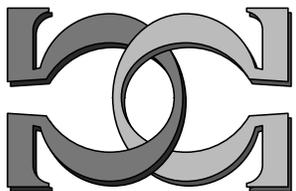
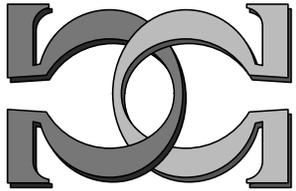
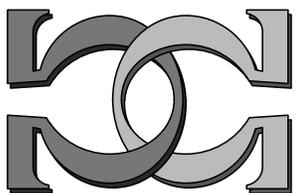
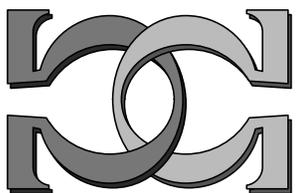


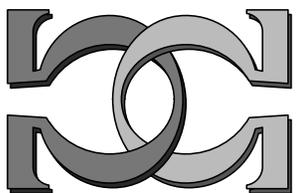
**CDMTCS
Research
Report
Series**



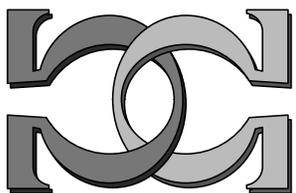
**Construmath South 2012
Applications of Non-Classical
Logic
Proceedings of the Workshop
at Westport**



**Maarten McKubre-Jordens,
Raazesh Sainudiin (Eds.)
University of Canterbury, NZ**



CDMTCS-420
19 April 2012



Centre for Discrete Mathematics and
Theoretical Computer Science

CONSTRUMATH SOUTH 2012

APPLICATIONS OF NON-CLASSICAL LOGIC

PROCEEDINGS OF THE WORKSHOP AT WESTPORT

26–28 January 2012



Maarten McKubre-Jordens & Raazesh Sainudiin (Eds.)

This meeting was supported by Marie Curie grant PIRSES-GA-2008-230822 from the European Union, with counterpart funding from the Ministry of Research, Science and Technology of New Zealand, for the project Construmath. Additional financial and infrastructure support was generously provided by the University of Canterbury.

Co-Organizers of Construmath South 2012

Maarten McKubre-Jordens & Raazesh Sainudiin

Proceedings Compiled by

Maarten McKubre-Jordens & Raazesh Sainudiin

Administrative Support

Penelope Goode

Technical Support

Steve Gourdie

Copyright © 2012

The copyright of any material in this booklet (including without limitation the text, computer code, artwork, photographs, and images) is owned by the respective authors and/or the editors. You may request permission to use the copyright materials in this booklet by contacting the respective authors and/or the editors.

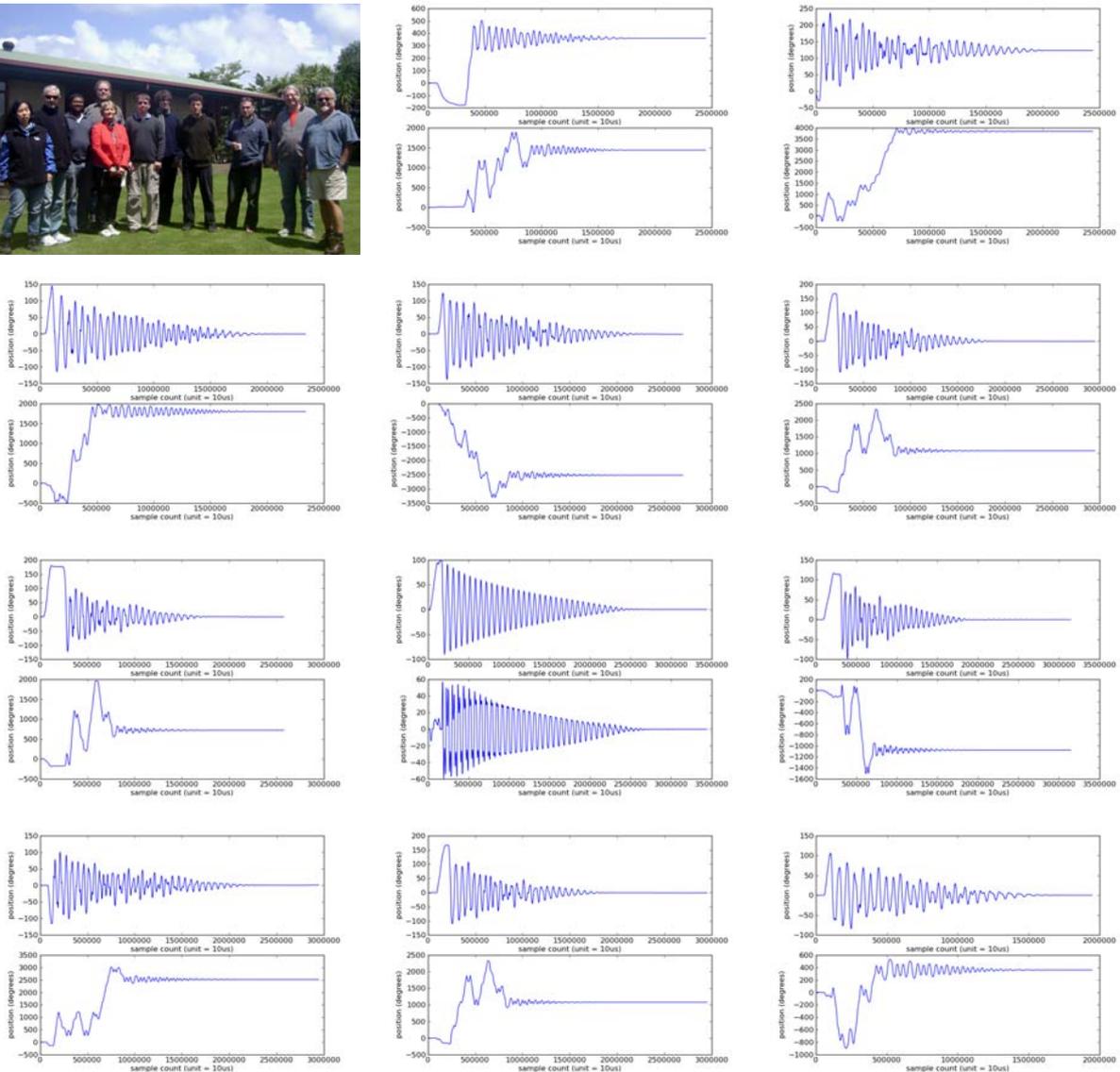
Foreword



This meeting was held on 26–28 January, 2012, at the Westport Field Station of the University of Canterbury, on the South Island of New Zealand. It was aimed at fostering the exchange of ideas between various disciplines, emphasizing links between mathematics, computer science, philosophy and statistics.

Tutorials and talks on various aspects of non-classical logics were run, with a view to using these aspects in other areas of research. Talks generally ran for about an hour, with a generous period of discussion and questions following. Thanks to the wide-ranging nature of the research interests of the group, this format proved to be very conducive to generating cross-disciplinary ideas and constructive critique. Participants also had the opportunity to explore the seal colony near Westport and walked from the colony to the lighthouse.

Participants of *ConstruMath South 2012: Applications of Non-Classical Logic and Their Double Pendulum Release Signatures*



The participants from left to right in the first image are: Ruriko Yoshida, Cris Calude, Raazesh Sainudiin, Maarten McKubre-Jordens, Elena Calude, Nicholas Duncan, James Dent, Ty Baen, Zach Weber, Ed Mares and Bruce Burdick. Each participant released a mechatronically measurable double pendulum. The remaining eleven images (from left to right and row by row) show the positions of each arm of the double pendulum through time upon release by each participant in the above list order.

Contents

Parameter Estimation in Epistemologically Valid Machine Interval Experiments

Raazesh Sainudiin (with Alex Danis and Warwick Tucker)
University of Canterbury

1

What's the Deal with Relevance?

An Introduction to Relevant Logic

Edwin Mares
Victoria University of Wellington

6

Paraconsistent Mathematics

Zach Weber
University of Otago

17

Constructive Methods in Mathematics

Maarten McKubre-Jordens
University of Canterbury

19

A Constructive Approach to the Complexity of Mathematical Problems

Cristian Calude and Elena Calude
University of Auckland / Massey University

21

Abstract Stone Duality - A Logic for Topology

Nicholas Duncan
University of Canterbury

27

Parameter Estimation in Epistemologically valid Machine Interval Experiments

Raazesh Sainudiin

Department of Mathematics and Statistics,
University of Canterbury, Private Bag 4800,
Christchurch, New Zealand

joint work with

Alexander Danis and Warwick Tucker
Department of Mathematics,
Uppsala University, Box 480 Uppsala, Sweden

ConstruMath South 2012, Westport, New Zealand

The Dualistic Context (“The Bigger Picture”)

- **Tradition:** Modern European Empiricism
- **Internal Consistency :** Aristotelean Logic
- **Universe of Hypotheses:** Popper’s Falsifiability
- **Empirical Resolution :** Mechatronically Measured Data \mathcal{D}_o
- **Model :** Deterministic ODE-IVPs
- **Parameter Space :** finite dimensional parameter space
- **Approach:** Statistical Decision Theory (set-valued approach)
- **Engineering Constraints:** Resource-limited Info. Proc.
- **Objective :** Epistemologically Valid Parameter Estimation
- **Solution:** Computer-aided Proofs & Interval Analysis

Outline

Statement (Hume, 1777)

“But where different effects have been found to follow from causes, which are to appearance exactly similar, all these various effects must occur to the mind in transferring the past to the future, and enter into our consideration, when we determine the probability of the event.” [1]

- Epistemologically valid experiment
- Data from a double pendulum
- Model with parameter space $\Theta \subsetneq \mathbb{R}^k, k < \infty$
- Action space of point estimation $\mathcal{A} = \Theta$
- Solution
 - MLE is CSP with epistemologically valid action space $\mathbb{I}\Theta^*$
 - Set-valued integrators, (T,F,?))-based estimators
- Blabber on Ongoing Work

Limits on Numerical resolution (LNR)
Limits on Empirical Resolution
Limits on Empirical Resolution

Epistemologically valid experiment

Definition

Epistemology is the study of the nature and grounds of knowledge especially with reference to its limits and validity.

Definition

A **statistical experiment** $\mathcal{E}_{\mathcal{P}}$ is the triple $(\mathbf{X}, \mathcal{F}_{\mathbf{X}}, \mathcal{P})$ consisting of a sample space \mathbf{X} of all possible empirically observable realizations of a natural phenomenon Φ , a sigma-algebra $\mathcal{F}_{\mathbf{X}}$ on \mathbf{X} , and a family of probability measures $\mathcal{P} = \{P_{\theta}, \theta \in \Theta\}$, where each P_{θ} is a probability measure on the measurable space $(\mathbf{X}, \mathcal{F}_{\mathbf{X}})$. The θ is an index belonging to the index set Θ . The index map $d(\theta) = P_{\theta} : \Theta \rightarrow \mathcal{P}$ associates every $\theta \in \Theta$ with $P_{\theta} \in \mathcal{P}$, in an arbitrary manner that even allows for the index map d to be the identity map with $\Theta = \mathcal{P}$.

Phenomenon: Damped Double Pendulum Trajectories

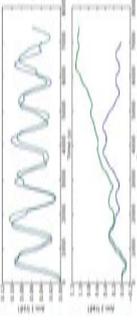
A: DP Schematic



B: Streaming DP data



C: Enclosures of two initially close trajectories



ODE Model: Damped Single Pendulum Trajectories

Kinetic energy of the arm consists of only rotational kinetic energy $T = \frac{1}{2}I\dot{\varphi}^2$, The potential energy of the pendulum is calculated by considering the geometric position of the centre of mass above the equilibrium position, $V = m l_c g(1 - \cos \varphi)$ Lagrangian of the single pendulum:

$$\mathcal{L} = T - V \quad (1)$$

$$= \frac{1}{2}I\dot{\varphi}^2 - m l_c g(1 - \cos \varphi). \quad (2)$$

The Euler-Lagrange form:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}} \right) - \frac{\partial \mathcal{L}}{\partial \varphi} = 0, \quad (3)$$

giving the equation of motion for the single pendulum system,

$$I\ddot{\varphi} + m g l_c \sin \varphi = 0 \quad (4)$$

or,

$$\ddot{\varphi} = -\xi^2 \sin \varphi \quad (5)$$

where $\xi = \sqrt{\frac{m l_c g}{I}}$.

