in \tau$ (closure under finite intersections)

$\forall \Omega \subseteq \tau \,.\, \bigcup \Omega \in \tau$ (closure under arbitrary unions)

Each $A \in \tau$ is termed an *open* set of the *topology* τ, and each complement $X - A$ a *closed* set.

Definition (8)

For each topological space $(X,\ \tau)$ a **basis** of τ is a $\Omega \subseteq \tau$ such that each member of τ is a union of members of Ω.

Example (2)

The finite open intervals (x, y) are a basis for the usual topology on $(-\infty, +\infty)$.

WARWICK

Definition (9)

For each metric space (X, d) , $a \in X$, and $\epsilon > 0$
an **open ball** $B_\epsilon(a) = \{x \in A : d(x, a) < \epsilon\}$.

Lemma (1)

For each metric space $(X,\ d)$ *the open balls form the basis for a topology* τ_d *over* X *.*

Definition (10)

A topological space $(X,\ \tau)$ is **Hausdorff separable** (aka T_2) if,

$$a \neq b \quad \Rightarrow \quad \exists \epsilon, \delta \ . \ B_\epsilon(a) \cap B_\delta(b) = \{\}$$

(i.e. distinct points in a Hausdorff separable space can be separated by disjoint neighbourhoods)

1 Metric spaces versus computing

1.1 Necessary zero self-distance

- Although *weaker* separation axioms are well known it is rare to find much research interest in *non-Hausdorff* topology. Why?
- This is pre-computing mathematics, a world where *ontology* (that which exists) is synonymous with *epistemology* (that which is known).
- This is an earlier world of consistent, well defined truths (in logic & mathematics). A world of binary choices, *true* or *false*. No *halfway house*, no *maybe*, no *sort of*, no *inbetween* truths or knowledge allowed in this earlier world of perfect truths.

Example (3)

Let d be the metric on the set $\{F, T\}$ of truth values *false* & *true* such that $d(F, T) = 1$. Then $\tau_d = 2^{\{F,T\}}$, and $\{\{F\}, \{T\}\}$ is a basis for τ_d .

WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance

- The past century has taught us that some problems are **undecidable** (i.e. there are more *truths* in the world than there are sound reasoning of proofs to match).
- More recently **fallacies** (i.e. presently believed truths but subsequently proved falsehoods) such as might arise in large scale data processing are becoming unavoidable in practice.
- How well prepared are the mathematical certainties of metric spaces for today's information processing age? How can we incorporate the *cost* of deriving mathematical truths? How can we incorporate ever more unsound reasoning into Computer Science?

WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance

The ideal of *logic* for ascertaining perfect truth in Star Trek's portrayl of the 23rd century is a wonderful caricature of how we appreciate undecidable problems and fallacies in the information age of the 20-21st centuries.

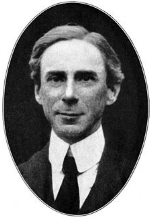
Captain Kirk arbitrates between his closest confidants, the totally logical Mr Spock and the passionate Dr McCoy.

Mr Spock's reasoning is infallible but as a result incomplete. Dr McCoy's reasoning is emotional, daring, wider reaching, but as a result fallible.

Just as Kirk needs Spock and McCoy to explore the galaxy so we need to bring together the infallibility of logical reasoning with the abstract expressiveness of (say) topology.

1 Metric spaces versus computing
1.2 Non-zero self-distance

- **Approximation** in point set topology by neighbourhoods is approximation of a totally known *limit point* by totally known *open sets*.
- So, is not an open set such as an open ball $B_\epsilon(a)$ a sufficient notion of approximation for each and every point $x \in B_\epsilon(a)$? The answer would be **YES!** if ontology and epistemology of naive set theory were consistent and synonymous.
- *Russell's paradox* of 1901 argues that if we could define $R = \{x | x \notin x\}$ then $R \in R \Leftrightarrow R \notin R$.



Russell in 1916

- This paradox (i.e. self contradiction) known to Russell et.al. led to our understanding of *incompleteness* in logic and *computability theory*.

WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance

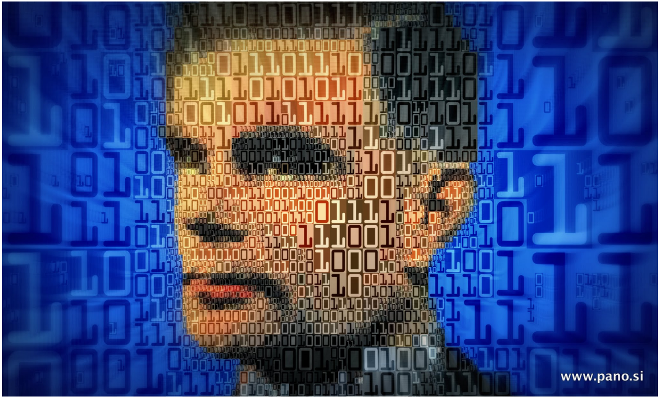
- And so *Russell's paradox* of 1901 was subsequently addressed through restriction to purely consistent knowledge of truth (e.g. typed sets) and exclusion of known (troublesome) contradictions.
- In our age touch sensitive computer screens are a playground of human mistakes, contradictions, changing our mind, etc.
- How can metric spaces & point set topology keep up with IT? To start with we need,

  Neighbourhood Approximation (i.e. traditional Hausdorff separable open sets)
  $+$ *Consistent Approximation* (e.g. T_0 separable topology for consistent computation)
  $+$ *Cost of Approximation* (cost-sensitive topology)
  $+$ *Inconsistent Approximation* (topology with mistakes)

WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance



WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance

Hausdorff separability was taken for granted until the 20th century story of *incompleteness*, Bletchley Park *code breaking*, and *Computer Science* developed.

