Research Statement

Cameron Calk

My research is centered around *topological* methods in *computer science*. Topological methods have proven useful in studying computational systems and other subjects within computer science. Order theory plays a fundamental role in logic and is pertinent in modelling *relative computational content* of various states of computation. Topology generalises order theory, allowing the incorporation of limiting behaviour to model unbounded iteration of computational loops.

During my career, I have studied applications of topology to multi-agent systems such as distributed and concurrent computing, as well as the computational properties of rewrite systems. In these domains, algebraic and combinatorial approaches to topology are useful in modelling *forbidden* or *inconsistent configurations* in computational systems. Topology not only provides insight into the *structure* of computations, but also has been instrumental in providing *no-go* results.

1. Global research theme: computational invariants via topology

My interest in the role of topology in computer science began during my doctorate, in which I described the algebraico-topological properties of *rewriting systems* via novel algebraic techniques [9, 11, 6], and participated in the formalisation of these structures in proof-assistants [14, 15], see Section 2. In parallel, I developed and refined topological and order-theoretic approaches to *concurrent computing* [10, 8], which I explain in detail in Section 3. In both of these fields, my work focussed on identifying obstructions in computational space using topological invariants. In the years since my doctorate, I pursued these research directions [13, 12], but also, as explained in Section 4, applied my expertise in topological approaches to multi-agent systems in the context of epistemic properties of *distributed systems* [7]. Below, I provide a brief description of these domains and my contributions in each.

Rewriting systems. Rewriting systems constitute an abstract model of computation, widely used in algebra, logic and CS. One of the fundamental questions rewriting adresses is the decidability of equivalence: given two freely generated terms subject to an equivalence relation, is there an algorithm determining whether or not they are equivalent? This is known as the *word problem*, and was first studied by Thue in 1914 [54] in the context of monoids presented by *generators* and *relations*.

One of the goals of rewriting theory [2, 5] is to identify the properties of a system of calculation necessary to respond to this question. The idea is to *orient* the equivalence relation in order to produce a directed system of computational steps which reduce a given element to a *normal form*, *i.e.* a unique representative of its equivalence class. When such a reduction is possible in a finite number of computational steps, we say that the rewrite system is *convergent*. When this property is satisfied, the word problem is decidable: simply reduce elements to their normal forms and compare them.

The existence of a finite, convergent rewriting system thus provides a computational solution to the word problem. This led to the converse question of *universality* in convergent rewriting: given a monoid, finitely presented by generators and relations, does decidability of its word problem imply the existence of an associated convergent rewriting system? In the 1990s, Squier answered this question in the negative [49] by exhibiting a *topological invariant* of finitely presented monoids.

This algebraico-topological method has since been extended to categories [1], which can be thought of as multi-sorted monoids. This allows these methods to be applied to a larger family of algebraic structures presented by generators and relations [26, 32, 33]. In parallel, classic rewriting results were formulated and proved [17, 51] in *Kleene algebras*, which generalise relational algebras and are widely used to describe computational properties [19, 40, 55]. My contributions in this area unifies these two approaches, by introducing a *formal, algebraic setting* for proving topological invariants of rewriting systems [9, 11]. From this, we developed a correspondence theory relating these structures to their categorical counterparts [6, 12]. In parallel, I participated in the formalisation of these structures and their correspondences in the proof assistant Isabelle [14, 15]. These contributions are further detailed in Section 2.

Concurrency. While rewriting theory examines computational properties in the presence of many *local choices* in calculation, concurrency theory tackles a similar question in the presence of *simultaneous* calculations. In a multi-agent system, for example a multi-core processor or computer network, changes to the temporal order of events can affect the global state. Managing the *scheduling* of events and identifying *forbidden* temporal configurations is central to concurrency theory.

A topological approach to this domain has been developed [23, 28], in which executions of such a system are represented as *directed paths* through a space of computational states. The geometric and topological properties of such spaces of directed paths correspond to behavioural properties of the concurrent system. As in the case of rewriting systems, *topological invariants* have been used to behaviourally classify such systems.

In 2017, invariants known as *natural homology* and *homotopy* had recently been introduced [20], the former being used to establish bisimulations between concurrent systems [22]. However, these invariants did not capture important properties of the system, and, due to the state explosion problem in concurrent systems, are in general difficult to calculate. In Section 3, I detail my work in this domain, firstly in refining invariants of directed spaces [8] and then in relating them to more computationally tractable theories [10]. After my doctorate, I continued this research, relating such invariants to the congruences of certain lattices by means of *topological dualities* [13].

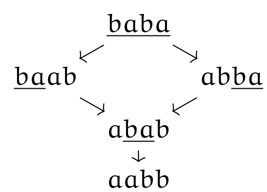
Distributed computing. Distributed systems are examples of concurrent systems, in which several independent processes attempt to coordinate via some means of communication in order to solve a task[refs]. In this domain, the focus is less on questions of scheduling, but rather on tracking the possible *epistemic states* in each possible concurrent execution. A fundamental result in distributed computing is that the *consensus task*, in which each machine starts with some value and they must collectively decide on precisely one of these values, is not solvable in the presence of crash failures [25].

Once again, topological methods have shed light into this result, as well as other questions of task-solvability. By organising local and global epistemic states into combinatorial objects with a geometric flavour called *simplicial complexes*, Herlihy and Shavit characterised task-solvability for a model of distributed communication using topology, earning them the 2004 Gödel prize for the Asynchronous Computability Theorem (ACT) [34].

However, their result only applies to one model of communication, while extensions to other models were proved using case-by-case algorithmic simulations. Using *topological duality* and *non-Hausdorff topologies*, we have established a novel approach to distributed task-solvability, thereby extending the scope of the ACT and providing many new avenues of research in this domain [7].These contributions are further elaborated in Section 4.