Phenomenon: Damped Double Pendulum Trajectories

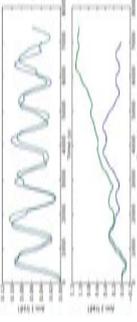
A: DP Schematic



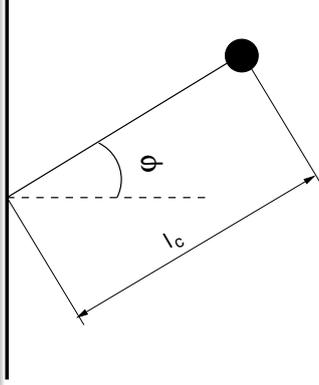
B: Streaming DP data



C: Enclosures of two initially close trajectories



ODE Model: Damped Single Pendulum Trajectories



- Model the arm as a distributed mass with centre of mass located at a distance l_c from the pivot,
- moment of inertia the arm is I and its mass is m .
- the acceleration due to gravity is $g \approx 9.81 \text{ms}^{-2}$,
- φ is the angular position,
- $\dot{\varphi}$ is the angular velocity,

ODE Model: Damped Single Pendulum Trajectories

To numerically integrate the equation of motion, we convert (5) into a system of first order equations by letting $\dot{\varphi} = \omega$ and differentiating, $\dot{\omega} = \ddot{\varphi}$. Thus we have the system of first order equations,

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \omega \\ -\xi^2 \sin \varphi \end{bmatrix} \quad (6)$$

Friction may be added to the system by adding another term to (4). The friction in this case is modeled as proportional to the angular velocity, the torque produced is given by,

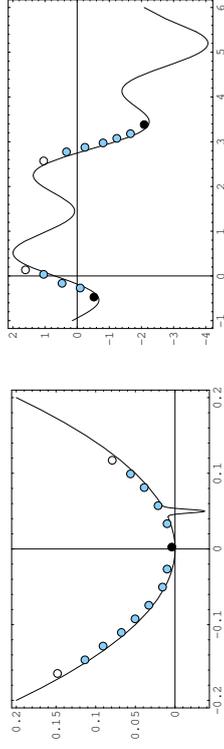
$$\tau_b = \mu \dot{\varphi}$$

giving (4) as,

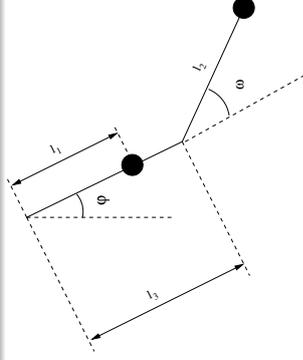
$$I\ddot{\varphi} + \mu \dot{\varphi} + m g l_c \sin \varphi = 0 \quad (7)$$

Numerical Errors due to LNR

- Overflow Error ($12! = 479001600, 13! \neq 1932053504$)
- Rounding Error (actual result - computed result)
- Cancellation Error (accumulated round-off error)
- Truncation Error (from doing only finitely many operations)
- Conversion Error (decimal to finite set of binary numbers)
- Heuristic punctual local optimization is not rigorous!



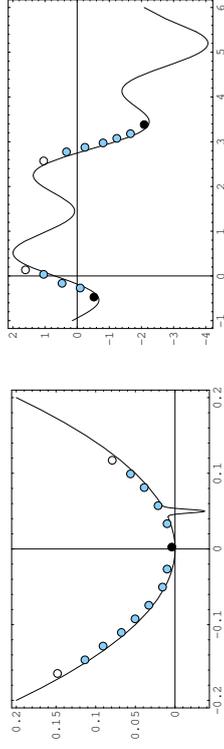
ODE Model: Passive Double Pendulum Trajectories



- the centre of mass of the inner arm is distance l_1
- the distance between pivots of the inner arm is l_3
- the centre of mass of the outer arm is distance l_2
- top arm has mass m_1 and moment of inertia of I_1
- similarly for the outer arm they are m_2 and I_2

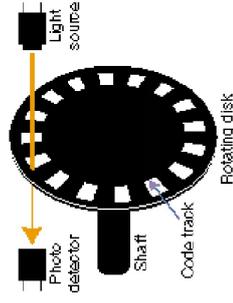
Limits on Empirical Resolution

- Overflow Error ($12! = 479001600, 13! \neq 1932053504$)
- Rounding Error (actual result - computed result)
- Cancellation Error (accumulated round-off error)
- Truncation Error (from doing only finitely many operations)
- Conversion Error (decimal to finite set of binary numbers)
- Heuristic punctual local optimization is not rigorous!



Limits on Empirical Resolution

In order to make the ground of knowledge about Φ with LER epistemologically sound, the empirically indiscernible sets must be allowed to enter the statistical experiment as data.



ODE Model: Passive Double Pendulum Trajectories

After some work...
Derivation of equations via the Euler-Lagrange equations of motion follows in a manner analogous to that presented for the passive single pendulum...
We can this parametric family of vector fields for our statistical experiment with data $\{x(t_i)\}_{t_i \in T}$ as follows:

$$\Theta \ni \theta, \quad x(t) = \int f(x, \dots, \theta)$$

Here, x_i is a sample time and $y_i = (\varphi_1, \varphi_2)$ gives the angular positions of each arm at time x_i

Epistemologically Valid Experiment

We want an **epistemologically valid experiment** that accounts for the physical limits on

- empirical resolution (“show what you can actually see”)
- numerical resolution (“compute what you actually can”)

Epistemologically valid experiment

Definition

Epistemology is the study of the nature and grounds of knowledge especially with reference to its limits and validity.

Epistemological Considerations:

- Limits on Numerical Resolution
- Limits on Empirical Resolution
- Limits on Linguistic Resolution (future work!)
- Limits on ...

Solution

- Action Space \mathcal{A} of the classical estimation problem is merely the parameter space Θ
- Epistemologically valid action space \mathcal{A} is a machine-representable Hausdorff-extension the Parameter Space

$$\Theta \xrightarrow{\text{EN}} \mathbb{I}\Theta^* := \mathbb{I}\Theta \cup \emptyset$$

- $\mathbb{I}\Theta$ is the set of all compact boxes in Θ .
- \emptyset has to be added to our epistemologically valid \mathcal{A}
- identifiability of the extended experiment indexed by $\mathbb{I}\Theta$ in terms of symmetric set difference follows from identifiability of the original experiment indexed by Θ and inclusion monotony of the index map (likelihood or conditional probability of data given parameter)

Limits on Numerical resolution (LNR)



Computers support a finite set of fixed length floating-point numbers of the form

$$x = \pm m \cdot b^e = \pm 0.m_1 m_2 \dots m_p \cdot b^e$$

where, m is the signed mantissa of precision p , b is the base (usually 2) and e , bounded between \underline{e} and \bar{e} , is the exponent. When $b = 2$, the digits of the mantissa $m_i = 1$ and $m_i \in \{0, 1\}, \forall i, 1 < i \leq p$ [3].

Thanks

- Many thanks to:
 - Piers Lawrence for completing the physical double pendulum in Civil Engg Dept.'s Lathe (Alan Nicholson), Richard Brown coordinated Electronic design and Mike Stuart did it
 - UCSDMS for supporting the double pendulum project (especially)
 - Bob Broughton (logistics, parts order, etc)
 - David Wall (\$ and kind words)
 - Douglas Bridges et al's Construmath Grant for UppsalaCAPA-CanterburyUCDMS air-traffic

Data

- lossless compression (minimal sufficient statistic) of the trajectory
- the measurable discrete state transitions along with the transition time
- time stamps, arm-position states are integers representing intervals

```
sample_number, encoder1, encoder2
26 0 0
1042 -1 0
1578 -1 -1
6752 -2 -1
.
.
1222243 -2 20480
1229330 -1 20480
```

Bibliography

-  D Hume.
An enquiry concerning human understanding: Section VI - OF PROBABILITY.
In C.W Eliot, editor, *The Harvard classics: English philosophers of the seventeenth and eighteenth centuries*, 1910 edition, volume 37. The Collier Press, 1777.
-  A Neumaier.
Interval methods for systems of equations.
Cambridge university press, 1990.
-  IEEE Task P754.
ANSI/IEEE 754-1985. Standard for Binary Floating-Point Arithmetic.
IEEE, New York, 1985.
-  AN Shiryaev.
Probability.
Springer-Verlag, 1989.
-  Piers Lawrence, Michael Stuart, Richard Brown, Warwick Tucker and Raazesh Sainudin.
A mechatronically measurable double pendulum for machine interval experiments.
Indian Statistical Institute Technical Report, sibang/ms/2010/11, October 25, 2010

Limits on Empirical Resolution

Words of Vladik Kreinovich (two recent Los Alamos Reports on Measurement Errors)

In many such situations, the only thing we know is the upper bound d on the measurement error. Thus, after we get the measured value X , the only information that we have about the actual (unknown) value x is that x belongs to the interval $[X - d, X + d]$. Here, we have two choices:

- we can ask an expert and come up with a subjective probability distribution on this interval. However, there is no guarantee that this distribution is correct, and that the recommendations based on this subjective expert distribution are valid for the actual (unknown) distribution of the measurement error.
- Another approach is to use robust statistics – a special type called interval computations. We do not know the exact distribution, we only know that this distribution is located on the interval. So, we want to make conclusions which are valid no matter what this distribution is.

What's the Deal with Relevance?

An Introduction to Relevant Logic

Edwin Mares
Victoria University of Wellington

Relevant Logics are logical systems that reject the so-called paradoxes of material and strict implication. They also brand certain inferences valid in classical or intuitionist logic as *fallacies of relevance*. Consider, for example, the inference

$$\frac{A}{\therefore B \rightarrow B}.$$

This inference is valid in classical and intuitionist logic because $B \rightarrow B$ is provable in any context (read ‘context’ as possible world for classical logic, evidential situation for intuitionist logic). The proof of $B \rightarrow B$ need have nothing to do with A , but this does not matter according to classical or intuitionist logic. The premise in an inference that is considered to be deductively valid in relevant logic, on the other hand, has *really to be used* in the proof of the conclusion. It is this notion of real use that is the key concept of relevant logic.

The notion of real use can be understood in various ways. In terms of a Gentzen-style sequent calculus, for example, it can be understood at least in part in terms of the rejection of weakening on the left-hand side of the turnstile. In terms of Fitch-Lemmon style natural deduction system, it can be understood in terms of labels that are employed to keep track of the use of hypotheses. For example, the following is a relevant deduction:

1.	$A \rightarrow (B \rightarrow C)_{\{1\}}$	<i>hyp.</i>
2.	$A \rightarrow B_{\{2\}}$	<i>hyp.</i>
3.	$A_{\{3\}}$	<i>hyp.</i>
4.	$A \rightarrow (B \rightarrow C)_{\{1\}}$	1, <i>reit.</i>
5.	$B \rightarrow C_{\{1,3\}}$	3, 4, $\rightarrow E$
6.	$A \rightarrow B_{\{2\}}$	2, <i>reit.</i>
7.	$B_{\{2,3\}}$	3, 6, $\rightarrow E$
8.	$C_{\{1,2,3\}}$	5, 7, $\rightarrow E$
9.	$A \rightarrow C_{\{1,2\}}$	3 – 8, $\rightarrow I$
10.	$(A \rightarrow B) \rightarrow (A \rightarrow C)_{\{1\}}$	2 – 9, $\rightarrow I$
11.	$(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))_{\emptyset}$	1 – 10, $\rightarrow I$

The treatment of the subscripted labels can be tricky, especially in the rules concerning conjunction (see the slides for the talk), but the basic idea is quite simple. When a hypothesis is introduced, it is given a new number. The hypothesis has to be used in the proof of a conclusion for it to be discharged, and this use is evident from the appearance of its number in the subscript of the conclusion. Similarly, if we leave a hypothesis undischarged – as a premise in an argument – its number must appear in the subscript of the conclusion in order for the deduction to be considered relevantly valid.