Bletchley Park (Draco, 2008)

- $\forall\, x \,.\, x = x$ (identity)
- $\forall\, x, y \,.\, x = y \Rightarrow y = x$ (symmetry)
- $\forall\, x, y, z \,.\, x = y \wedge y = z \Rightarrow x = z$ (associativity)

T_2 separability is consistent with a two valued logic of truth (*false* and *true*). As a means of mathematically describing many structures in our natural 3D world metric spaces remain a powerful tool. However, increasingly within our everyday world people need to be as aware of how such structures are to be effectively represented within a computer as with what is their inherent nature. (`bletchleypark.org.uk`)

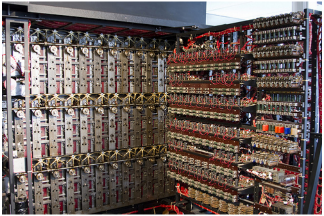
WARWICK

1 Metric spaces versus computing

1.2 Non-zero self-distance

"The bombe was an electromechanical device used by British cryptologists to help decipher German Enigma machine encrypted signals during World War II. The US Navy and US Army later produced machines to the same functional specification, but engineered differently. The initial design of the bombe was produced in 1939 at the UK Government Code and Cypher School (GC& CS) at Bletchley Park by Alan Turing, with an important refinement devised in 1940 by Gordon Welchman. The engineering design and construction was the work of Harold Keen of the British Tabulating Machine Company. ..."

Machine Room, Hut 6, 1943



Bombe back panel

WARWICK

"... It was a substantial development from a device that had been designed in 1938 by Polish Cipher Bureau cryptologist Marian Rejewski, and known as the "cryptologic bomb" (Polish: "bomba kryptologiczna"). The bombe was designed to discover some of the daily settings of the Enigma machines on the various German military networks: specifically, the set of rotors in use and their positions in the machine; the rotor core start positions for the message, the message key, and one of the wirings of the plugboard." (Wikipedia)

Enigma machine

Definition (11)

A topological space $(X,\ \tau \subseteq 2^X)$ is T_0 **separable** if

$$a \neq b \Rightarrow \exists O \in \tau \,.\, (a \in O \wedge b \notin O) \vee (b \in O \wedge a \notin O)$$

(distinct points can be separated by a neighbourhood)

Lemma (2)

$T_2 \Rightarrow T_0$

Inspired by Dana Scott in the 1960s T_0 separable spaces became a topological model for the logic of computable functions.

Dana Scott

Definition (12)

The **information ordering** (in the study of *consistent approximation*) is,

$$a \sqsubseteq b \Leftrightarrow (\forall O \in \tau \,.\, a \in O \Rightarrow b \in O)$$

- $a \sqsubseteq b$ is read as *compute up from* a *to* b. This is a progressive model for *consistent approximation*. **BUT!** No mistakes are possible, and there is no way to undo a computational step. E.g. there is no way to return from b back down to a.
- This model is consistent with the 1901 time of paradoxes where contradictions could be prevented by exclusion. It was barely viable in the 1970s where fallible programmers of mainframe computers could be made responsible for managing their mistakes. Now try telling an iPad user that they cannot make mistakes, or that there is no *undo* option in their favourite *app*.

1 Metric spaces versus computing

1.2 Non-zero self-distance

- *Consistent approximation* is a one-way street, an asymmetric half of Hausdorff separable mathematics (such as metric spaces) so to speak.
- From the 1920s to 1970s consistent approximation was an essential step forward, but now we need another to live in today's computing world.
- *Consistent approximation* does not scale up to meet today's practical demands for large scale data processing and inconsistent computation.

$$a \sqsubset b \quad \Rightarrow \quad b \not\sqsubseteq a$$

- So, how can the Hausdorff separable world of metric spaces & Banach's fixed point theorem respond to T_0 separable spaces, consistent approximation, & the practical demands of computing?

A difficult experience of our research has thus been how to travel the following path.

1.3.1 **Strong mathematics** (ontology $\equiv$ epistemology)

1.3.2 **Computability** (fundamental limitations upon knowledge)

1.3.3 **Mathematics** $+$ **computability** (e.g. partial metric spaces)

1.3.4 **Mathematics** $+$ **scalability** (e.g. intensional pmetric spaces)

1.3.1 Strong mathematics: from pre-computing mathematics a familiar misconception is that there is insufficient new metric analysis or general topology within the new pretender of partial metric spaces to justify it's introduction. For example, the following variation of a *metric space* (with which we will shortly demonstrate a close relationship with a partial metric space) can be offered by strong mathematics to argue the point.

Definition (13)

A **weighted metric space** is a tuple $(X,\ d,\ |\cdot| : X \to (-\infty, \infty))$ such that $(X,\ d)$ is a metric space.

1.3.2 Computability: in contrast the pre-Computer Science view (of logic & set theory) of *partial metric spaces* is that metric spaces & general topology can be, and indeed must be for the sake of realism, meaningfully combined with existing research into the logic of computable functions. Thus (1969) Dana Scott's groundbreaking T_0-separable *domain theory* model for **computable functions** in mathematics. Sufficient for the $x \sqsubseteq y$ information ordering of computability theory, but a one-way street insufficient for today's world of *Apps* & tablet computers.