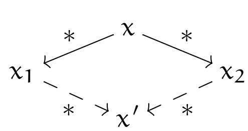
2. Computational coherence, Kleene algebras and correspondences

2.1. Rewriting as computation. The coordinate-wise addition on $\mathbb{N}^2$ can be described by considering the free words on the two letter alphabet $\{a, b\}$. Interpreting a word $w = a^{n_1} b^{m_1} \cdots a^{n_k} b^{m_k}$ as the sum $\Sigma_i(n_i, m_i)$, and considering the relation $ba = ab$, we can reorganise w as $w' = a^{n_1} \cdots a^{n_k} b^{m_1} \cdots b^{m_k}$, which corresponds to the coordinate-wise addition $(\Sigma_i n_i, \Sigma_i m_i)$. The word w' is the *normal form* obtained by considering the directed relation *generated* by applying $ba \to ab$ to each possible context in a given word. As shown on the right, this leads to several different ways of calculating the output, but always leads to a unique answer.



$(1, 1) + (1, 1) = (2, 2)$

A good system of calculation, given by a relation $\to$, should provide a *unique* output, no matter how this output is calculated. This property, called *confluence*, states that *for all* branchings, *i.e.* local choices in computation, *there exists* a confluence leading to a common result. Denoting by $\xrightarrow{*}$ finite iterations of computational steps of $\to$, and by $\leftarrow$ its inverse, the for-all/exists condition of confluence can be expressed via an inclusion of relations: if x_1 and x_2 are related by the *branching relation* $\xleftarrow{*} \cdot \xrightarrow{*}$, where $\cdot$ denotes relational composition, then they are also related by the *confluence relation* $\xrightarrow{*} \cdot \xleftarrow{*}$. This inequality, along with its diagrammatic counterpart, are pictured below on the left. *Local* confluence is similarly defined, but with one-step branchings $\leftarrow \cdot \to$.



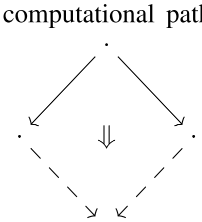
$\xleftarrow{*} \cdot \xrightarrow{*} \;\subseteq\; \xrightarrow{*} \cdot \xleftarrow{*}$

Confluence

When a rewriting system is both confluent and *terminating*, *i.e.* a normal form is reached in a finite number of steps, we say that it is *convergent*. In this case, equivalence can be decided by comparing normal forms. Newman's lemma and the Church-Rosser theorem, two central results in rewriting, together imply that a locally confluent, terminating rewriting system is convergent. In the presence of termination, this means we only need to check *local* confluence in order to prove convergence.

2.2. Coherent calculation. Rewriting thus allows us to calculate on *free* algebraic objects using directed rules. As we saw above, the rules themselves are often also generated by the ambient algebraic structure. However, the fact that several calculations produce the same outcome induces relations between the rewrite rules, meaning that they are not *freely* generated. This leads to the domain of *higher rewriting* [1], in which these relations are considered as rewrite rules on computational paths.

Interpreting confluence diagrams as *holes* in the space of rewriting paths led Squier to introduce algebraico-topological invariants of rewriting systems [49] in order to address the problem of *universality*, described in Section 1. Higher dimensional holes appear as confluence diagrams of higher rules, leading to the so-called *coherence theorem*: if the original rules are convergent, then every such hole, *i.e.* obstruction to free calculation, can be filled by some higher cell. As in the classical case, higher rewriting also provides local-to-global results, stating that it suffices to fill *local* confluences in order to fill every hole.

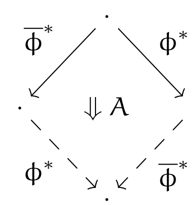


Mac Lane's coherence theorem for monoidal categories [42, p. 257-260] is a corollary of this theorem, which has also been applied, for instance, to give constructive proofs of coherence in complicated monoidal structures [32, 33], monoidal categories [29, 30] and even has applications in homotopy type theory [41]. This categorical approach reunites all dimensions of rewriting: types, algebraic structure, relations and homological properties in any dimension. It also has a strong topological flavour: a rewriting system is thought of as a higher-dimensional space in which holes represent obstructions to free calculation.

As described above, rewriting was traditionally formalised in terms of algebras of binary relations: confluence properties are described by union, composition and iteration operations, which are generalised by *Kleene algebras*. A Kleene algebra $(\mathsf{K}, +, \cdot, (-)^*)$ is equipped with monoidal addition and multiplication operations, as well as the *Kleene star*, which models unbounded finite iteration. Kleene algebras are known for their ability to capture complex computational properties by simple equational specifications and reasoning [19, 40, 53, 55] and their capacity to unify various semantics of computational interest, including formal languages, binary relations, path algebras or execution traces of automata [37]. In [17, 51, 52], Struth showed that classical confluence results such as the Church-Rosser theorem or Newman's lemma can be proved in Kleene algebras augmented with modal operators.

2.3. Higher Kleene algebras. In short, while a formal, algebraic setting for *confluence* theorems had been developed [17, 51, 52], the approach to *coherence* theorems was expressed in the language of strict ω-categories [31]. During my doctorate, myself, Malbos and Struth set out to combine the two lines of research on Kleene-algebraic and higher-dimensional rewriting into a unified framework.