I interpret the subscripts in terms of the theory of *situations*. A situation is a partial representation of a universe. A situation need not contain all the information about a universe in it. For example, as I write this, I have no idea what the weather is in New

York; that information is not available to me and so it is not in this situation (I could be considered to be also in various other situations, some of which include the current weather in New York, but I will leave that for now). A step in a relevant proof, say, $A_{\{1\}}$ says that a particular situation, s_1 , contains the information that A . In the proof above, we have the hypotheses that the information that $A \rightarrow (B \rightarrow C)$ is contained in s_1 , $A \rightarrow B$ is contained in s_2 , and A is contained in s_3 . We also are assuming that these three situations obtain in the same world. On the basis of this, we infer, for example, that $B \rightarrow C$ is contained in a situation (labelled in the proof as $\{1, 3\}$) in the same world.

The logic described in the foregoing paragraphs is the logic R of *relevant implication*. Not all relevant logicians accept R as representing the last word on relevance. Many accept weaker logics. One reason for doing so is that they want a logic to act as a basis for a naive theory of truth or a naive set theory. Here I will only treat theories of truth, since the chapter by Zach Weber treats naive set theory. I don't need to go through all the issues concerning the theory of truth, but I will present the key problem, that is, the Curry paradox. Consider the Curry sentence,

(C) If this sentence is true, then the moon is made of green cheese.

Let p mean 'the moon is made of green cheese'. We know that, by virtue of the meaning of C that is is logically equivalent to $C \rightarrow p$. So, the following proof is valid in R:

1.	$C_{\{1\}}$	<i>hyp.</i>
2.	$C \leftrightarrow (C \rightarrow p)_\emptyset$	<i>stipulation</i>
3.	$C \rightarrow p_{\{1\}}$	1, 2, $\leftrightarrow E$
4.	$p_{\{1\}}$	1, 3, $\rightarrow E$
5.	$C \rightarrow p_\emptyset$	1 – 4, $\rightarrow I$
6.	$C \leftrightarrow (C \rightarrow p)_\emptyset$	<i>stipulation</i>
7.	C_\emptyset	5, 6, $\leftrightarrow E$
8.	p_\emptyset	5, 7, $\rightarrow E$

In order to bar this derivation, some relevant logicians to replace the sets in the subscripted labels with *multisets*. In a multiset, the same number can occur twice. The proof cannot be completed now:

1.	$C_{[1]}$	<i>hyp.</i>
2.	$C \leftrightarrow (C \rightarrow p)_\square$	<i>stipulation</i>
3.	$C \rightarrow p_{[1]}$	1, 2, $\leftrightarrow E$
4.	$p_{[1,1]}$	1, 3, $\rightarrow E$
5.	$C \rightarrow p_{[1]}$	1 – 4, $\rightarrow I$
6.	????	

We only have one hypothesis to discharge, but it was used twice to prove $C \rightarrow p$. Thus we have part of a means of banning the derivation of Curry's paradox. But the question is: how can we interpret logics with this restriction?

Further Reading

The natural deduction system for the relevant logic R is set out in Anderson and Belnap, *Entailment*, volume I (Princeton: Princeton University Press, 1975). Natural deduction systems for alternative relevant logics are set out in Ross Brady (ed.), *Relevant Logic and its Rivals*, volume 2 (Farnham, Surrey: Ashgate, 2003). Philosophical interpretations of relevant logics are found in Stephen Read, *Relevant Logic: A Philosophical Interpretation*

of Inference (Oxford: Blackwell, 1989) and Edwin Mares, *Relevant Logic: A Philosophical Interpretation* (Cambridge: Cambridge University Press, 2004). Greg Restall, *Introduction to Substructural Logic* (London: Routledge, 2000) places relevant logic in a more general context.

So What's the Deal With Relevance? An Introduction to Relevant Logic

Ed Mares
Philosophy Programme
and

The Centre for Logic, Language, and Computation

Victoria University of Wellington

$$\frac{B}{\therefore A \rightarrow B}$$

$$\frac{A}{\therefore B \vee \neg B}$$

$$\frac{\neg A}{\therefore A \rightarrow B}$$

... These are called the *fallacies of relevance*.

Relevant Logic

Relevant Logic is a subsystem of classical logic created to avoid the so-called paradoxes of material and strict implication, such as

- $p \rightarrow (q \rightarrow p)$
- $(p \wedge \neg p) \rightarrow q$
- $p \rightarrow (q \vee \neg q)$
- $p \rightarrow (q \rightarrow q)$
- $(p \rightarrow q) \vee (q \rightarrow r)$

The Proof-Theoretic Framework: Natural Deduction

Following Anderson and Belnap, I use a Fitch-style natural deduction system. Consider an ND proof of one of the paradoxes:

1. A *hyp*
2. B *hyp*
3. A *1, reit*
4. $B \rightarrow A$ *2-3, $\rightarrow I$*
5. $A \rightarrow (B \rightarrow A)$ *1-4, $\rightarrow I$*

Implication elimination

Relevant logics stop this by adding subscripts to steps in the proof and adding restrictions that utilize the subscripts.

When we introduce a hypothesis, we give it a number (that is new to the proof). We keep track of the hypotheses that are used to produce a given line of a proof.

$$\begin{array}{l} A \rightarrow B_\alpha \\ A_\alpha \\ B_\alpha \cup \beta \end{array} \rightarrow \mathbf{E}$$

Implication introduction

$$\frac{\begin{array}{l} A_{\{k\}} \\ \vdots \\ B_\alpha \end{array}}{A \rightarrow B_{\alpha - \{k\}}} \rightarrow \mathbf{I}$$

where $k \in \alpha$.

Real Use

The key notion that is added to the classical system in order to make it relevant is that of the **real use of hypotheses**.

This concept is not explicitly defined, but we use an intuitive understanding of real use to allow or ban certain rules.

Consider the classical conjunction rules:

$$\begin{array}{l} A_\alpha \\ B_\beta \\ \hline \therefore A \wedge B_{\alpha\cup\beta} \quad \wedge I \end{array}$$

$$\begin{array}{l} A \wedge B_\alpha \\ \hline \therefore A_\alpha \end{array} \quad \wedge E \quad \begin{array}{l} A \wedge B_\alpha \\ \hline \therefore B_\alpha \end{array}$$

Two Sorts of Conjunction: 1. Extensional Conjunction

$$\begin{array}{l} A_\alpha \\ B_\alpha \\ \hline \therefore A \wedge B_\alpha \quad \wedge I \end{array}$$

$$\begin{array}{l} A \wedge B_\alpha \\ \hline \therefore A_\alpha \end{array} \quad \wedge E \quad \begin{array}{l} A \wedge B_\alpha \\ \hline \therefore B_\alpha \end{array}$$

2. Intensional Conjunction (Fusion)

$$\begin{array}{l} A_\alpha \\ B_\beta \\ \hline \therefore A \circ B_{\alpha\cup\beta} \quad \circ I \end{array}$$

$$\begin{array}{l} A \circ B_\alpha \\ A \rightarrow (B \rightarrow C)_\beta \\ \hline \therefore C_{\alpha\cup\beta} \quad \circ E \end{array}$$

1. A
 2. B
 3. A
 4. $A \wedge B$
 5. A
 6. $B \rightarrow A$
 7. $A \rightarrow (B \rightarrow A)$
- hyp*
hyp
 1, *reit*
 2, 3, $\wedge I$
 4, $\wedge E$
 2 – 5, $\rightarrow I$
 1 – 6, $\rightarrow I$

$$(A_1 \circ \dots \circ A_n \circ B) \rightarrow C \dashv\vdash (A_1 \circ \dots \circ A_n) \rightarrow (B \rightarrow C)$$

Negation Rules

Negation Introduction:

$$\frac{A_{\{k\}} \quad \vdots \quad f_{\alpha}}{\neg A_{\alpha-\{k\}} \quad \neg I}$$

where $k \in \alpha$.

Negation Elimination:

$$\frac{\neg A_{\alpha} \quad A_{\beta}}{\therefore f_{\alpha \cup \beta} \quad \neg E_1}$$

$$\frac{\neg A_{\{k\}} \quad \vdots \quad f_{\alpha}}{A_{\alpha-\{k\}} \quad \neg E_2}$$

where $k \in \alpha$.

Relevant Logics do not contain

$$f_{\alpha} \quad \therefore A_{\alpha}$$

The Semantics

- The semantics is a frame theory, in a sense similar to Kripke's semantics for modal and intuitionist logic.
- The points in the relevant frame are *situations* (in the sense of Barwise and Perry).
- A concrete situation is a part of a world. For example, this room from 3-3:45pm today.
- This situation contains certain information (e.g. what colour these chairs are now, what is currently on the screen, ...)
- And it fails to contain other information (e.g. whether it is raining in Wellington right now, ...).
- There is true information that this situation does not contain.

A Relevant Problem with Truth

What is wrong with the following inferences?

$$\frac{p \wedge \neg p}{\therefore q}$$
$$\frac{p}{\therefore q \rightarrow q}$$

Our worry is not about whether the premise can be true or the conclusions can be false.

Rather, it is the worry that the conclusions do not follow from the premises.

One way of understanding this is to say that the premises do not contain the information that the conclusions hold.

Containing Information

What is it for a situation to contain information?

Information is a relational notion What counts as information in an environment relative to an agent are the features of that environment that she could know about given her cognitive and sensory capacities.

Abstract Situations

On the informational interpretation, the points of the relevant semantics are *abstract* situations.

We abstract from the salient features of real situations to create a general notion of a situation, and then use these features to determine whether or not they contain particular information.

Logical validity is not truth preservation, but **information preservation**.

Information Conditions

- The information condition associated with a connective is to be distinguished from its truth condition.
- An information condition is a condition under which some information of a given type is in a situation.

Information Conditions, Not Truth Conditions. – Jon Banwise



A step in a proof

A_k
 is read as saying that a situation s_k is such that
 $s_k \models A$.

When we write a hypothesis $A_{\{k\}}$, we are saying in effect, “suppose that there is some situation s_k satisfies A ”.

Implication

When we have

$$A \rightarrow B_k$$

in a line in a proof, we are saying that, at s_k we have available information that perfectly reliably allows us to infer from there being a situation in the same world as s_k that contains the information that A to there also being a situation in that world that contains the information that B .

The information condition for extensional conjunction is straightforward:

$$s \models A \wedge B$$

iff

$$s \models A \text{ and } s \models B$$

Intensional Conjunction

But the information condition for intensional conjunction is a bit more difficult:

$$s \models A \circ B$$

iff

s contains all the consequences of a situation that contains A put together with a situation that contains B . (I.e., there are situations t and u such that $t \models A$ and $u \models B$, and if we were to hypothesize that t and u were to coexist in some world, we could infer that a subsituation of s would also exist in that world.)

The Falsum

$s \models f$
iff

s is an impossible situation.

The present system validates **contraction**:

$$\frac{X, A, A, Y \vdash C}{X, A, Y \vdash C}$$

and some people think this is bad. (But it's not.)

We can modify the natural deduction system in several ways

One of the easiest is by changing the nature of the subscripts:

- We can make the subscripts multisets rather than sets
- We can make the subscripts sequences
- We make the subscripts binary trees (structures, in the sense of Slaney-Restall)

But if we change the nature of the subscripts, we also have to come up with a different interpretation of the system.

Paraconsistent Mathematics

Zach Weber
University of Otago

Overview

When we practice mathematics, we make some very intuitive assumptions that can trigger contradictions. Well known examples include the original infinitesimal calculus and naive set theory, the latter based on *naive comprehension*:

$$\exists y \forall x (x \in y \leftrightarrow A(x))$$

Paraconsistency is a method for preserving our original mathematical intuitions, by controlling for inconsistency with a weaker logical consequence relation, \vdash .