1.3.3 Mathematics $+$ computability: the strong Math respect for metric spaces identified with (Hausdorff) T_2-separability understandably can see little value in the weaker T_0-separability of Scott topology. The computability view in common with domain theory in general has not developed in tandem with the self-evident progress of research into applied computing to be of service therein. The result is that (yes I will define them eventually!) partial metric spaces have deservedly fallen between the two schools of *strong mathematics* & *computability theory*. Thus, partial metric spaces have failed to be a significant development of either metric spaces or of domain theory. Similarly, when considered in combination, partial metric spaces have failed to advance in tandem with the mullti-disciplinary lead of applied computing.

1.3.4 Mathematics $+$ scalability: the premise of this talk is that while positive self-distance in (I promise soon to be defined!) *partial metric spaces* successfully quantifies information content in the sense of Scott's domain theory it does not go far enough to be able to discuss more sophisticated notions of *cost* which have subsequently & rightly become orthodox features of today's applied computing life. By way of gaining research experience our first cost to consider, and the one to be our working example, is that of *time*. We seek to demonstrate that in a cost-sensitive mathematics of computation *failure* to prove the existence of a truth is in general unavoidable as in logic programming, partial information in the sense of Scott domains, and a quantifiable form of knowledge.

In the pre-computing cost-free world of two-valued truth logic we can take for granted the property $P \Leftrightarrow \neg\neg P$. Later in *logic programming*, still a two-valued logic, a term `not P` is assumed to hold (rightly or wrongly) from the so-called *failure* to prove (by computable means) the truth of `P`. The following example is adapted from a Prolog manual.

```
unmarried_student(x) :- not(married(x)), student(x)
student(joe)
married(john)
```

`unmarried_student(joe)` holds as `not(married(joe))` holds as attempting to prove `married(joe)` fails.

2 Partial metric spaces (computable maths)

2.2 Cost-free 2, 3, & 4 valued truth logic

$$\begin{tikzcd}
 & \top & \\
F \arrow[ur] \arrow[rr, dashed, no head] & & T \arrow[ul] \\
 & \bot \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

- $\top$ (pronounced *top* in lattice theory and *both* in four valued logic) introduces the possibility of an *overdetermined* mathematical value,
- F, T (pronounced *false*, *true*) typifies two well defined consistent distinct values in a mathematical theory.
- $\bot$ (pronounced *bottom* in lattice theory and *either* in four valued logic) introduces the possibility of an *underdetermined* mathematical value.

2 Partial metric spaces (computable maths)

2.2 Cost-free 2, 3, & 4 valued truth logic

$$
\begin{tikzcd}
& \top & \\
F \arrow[ur] \arrow[rr, dash, dashed] & & T \arrow[ul] \\
& \bot \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$\top$ as computing power and *apps* grow so does the scope for humans to enter and expect computers to (consistently) process their inconsistent data.

F, T the traditional idealised realm of mathematics in which all values, axioms, theorems, etc. are presumed to be logically consistent.

$\bot$ perhaps the wait is finite, perhaps infinite, we just cannot know how long if ever it will take to compute the desired mathematical value.

2 Partial metric spaces (computable maths)

2.2 Cost-free 2, 3, & 4 valued truth logic

$$
\begin{tikzcd}
 & \top & \\
F \arrow[ur] \arrow[rr, dash, dashed] & & T \arrow[ul] \\
 & \bot \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

We assume that information is **partially ordered.**

$\bot \sqsubset F \sqsubset \top$ and $\bot \sqsubset T \sqsubset \top$.

$F \not\sqsubseteq T$ and $T \not\sqsubseteq F$ (two valued logic is unchanged).

$x \sqsubseteq y \Rightarrow f(x) \sqsubseteq f(y)$ (functions are *monotonic*).

We envisage a vertical structure of information processing orthogonal to a given horizontal mathematical structure.

Definition (14)

A **partially ordered set** (**poset**) is a relation $(X, \sqsubseteq \subseteq X \times X)$ such that

$$x \sqsubseteq x \quad \text{(reflexivity)}$$
$$x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y \quad \text{(antisymmetry)}$$
$$x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z \quad \text{(transitivity)}$$

Definition (15)

For each poset $(X, \sqsubseteq)$, $(X, \sqsubset)$ is the relation such that, $x \sqsubset y \Leftrightarrow x \sqsubseteq y \wedge x \neq y$.

Example (4)

The real numbers $(-\infty, \infty)$ are partially ordered by the relation $x \sqsubseteq y$ iff $x \geq y$.

Definition (16)

A **lattice** is a partially ordered set in which each pair of points x, y has a unique greatest lower bound (aka *infinum* or *meet*) denoted $x \sqcap y$, and unique lowest upper bound (aka *supremum* or *join*) denoted $x \sqcup y$.

Example (5)

A set is a lattice when partially ordered by set inclusion. Infinum is set intersection, and supremum is set union.

Example (6)

The extension of two valued truth logic from $\{F, T\}$ to $\{F, \bot, \top, T\}$ where $F \sqcap T = \top$ and $F \sqcup T = \bot$ is a lattice.

Definition (17)

A two argument function (in infix notation) op is **symmetric** if $x\ op\ y = y\ op\ x$ for all x and y .

Definition (18)

A two argument function (in infix notation) op is **associative** if $x\ op\ (y\ op\ z) = (x\ op\ y)\ op\ z$ for all x, y, & z .

Example (7)

In a **distributive** lattice, *meet* & *join* exist, are symmetric, are associative, and $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$ and $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$.

Definition (19)

A function f over a lattice is **distributive** if $f(x \sqcap y) = (f\ x) \sqcap (f\ y)$ and $f(x \sqcup y) = (f\ x) \sqcup (f\ y)$.