To achieve this, we introduced *higher Kleene algebras* (HKA) [11], generalising both modal Kleene algebras [18] and concurrent Kleene algebras [36]. In order to formulate and prove the coherence theorem in this context, we had to describe the *filling* of confluence diagrams in HKA. We achieved this using the *modal operators* $\Diamond_{(-)}$ inherent to these structures, thereby expressing the for-all/exists relationships between branchings and the associated confluences and higher cells, as pictured on the right. These structures algebraically capture the semantics of higher rewriting, providing a point-free, algebraic approach to coherence which lends itself to formalisation due to the relatively simple algebraic signature. In [11], we introduced this algebraic setting and proved coherent versions of Newman's lemma and the Church-Rosser theorem, *i.e.* purely algebraic proofs of these results using higher dimensional witnesses. We then formulated and proved the coherence theorem for abstract rewriting systems in these higher algebras using a novel characterisation of rewriting strategies [9].



$$\overline{\phi}^* \cdot \phi^* \leq \Diamond_{A}(\phi^* \cdot \overline{\phi}^*)$$

2.4. A correspondence between categories and algebras. HKA were introduced to algebraically describe pasting schemes in higher categories, but an a priori justification of this correspondence was lacking. During the course of the work described above, we noticed that lifting categorical compositions to sets of higher cells resulted in models of HKA, *i.e.* that the powerset of a higher category has the structure of an HKA. In the 1950s, powerset lifting of relational structures was developed into a rich and powerful duality theory by Jónsson and Tarski [38, 39] for the purpose of obtaining concrete representations of abstract Boolean algebras with additional operations (BAOs), prime examples of which are relational and modal algebras. In fact, every modal algebra embeds in a canonical way in a powerset, with the modalities defined by powerset liftings of relations. This is at the heart of the very powerful methods of Kripke semantics in modal logic, see e.g. [4].

Using these ideas, we set out to provide a systematic construction of higher algebras and a justification of their axioms relative to the underlying ω-categories. With Struth, Fahrenberg, Johansen and Ziemański [6], and then later with Malbos, Pous and Struth [12], I studied extensions of the modal correspondences associated with Jónsson-Tarski duality in the special case of ternary relations and binary modal operators. To apply the tools for these constructions, we generalised from ω-categories to ω-*catoids*, which are isomorphic to ternary-relational structures, and specialised from HKA to ω-*quantales*, whose multiplications we consider as binary modalities. Quantales are complete, distributive lattices equipped with a multiplication operation which commutes with suprema, and constitute a special case of complete Kleene algebras.

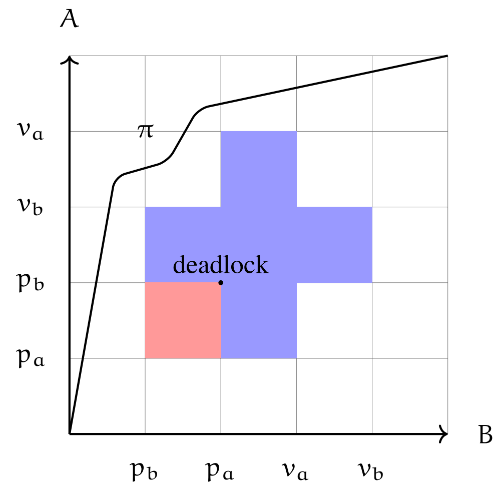
This work began with the one dimensional case, in which we introduced catoids and related them to quantales [6], and was later developed into a series of correspondence theorems between generalised categories, higher quantales and the associated convolution algebras [12].

2.5. Formalisation. Alongside these results, we have formalised HKA, ω-quantales and ω-catoids in Isabelle, an interactive theorem prover based on HOL. An added benefit was that it allowed us to experimentally tweak and reduce the axioms we employed. We added this formalisation of the theory of (higher) quantales and catoids, as well as HKA, to the archive of formal proofs [14, 15].

My role. During my doctorate, under the supervision of Goubault and Malbos, I collaborated closely with Struth on [11], after which I characterised rewriting strategies and thereby proved the coherence theorem in HKA [9]. This was the basis of my contribution to [6] due to the strong links between my work relating Kleene algebras to categories, and the Jónnson-Tarski correspondences studied by my co-authors. After my doctorate, the intuitions gained by studying this correspondence in the context of higher dimensional structures led to my contribution in [12].

3. Topological invariants of directed spaces

3.1. Geometric semantics of concurrency. Researchers have, for almost thirty years now, attempted to understand the semantics of *concurrent programs* via algebraic topology [23]. In concurrent programs, several processes run in parallel, which increases computing power but can also lead to problematic scenarios. Indeed, since the processes share the same memory and resources, conflicts can arise, for example when two processes attempt to use the same resource simultaneously. One way of managing these problems is through the use of *mutexes*, which block the action of certain threads while another performs an action. The use of these may, however, also lead to unfavourable situations such as *deadlock*.



In order to determine the behaviour of such programs, we interpret them as topological spaces, augmented with a notion of *direction* capturing the *temporal order* of executions. Pratt's higher dimensional automata (HDA) [45] provide a natural way to view concurrent programs as *directed topological spaces* [28], given by a pair (X, dX), where X is a topological space and dX is a set of *directed paths*, or *dipaths*. In this semantics, the possible executions, or *schedulings*, of the concurrent program are interpreted as the directed paths in the associated space.

The figure on the right corresponds to a two-agent concurrent system, each of which executes four actions. The command p_a corresponds to "taking" the mutex a, while v_a corresponds to "letting it go". An agent can hold several mutexes at once, but each mutex can only be held by one agent at a time. Paths which increase in each coordinate going from the bottom-left to the top-right of the space correspond to executions: for example, the path π in the figure corresponds to an execution in which A takes the mutex a and then b, and then frees b, after which B takes b, A frees a, thereby allowing B to take a and then release both mutexes. Any execution which enters the red zone is *doomed*, in the sense that it will end with deadlock: A waiting for B to release b while B waits for A to release a. *Forbidden temporal configurations* appear as holes in this space of executions, represented in the above example by the blue cross in the figure, since A cannot take a mutex while B is holding it, and vice versa.