‘Classical’ inferences

In a paraconsistent setting, classical inferences like ex falso quodlibet ($A, \neg A \vdash B$) and disjunctive syllogism ($A, \neg A \vee B \vdash B$) are not in general valid. Nevertheless, because paraconsistent theories are not trivial (i.e. some sentences are not satisfied), these inferences can be restored in appropriate forms. An absurdity constant is defined

$$\perp := \forall x \forall y x \in y$$

yielding the property that $\perp \vdash A$ for any sentence A . Then ex falso and disjunctive syllogism are both valid when $\neg A$ is replaced by the property that A entails \perp . If we further identify

$$0 := \{x : \perp\}, \quad 1 := \{0\}$$

then we find a *consistency point* at the bottom of the number line: $0 = 1$ is absurd, and thus so is any sentence that implies $0 = 1$. Using this consistency point, we can confirm some structural facts that are very ‘far away’, such as \mathbb{N} being unbounded in \mathbb{R} , König’s Lemma (and Brouwer’s Fan Theorem), and the Heine-Borel Theorem. A complementary consistency point is generated at the top of the number line, at the universal set $\mathcal{V} = \{x : \exists y x \in y\}$.

Connections with other areas

Paraconsistent mathematics thus offers a way to control arguments in a more nuanced way (especially when the underlying logic is a *relevant* logic). The logic makes ‘intensional’ distinctions’, which is especially clear when we look at non-equivalent definitions of empty sets, such as $\{x : \perp\}$ and $\{x : x \neq x\}$. (The latter may have some members, even though it has no members.)

Paraconsistency is a natural dual to constructive mathematics, but it is not opposed to constructivism – in fact, constructive techniques are particularly powerful in paraconsistent settings. The goals of the program are to recapture classical results, and extend them into the study of the inconsistent, which is intrinsically interesting and beautiful in its own right, and which may yet find applications in any domain where inconsistency is possible.

References

Getting started:

Inconsistent Mathematics, Internet Encyclopedia of Philosophy: <http://www.iep.utm.edu/math-inc/>

Recent papers:

Weber, Zach (2010). Transfinite Numbers in Paraconsistent Set Theory. *Review of Symbolic Logic* 3(1): 71-92.

McKubre-Jordens, Maarten and Weber, Zach (2012). Real Analysis in Paraconsistent Logic. *Journal of Philosophical Logic*, to appear.

See also:

Brady, Ross (2006). *Universal Logic, CSLI*. [**** *This book includes the classical model theoretic proofs that show paraconsistent mathematics is not trivial.*****]

Mortensen, Chris (1995). *Inconsistent Mathematics*. Kluwer Academic Publishers.

Priest, Graham (2006). In *Contradiction: A Study of the Transconsistent*. Oxford University Press. second edition.

Constructive Methods in Mathematics

Maarten McKubre-Jordens
University of Canterbury

In Brief

The point of using constructive methods in mathematics is to explicitly exhibit any object or algorithm that the mathematician claims exists; so constructive proof provides, in principle, a mechanical method. Loosely speaking, one replaces the absolute notion of *truth* in mathematics, with (algorithmic) *provability*. Constructive proofs:

1. embody (in principle) an algorithm (for computing objects, converting other algorithms, etc.), and
2. prove that the algorithm they embody is correct (i.e. that it meets its design specification).

Constructive techniques

Upon adopting only constructive methods, we lose some powerful proof tools in our arsenal, such as unrestricted use of the Law of Excluded Middle (LEM) and anything which validates it, such as double negation elimination and unrestricted use of proof by contradiction¹. We cannot, in general, constructively prove $\exists xP(x)$ by assuming $\neg\exists xP(x)$ and deriving a contradiction; that doesn't compute the required x .

However the news isn't all bad. In a lot of cases, constructive alternatives to non-constructive classical principles in mathematics, leading to some very strong results. For example, the classical least upper bound principle is not constructively provable.

LUB Any nonempty set of reals that is bounded from above has a least upper bound.

However the constructive least upper bound principle *is* provable.

CLUB Any order-located nonempty set of reals that is bounded from above has a least upper bound.

A set is order-located if given any real x , the distance from x to the set is computable. It is quite common for a constructive alternative to be *classically* equivalent to the classical principle; and, indeed, classically every nonempty set of reals is order-located.

To see why LUB is not provable, we may consider a so-called *Brouwerian counterexample* (or *weak* counterexample), such as the set

$$S = \{x \in \mathbb{R} : (x = 2) \vee (x = 3 \wedge P)\}$$

where P is some as-yet unproven statement, such as Goldbach's conjecture. If the set S had a computable LUB, then we would have a quick proof of the Goldbach conjecture's truth or of its unprovability. A Brouwerian counterexample is an example which

¹Which is not to say that LEM is *false*. Both Russian recursive mathematics, in which LEM is provably false, and classical mathematics, in which it is logically true, are models of constructive mathematics—so in a way, LEM is *independent* of constructive mathematics, and hence non-constructive.

shows that if a certain property holds, then it is possible to constructively prove a non-constructive principle (such as LEM); and thus the property itself must be essentially non-constructive.

It is often the case that a classical theorem becomes more enlightening when seen from the constructive viewpoint². For example, in the least upper bound principle the extra computational information provided by being order-located is enough to guarantee the computability of the least upper bound.

Within constructive mathematics a number of methods has been developed, enriching the subject to a degree where it is comparable to its classical counterpart in complexity, and often exceeds it in computational informativity.

Connections with other disciplines

The connection of constructive mathematics with computer science and programming is clear. A major upshot of the constructive approach is to identify with relative ease the sorts of things that computers cannot do (it is usually easier to prove a negative result), and so to guide the programmer to focus on what *is* achievable.

Like paraconsistency, constructivism brings out finer-grained details of proof that are often casually dismissed in classical proofs. In fact, a single classical theorem can lead to several constructively discernible *different* theorems, where the constructive techniques bring to the fore extra computational strength required in the hypotheses, or further information contained in the conclusion.

References

For a more in-depth introduction:

Bridges, D.S. *Constructive Mathematics*. Stanford Encyclopedia of Philosophy: <http://plato.stanford.edu/entries/mathematics-constructive/>

Further reading:

Aberth, O. (1980) *Computable Analysis*. New York: McGraw-Hill.

Aczel, P., and Rathjen, M. (2001) *Notes on Constructive Set Theory*. Report No. 40, Institut Mittag-Leffler, Royal Swedish Academy of Sciences.

Beeson, M.J. (1985) *Foundations of Constructive Mathematics*. Heidelberg: Springer-Verlag.

Bishop, E. and Bridges, D.S. (1985) *Constructive Analysis*. Grundlehren der math. Wissenschaften, Heidelberg: Springer-Verlag.

Bridges, D.S. and Richman, F. (1987) *Varieties of Constructive Mathematics*. Cambridge: Cambridge University Press.

Bridges, D.S. and Vîță, L.S. (2006) *Techniques of Constructive Analysis*. Universitext, Heidelberg: Springer-Verlag.

Dummett, M. (2000) *Elements of Intuitionism*. Oxford Logic Guides 39, Oxford: Clarendon Press.

Martin-Löf, P. (1968) *Notes on Constructive Analysis*. Stockholm: Almqvist & Wixsell.

Weirauch, K. (2000) *Computable Analysis*. EATCS Texts in Theoretical Computer Science, Heidelberg: Springer-Verlag.

²Although it would be unfair to say that constructive mathematics is revisionist in nature. Indeed, Brouwer proved his fan theorem intuitionistically in 1927, but the first proof of König's lemma (its classical equivalent) was published in 1933.

A constructive approach to the complexity of mathematical problems

C. S. Calude (UoA) and E. Calude (Massey U)
ConstruMath South 2012, Wesport

1 / 199

Do the following statements

- ▶ the four colour theorem,
- ▶ Fermat's great theorem,
- ▶ the Riemann hypothesis,
- ▶ the Collatz conjecture

share a common mathematical property?

And, if there is such a property, how can we use it for a better understanding of these statements?

3 / 199

This talk presents an overview of results obtained with an algorithmic uniform method to measure the complexity of a large class of mathematical problems and discusses a few open problems.

2 / 199

Computability and Complexity

1

Universality theorem. There exists (and can be constructed) a (Turing) machine U —called *universal*—such that for every machine V there exists a constant $c = c_{U,V}$ such that for every program σ there exists a σ' for which the following two conditions hold:

- ▶ $U(\sigma') = V(\sigma)$,
- ▶ $|\sigma'| \leq |\sigma| + c$.

4 / 199

A universal Turing machine

1

A simple, minimal (each instruction is essential) universal Turing machine U can be designed using the following five instructions:

```
=r1,r2,r3 (branching instruction)
&r1,r2 (assigning instruction)
+r1,r2 (sum)
!r1 (read one bit)
% (halt)
```

5 / 199

An example of a program for U

The following program computes in d the product of two non-negative integers stored in a and b :

number	instruction
0	&h,e
1	&d,0
2	=b,0,8
3	&e,1
4	+d,a
5	=b,e,8
6	+e,1
7	=a,a,4
8	&e,h
9	=a,a,c

7 / 199

A universal Turing machine

2

A register machine program consists of a finite list of labeled instructions from the above list, with the restriction that the halt instruction appears only once, as the last instruction of the list. The input data (a binary string) follows immediately after the halt instruction. A program not reading the whole data or attempting to read past the last data-bit results in a run-time error. Some programs (as the ones presented in this paper) have no input data; these programs cannot halt with an under-read error.

6 / 199

Computability and Complexity

2

The **halting problem** for a machine V is the function Λ_V defined by

$$\Lambda_V(\sigma) = \begin{cases} 1, & \text{if } V(\sigma) = \infty, \\ 0, & \text{otherwise.} \end{cases}$$

Undecidability theorem. If U is universal, then Λ_U is incomputable, i.e. the halting problem for a universal machine is undecidable.

8 / 199

Π_1 -problems

A problem π of the form

$$\forall \sigma P(\sigma),$$

where P is a computable predicate is called a Π_1 -problem.

- ▶ Any Π_1 -problem is finitely refutable.
- ▶ For every Π_1 -problem $\pi = \forall \sigma P(\sigma)$ we associate the program

$$\Pi_P = \inf\{n : P(n) = \text{false}\}$$
 which satisfies:
 - π is true iff $U(\Pi_P) = \infty$.
- ▶ Solving the halting problem for U solves all Π_1 -problems.

9 / 199

Complexity

Complexity

$$C_U(\pi) = \min\{|\Pi_P| : \pi = \forall n P(n)\}.$$

Invariance theorem. If U, U' are universal, then there exists a constant $c = C_{U,U'}$ such that for all $\pi = \forall n P(n)$, P computable:

$$|C_U(\pi) - C_{U'}(\pi)| \leq c.$$

Incompleteness theorem. If U is universal, then C_U is incomputable.

11 / 199

Examples

The problems

- ▶ the four colour theorem,
- ▶ Fermat's great theorem,
- ▶ the Riemann hypothesis,
- ▶ the Collatz's conjecture

are all Π_1 -problems.

Of course, not all problems are Π_1 -problems. For example, the twin prime conjecture.

10 / 199

Computing the size of the program MULT

number	instruction	code	length
0	&h, e	01 0001001 00110	14
1	&d, 0	01 00101 100	10
2	=b, 0, 8	00 011 100 1110010	15
3	&e, 1	01 00110 101	10
4	+d, a	111 00101 010	11
5	=b, e, 8	00 011 00110 1110010	17
6	+e, 1	111 00110 101	11
7	=a, a, 4	00 010 010 11010	13
8	&e, h	01 00110 0001001	14
9	=a, a, c	00 010 010 00100	13

Total length: 128.