Many valued truth logic has an important role in the joint progress of mathematics and computer science. Before the age of computing, mathematics had to find its own way (for us metric spaces). When computing gathered pace in the 1960s it was from the presumption that mathematics was always to be computed bottom-up upon a single machine architecture from the consistent nothing of $\perp$. Now, in today's world of parallel network based computing, there is an additional increasingly demanding necessity that pre-computed possibly inconsistent information may arrive top-down from other sources (machine or human) to be reconciled consistently with a mathematical model below.

Truth table for *negation*.

P	$\perp$	F	T	$\top$
$\neg$**P**	$\perp$	T	F	$\top$

Negation is monotonic and distributive.

Truth table for *sequential and* (computing left-to-right).

$\mathbf{P} \wedge \mathbf{Q}$	$\bot$	F	T	$\top$
$\bot$	$\bot$	F	$\bot$	$\bot$
F	$\bot$	F	F	F
T	$\bot$	F	T	$\top$
$\top$	$\bot$	F	$\top$	$\top$

Sequential and is monotonic, not symmetric as $\bot \wedge F \neq F \wedge \bot$, and $\bot \wedge \mathbf{Q} = \bot$ for each $\mathbf{Q}$.

Truth table for *parallel and* (Belnap logic).

$\mathbf{P} \wedge \mathbf{Q}$	$\bot$	F	T	$\top$
$\bot$	$\bot$	F	$\bot$	F
F	F	F	F	F
T	$\bot$	F	T	$\top$
$\top$	F	F	$\top$	$\top$

Parallel and is monotonic, symmetric, distributive, and above *sequential and*.

- And so we see that the structure of two valued truth logic $\{F, T\}$ can be generalised in such a way that renders it more relevant to incorporate concepts which become both obvious and necessary in the age of computing.
- It is, in effect, sufficient to use our intuition to extend two valued truth logic. Sadly it is more challenging to see how to extend more sophisticated mathematical structures such as *metric spaces*.
- The insight here is to appreciate that the metric concept of *self-distance* could meaningfully (as in each of *mathematics* and *CS*) be non-zero as well as zero.

Definition (20)

A **partial metric space** (Matthews, 1992) is a tuple $(X,\ p : X \times X \to [0, \infty))$ such that,

$$p(x,x) \le p(x,y) \quad \text{(small self-distance)}$$
$$p(x,x) = p(y,y) = p(x,y) \Rightarrow x = y \quad \text{(equality)}$$
$$p(x,y) = p(y,x) \quad \text{(symmetry)}$$
$$p(x,z) \le p(x,y) + p(y,z) - p(y,y) \quad \text{(triangularity)}$$

Definition (21)

An **open ball** $B_\epsilon(a) = \{x \in A : p(x,a) < \epsilon\}$.

Lemma (3)

The open balls form the basis for a topology τ_p *. This is asymmetric in the sense that there may be* x, y *such that* $y \in cl\{x\} \wedge x \notin cl\{y\}$ *(i.e.* T_0 *separation).*

Definition (21)

For each partial metric space $(X,\ p)$, $(X,\ \sqsubseteq_p \subseteq\ X \times X)$ is the relation such that $x \sqsubseteq_p y\ \Leftrightarrow\ p(x,x)\ =\ p(x,y)$.

Lemma (4)

$(X,\ \sqsubseteq_p)$ *is a partially ordered set.*

Lemma (5)

A metric space is (in the sense of 1.3.1 Strong mathematics) precisely a partial metric space for which each self-distance is 0 *. In such a space the partial ordering is equality.*

Thus the notion of *partial metric space* is a generalisation of the notion of *metric space* through introducing *non-zero self-distance* and motivated by the glorious history of topology, logic, and computing.

WARWICK

- Note that a metric space (X, d) is not **linear** in the sense that the function $d'(x, y) = a \times d(x, y) + b$ for arbitrary $a > 0$, $b \geq 0$ is not in general a metric; this being because of the *necessary zero self-distance* axiom $d(x, x) = 0$.
- In contrast a partial metric space (X, p) is linear as the function $p'(x, y) = a \times p(x, y) + b$ is also a partial metric, and furthermore $\tau_{p'} = \tau_P$.
- Thus now we can generalise $(X,\ p : X \times X \to [0, \infty))$ to $(X,\ p : X \times X \to (-\infty, \infty))$ to express *negative distances*.

Definition (13)

A **weighted metric space** is a tuple $(X,\ d,\ |\cdot| : X \to (-\infty, \infty))$ such that $(X,\ d)$ is a metric space.

Lemma (6)

If $(X,\ d,\ |\cdot|)$ *is a weighted metric space then*

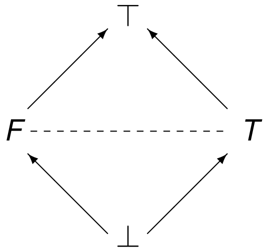
$$p(x, y) \ := \ d(x, y) + \frac{|x| \ + \ |y|}{2}$$

is a partial metric such that $p(x, x) \ = \ |x|$ *for each* $x \in X$ *.*
If $(X,\ p)$ *is a partial metric space and*

$$d_p(x, y) \ := \ p(x, y) - \frac{p(x, x) + p(y, y)}{2} \ , \ \ |x|_p \ := \ p(x, x)$$

then $(X,\ d_p,\ |\cdot|_p)$ *is a weighted metric space.*

WARWICK



Example (8)

A partial metric for (cost-free) 2, 3, & 4 valued truth logic.

$$
\begin{aligned}
p_B(\top,\top) &= -1 \\
p_B(F,F) &= p_B(F,\top) = p_B(T,T) = p_B(T,\top) = 0 \\
p_B(\bot,\bot) &= p_B(\bot,F) = p_B(\bot,\top) = p_B(\bot,T) = p_B(F,T) = 1
\end{aligned}
$$

We choose this particular partial metric in order that the induced weighted metric space has the familiar metric $d_{p_B}(F,T)=1$.