3.2. Directed algebraic topology. The goal of *directed algebraic topology* is to behaviourally distinguish concurrent programs via topological invariants of directed spaces. Determining which dipaths can be *continuously deformed* from one to the other allows us to conclude, in the above example, that there are essentially three types of executions: those in which either A or B takes both mutexes first, or those that end in deadlock. These correspond to the paths which go above or below the hole, or those which enter the red zone, respectively.

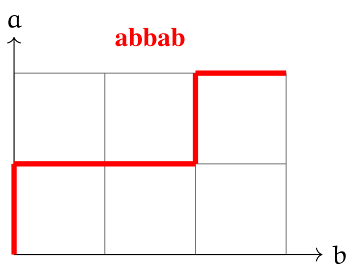
My work in this domain has mainly concerned the refinement and computation of *natural homotopy* and *natural homology*, introduced in [21]. These invariants of directed spaces track the existence of holes in the space of paths between two given points *as the points move throughout the space* along permitted directions. This classifies the behaviour [22] of the concurrent system in question by detecting forbidden temporal configurations, but also encodes how these obstructions are related temporally.

3.3. Refinement and tractability of invariants. During my M2 internship, under the guidance of Eric Goubault, I solved the problem of invariance under *time-reversal* [8], which was remarked by Hess and Fajstrup [24]: the natural homotopy of a directed space was isomorphic to that of its time-reversal, *i.e.* the space in which all directed paths run in the opposite direction. This is not desirable for an invariant of directed spaces, and I was able to correct this by considering *composition pairings*, a notion coming from Porter's study of natural systems augmented with the structure of lax functors [43]. This extra structure tracks the order of events by linking products of homology or homotopy groups to the concatenation of directed paths, thereby detecting time-reversal.

My contributions in this area are in the refinement and enrichment of the theory of natural homotopy and homology. In [8], we showed that natural homology and natural homotopy can be equipped with composition pairings. Using this extra structure, the invariants associated with a space are dual to those associated with its time-reversal. Therein, we also developed a notion of *relative* natural homotopy and homology, obtaining a *long exact sequence* as in the classical case. This construction aids in the calculation of homotopy and homology groups of a given space by reducing to simpler subspaces.

The natural homology of a directed space is in general hard to compute due to the multiplicity of possible extensions of a given execution path. We approached this difficulty by studying links to the efficiently computable invariant known as *persistent homology*, which was originally conceived as a tool for determining topological features of large, higher dimensional data sets [16]. In the final year of my doctorate, Goubault and I established an explicit link between persistent and directed homology theories [10], thereby making the first step toward the tractable computation of homological invariants for directed spaces.

3.4. Binomial lattices. In my first post-doctoral position, I studied the congruences of binomial lattices and their continuous counterparts with Luigi Santocanale, in particular their relation to invariants of directed topology.



Binomial lattices, and more generally, multinomial lattices, were introduced by Bennet and Birkhoff [3] as lattice-theoretic descriptions of commutativity. Indeed, given the set of words on $\{a, b\}$ with n occurrences of a and m occurrences of b, consider the rewrite relation $w \cdot ba \cdot w' \to w \cdot ab \cdot w'$. Its reflexive, transitive closure results in an ordering on these words which has the structure of a lattice. Multinomial lattices are defined similarly, and generalise *permutohedra* [44], *i.e.* the permutations on a finite set equipped with the *weak Bruhat order*.

However, these lattices also have a concrete *geometric* interpretation as lattices of certain directed paths [47], presenting a link to directed topology. Words are interpreted as interleaving sequences of actions, each letter corresponding to an agent. This means that elements of the lattice can be represented as dipaths in a *cubical complex*, which can also be interpreted as *higher dimensional automata* (HDA). These were introduced by Pratt [45] as a semantics for concurrent systems. For example, the HDA corresponding to the binomial lattice for $n = 2$ and $m = 3$ is pictured on the left, along with the dipath corresponding to the word $abbab$. The *congruences* of this lattice present analogies with the notion of *directed homotopy*. Moreover, the associated equivalence relations are of general interest in concurrency, since they correspond to sequences of actions that can be (discretely) deformed from one to the other via a series of commuting actions, *i.e.* actions which are not separated by a hole in the associated HDA.

These lattices also admit *continuous analogs*, as described in [48]. In the two-dimensional case, these are given by *suprema-preserving endomorphisms* of the ordered unit interval I, equipped with the point-wise order, called the *dominance order*, and is denoted by $Q_\vee(I)$. These continuous lattices and their higher dimensional analogs have strong connections with Linear Logic (LL) [57], since functional composition equips them with the structure of *Girard quantales* [48].

3.5. Lattice quotients and directed homotopy. To sum up, strong connections had been conjectured between multinomial lattices, their continuous analogs and their congruences on the one hand, and directed topological spaces and their algebraic invariants on the other. However, the scope of this correspondence had not been formally investigated.

Santocanale and I tackled the two-dimensional case, relating binomial lattice congruences and cubical directed homtopy. This was achieved by introducing certain cubical complexes called *binomial complexes*, which represent the discrete grids the lattice paths live in. We also studied their continuous counterpart $Q_\vee(I)$, relating its congruences to (directed) topological properties of the unit square by using *topological duality techniques* such as Priestley duality [46] and frame duality [50], which relate topological spaces and algebraic structures such as lattices.