12 / 199

Complexity Classes

Because of the incompleteness theorem, we work with upper bounds for C_U . As the exact value of C_U is not important, we classify Π_1 -problems into the following classes:

$$\mathcal{C}_{U,n} = \{\pi : \pi \text{ is a } \Pi_1\text{-problem, } C_U(\pi) \leq n \text{ kbit}\}.$$

13 / 199

Riemann hypothesis predicate

The negation of the Riemann hypothesis is equivalent to the existence of positive integers k, l, m, n satisfying the following:

1. $n \geq 600$,
2. $\forall y < n [(y + 1) \mid m]$,
3. $m > 0 \ \& \ \forall y < m [y = 0 \vee \exists x < n [\neg [(x + 1) \mid y]]]$.
4. $\text{explog}(m - 1, l)$,
5. $\text{explog}(n - 1, k)$,
6. $(l - n)^2 > 4n^2k^4$,

where $x \mid z$ means "x divides z" and $\text{explog}(a, b)$ is the predicate

$$\exists x [x > b + 1 \ \& \ (1 + 1/x)^{xb} \leq a + 1 < 4(1 + 1/x)^{xb}].$$

15 / 199

Some Results

- ▶ $\mathcal{C}_{U,1}$: *Legendre's conjecture* (there is a prime number between n^2 and $(n + 1)^2$, for every positive integer n), *Fermat's last theorem* (there are no positive integers x, y, z satisfying the equation $x^n + y^n = z^n$, for any integer value $n > 2$) and *Goldbach's conjecture* (every even integer greater than 2 can be expressed as the sum of two primes)
- ▶ $\mathcal{C}_{U,2}$: *Dyson's conjecture* (the reverse of a power of two is never a power of five)
- ▶ $\mathcal{C}_{U,3}$: *the Riemann hypothesis* (all non-trivial zeros of the Riemann zeta function have real part $1/2$), *Euler's integer partition theorem* (the number of partitions of an integer into odd integers is equal to the number of partitions into distinct integers).
- ▶ $\mathcal{C}_{U,4}$: *the four colour theorem* (the vertices of every planar graph can be coloured with at most four colours so that no two adjacent vertices receive the same colour)

14 / 199

The Collatz conjecture

Given a positive integer a_1 there exists a natural N such that $a_N = 1$, where

$$a_{n+1} = \begin{cases} a_n/2, & \text{if } a_n \text{ is even,} \\ 3a_n + 1, & \text{otherwise.} \end{cases}$$

The Collatz conjecture is a Π_1 -statement, but the proof is *non-constructive*! Writing The Collatz conjecture as a Π_2 -statement (i.e. of the form $\forall n \exists i R(n, i)$, where $R(n, i)$ is a computable predicate) is easy and *constructive*.

How to generalise the complexity method for Π_2 -statements?

16 / 199

Inductive Complexity and Complexity Classes of First Order

By transforming each program Π_P for U into a program $\Pi_P^{ind,1}$ for U^{ind} (U working in "inductive mode") we can define the inductive complexity of first order by

$$C_U^{ind,1}(\pi) = \min\{|\Pi_P^{ind,1}| : \pi = \forall n P(n)\},$$

the inductive complexity classes of order one by

$$e_{U,n}^{ind,1} = \{\pi : \pi \text{ is a } \Pi_1\text{-statement, } C_U^{ind,1}(\pi) \leq n \text{ kbit}\},$$

and prove that

$$e_{U,n} = e_{U,n}^{ind,1}.$$

Inductive computation of the Collatz conjecture

Define the function

$$C(n) = \begin{cases} n, & \text{if } \exists i (F^i(n) = 1), \\ 1, & \text{otherwise,} \end{cases}$$

where

$$F(x) = \begin{cases} x/2, & \text{if } x \text{ is even,} \\ 3x + 1, & \text{otherwise,} \end{cases}$$

and F^i is the i th iteration of F .

Next we define the inductive Turing machine $M_{Collatz}^{ind,2}$ by

$$M_{Collatz}^{ind,2} = \begin{cases} 0, & \text{if } \forall n \geq 1, C(n) = n, \\ 1, & \text{otherwise.} \end{cases}$$

Inductive Complexity and Complexity Classes of Higher Orders

By allowing inductive programs of order 1 as routines we get inductive programs of order 2, so we can define the inductive complexity of second order (for more complex problems)

$$C_U^{ind,2}(\rho) = \min\{|M_R^{ind,2}| : \rho = \forall n \exists i R(n, i)\},$$

and the inductive complexity class of second order:

$$e_{U,n}^{ind,2} = \{\rho : \rho = \forall n \exists i R(n, i), C_U^{ind,2}(\rho) \leq n \text{ kbit}\}.$$

The Collatz conjecture is in the class $e_{U,1}^{ind,2}$

label	instruction	label	instruction	label	instruction
L1	&OR, 1 &T, 1 &OC, 1		&D, 0 &E, 1 &F, 1	L9	+G, F +G, F +G, 1
L2	&N, 1 &G, T	L6	=F, G, L8 +E, 1	L10	=E, E, L3 =G, 1, L11 +N, 1
L3	&K, N =K, 1, L10		+F, 1 =E, 2, L7	L11	=E, E, L2 &OC, T =OC, T, L12
L4	&E, 1 &F, E	L7	=E, E, L6 &E, 0		&OR, 0 =E, E, L13 +T, 1
	+F, 1 =F, K, L5	L8	=E, 1, L9 &G, D	L12	=E, E, L1 %
L5	+E, 1 =E, E, L4 &K, E		=E, E, L3	L13	

Inductive program for the Collatz conjecture

Two open problems

What is the complexity of

- ▶ P vs NP problem?
- ▶ Poincaré's theorem (Perelman)?

21 / 199

References 2

- C. S. Calude, E. Calude, K. Svozil. The complexity of proving chaoticity and the Church-Turing Thesis, *Chaos* 20 037103 (2010), 1–5.
- C. S. Calude, M. J. Dinneen. Exact approximations of omega numbers, *Int. Journal of Bifurcation & Chaos* 17, 6 (2007), 1937–1954.
- E. Calude. The complexity of Riemann's Hypothesis, *Journal for Multiple-Valued Logic and Soft Computing*, (2012), to appear.
- E. Calude. Fermat's Last Theorem and chaoticity, *Natural Computing*, (2011), DOI: 10.1007/s11047-011-9282-9.
- M. J. Dinneen. A program-size complexity measure for mathematical problems and conjectures, in M. J. Dinneen, B. Khoussainov, A. Nies (eds.): *Computation, Physics and Beyond*, Springer, Heidelberg, 2012.
- J. Hertel. On the Difficulty of Goldbach and Dyson Conjectures, *CDMTCS Research Report* 367, 2009, 15pp.

23 / 199

Two open problems

What is the complexity of

- ▶ P vs NP problem?
- ▶ Poincaré's theorem (Perelman)?

21 / 199

References 2

- C. S. Calude, E. Calude, K. Svozil. The complexity of proving chaoticity and the Church-Turing Thesis, *Chaos* 20 037103 (2010), 1–5.
- C. S. Calude, M. J. Dinneen. Exact approximations of omega numbers, *Int. Journal of Bifurcation & Chaos* 17, 6 (2007), 1937–1954.
- E. Calude. The complexity of Riemann's Hypothesis, *Journal for Multiple-Valued Logic and Soft Computing*, (2012), to appear.
- E. Calude. Fermat's Last Theorem and chaoticity, *Natural Computing*, (2011), DOI: 10.1007/s11047-011-9282-9.
- M. J. Dinneen. A program-size complexity measure for mathematical problems and conjectures, in M. J. Dinneen, B. Khoussainov, A. Nies (eds.): *Computation, Physics and Beyond*, Springer, Heidelberg, 2012.
- J. Hertel. On the Difficulty of Goldbach and Dyson Conjectures, *CDMTCS Research Report* 367, 2009, 15pp.

23 / 199

References 1

- M. Burgin, C. S. Calude, E. Calude. Inductive complexity measures for mathematical problems, in preparation.
- C. S. Calude, E. Calude, M. J. Dinneen. A new measure of the difficulty of problems, *Journal for Multiple-Valued Logic and Soft Computing* 12 (2006), 285–307.
- C. S. Calude, E. Calude. Evaluating the complexity of mathematical problems. Part 1 *Complex Systems*, 18-3 (2009), 267–285.
- C. S. Calude, E. Calude. Evaluating the complexity of mathematical problems. Part 2 *Complex Systems*, 18-4, (2010), 387–401.
- C. S. Calude, E. Calude. The complexity of the Four Colour Theorem, *LMS J. Comput. Math.* 13 (2010), 414–425.
- C. S. Calude, E. Calude, M. Queen. The complexity of the integer partition theorem, *Theoretical Computer Science*, accepted.

22 / 199

References 3

- J. Lagarias (ed.), *The Ultimate Challenge: The $3x + 1$ Problem*, AMS, 2010.
- C. Moore, S. Mertens. *The Nature of Computation*, Oxford University Press, Oxford, 2011.
- G. Perelman. Ricci Flow and Geometrization of Three-Manifolds, Massachusetts Institute of Technology, Department of Mathematics Simons Lecture Series, September 23, 2004.

24 / 199

Abstract Stone Duality - A Logic for Topology

Nicholas Duncan
University of Canterbury

Abstract

Abstract Stone Duality (ASD) is a logical system for reasoning and computing with topological spaces. Compared to other systems of logic the quantifiers \forall and \exists are restricted to compact and overt spaces, respectively, in ASD. The concept of overt spaces is not seen in topology, as all topological spaces are overt, but they play an important computational role in ASD.

In this paper we start with the definition of topology and relate it to computability. Then we focus on the construction of the type theory underlying ASD, starting with the types, which represent spaces, moving on to terms, which represent continuous functions, and finally reaching judgements, which deal with proving results in the calculus.

Next we will consider how local compactness allows computation in the calculus and the connection to interval arithmetic. Finally we compare this system to other computable systems, like Recursive Analysis.

Introduction

Abstract Stone Duality (ASD) is a logical system created by Paul Taylor for reasoning and computing with topological spaces. It is named after Stone's duality between Boolean algebras and Stone spaces. This duality manifests itself in the fact that spaces are also algebras, so we can avoid the use of sets by using the spaces as carriers of those algebras. However, in this article we will not dwell on this important aspect of ASD, instead we will focus on the logic. This system is given as an example of a logic that is suitable for a specific domain in mathematics.

Instead of starting with a large all-encompassing system, such as a set theory, and then placing continuous and computable structures on sets, we begin with a logical system in which everything is computable and continuous and the operations of the system preserve these properties.

This system is a type theory whose types represent topological spaces, and the terms of the calculus represent continuous functions. Everything that can be constructed preserves continuity and computability.

A major restriction in this system is that quantifiers may only range over certain kinds of spaces. The universal quantifier \forall can only quantify over compact spaces, like the unit interval $[0, 1]$, but not \mathbb{N} or \mathbb{Q} . The existential quantifier \exists can only quantify over overt spaces. Classically all topological spaces are overt, however in some constructive topological settings like locale theory not all spaces are overt. Many of the spaces in ASD are overt, like \mathbb{N} , \mathbb{Q} , and \mathbb{R} , however not all spaces are overt. Indeed those that are overt embody some computational process to access their elements. The subspace of all zeros of a function is often not overt, since deciding whether a real number is equal to zero is not computable.

Parts of this calculus, like the real numbers, can be transformed into programs using interval arithmetic. The calculus insures that these programs succeed and return an interval solution to within a given tolerance. This way results can be extracted from the calculus.

In comparison with other approaches to computable topology the closed interval $[0, 1]$ is compact in ASD. This allows us to use the universal quantifier over closed bounded intervals, which is vital for the translation to interval arithmetic.

Topology and Observability

We start with the definition of a topological space.

Definition 1. A *topological space* consists of a set X of points along with a set \mathcal{T} of subsets of X , whose elements are called open subsets, such that

- The subsets \emptyset and X are open subsets.
- If U and V are open subsets so is $U \cap V$.
- If $\{U_i\}_{i \in I}$ is a collection of open subsets then $\bigcup_{i \in I} U_i$ is an open subset.

Example 2. The real numbers \mathbb{R} has the Euclidean topology where the open sets are arbitrary unions of open intervals (a, b) :

$$U = \bigcup_i (a_i, b_i), \text{ where } (a_i, b_i) = \{x \in \mathbb{R} \mid a_i < x < b_i\}$$

We compare the definition of open subsets with the concept of observable properties of a set. An observable property is some subset in which membership is semi-decidable. A non-rigorous definition is the following:

Definition 3. Let X be a set. A subset S of X is *observable* if there is a computer program, such as a Turing machine, which when given an encoding of an element x of X , halts if $x \in S$, or loops (runs forever) otherwise.