The initial interest in non-zero self-distance for metric spaces followed from pioneering research of Ashcroft & Wadge in the *Lucid* programming language.

Example (9)

A **Lucid partial metric space** (Matthews, 1992) is a pair $(X^{*\omega},\ p_L\ :\ X^{*\omega} \times X^{*\omega} \to [0,1])$ such that,

$$
\begin{array}{rcll}
\omega & := & \{0, 1, \ldots\} & \\
X^n & := & \{\ x : \{0, 1, \ldots, n-1\} \to X)\ \} & \quad \textit{for } n \in \omega \\
X^* & := & \bigcup_{n\in\omega}\ X^n & \\
X^{*\omega} & := & X^* \ \cup\ X^\omega & \\
\overline{x} & := & \begin{cases} n & \textit{if } x \in X^n \\ \infty & \textit{if } x \in X^\omega \end{cases} & \\
p_L(x,y) & := & 2^{-sup\{n \mid n \le \overline{x},\ n \le \overline{y},\ \forall k < n\ .\ x_k = y_k\}} &
\end{array}
$$

2 Partial metric spaces (computable maths)

2.4 Fixed points and partial metric spaces

$$
\begin{tikzcd}
& \top & \\
F \arrow[ur] \arrow[rr, dash, dashed] & & T \arrow[ul] \\
& \bot \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

In the least fixed point domain theory tradition of Kleene, Alfred Tarski (1926), & Scott $\neg T = F$, $\neg F = T$, $\neg\bot = \bot$, $\neg\top = \top$, and $\sqcup_{n \geq 0} \neg^n(\bot)$ can be used to define the ideal meaning for ψ recursively defined by $\psi = \neg\psi$.

The problem in 1980 for myself was how to reconcile the Banach contraction mapping theorem of metric spaces with Tarski's least fixed point theorem of chain complete posets.

Definition (22 (Matthews, 1992))

For each partial metric space (X, p), and for each $x \in X^\omega$, x is a **Cauchy sequence** if,

$$\forall \epsilon > 0 \, \exists k \in \omega \, \forall n, m > k \, . \; p(x_n, x_m) < \epsilon$$

Definition (23 (Matthews, 1992))

A partial metric space is **complete** if for each Cauchy sequence $x \in X^\omega$ there exists $a \in X$ s.t.,

$$\exists \lim_{n \to \infty} p(x_n, a) = 0$$

Note: these definitions (and the fixed point theorem below) work for non-negative valued partial metric spaces.
We have yet to find a way to generalise them further.

Theorem (2 (Matthews, 1995))

For each complete partial metric space (X, p) *and for each function* $f : X \to X$ *s.t.*

$$\exists 0 \leq c < 1 \,\forall x, y \in X \,.\; p(f(x), f(y)) \leq c \times p(x, y)$$

called a **contraction** *, firstly has a unique* $a \in X$ *s.t.* $a = f(a)$ *, and secondly* $p(a, a) = 0$ *.*

Thus each *chain* $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \ldots$ has a least upper bound in the sense of Tarski's least fixed point theorem, a unique fixed point in the sense of Banach's contraction mapping theorem, and zero self-distance in the sense of partial metric spaces.

Note: negative distance for partial metric spaces was introduced after this theorem for non-negative distances in $[0, \infty)$ **.**
More evidence of the need to rethink traditional consistent-only mathematics.

3 Failure takes time (scalable maths)

3.1 The *cost* of computing information

- A partial metric space (X, p) generalises Fréchet's landmark notion of *metric space* (X, d) through generalising the notion of *necessary zero self distance*.
- Just another moment in a glorious history of metric spaces, topology, logic, incompleteness, codebreaking, early theoretical computing, and now the practical demands of computing.
- $x \sqsubseteq_p y$ iff $p(x,x) = p(x,y)$ models the computing notion that the data content of x can be increased to that of y.
- Very relevant to classical **what** problems of recursion in logic, computability theory, and programming language design of the 1960s. But, these problems are now largely resolved, and computing faces very different **how** challenges.

WARWICK

3 Failure takes time (scalable maths)

3.1 The *cost* of computing information

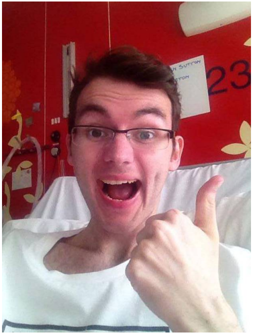
Jeff Crouse *"Recursion shirt"*

Defining ψ to be $\neg\psi$ is $\psi = \bot$ in Tarski's ideal world of least fixed points, but **NOT** in the real world of today's computing. Any attempt to compute the value of ψ up from $\bot$ would sooner or later *time out* with an error message such as *Control Stack Overflow*.

3 Failure takes time (scalable maths)

3.1 The *cost* of computing information

- The **what** truth of $x \sqsubseteq_p y$ is rarely sufficient to determine the **how** algorithm used to compute from x to y.
- With the benefit of hindsight it is easy to conclude that T_0 separable spaces (and partial metric spaces in particular) could not progress from modelling the **what** into the **how** of computation.



"I don't see the point in measuring life in terms of time anymore. I'd rather measure life in terms of making a difference.

Stephen Sutton (1994-2014)
raised £4m and inspired his generation.