Guided by my experience with cubical directed homotopy and Santocanale's expertise in lattice theory, we established a tight correspondence in the discrete, two-dimensional case. Indeed, the directed homotopy groups of the subcomplexes of a binomial complex correspond exactly to the congruences of the associated binomial lattice [13]. This result establishes interesting links between finite lattice theory, combinatorics, and concurrency theory. In the same article, we showed that the dominance order on $Q_\vee(I)$ can be characterised by a point-wise order on the unit square, similar to that given in the discrete case, by constructing *simultaneous parametrisations*. Using topological dualities, we also determined the properties of *spatial* congruences via this ordered topology on the unit square. Finally, we established the impossibility of a complete characterisation via directed homotopy types.

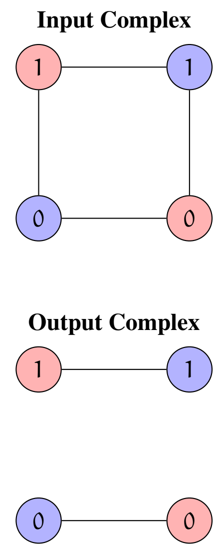
My role. I began my doctorate by establishing the relationship between natural homotopy and composition pairings as suggested my advisors. This led me to solve the problem of time-reversal and further enrich the theory of directed homotopy by introducing the notion of relative natural homotopy, resulting in my first publication, namely [8]. Later, discussions with my advisors led me to study links between persistent homology and natural homology, which led me to establish close links between these theories [10]. The investigation of the congruences of binomial lattices and those of $Q_\vee(I)$ in [13] was a collaboration between Santocanale and I during my first post-doctoral position in the context of the LambdaComb ANR project.

4. Topological approaches to distributed computing

Fault-tolerant distributed computing endeavours to design algorithms in which many different machines, each with a proper local memory, interact via some method of communication in order to solve so-called *decision tasks*, in the presence of crash-failures. It was shown by Fischer, Lynch and Paterson in 1985 [25] that even the primordial *consensus task*, in which each machine starts with an initial value and must *all* end with the same final value, cannot be solved in the presence of a single crash.

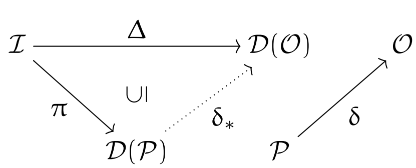
Since then, determining which kinds of tasks can be solved in a given model of communication, *i.e. distributed computability*, has become an important question in the domain of distributed computing. As opposed to the sequential case, distributed computability is not limited by computational power, but by the lack of global information. A given process only knows its local value, and may have the same value in different communication scenarios, meaning that it cannot distinguish between these possible worlds. This epistemic ambiguity creates an obstruction to task solvability. Since the turn of the century, topological structures have been employed to encode these epistemic configurations and have led to groundbreaking computability results, using, for example, connectivity arguments to prove impossibility results, or ideas from topology to determine solvability of a given task [35].

4.1. The role of simplicial topology. While sequential program specifications relate input values to output values, their distributed counterpart, *tasks*, relate sets of input values to permitted sets of output values. Many important communication models are *wait-free*, meaning that any non-faulty process can eventually decide. In these models, any subset of values present in the system also constitutes a valid configuration: sets of values are *down-closed*. Such down-closed collections of sets have been widely studied in the fields of topology and combinatorics under the name *simplicial complexes*. These structures generalise graphs, and can be thought of as spaces, considering singleton sets $\{x\}$ as points, sets $\{x, y\}$ as lines connecting points, sets $\{x, y, z\}$ as triangles spanning three points, etc. Elements of a simplicial complex are called *simplices*.



Simplicial complexes thus encode the information present in the system. More importantly, they capture the *ambiguity* of local knowledge via neighbourhoods: in the input complex above to the right, when the red process has the value one, the lines connecting it to blue states with 0 and 1 encode the fact that the red process cannot distinguish between these two cases. Given a such complexes $\mathcal{I}$ and $\mathcal{O}$ of input and output values, a task is a relation $\mathcal{I} \times \mathcal{O}$ which respects the simplicial structure. By Currying, we can encode such a relation as a monotone map $\Delta : \mathcal{I} \to \mathcal{D}(\mathcal{O})$, where $\mathcal{D}(\mathcal{O})$ is the set of subcomplexes of $\mathcal{O}$, ordered by inclusion. For the consensus task with two processes, the input and output complexes are pictured on the right. The task specifies that if both processes start with 0 or 1, *i.e.* the horizontal edges, then they must decide on that value, whereas if they start with distinct values, the vertical edges, then they can decide either value.

A distributed protocol is interpreted similarly, and can be thought of as a transformation of the input complex, relating it to the so-called *view* or *protocol complex*. Concretely, a protocol $(\pi, \mathcal{P})$ consists of a (protocol) complex $\mathcal{P}$ and a (protocol) map $\pi : \mathcal{I} \to \mathcal{D}(\mathcal{P})$, relating an input configuration to the configurations reachable therefrom under the action of the protocol. The task $(\mathcal{I}, \mathcal{O}, \Delta)$ is *solved* by a protocol $(\pi, \mathcal{P})$ if there exists a simplicial map $\delta : \mathcal{P} \to \mathcal{O}$, whose lifting δ_* to subcomplexes makes the diagram weakly commute, *i.e.* $\delta_*(\pi(\sigma)) \subseteq \Delta(\sigma)$ for all $\sigma \in \mathcal{I}$. This means that the *decision map* δ takes reachable states $\pi(\sigma)$ from some input configuration σ to output configurations which are permitted by the specification, *i.e.* are elements of $\Delta(\sigma)$. The figure below on the left represents this situation diagrammatically.