Example 4. The standard example of observable subsets are the recursively enumerable subsets of \mathbb{N} . Every recursively enumerable subset is given by a computer program, and an element x is in a recursively enumerable subset if and only if this computer program halts on input x .

There are some similarities between open subsets of a topological space and observable subsets:

- The subset \emptyset is observable, just use a program which runs forever. This program never halts, so it does not accept any element of X .
- The subset X is observable, use the program which immediately halts.
- If U and V are observable with programs u and v , then $U \cap V$ is observable. Take the program which runs u until it halts, then it runs v . If x is in U and V then both programs halt, so x is in $U \cap V$. If x is not in both U and V then one of the programs will loop on input x . This shows that $U \cap V$ is observable.
- Let $(U_i)_{i \in \mathbb{N}}$ be a sequence of observable subsets, where the sequence of accepting programs is also computable (e.g. if the computer programs are encoded by numbers u_i , then the function $i \mapsto u_i$ should be computable). Then $U = \bigcup_{i \in \mathbb{N}} U_i$ is also observable.

To see this let (u_i) be the sequence of computer programs in which u_i accepts U_i . We construct a computer program u that takes an input x and interleaves the computations of each $u_i(x)$, terminating as soon as one $u_i(x)$ terminates. This program could perform one step of $u_0(x)$, then one step of $u_0(x)$ and $u_1(x)$, then another of $u_0(x)$, $u_1(x)$ and $u_2(x)$, and so on. Each loop we introduce a new program in the sequence. Since the sequence is computable this method of introducing new programs is also computable.

If x is in some U_i then eventually the program u_i will halt. If x is not in any of the subsets U_i then each program u_i will loop on input x , so the program u above will also loop on x . Hence the union is observable.

So we see that observable properties have binary intersections, but only some unions, and the indexing set of these unions must somehow be computable. If the sequence (U_i) was not computable then we cannot construct a new program in each iteration of the program u above. Furthermore, if the indexing set of the union is not countable then we cannot interleave the operations like we did in u , and we would miss some elements of the indexing set.

In ASD we do not have all unions, but we do have computable unions like in the observable subsets case. The indexing spaces of these unions will be overt spaces, which we will define shortly. If a collection of open subspaces is indexed by an overt space then the union is open. However not all spaces are overt, so we do not have all unions.

Open Subsets vs Predicates vs Functions

Instead of treating observable properties as subsets we will treat them as continuous functions of a special kind. This will reduce the number of primitive concepts that we have to consider. First we give the definition of a continuous function.

Definition 5. Given spaces X and Y a *continuous function* $f: X \rightarrow Y$ is a function such that for every open subset U of Y the *inverse image* $f^{-1}(U)$ is an open subset of X . The inverse image of U is the set:

$$f^{-1}(U) = \{x \in X \mid f(x) \in U\}$$

To treat open subsets as continuous functions we need a special topological space, called the Sierpinski space.

Definition 6. The *Sierpinski space* Σ classically consists of two points, which we call T and F , and the topology consists of the three subsets \emptyset , Σ and $\{T\}$. Note that $\{F\}$ is not open.

Given a continuous function $\phi: X \rightarrow \Sigma$ we have an open subset $\phi^{-1}(\{T\})$ of X . Conversely, given any open subset U of X we can define a continuous function $X \rightarrow \Sigma$, which classically is defined by $f(x) = T$ if $x \in U$, or $f(x) = F$ otherwise.

This construction gives us a correspondence between continuous functions $X \rightarrow \Sigma$ and open subsets of X . This can be further extended to closed subsets of X , which classically are the set complements of open subsets. A closed subset is given by the inverse image of the closed set $\{F\}$.

Now let us consider the continuous functions $\Sigma \rightarrow \Sigma$. We know that these correspond to open subsets of Σ , of which classically there are three: \emptyset , Σ and $\{T\}$. They correspond

to the continuous functions F (constantly false), T (constantly true), and the identity function.

Notice that there is no continuous function which swaps T and F . This has important computational significance. The space Σ can be thought of as the space of termination possibilities of a computer program - either a program terminates (T), or a program loops (F). If we think of a space X as the space of inputs of a program, and $f: X \rightarrow \Sigma$ as a computer program recognising an observable property, then $f(x) = T$ if the program halts, or $f(x) = F$ if the program loops.

If we had a function $\neg: \Sigma \rightarrow \Sigma$ which swaps T and F then the set complement of any observable property would also be observable, just take the corresponding function $f: X \rightarrow \Sigma$ and post-compose with \neg . In the recursively enumerable subset example this would mean that co-recursively enumerable subsets would also be observable, so the halting problem would be decidable. This is not computable, so we would lose computational ability if \neg was an acceptable function. Luckily the topology of Σ prevents this behaviour.

Objects of ASD

We have seen that open subsets of a space X can be represented by a continuous function $X \rightarrow \Sigma$. The calculus allows us to abstract away from the set theoretic nature of topologies, and so instead of considering open subsets of a space X we will consider continuous functions $X \rightarrow \Sigma$. Since we also want to abstract away from the set-theoretic nature of functions we use the word *morphism* instead of continuous function.

ASD is a type theory whose types represent spaces. How do we represent the topology on a space X ? Classically it is given by a collection of subsets of X , but we have seen that these subsets may be represented by morphisms $X \rightarrow \Sigma$. To avoid the use of sets this collection of morphisms, which we denote Σ^X , should itself be a space. So the topology of a space X in ASD is itself a space, Σ^X . Classically, for this to be a suitable space X must be a locally compact topological space, and then we can give Σ^X the Scott topology. Locally compact spaces will be considered later on in this article, but we will not cover the Scott topology. See [6] for details. So the classical model of ASD will interpret the types as locally compact topological spaces.

Suppose we have interpretations for the types X and Σ^X , how do we ensure that Σ^X is the topology on X ? For this we use a notion from category theory called a monad. If you do not know about monads then feel free to skip this paragraph. Monads allow us to define algebras whose carriers are not necessarily sets, and the arity of the operations in the algebra do not need to be indexed by sets. We require that the adjunction $(\Sigma^{(-)} \dashv \Sigma^{(-)})$ be monadic, where $\Sigma^{(-)}$ is the exponential functor. This makes the objects Σ -algebras, and this method bypasses any requirements of underlying sets. For more details see [5].

This leads us to our first axiom, which gives the types of ASD. The calculus of ASD consists of four syntactic elements: types, terms, statements, and judgements. Many of these depend on each other, so the formal definition of the calculus requires a mutually inductive definition. We will give parts of the calculus independently, and some stages may refer to future stages of definition. This is not intrinsic to ASD itself, as other type theories have this difficulty.

Axiom 1. The *types* of ASD consist of the following:

- The basic types 1 , Σ and \mathbb{N} .

- If X is a type then so is Σ^X .
- If X and Y are types, so is $X \times Y$.
- Technical condition: If X is a type then any Σ -split subspace is also a type. This construction comes from the monad above, and allows the construction of a variety of derived types. These types are denoted by $\{X \mid E\}$, where E is a special term of type $\Sigma^{X \times \Sigma^X}$ called a *nucleus*. See [4] for details on this construction.

The derived types of ASD can be constructed from the type constructors above, and they include the empty space $\mathbf{0}$, \mathbb{Q} , \mathbb{R} , $[0, 1]$, and many other spaces.

In a model of ASD these types are sent to certain objects, but note that the derived type \mathbb{R} need not necessarily be interpreted as the real numbers. In certain constructive settings the closed interval $[0, 1]$ is not compact, so \mathbb{R} can not be interpreted as the standard real numbers in such a setting. Also note that the classical interpretation of \mathbb{Q} is with the discrete topology, not the order topology. We will see more of this later.

Logical Terms of the Calculus

Now we will consider the logical terms of the calculus, which will represent logical properties and subspaces. The terms of type Σ are called propositions, and the terms of type Σ^X are called predicates.

Axiom 2. The *logical terms* of ASD consist of the following:

- *Variables:* The types Σ and Σ^X all have a countable supply of variables, often denoted σ, τ for propositions and ϕ, ψ for predicates. Each variable has an associated type.
- *Constants:* \top, \perp are terms of type Σ , which represent true and false.
- *Connectives:* if σ and τ are terms of type Σ , the connectives $\sigma \vee \tau$ and $\sigma \wedge \tau$ are terms of type Σ , representing disjunction and conjunction, respectively.
- *λ -abstraction:* if $\phi(x)$ is a term of type Σ with a free variable x of type X then $\lambda x. \phi(x)$ is a term of type Σ^X . λ -abstraction is used to construct functions in type theory.
- *λ -application:* if ϕ is a term of type Σ^X and a is a term of type X , then $\phi(a)$ is a term of type Σ . This term is also denoted ϕa .
- *Equality:* if N is a *discrete* space and n and m are terms of type N , then $n =_N m$ is a term of type Σ .
- *Inequality:* if H is a *Hausdorff* space and n and m are terms of type H , then $n \neq_H m$ is a term of type Σ .
- *Universal quantification:* If X is *compact* and $\phi(x)$ is a term of type Σ with a free variable x of type X , then $\forall x: X. \phi x$ is a term of type Σ .
- *Existential quantification:* Similarly, if X is *overt*, then $\exists x: X. \phi x$ is a term of type Σ .

Note that we do not have the connectives \neg or \rightarrow of type Σ . These would lead to non-computability, as we have seen earlier.

These logical terms require us to consider certain kinds of spaces. The discrete, Hausdorff and compact spaces have their regular interpretation in topology. However, classically all spaces are overt, so the concept does not show up in classical topology.

First we consider the *discrete* spaces. These are spaces in which equality is observable. Classically this corresponds to spaces whose diagonal subset

$$\{(x, x) \in X \times X \mid x \in X\}$$

is open, which implies that all subsets are open. In ASD the spaces \mathbb{N}, \mathbb{Z} and \mathbb{Q} are discrete. Note that \mathbb{R} is not discrete. If we consider real numbers as infinite decimal expansions then to check equality we are required to check the entire expansion, which is not observable as it would take an infinite amount of time.

Next are the *Hausdorff* spaces. In these spaces inequality is observable. Classically these correspond to Hausdorff topological spaces, where the diagonal is closed. In ASD the spaces $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$, and $[0, 1]$ are all Hausdorff. Classically all discrete spaces are Hausdorff, but this is not so in ASD. Open subsets are not closed under arbitrary unions, so the classical proof that discrete implies Hausdorff does not apply.

The *compact* spaces correspond to the compact spaces in topology, which classically are given by the finite subcover property:

Definition 7. A topological space X is *compact* if for any family of open subsets $\{U_i\}_{i \in I}$ whose union is the whole space X , there is a finite subset J of I such that the subfamily $\{U_j\}_{j \in J}$ also covers the whole space X .

Note that we do not require compact spaces to be Hausdorff. In ASD the bounded closed intervals $[a, b]$ are compact, as well as the Sierpinski space Σ .

The *overt* spaces are invisible in classical topology. A topological space X is overt if the unique continuous function $X \rightarrow 1$ sends open subsets to open subsets. Classically all topological spaces are overt, however in constructive locale theory not all spaces have this property. Earlier terminology from locale theory called overt spaces open spaces, as the unique map $X \rightarrow 1$ is open. However this clashes with the terminology for open subspaces, so overt spaces are the preferred terminology.

In locale theory an overt locale has a *positivity predicate* $\text{Pos}(a)$, which holds if a is inhabited. Constructively not all non-empty subsets are inhabited, i.e. have an element, so overt spaces have a way of recognizing when an open is inhabited.

Here is a chart taken from [1], Examples 4.26, which gives a variety of different types of ASD and the properties they have:

space	overt	discrete	compact	Hausdorff
$\mathbb{N} \times \Sigma$	✓	✗	✗	✗
\mathbb{R}, \mathbb{R}^n	✓	✗	✗	✓
Σ	✓	✗	✓	✗
$[a, b], 2^{\mathbb{N}}$	✓	✗	✓	✓
free SK -algebra	✓	✓	✗	✗
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}$	✓	✓	✗	✓
K-finite	✓	✓	✓	✗
finite	✓	✓	✓	✓
set of codes of non-terminating programs	✗	✓	✗	✓

The *free SK-algebra* is the free algebra with a non-associative operation $x \cdot y$ and symbols S and K such that the two equalities:

$$((S \cdot x) \cdot y) \cdot z = (x \cdot z) \cdot (y \cdot z) \text{ and } (K \cdot x) \cdot y = x$$

hold. This represents combinatory logic with the combinators S and K . Equality is observable, as we can loop through all possible equalities. However, inequality is not observable, as programs can be represented as combinators.