TEENAGE
CANCER
TRUST

WARWICK

3 Failure takes time (scalable maths)

3.1 The *cost* of computing information

Dana Scott's inspired work has given us the T_0 topology to model partial information. But, a computable function could still be an *impossible object* as depicted in Escher's *Waterfall* which appears to produce an unending flow of water.

Equivalent to the paradox of the *Penrose Triangle* impossible object.

The supposed *paradox* disappears in our *intensional partial metric spaces* where (dynamic) cost can be composed with (static) data content.

Waterfall

Lithograph by M.C. Escher (1961)

WARWICK

- Introducing the notion of **cost** to computing a partial metric distance $p(x, y)$ is an important initial step in developing a future theory of metric spaces that may be termed dynamic, interactive, adaptive, or intelligent.
- For example, in classical logic we want to retain the double negative elimination theorem $\neg\neg P \leftrightarrow P$ while in addition asserting that $\neg\neg P$ is more costly to compute than P.
- Applying this idea to our earlier definition $\psi = \neg\psi$ would mean that we could retain the ideal Kleene, Tarski, & Scott domain theory meaning $\sqcup_{n \geq 0} \neg^{n}(\perp)$ of least fixed points and use *cost* as a criteria for introducing an error *Control Stack Overflow*.

3 Failure takes time (scalable maths)

3.2 A *cost* for computing negation

- In 2 & 3 valued propositional logic $\neg\neg P \equiv P$ is necessarily a tautology, as these logics are *cost-free*.
- However, in computer programming $\neg\neg P$ is distinguishable from P in the sense that the former inevitably takes more time to compute than does the latter.
- Thus in Computer Science the whole of $\neg\neg P$ is necessarily greater than the sum of its parts $\neg$, $\neg$, & P as would be perceived in logic, mathematics, or philosophy.
- Or, to put it another way, computer programmers would argue that an essential part is missing.

WARWICK

- For example, in classical logic we want to retain the double negative elimination theorem $P \equiv \neg\neg P$ while in addition asserting that $\neg\neg P$ is more costly to compute than P .
- Applying this idea to our earlier definition $\psi \equiv \neg\psi$ would mean that we could retain the ideal Kleene, Tarski, & Scott domain theory meaning $\sqcup_{n\geq 0} \neg^n(\perp)$ of least fixed points and use *cost* as a criteria for defining error messages such as *Control Stack Overflow*.

3 Failure takes time (scalable maths)

3.3 Wadge's *hiaton* for computing

Bill Wadge

- Wadge is much respected for his PhD UC Berkeley research known as the *Wadge hierarchy*, levels of complexity for sets of reals in descriptive set theory.
- Wadge's later insight that a *complete object* is "one that cannot be further completed" led from metric spaces (of complete objects), to *Lucid* (for programming over metric spaces), to partial metric spaces (domain theory for metric spaces), and now to discrete partial metric spaces (complexity theory for domains).
- "I don't know if *infinitesimal logic* is the best idea I've ever had, but it's definitely the best name. So here's the idea: a multivalued logic in which there are truth values that are not nearly as true as 'standard' truth, and others that are not nearly as false as 'standard' falsity" (Bill Wadge's blog, 3/2/11).

3 Failure takes time (scalable maths)

3.3 Wadge's *hiaton* for computing

- In 1977 Ashcroft & Wadge introduced a functional programming language called *Lucid* in which each input (resp. output) is a finite or infinite sequence of data values termed a *history*.
- In domain theory parlance,

  $$\langle\rangle \sqsubset \langle a\rangle \sqsubset \langle a,b\rangle \sqsubset \langle a,b,c\rangle \sqsubset \langle a,b,c,d\rangle \sqsubset \ \ldots$$

  where the totally defined inputs are precisely the infinite sequences, and the partially defined inputs are precisely the finite sequences.
- Subsequently in *partial metric space* terms $p_L(x,y) = 2^{-n}$ where n is defined to be the largest integer (or ∞ if $x = y$ is an infinite sequence) such that for each $0 \leq i < n \; x_i = y_i$. Then $(X^{*\omega}, p_L)$ is a partial metric space inducing the required *initial segment* ordering.

WARWICK

3 Failure takes time (scalable maths)

3.3 Wadge's *hiaton* for computing

- An unavoidable implication is that each data value in a sequence is presumed to take the same amount of *time* to input (resp. output).
- Suppose now that *time* is to be our experimental notion of cost for executing Lucid programs.
- Wadge & Ashcroft recognised full well that defining a notion of *cost* synonymous with the data content of an input (resp. output) sequence is unrealistic in any non trivial programming language, and so presented their insightful vision of a *pause* in the execution of a program.
- For example, the following Lucid-like sequence seeks to introduce a special pause data value termed a *hiaton* denoted $*$ to Scott's domain theory.

$$\langle *, 2, 3, *, 5, *, 7, *, *, *, 11, \ \ldots \ \rangle$$

WARWICK

3 Failure takes time (scalable maths)

3.3 Wadge's *hiaton* for computing

Wadge appreciated that *"When you look for long into an abyss, the abyss also looks into you"* (attributed to Friedrich Nietzsche).