The challenge is therefore to build the map δ, or disprove its existence, thereby determining solvability. In particular, this depends on the topological structure of the protocol complex. In the *Immediate, Iterated Snapshot* (IIS) model, the protocol complex is a *subdivision* of the initial complex $\mathcal{I}$, denoted by $\mathcal{B}(\mathcal{I})$ [34]. Morally, this means that the topology of the protocol complex is identical to that of the initial complex, the only difference being how much information is encoded. The construction of $\delta_{\mathcal{B}}$ can therefore be related to the existence of a continuous map via simplicial approximation.

4.2. Continuity and computability. A classic example of the connection between topology and computer science relates *computability* to *continuity*. Indeed, in computational analysis, one of the basic results is that every computable function, in the sense of Turing machines, from $\mathbb{R}$ to $\mathbb{R}$ is in fact continuous [56, p. 6]. In other words, for a computable function, a small change in the input will result in a small change in its output. A similar result in the domain of distributed computing, for which Herlihy and Shavit won the Gödel prize in 2004 [34], relates distributive computability to the existence of a continuous function:

Theorem 1 (Asynchronous computability theorem (ACT) [34])**.** *A task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *can be solved by the Iterated Immediate Snapshot protocol, if, and only if, there exists a continuous map* $|\mathcal{I}| \to |\mathcal{O}|$.

where $|\mathcal{C}|$ denotes the *geometric realisation* of a complex $\mathcal{C}$, *i.e.* its interpretation as a subspace of $\mathbb{R}^n$ made up of points, lines, triangles, and their higher-dimensional analogs. In particular, this result shows that the consensus task described above is not solvable by IIS, since the input complex is connected, while the output complex is disconnected.

Note that this topological characterisation applies *only* to IIS, but has been extended other models by establishing *simulations* to IIS. For now, these simulations, *i.e.* ways of reducing one model to another, have only ad-hoc algorithmic justifications, such as the extension to the asynchronous read-write model [34]. My goal in this field is to provide a deeper conceptual understanding of the topological approach to distributed computing.

4.3. Spectral semantics. Working with Godard, we decided to first limit our investigation to *colourless* models and tasks, *i.e.* those in which process identities are forgotten, and only values present in the system are encoded. Such protocols and tasks are nonetheless important in distributed computing, the consensus task being an example thereof.

We remarked that the use of real topologies, such as the geometric realisation, produces *Hausdorff* topological spaces, *i.e.* those in which points can be separated by neighbourhoods. As described above, simplicial complexes are useful in describing distributed knowledge exactly because points *cannot* in general be separated by neighbourhoods. This indicated the need for more general topologies to classify computability for general communication models.

As a first step, we have extended the ACT to a wider range of colourless round-based protocols [7]. We realised that many of the known communication models, including IIS, can be encoded as endofunctors on the category sComp of simplicial complexes. This led to an interpretation of a protocol as a pair (π, Π), where Π is an endofunctor on sComp and π is a natural transformation, giving a protocol map $\pi_{\mathcal{I}} : \mathcal{I} \to \mathcal{D}(\Pi(\mathcal{I}))$ for every input complex $\mathcal{I}$. Using *Stone duality*, these maps can be "turned around", yielding projection maps $\rho_{\mathcal{I}} : \Pi(\mathcal{I}) \to \mathcal{I}$. By naturality, we obtain a map $\rho_{\Pi^n(\mathcal{I})} : \Pi^{n+1}(\mathcal{I}) \to \Pi^n(\mathcal{I})$ for every $n \geq 0$. Together, these define a projective limit system, depicted below, which results in a so-called *spectral space*. This construction defines a functor Π^∞ taking input complexes to spectral spaces.

$$\mathcal{I} \xleftarrow{\rho_0} \Pi(\mathcal{I}) \xleftarrow{\rho_1} \Pi^2(\mathcal{I}) \xleftarrow{\rho_2} \cdots \xleftarrow{\rho_{n-2}} \Pi^{n-1}(\mathcal{I}) \xleftarrow{\rho_{n-1}} \Pi^n(\mathcal{I}) \xleftarrow{\rho_n} \cdots \qquad \Pi^\infty(\mathcal{I}) \to \Pi^k(\mathcal{I}) \text{ for all } k$$

Spectral spaces are ordered topological spaces which can be thought of as infinitary refinements of ordered structures. In this optic, the space $\Pi^\infty(\mathcal{I})$ encodes the transformation of $\mathcal{I}$ under all possible infinite executions of the protocol. This spectral space determines the distributed computability of the associated protocol, as expressed in the following theorem, which is the main result we have thus far achieved, generalising the famous ACT. Indeed, we show that task-solvability is equivalent to the existence of a *spectral map*, a continuous map which is also monotone with respect to the ordering:

Theorem 2 ([7])**.** *A task* $(\mathcal{I}, \mathcal{O}, \Delta)$ *is solved by a protocol* (π, Π) *if, and only if, there exists a spectral map* $f : \Pi^\infty(\mathcal{I}) \to \mathcal{O}$ *compatible with* Δ.

This first result of *spectral semantics* for distributed systems is a major step in the topological approach to distributed computability. We believe that this newly established link to spectral topologies and duality theory has the potential to transform research in the domain. Future directions for this line of research is described in my Research Project.

My role. I began studying distributed computability in my second post-doctoral position, working with Chalopin and Godard in the context of the DUCAT ANR project. Inspired by the approach in [27] linking directed topological semantics of concurrency to distributed computability, we began by studying *interval orders*, which were used as the formal bridge between these two domains. The spectral approach was subsequently developed by Godard and myself, based on the interplay between his intimate knowledge of distributed computing and my experience with categorical and topological methods. This led to our collaboration in establishing the generalised ACT [7].