K-finite spaces correspond to spaces in which all the elements can be finitely listed, but there may be repetitions, as inequality is not observable. Some examples are subspaces of a finite space given by a semi-decidable predicate.

Other Terms of the Calculus

Now that we have considered the logical terms we move on to the other terms of the calculus.

Axiom 3. Non-logical terms of ASD.

Variables: Every term has a countable supply of *variables*. Each variable has a distinguished type.

Product terms:

- If s is a term of type S , and t is a term of type T , then $\langle s, t \rangle$ is a term of type $S \times T$, representing an ordered pair.
- If x is a term of type $S \times T$, then $\pi_1 x$ is a term of type S and $\pi_2 x$ is a term of type T . These represent projections from an ordered pair to one of its components.

Numerical terms:

- *Zero:* 0 is a term of type \mathbb{N} .

- *Successor*: If n is a term of type \mathbb{N} , then Sn is a term of type \mathbb{N} .
- *Definition by description*: If $\phi(n)$ is a term of type Σ with a free variable n of type N then **the** $n.\phi n$ is a term of type \mathbb{N} if the two statements:

$$\phi n \wedge \phi m \Rightarrow n =_N m \quad \text{and} \quad \exists n: \mathbb{N}.\phi n \Leftrightarrow \top$$

hold. Since we have not covered statements yet these can be read as the uniqueness and existence properties of definition by description.

- We also have terms for *primitive recursion* over any type.

Note that definition by description turns logical predicates into terms of type \mathbb{N} . This allows unbounded minimization to be represented in the calculus, so we can represent all partial recursive functions. However unbounded minimization implies that equality in the calculus is no longer decidable, but we wanted to represent all computable functions, so this is not a problem.

We can derive more predicate terms from the ones above, like $\leq, \geq, <, >$ for \mathbb{N} and \mathbb{Q} , by using primitive recursion and equality.

There are also derived terms for the derived types. For example the real numbers have the following derived terms, which we give as an axiom:

Axiom 4. The type \mathbb{R} has the following terms:

- *Constants*: 0 and 1 are terms of type \mathbb{R} . Note that 0 is different from the term 0 of type \mathbb{N} , so we use context to determine which term we mean.
- *Operators*: If x and y are terms of type \mathbb{R} , then $x + y, x \times y, x - y$ are terms of type \mathbb{R} . If we have the judgement $y > 0 \vee y < 0 \Leftrightarrow \top$, then $x \div y$ is a term of type \mathbb{R} .
- *Dedekind cuts*: Given predicates δ and ν of type $\Sigma^{\mathbb{Q}}$, or even of type $\Sigma^{\mathbb{R}}$, then **cut** $du.\delta d \wedge \nu u$ is a term of type \mathbb{R} if the following six judgements hold:

$$\begin{array}{ll} \exists e.(d < e) \wedge \delta e \Leftrightarrow \delta d & \exists t.\nu t \wedge (t < u) \Leftrightarrow \nu u \\ \exists d.\delta d \Leftrightarrow \top & \exists e.\nu e \Leftrightarrow \top \\ \delta d \wedge \nu u \Rightarrow d < u & \delta d \vee \nu u \Leftarrow (d < u) \end{array}$$

These judgements have been organised into two columns to illuminate the symmetry between them. The first line states that the cuts are *rounded*, so have no maximum or minimum elements. The next line states that the cuts are *inhabited*. The fifth judgement states that cuts are *disjoint*, and the final judgement states that cuts are *order-located*.

This axiom is not strictly necessary, as the reals can be constructed in the calculus, but it is useful to see the properties of the real numbers. See [1] for the construction of \mathbb{R} from the basic types.

Statements and Judgements

We have given many of the terms of the calculus, the next thing to consider are the statements. These describe a relationship, such as equality, between two terms.

Definition 8. There are two types of *statements*:

- If a and b are terms of type X , then $a = b$ is a statement. Note we have no subscript on the equality sign. Such a statement expresses the fact that a and b represent the same element of X .
- If α and β are terms of type Σ^X , then $\alpha \Rightarrow \beta$, $\alpha \Leftarrow \beta$ and $\alpha \Leftrightarrow \beta$ are statements. The statement \Rightarrow states that the open subspace corresponding to α is included in the open subspace corresponding to β . The statement \Leftrightarrow is the same as $=$, but we will use the form \Leftrightarrow for predicates.

We stated earlier that \neg and \rightarrow are not terms of type Σ , so are not propositional connectives. However we can use statements to give some form of negation or implication. The proposition $a \rightarrow b$ can be represented as $a \Rightarrow b$, and $\neg a$ can be represented by $a \Rightarrow \perp$. However the logic is limited, in which we can only have one implication or negation, and it must occur as the outermost connective.

Finally we reach judgements, which are used to express some logical truth in our system.

Definition 9. A *context* Γ for a judgement consists of two lists. One list of variable declarations, e.g $n: \mathbb{N}$, and another list of statements, where the free variables of the statements occur in the first list.

Example 10. The following is a valid context:

$$n: \mathbb{N}, \phi: \Sigma^{\mathbb{N}}, \phi n \Leftrightarrow \top$$

Definition 11. There are three types of *judgements*:

- *Valid type formation:* This has the form $\vdash X: \text{type}$, and states that X is a valid type.
- *Term formation:* This has the form $\Gamma \vdash a: X$, which states that a is a valid term of type X in the context Γ .
- *Statement formation:* Similarly this has the form $\Gamma \vdash s: X$, where s is a statement between terms of type X . This states that the statement s holds in the context Γ .

Note that type formation does not have a context, so we cannot form types which depend on terms, or dependent types as they are called in type theory. Future extensions of the calculus may allow such types.

We have given a large part of the syntax of the calculus, so now we will consider how to interpret this syntax. A type judgement can just be interpreted as a space of some sort, such as a topological space, a locale, or a more exotic object. We will assume that we are interpreting the types of the calculus as topological spaces.

The representation of a context is similar to the type judgement. The variable declarations are represented by a product of topological spaces, one for each variable. The list of statements then forms a subspace of that topological space. For the context in Example 10 above the representation will be the subspace

$$\{(n, \phi) \in \mathbb{N} \times \Sigma^{\mathbb{N}} \mid \phi(n) = \top\}.$$

A term judgement $\Gamma \vdash a: X$ is represented by a continuous function from the representation of Γ to the representation of X . So in this calculus terms are continuous functions.

The empty context is interpreted as the one element topological space $\{*\}$, so term judgements of the form $\vdash a : X$ represent continuous functions $\{*\} \rightarrow X$. But such functions correspond to elements of X , hence term judgements with empty contexts represent elements of a space.

Finally for statement judgements we have to consider the structure of Σ . Every topological space is equipped with a preorder called the *specialization preorder*, denoted by the binary relation \leq (rather than \leq). The specialization preorder on Σ is $F \leq T$, and on Σ^X it is

$$f \leq g \text{ if and only if } \forall x \in X. f(x) \leq_{\Sigma} g(x)$$

The judgement $\Gamma \vdash \alpha \Rightarrow \beta : \Sigma^X$ states that the functions f and g which represent α and β , respectively, satisfy $f \leq g$. The judgement $\Gamma \vdash \alpha \Leftrightarrow \beta : \Sigma^X$ similarly state that the functions f and g are equal.

The specialization preorder on any Hausdorff spaces is trivial, i.e. $x \leq y$ implies $x = y$, so this preorder is not often seen in topology.

Aside. From the category theory viewpoint the category of topological spaces is an enriched category over preorders. This means that every homset carries a preorder, and function composition preserves this order. For the category of locally compact topological spaces the preorder is a partial order, and the category is enriched over posets.

Example 12. An example of a judgement is one for ϵ - δ continuity. Let f be the representation of a term

$$x : \mathbb{R} \vdash f(x) : \mathbb{R}$$

Classically the function f is continuous at x if for all $\epsilon > 0$ there exists a $\delta > 0$ such that for all y , if $|x - y| < \delta$ then $|f(x) - f(y)| < \epsilon$. To convert this into the ASD calculus we run into a few problems. First ϵ is given by quantifying over all positive real numbers. However the space $(0, \infty)$ or even \mathbb{R} is not compact, so we cannot perform this quantification. This can be fixed by converting it to a statement, but note that we are no longer allowed to use implication or negation in the rest of the term.

The existence of δ is fine, but the quantification over the interval $(x - \delta, x + \delta)$ leads to another problem. We can fix this by quantifying over the compact interval $[x - \delta, x + \delta]$ instead. We end up with the judgement:

$$x : \mathbb{R}, \epsilon : \mathbb{R} \vdash \epsilon > 0 \Rightarrow \exists \delta : \mathbb{R}. \delta > 0 \wedge \forall y : [x - \delta, x + \delta]. |f(x) - f(y)| < \epsilon$$

One slight problem with this formulation is that the type $[x - \delta, x + \delta]$ depends on the terms x and δ but we have not included dependent types in the calculus. A future adjustment of the calculus may allow such types, but for the cases of quantifying over bounded intervals in the reals we have the following translation: Given a predicate ϕ of type $\Sigma^{\mathbb{R}}$ we transform

$$\forall x : [a, b]. \phi x$$

into the following acceptable form:

$$\forall x : [0, 1]. \phi(ax + b(1 - x))$$

This translation is straightforward, so we prefer to use the version above.

The Logical Axioms of ASD

We have given the syntax of the calculus and how to interpret the syntax, so now we show how to reason with the calculus. This is done through proof rules, which have the two forms

$$\frac{\text{Judgement 1}}{\text{Judgement 2}}$$

$$\frac{\text{Judgement 1}}{\underline{\text{Judgement 2}}}$$

The left rule states that if Judgement 1 holds then we can assert that Judgement 2 holds as well. The double line version states that if Judgement 1 holds then we can assert that Judgement 2 holds, and vice-versa.

We start with the proof rules for propositions, which are the terms of type Σ . We have the connectives \wedge and \vee representing conjunction and disjunction.

Axiom 5. The logical connectives \wedge and \vee satisfy the following proof rules:

$$\frac{\Gamma, \sigma \Leftrightarrow \top \vdash \alpha \Rightarrow \beta}{\Gamma \vdash \sigma \wedge \alpha \Rightarrow \beta} \qquad \frac{\Gamma, \sigma \Leftrightarrow \perp \vdash \beta \Rightarrow \alpha}{\Gamma \vdash \beta \Rightarrow \sigma \vee \alpha}$$

If we interpret the spaces as sets then the first rule is constructively valid, but the second is not. This proof rule can be interpreted as the equivalence between $\neg\sigma \rightarrow (\beta \rightarrow \alpha)$ and $\beta \rightarrow (\sigma \vee \alpha)$. If we take β to be true, and α to be $\neg\sigma$, then we get $(\sigma \vee \neg\sigma)$. However we cannot express negation in ASD, so we do not have the law of excluded middle. In fact this proof rule represents a relationship between topological properties of spaces. While the second rule is not constructively valid if interpreted as sets, it is constructively valid if we interpret spaces as locales.

Next we consider the logical properties of the space Σ . We have seen that classically there are three open subsets of Σ . We want to avoid mentioning subsets, so instead we consider continuous functions $\Sigma \rightarrow \Sigma$. Classically these functions are determined by their values on T and F . The Phoa principle represents this property, and is a major axiom in the system.

Axiom 6. *The Phoa principle:* Functions from Σ to Σ are determined by their values on \perp and \top :

$$\Gamma, F: \Sigma^\Sigma, \sigma: \Sigma \vdash F\sigma \Leftrightarrow F\perp \vee \sigma \wedge F\top$$

This has the consequence that F is monotone: $F\perp \Rightarrow F\top$, therefore we cannot represent a negation function.