> *"When we discovered the dataflow interpretation of Lucid (see post, Lucid the dataflow language) we thought we'd found the promised land. We had an attractive computing model that was nevertheless faithful to the declarative semantics of statements-as-equations. However, there was a snake in paradise, as David May explained in the fateful meeting in the Warwick Arts Center Cafeteria. ... And it's the need to discard data that leads to serious trouble. It could be that a huge amount of resources were required to produce ..., resources that were wasted because we threw it out. But a real catastrophe happens if the data to be discarded requires infinite resources. Then we wait forever for a result that is irrelevant, and the computation deadlocks." (Bill Wadge's blog, 6/12/11)*

3 Failure takes time (scalable maths)

3.3 Wadge's *hiaton* for computing

- But $*$ is neither a well defined null data value (such as is the number 0) nor is say $\langle *, 2\rangle$ a partial data value comparable to $\langle 2\rangle$ in the partial ordering of domain theory. And so, what is a *hiaton* ?
- The following table is an example history of how pauses could be envisaged as Lucid sequences.

Time point	*Scott domain ordering*	*Hiaton intuition*
0	$\langle\rangle$	$\langle\rangle$
1	$\sqsubseteq \langle 1\rangle$	$\langle 1\rangle$
2	$\sqsubseteq \langle 1\rangle$	$\langle 1, *\rangle$
3	$\sqsubseteq \langle 1, 2\rangle$	$\langle 1, *, 2\rangle$
4	$\sqsubseteq \langle 1, 2\rangle$	$\langle 1, *, 2, *\rangle$
5	$\sqsubseteq \langle 1, 2, 3\rangle$	$\langle 1, *, 2, *, 3\rangle$
6	$\sqsubseteq \langle 1, 2, 3\rangle$	$\langle 1, *, 2, *, 3, *\rangle$
7	$\sqsubseteq \langle 1, 2, 3\rangle$	$\langle 1, *, 2, *, 3, *, *\rangle$
...	...	......

- Sadly in 1977 this correct computational insight did not have the metric mathematics to back it up!

- Partial metric spaces can describe information content as envisaged in domain theory, but not in addition a temporal behaviour of how that data is actually computed.
- Wadge's notion of *hiaton* is shown to be ahead of its time, and a plea for an intelligent integration of Fréchet/Banach and Tarski/Scott who came before.
- The insightful temporal interpretation of functional programming of Ashcroft & Wadge (1977) can now be reconciled with the domain theory of Scott (1969) using a temporal generalisation of partial metric spaces.
- With *intensional partial metric spaces* we can, most importantly, begin to efficiently computerise mathematics.

Definition

An **intensional partial metric space** is a tuple $(X, \mathbf{I}, p : \mathbf{I} \rightarrow p^{\mathbf{I}})$ such that,

(1) X is the set (or category) of so-called **extensions** ,

(2) $\mathbf{I}$ is a category whose objects are called **intensions**, and having an initial object denoted $\perp$,

(3) For each object i in $\mathbf{I}$ there is an associated partial metric space (X, p^i) ,

(4) $p^{\mathbf{I}}$ is a category with objects (X, p^i) (for each object i in $\mathbf{I}$) and arrows $(X, p^i) \rightarrow (X, p^j)$ if $i \rightarrow j$ is an arrow in $\mathbf{I}$ and for all $x, y \in X$ $p^i(x, y) \geq p^j(x, y)$ ($p : \mathbf{I} \rightarrow p^{\mathbf{I}}$ is thus a functor using somewhat overloaded notation).

- An *intensional partial metric space* is thus a categorical construction intended to formally demarcate the *extensional* (i.e. fixed) structure of data from the *intensional* (i.e. changing) structure.
- Why this particular construction? Presently we have mere guesswork to guide our intuitions, when what we need most is accumulating experience of *Apps* to characterise our requirements for the *extensional* & *intensional* structure of data.
- From metric spaces we generalised to partial metric spaces, and from there we need more solid intensional mathematics to realise computing intuitions such as those of Wadge & Ashcroft.

Example (10)

A **hiatus partial metric space** is an intensional partial metric space $(X^{*\omega}, \ \mathbf{H}, \ p^{\mathbf{H}})$ such that,

(1) **H** is the category of **hiatus-histories** whose objects are all strictly monotonic functions in ω^{ω} . $i \rightarrow j$ is an arrow iff for each $k \in \omega \ i_k \leq j_k$. $\perp$ is the function s.t. for each $k \in \omega \ \perp_k = k$.

(2) For each object i in **H** and for all $x, y \in X^{*\omega}$,

$$p^{i}(x, y) \ := \ 2^{-sup\{i_n \ | \ n \leq \overline{x}, \ n \leq \overline{y}, \ \forall k < i_n \ . \ x_k \ = \ y_k\}}$$

Note that $p^{i}(x, x) \ = \ 2^{-(i_{\overline{x}})}$. Also note how a hiatus partial metric space has in contrast to Wadge & Ashcroft no extensional concept of a *hiaton* data value, and thus no ambiguity between extensional & intensional mathematics.

- The story so far for Lucid? Lucid was reconciled with Scott's domain theory, and (later) metric spaces. Our *hiatus partial metric spaces* reconcile partial metric spaces with Wadge & Ahscroft's *hiaton*.
- At this point our notion of *time* is a compile-time one, where it can be determined in advance.
- From the property $p^i(x,x) = 2^{-(i_{\overline{x}})}$ of a hiatus partial metric space $(X,\mathbf{H},p^{\mathbf{H}})$ & $i \in \mathbf{H}$ we can infer that it is not possible in general to determine each of the amount of data content in x and the number of time delays in x.
- Can we go further with our intensional framework and consider run-time notions of time?

Example (11)

A **cost-hiatus partial metric space** is an intensional partial metric space $(X^{*\omega}, \mathbf{H}, p^{\mathbf{H}})$ such that suppose n denotes $\overline{x \sqcap y}$ for some $x, y \in X^{*\omega}$. Then,

$$p^i(x,y) \;:=\; 2^{-n-1+2^{-i_n}}$$

Here i_n is to be read as the total number of additional delays so far in producing each data item up to and including the $n'th$ in intension i .