References

[1] Dimitri Ara, Albert Burroni, Yves Guiraud, Philippe Malbos, François Métayer, and Samuel Mimram. *Polygraphs: From Rewriting to Higher Categories*, volume 495 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2025. URL: https://hal.science/hal-04322821.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1999. URL: https://books.google.fr/books?id=N7BvXVUCQk8C.

[3] M. K. Bennett and Garrett Birkhoff. Two families of Newman lattices. *Algebra Universalis*, 32(1):115–144, 1994. doi:10.1007/BF01190819.

[4] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2002. URL: https://books.google.fr/books?id=-n7uBgAAQBAJ.

[5] Ronald Book and Friedrich Otto. *String-rewriting systems*. Texts and Monographs in Computer Science. Springer-Verlag, 1993.

[6] Cameron Calk, Uli Fahrenberg, Christian Johansen, Georg Struth, and Krzysztof Ziemiański. lr-multisemigroups, modal quantales and the origin of locality. In *Relational and Algebraic Methods in Computer Science*, pages 90–107, Cham, 2021. Springer International Publishing. URL: https://hal.science/hal-05450819, doi:10.1007/978-3-030-88701-8_6.

[7] Cameron Calk and Emmanuel Godard. Stone duality proofs for colorless distributed computability theorems, 2025. arXiv:2511.03609.

[8] Cameron Calk, Eric Goubault, and Philippe Malbos. Time-reversal homotopical properties of concurrent systems. *Homology Homotopy Appl.*, 22(2):31–57, 2020. URL: https://hal.science/hal-02015088v1, doi:10.4310/hha.2020.v22.n2.a2.

[9] Cameron Calk, Eric Goubault, and Philippe Malbos. Abstract strategies and coherence. In *Relational and Algebraic Methods in Computer Science*, pages 108–125, Cham, 2021. Springer International Publishing. URL: https://hal.science/hal-03334734, doi:10.1007/978-3-030-88701-8_7.

[10] Cameron Calk, Eric Goubault, and Philippe Malbos. Persistent homology of partially ordered spaces. *Journal of Applied and Computational Topology*, 8(8):2247–2284, 2024. arXiv:2305.03357, doi:10.1007/s41468-024-00190-9.

[11] Cameron Calk, Eric Goubault, Philippe Malbos, and Georg Struth. Algebraic coherent confluence and higher globular Kleene algebras. *Logical Methods in Computer Science*, 18(4), November 2022. URL: https://hal.science/hal-03161580, doi:10.46298/lmcs-18(4:9)2022.

[12] Cameron Calk, Philippe Malbos, Damien Pous, and Georg Struth. Higher catoids, higher quantales and their correspondences. *Applied Categorical Structures*, 33(4):25, 2025. arXiv:2307.09253, doi:10.1007/s10485-025-09817-z.

[13] Cameron Calk and Luigi Santocanale. Complete congruences of completely distributive lattices. In Uli Fahrenberg, Wesley Fussner, and Roland Glück, editors, *Relational and Algebraic Methods in Computer Science*, pages 101–118, Cham, 2024. Springer Nature Switzerland. URL: https://hal.science/hal-05450869, doi:10.1007/978-3-031-68279-7_7.

[14] Cameron Calk and Georg Struth. Higher Globular Catoids and Quantales. Archive of Formal Proofs. URL: https://www.isa-afp.org/entries/OmegaCatoidsQuantales.html.

[15] Cameron Calk and Georg Struth. Modal quantales, involutive quantales, Dedekind Quantales. Archive of Formal Proofs. URL: https://www.isa-afp.org/entries/Quantales_Converse.html.

[16] Gunnar Carlsson. Topology and data. *Bull. Amer. Math. Soc. (N.S.)*, 46(2):255–308, 2009. doi:10.1090/S0273-0979-09-01249-X.

[17] Jules Desharnais, Bernhard Möller, and Georg Struth. Termination in modal Kleene algebra. In *IFIP Advances in Information and Communication Technology*, volume 155, pages 647–660, 01 2004. doi:10.1007/1-4020-8141-3_49.

[18] Jules Desharnais and Georg Struth. Internal axioms for domain semirings. *Sci. Comput. Programming*, 76(3):181–203, 2011. doi:10.1016/j.scico.2010.05.007.

[19] Henk Doornbos, Roland Carl Backhouse, and Jaap van der Woude. A calculational approach to mathematical induction. *Theor. Comput. Sci.*, 179(1-2):103–135, 1997.

[20] Jérémy Dubut. *Directed homotopy and homology theories for geometric models of true concurrency*. PhD thesis, 2017.

[21] Jérémy Dubut, Eric Goubault, and Jean Goubault-Larrecq. Natural homology. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, pages 171–183, 2015.

[22] Jérémy Dubut, Éric Goubault, and Jean Goubault-Larrecq. Directed homology theories and Eilenberg-Steenrod axioms. *Applied Categorical Structures*, pages 1–33, 2016. doi:10.1007/s10485-016-9438-y.

[23] L. Fajstrup, E. Goubault, E. Haucourt, S. Mimram, and M. Raussen. *Directed Algebraic Topology and Concurrency*. Springer, 2016.

[24] Lisbeth Fajstrup and Kathryn Hess. Time reversibility of natural homology. Private communication, July 2018.

[25] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, apr 1985. doi:10.1145/3149.214121.

[26] Stéphane Gaussent, Yves Guiraud, and Philippe Malbos. Coherent presentations of Artin monoids. *Compos. Math.*, 151(5):957–998, 2015. doi:10.1112/S0010437X14007842.