Now we consider the equational axioms of the system. Many of the equality statements are β or η rules for type constructors. See [3] chapter 7.2 for an overview of these types of equalities in type theories.

Axiom 7. Equational axioms:

- *Lattice structure:* The predicates of a type Σ^X form a *distributive lattice*. The conjunction of two predicates ϕ and ψ is the predicate $\lambda x. \phi x \wedge \psi x$ and the definition for disjunction is similar. We have the following proof rules linking the specialization order with these connectives:

$$\frac{\Gamma, \phi, \psi: \Sigma^X \vdash \phi \Rightarrow \psi}{\frac{\Gamma, \phi, \psi: \Sigma^X \vdash \phi \wedge \psi \Leftrightarrow \phi}{\Gamma, \phi, \psi: \Sigma^X \vdash \phi \vee \psi \Leftrightarrow \psi}}$$

- *Application:* The β and η rules hold for λ -abstraction and application.

$$(\lambda x.\phi)a = \phi[x := a] \qquad \lambda x.\phi x = \phi$$

where the notation $\phi[x := a]$ means that we have replaced all the free-variables x with the term a .

- *Projections:* The β and η rules hold for projections and pairing:

$$\pi_1 \langle a, b \rangle = a, \pi_2 \langle a, b \rangle = b, \text{ and } \langle \pi_1 p, \pi_2 p \rangle = p$$

- *Recursion:* The β and η rules hold for recursion over \mathbb{N} .

Now we will consider the logical axioms for discrete and Hausdorff spaces.

Definition 13. A space X is *discrete* if the left proof rule holds, and it is *Hausdorff* if the right proof rule holds:

$$\frac{\Gamma \vdash n = m : X}{\Gamma \vdash (n =_X m) \Leftrightarrow \top : \Sigma} \qquad \frac{\Gamma \vdash n = m : X}{\Gamma \vdash (n \neq_X m) \Leftrightarrow \perp : \Sigma}$$

These proof rules allow us to pass from the equality/inequality predicate to an equality statement and vice-versa.

For compact and overt spaces we also require proof rules.

Definition 14. A space X is *overt* if the left proof rule holds, and it is *compact* if the right rule holds:

$$\frac{\Gamma, x : X \vdash \phi x \Rightarrow \sigma}{\Gamma \vdash \exists X. \phi x \Rightarrow \sigma} \qquad \frac{\Gamma, x : X \vdash \sigma \Rightarrow \phi x}{\Gamma \vdash \sigma \Rightarrow \forall X. \phi x}$$

With these proof rules we can use the quantifiers in similar ways to how they are normally used. For example, if we have the judgement $\vdash \forall x : X. \phi x \Leftrightarrow \top : \Sigma$, and we have a term judgement $\vdash a : X$, then we may assert $\vdash \phi a \Leftrightarrow \top : \Sigma$.

In constructive mathematics if we have a term of the form $\exists x : X. \phi x$ then there exists a term $a : X$ such that ϕa holds. This result does not hold for ASD. Even in the term model, whose types and terms only consist of those that can be constructed from the axioms of the system, we do not have this property. However if the context Γ only consists of overt spaces then in the term model iff $\Gamma \vdash \exists x : X. \phi x \Leftrightarrow \top : \Sigma$ holds we can construct a term a such that $\Gamma \vdash \phi a \Leftrightarrow \top : \Sigma$ holds.

Furthermore, even if we prove $\vdash \phi a \Leftrightarrow \top : \Sigma$ for every term a of a compact space X it does not imply that we can assert $\vdash \forall x : X. \phi x \Leftrightarrow \top : \Sigma$, even in the term model. Effectively the spaces represented by the types in this system consist of more than their definable elements. This corresponds with the localic point of view where spaces are not determined by their points.

Now we will move on to a numerical axiom. The Archimedean axiom prevents non-standard models of the rationals, and allows us to extract numerical results from the calculus up to a given tolerance.

Axiom 8. The spaces \mathbb{Q} and \mathbb{R} satisfy the *Archimedean principle*. For p and q of type \mathbb{Q} or \mathbb{R} we have the judgement

$$q > 0 \Rightarrow \exists n: \mathbb{Z}. q(n - 1) < p < q(n + 1)$$

There are a few remaining axioms which we will not spend much time on, but they do play an important role in ASD. The Scott continuity axiom is a topological one and is based on the idea that continuous functions $\Sigma^X \rightarrow \Sigma$ preserve directed unions.

Another collection of axioms involve the term **focus** which we have not mentioned. This term is based on the monadic principle, and can be used to construct points of a space X from certain points of Σ^{Σ^X} . For more details see the papers [4] and [5].

Local Compactness and Interval Arithmetic

The classical model of ASD is given by locally compact topological spaces, and a constructive model is given by locally compact locales. The property of being locally compact allows us to perform some form of computation on the space. We start with the topological definition.

Definition 15. A topological space X is *locally compact* if for every $x \in X$ and every open subset U containing x , there exists an open subset V and a compact subset K such that

$$x \in V \subseteq K \subseteq U \subseteq X$$

A locally compact space is *computably generated* if the above open and compact subsets can be computed by a computer program. For the precise definition of computably generated locally compact space see [6].

Example 16. For the real numbers we can take the open subsets to be open intervals with rational endpoints, that is the open intervals (a, b) with $a, b \in \mathbb{Q}$, and the corresponding compact subset can then be taken to be $[a, b]$. As the open subsets of \mathbb{R} are given by unions of open intervals, each x is inside some open interval (a, b) . Now take rational numbers between a and x , and then another between x and b , to get an open interval with rational endpoints containing x .

The computable open subsets and compact subsets coming from local compactness will be called cells, and will be denoted with a bold symbol like \mathbf{x} . We will now convert the topological definition of locally compact into an ASD statement. We assume that all spaces are computably generated, so the cells will be indexed by an overt discrete space N . For example, the cells of \mathbb{R} can be indexed by $\mathbb{Q} \times \mathbb{Q}$ or even \mathbb{N} .

Local compactness states:

$$a: X, \phi: \Sigma^X \vdash \phi a \Leftrightarrow \exists \mathbf{x}: N. a \in \mathbf{x} \wedge \forall x: \mathbf{x}. \phi x: \Sigma$$

where $a \in \mathbf{x}$ means that a is in the open interval subspace corresponding to the cell \mathbf{x} . In other words this statement says that a is in the open subset corresponding to ϕ if and only if there is a cell \mathbf{x} such that a is in the open subset corresponding to that cell, and for all elements x in the compact subset corresponding to the cell \mathbf{x} , the element x is in the open subset corresponding to ϕ .

So how do we compute with locally compact spaces? The cells are indexed by an overt discrete space, so they can be represented on a computer. The locally compact property

allows us to replace terms ϕa with the right hand side above, which initiates a search for a cell which satisfies the necessary conditions.

The condition $a \in \mathbf{x}$ is observable, as \mathbf{x} represents an open subset, so a non-deterministic search will find cells which satisfy this. The other condition, $\forall x: \mathbf{x}. \phi x$ is a bit harder to satisfy. However, if the cell is chosen to be sufficiently small then this quantifier can be replaced with a different term involving interval arithmetic, which can be verified by computer. Any predicate ϕ can be transformed into one which only involves logical connectives, interval operations, and existential quantification over an overt discrete space like \mathbb{N} . The transformed predicate does not involve quantification over \mathbb{R} or $[a, b]$. For details on this translation see [2].

The process of converting a term like ϕ into an interval arithmetic program is involved, and one method requires a form of Prolog which includes non-deterministic branching, λ -calculus and interval constraints. The running time of such a program is not yet known, but the calculus guarantees that the program will eventually halt if given a required precision.

Comparison to Other Systems

There are a number of models of ASD, with the classical one being locally compact topological spaces. Constructive locale theory and formal topology can also provide models. Furthermore it is possible to use other meta-theories, like Bishop's Constructive Mathematics or Martin-Löf's Type Theory to construct the term model of ASD.

First we consider the soundness and completeness of the calculus. This system is sound since the axioms are derived from topological properties of locally compact spaces. For completeness we need to consider the term model of ASD, which consists of only the types and terms which can be constructed from the axioms alone. The term model can be characterised in terms of certain topological spaces:

Theorem 17 (Theorem 17.5 in [6]). *The category of types and terms of the term model of ASD is equivalent to the category of computably generated locally compact spaces and computable continuous functions.*

This implies that any computably generated locally compact space can be represented by a type in ASD, and any computable continuous function can be represent by a term, up to homeomorphism. However, one slight problem with this theorem is that it requires classical logic, as it uses the proof that locally compact locales and locally compact topological spaces agree.

Next we compare ASD to other systems which involve computation with real numbers. One such system is Recursive Analysis. A fundamental property of ASD is that the unit interval is compact. This does not hold in Recursive Analysis due to the the existence of singular covers. However ASD can be interpreted in Recursive Analysis, which may seem to cause a problem. The reason why there is no difficulty is that the real numbers in ASD differ from the real numbers in the meta-theory. Interpreting the reals from ASD will give a different object than the reals in Recursive Analysis.

Even though the two objects are different there still could be a problem with singular covers. In the term model of ASD constructed in Recursive Analysis it is possible to define a sequence of intervals (d_n, e_n) with rational endpoints, whose total length is bounded by $\frac{1}{2}$, and it is possible to prove in the meta-theory that (Remark 15.4 in [1])

$$\text{if } \vdash t: [0, 1] \text{ then } \vdash \exists n: \mathbb{N}. d_n < t < e_n$$

This would seem to say that every element of the unit interval is covered by one of the intervals in the sequence, therefore the unit interval is covered by intervals whose total length is less than $\frac{1}{2}$. However it is not possible to pass from these judgements to

$$t: [0, 1] \vdash \exists n: \mathbb{N}. d_n < t < b_n$$

The proof in the meta-theory only talks about definable elements of the unit interval, but this is not enough in ASD. Like locale theory, the unit interval consists of more than just its definable elements. In this way singular covers do not cause any difficulties for ASD, and there is no contradiction with a compact unit interval.

Conclusion

The logical calculus of Abstract Stone Duality is an interesting system from a logical perspective as it involves a fairly weak logic. The use of implication and negation is severely restricted, but we have seen that this is necessary to ensure continuity and computability. The quantifiers are also restricted to certain types, the compact and the overt types. Finally equality and inequality are also restricted, to the discrete and the Hausdorff types, respectively.

These restricted types all correspond to various properties of spaces: discrete, Hausdorff, compact, and overt. The overt spaces are not visible classically, as all topological spaces are overt. However in constructive settings overtness is a very useful property for a space to have. In this calculus overtness embodies a computational principle. Due to the computational properties of this system results can be computed and extracted, in a form which involves interval arithmetic.

For an example application of ASD to the Intermediate Value Theorem see the paper [7] where two versions of the Intermediate Value Theorem are given. One version is computational and the other is not. This is because the space of computable solutions is overt, whereas the space of non-computable solutions is not overt. The calculus has the ability to distinguish between the two types of solutions.

Abstract Stone Duality is a prime example of a foundational system which involves a restricted logic which reflects the domain that it models. Instead of using an all-encompassing system, which may have difficulties joining computation and continuity, a smaller system which involves only those principles which preserve computability and continuity may be a more appropriate system in various domains.

References

- [1] Andrej Bauer and Paul Taylor. The dedekind reals in abstract stone duality. *Mathematical Structures in Computer Science*, 19(04):757–838, 2009.
- [2] Paul Taylor. Interval analysis without intervals. <http://www.paultaylor.eu/ASD/intawi/intawi.pdf>.
- [3] Paul Taylor. *Practical foundations of mathematics*, volume 59 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999.
- [4] Paul Taylor. Sober spaces and continuations. *Theory and Applications of Categories*, 10:No. 12, 248300, 2002.
- [5] Paul Taylor. Subspaces in abstract stone duality. *Theory and Applications of Categories*, 10:No. 13, 301368, 2002.

- [6] Paul Taylor. Computably based locally compact spaces. *Logical Methods in Computer Science*, 2(1):1:1, 70, 2006.
- [7] Paul Taylor. A lambda calculus for real analysis. *Journal of Logic and Analysis*, 2:Paper 5, 115, 2010.