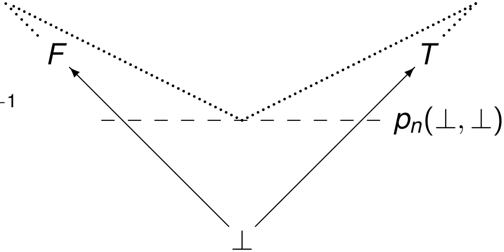
- A cost-hiatus partial metric space improves upon a hiatus partial metric space in the sense that $p^i(x, x)$ is a single number from which can be uniquely determined the amount of data content (namely $2^{-\overline{x}}$) and a cost for computing x (namely $2^{-1+2^{-i_n}}$) .
- However, our construction highlights a significant weakness that it is sufficient for computations where the delays are unconditional. However, more sophisticiation would be needed to generalise further for a a prototype conditional cost function.

3 Failure takes time (scalable maths)

3.4 Intensional partial metric spaces

Although $\neg\neg\bot = \bot$ in domain theory (as $\neg\neg P \equiv P$ in logic) we require a model for which the *cost* of computing $\neg\bot$ is discernibly greater than that of computing $\bot$. Consider the following sequence $p_0, p_1, \ldots$ of partial metrics.

$$
\begin{aligned}
p_n(T,T) &= 2^{-n}\\
p_n(F,F) &= 2^{-n}\\
p_n(F,T) &= 2^{-n}+2^{-1}\\
p_n(T,\bot) &= 2^{-n}+1\\
p_n(F,\bot) &= 2^{-n}+1\\
p_n(\bot,\bot) &= 2^{-n}+1
\end{aligned}
$$



p is *monotonic* in the sense that each p_n is a partial metric having the usual domain theory ordering $\bot \sqsubseteq F \wedge \bot \sqsubseteq T$, and $\forall x, y\ \forall 0 \leq n < m\ .\ p_n(x,y) > p_m(x,y)$

- In a *demand driven* (as opposed to *data driven*) programming language some potential catastrophes are never encountered. But, for the usual decidability reasons in computation and incompleteness of logic, not all catastrophes can be so avoided.
- Our monotonic treatment of *failure takes time* may thus be partially correct in the sense of domain theory, but is it still too weak to be useable in practice?
- A sequence $p_0, p_1, \ldots$ of consistent partial metrics as just described is an interesting step forward, but hardly a computable notion of partial metric. That is, is there a notion of partial metric that can express the best and worst of computation?

While *time* is a useful starting point for working bottom up toward a notion of cost, we also want to ask from what properties can be identified to work down?

> **Cost is observable.** Our first foundational assumption for *cost* is that it is a logic of observable properties (in the sense of a traditional *process calculus*.
>
> **Cost is monotonic.** Our second foundational assumption for *cost* is that it always increases with respect to whatever may be our chosen cost axis, usually time or space.

$$p_c(x, y) \ > \ p_{c+1}(x, y)$$

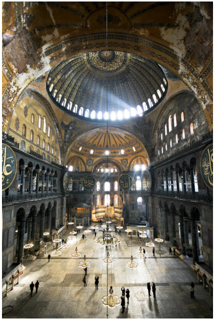
Note: terms such as $c + 1$ presume a suitable predefined algebra/logic of *cost*.

Cost is realistic. Our third foundational assumption for *cost* is that any given mathematical/logical model of *computational cost* that is wholly a static theory cannot be extended beyond it's own limits. In contrast real-world computing is dynamic, showing no signs yet of reaching whatever may prove to be it's ultimate limits and impact upon society. Our research continues to thrive upon the assertion that we refute an either all static or all dynamic approach, but rather voluntarily subject ourselves to building a balanced discipline of static & dynamic mathematics/logic for future use.

- **Conclusions:** metric spaces and topology are excellent for a former age of *strong mathematics* where *cost* could be ignored.
- Now we live in an age where the *cost* of globalisation is intrinsic to our very survival, calling for *compassion* back to the world we once ignored.
- I find the history of metric spaces, topology, logic, incompleteness, computer science, and partial metric spaces to be a fascinating personal journey of discovering *inclusion*, *compassion*, & *cost* in our overly competitive post-Fréchet world.

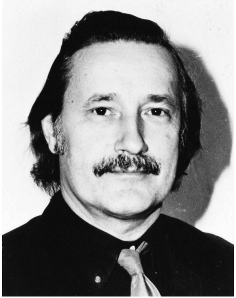
© Buddha of Compassion Society

- **Further work:** there has been an unfortunate cultural divide in Computer Science separating mathematical models & logic from the more successful complexity of algorithms.
- A pronounced *winner takes all* mentality reminiscent of tragic divides in *east meets west Holy Wars* pervades today's *IT Wars*.
- The history of **Hagia Sophia** (*Holy Wisdom*) from church, to mosque, to secular museum inspires our mathematical research to be a formalism of *disparate unity* for our world.



Hagia Sophia, Istanbul

"It has long been my personal view that the separation of practical and theoretical work is artificial and injurious. Much of the practical work done in computing, both in software and in hardware design, is unsound and clumsy because the people who do it have not any clear understanding of the fundamental design principles of their work. Most of the abstract mathematical and theoretical work is sterile because it has no point of contact with real computing. One of the central aims of the Programming Research Group as a teaching and research group has been to set up an atmosphere in which this separation cannot happen."



Christopher Strachey (1916-75)

Founder of the Oxford Programming Research Group (1965)