[27] Éric Goubault, Samuel Mimram, and Christine Tasson. Geometric and combinatorial views on asynchronous computability. *Distributed Computing*, 31(4):289–316, 2018. doi:10.1007/s00446-018-0328-4.

[28] Marco Grandis. *Directed Algebraic Topology, Models of non-reversible worlds*. Cambridge University Press, 2009.

[29] Yves Guiraud and Philippe Malbos. Higher-dimensional categories with finite derivation type. *Theory Appl. Categ.*, 22:No. 18, 420–478, 2009.

[30] Yves Guiraud and Philippe Malbos. Coherence in monoidal track categories. *Math. Structures Comput. Sci.*, 22(6):931–969, 2012. doi:10.1017/S096012951100065X.

[31] Yves Guiraud and Philippe Malbos. Polygraphs of finite derivation type. *Math. Structures Comput. Sci.*, 28(2):155–201, 2018. doi:10.1017/S0960129516000220.

[32] Nohra Hage and Philippe Malbos. Knuth's coherent presentations of plactic monoids of type A. *Algebr. Represent. Theory*, 20(5):1259–1288, 2017. doi:10.1007/s10468-017-9686-z.

[33] Nohra Hage and Philippe Malbos. Chinese syzygies by insertions. *Semigroup Forum*, 104(1):88–108, 2022. doi:10.1007/s00233-021-10244-4.

[34] M. Herlihy and N. Shavit. The topological structure of asynchronous computability. *Journal of the ACM (JACM)*, 46(6):858–923, 1999. doi:10.1145/331524.331529.

[35] Maurice Herlihy, Dmitry Feichtner-Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, Amsterdam, 2014. URL: http://www.sciencedirect.com/science/book/9780124045781.

[36] Tony Hoare, Bernhard Möller, Georg Struth, and Ian Wehrman. Concurrent Kleene algebra and its foundations. *J. Log. Algebr. Program.*, 80(6):266–296, 2011. doi:10.1016/j.jlap.2011.04.005.

[37] Peter Höfner and Georg Struth. Algebraic notions of nontermination: omega and divergence in idempotent semirings. *J. Log. Algebr. Program.*, 79(8):794–811, 2010. doi:10.1016/j.jlap.2010.07.016.

[38] Bjarni Jonnson and Alfred Tarski. Boolean algebras with operators. part II. *American Journal of Mathematics*, 74(1):127–162, 1952. URL: http://www.jstor.org/stable/2372074.

[39] Bjarni Jonsson and Alfred Tarski. Boolean algebras with operators. part I. *American Journal of Mathematics*, 73(4):891–939, 1951. URL: http://www.jstor.org/stable/2372123.

[40] Dexter Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, 1997.

[41] Nicolai Kraus and Jakob von Raumer. A rewriting coherence theorem with applications in homotopy type theory. *Mathematical Structures in Computer Science*, 32(7):982–1014, 2023. doi:10.1017/S0960129523000026.

[42] Saunders Mac Lane. *Categories for the working mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 1998.

[43] Timothy Porter. Group objects in $\mathsf{Cat}_{\Sigma_0}/\mathsf{B}$, 2012. Preprint.

[44] Alexander Postnikov. Permutohedra, associahedra, and beyond. *International Mathematics Research Notices*, 2009, 08 2005. doi:10.1093/imrn/rnn153.

[45] Vaughn Pratt. Modeling concurrency with geometry. In *Proceedings of the 18th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '91, pages 311–322. ACM, 1991. doi:10.1145/99583.99625.

[46] H. A. Priestley. Representation of distributive lattices by means of ordered stone spaces. *Bulletin of the London Mathematical Society*, 2(2):186–190, 1970. doi:10.1112/blms/2.2.186.

[47] Luigi Santocanale. On the join dependency relation in multinomial lattices. *Order*, 24(3):155–179, August 2007.

[48] Luigi Santocanale and Maria João Gouveia. The continuous weak order. *Journal of Pure and Applied Algebra*, 225:106472, 2021. URL: https://hal.archives-ouvertes.fr/hal-01944759.

[49] Craig C. Squier, Friedrich Otto, and Yuji Kobayashi. A finiteness condition for rewriting systems. *Theoret. Comput. Sci.*, 131(2):271–294, 1994. doi:10.1016/0304-3975(94)90175-9.

[50] Marshall Harvey Stone. Topological representations of distributive lattices and brouwerian logics. 1938. URL: https://api.semanticscholar.org/CorpusID:55241242.

[51] Georg Struth. Calculating Church-Rosser proofs in Kleene algebra. In *Relational methods in computer science*, volume 2561 of *Lecture Notes in Comput. Sci.*, pages 276–290. Springer, Berlin, 2002. doi:10.1007/3-540-36280-0_19.

[52] Georg Struth. Abstract abstract reduction. *J. Log. Algebr. Program.*, 66(2):239–270, 2006. doi:10.1016/j.jlap.2005.04.001.

[53] Georg Struth. Modal tools for separation and refinement. *Electron. Notes Theor. Comput. Sci.*, 214:81–101, 2008.

[54] Axel Thue. Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. *Kristiania Vidensk. Selsk, Skr.*, 10:493–524, 1914.

[55] Joakim von Wright. Towards a refinement algebra. *Sci. Comput. Program.*, 51(1-2):23–45, 2004.

[56] K. Weihrauch. *Computable Analysis: An Introduction*. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2000. URL: https://books.google.de/books?id=OPolVWVFDJYC.

[57] David N. Yetter. Quantales and (noncommutative) linear logic. *The Journal of Symbolic Logic*, 55(1):41–64, 1990. URL: http://www.jstor.org/stable/2274953.