

INFORMATION FLOW IN SPATIAL MODELS OF COMPUTATION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2016

By
Joseph A. B. Razavi
School of Computer Science

Contents

Abstract	5
Declaration	6
Copyright	7
Acknowledgements	8
1 Introduction	9
1.1 Motivation	9
1.1.1 A Model of Computation	13
1.1.2 A Notion of Space	14
1.1.3 A Definition of Information	16
1.2 Outline of this Thesis	19
2 Background	21
2.1 Category Theory	21
2.1.1 Presheaves, Sites, and Separation	21
2.1.2 The Category of Elements of a Presheaf	25
2.2 Neighbourhood Spaces	27
2.2.1 Open Sets and Topological Spaces	31
2.2.2 Cospecialization and Alexandroff Spaces	33
2.2.3 Coarse and Fine Structures, Limits and Colimits	35
2.3 Paths, Homotopies, and Directed Spaces	38
2.3.1 Directed Spaces	39
2.4 Simplices and Simplicial Complexes	42
2.4.1 Motivation	42
2.4.2 The Structure of Simplices	42

2.4.3	Simplicial Complexes and Subcomplexes	48
2.4.4	The Underlying Space of a Complex	49
2.4.5	Subdivisions	52
2.4.6	Full Subcomplexes and Extending Maps	54
3	Paths and Homotopies in Alexandroff Spaces	57
3.1	Introduction	57
3.2	Tools from Simplicial Complexes	60
3.2.1	Alexandroff Spaces	63
3.3	Constancy, Accommodation, and Relaxation	66
3.4	Combinatorial Approximation of a Map	68
3.4.1	Future Work	73
4	Information Flow in a Model of Computation	76
4.1	Introduction	76
4.1.1	Standard Formalizations of Cellular Automata	77
4.2	Cellular Automata on a Category	82
4.2.1	Alphabets, Encodings, and Automata	88
4.3	Local Information	90
4.3.1	Properties which Characterize Ken	100
4.3.2	Continuity and Computation	105
4.3.3	Towards a Spatial View of Computations	115
5	Conclusions and Future Work	117
5.1	Conclusions	117
5.2	Future Work	118
5.2.1	First Investigations of Homotopy in the Space	118
5.2.2	A Better Spatial Description	119
5.2.3	A Better Notion of Information	120
	Bibliography	121
	Word Count: 32525	

List of Figures

- 4.1 The von Neumann (left) and Moore (right) neighbourhoods of a cell. . 78
- 4.2 Depiction of the morphisms from a two-by-two square to a three-by-three square.. 83
- 4.3 A cover of the three by three square. 84
- 4.4 The action of erosion on a morphism. 85
- 4.5 Interior of a four-by-four region. 87

Abstract

INFORMATION FLOW IN SPATIAL MODELS OF COMPUTATION

Joseph A. B. Razavi

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2016

Some models of computation have a notion of underlying space. In this thesis, we study the way in which information flows over this space in the course of computation, with the aim of discovering some geometric structure in the set of possible computations. We select cellular automata as a particular example to study. We define an abstract notion of information flow and characterize those which come from cellular automata. A slight generalization of the properties involved in the characterization includes what can be interpreted as a continuity condition on the flow of information. This condition can be thought of as giving an Alexandroff neighbourhood space (a mild generalization of a topological space) whose points are distributions of information. Motivated by this we study aspects of the structure of Alexandroff neighbourhood spaces. We show that any map from a simplicial complex into an Alexandroff space is homotopic to one of a simple combinatorial form.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

- i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s policy on presentation of Theses

Acknowledgements

I would like to thank my supervisor Andrea Schalk; it would have been unreasonable to have asked for a supervisor half as good! To supervise this kind of open-ended work requires a heroic amount of effort, energy, and skill, and Andrea always supplies much more than what is required.

I would like to thank Harold Simmons for his extensive mathematical help and advice, which included preparing notes and talks on some of the areas needed for this research, and the discussion of certain proofs.

I would like to thank Francisco Lobo for his considerable help with mathematics, particularly as concerns matters of category theory, for spending lots of time preparing talks for my benefit, and for always being happy to be interrupted with questions of typesetting!

I would like to thank Nige Ray for meeting me to discuss simplicial complexes; these discussions were very useful for the work presented here.

I would like to thank Jonas Lorenz for his ability to refute my conjectures while they were still half-formulated, which saved months of effort, Ian Pratt-Hartmann for helpful discussions on topological matters, and Sam Dean for a very helpful discussion on presheaves.

Finally I would like to thank my parents and my sister Natalie, as well as Jim, Sylvia, and Annette for their support.

Chapter 1

Introduction

1.1 Motivation

In the study of computation, we are usually concerned with one of two things: there are computational problems, which describe *what* we want to compute, and programs, which describe *how* we intend to carry out the computation. Computational problems often come with some kind of high level structure. We may want to compute functions, for example, which can be composed and evaluated; in this thesis, we restrict ourselves to decision problems (with yes or no answers) which can be compared and combined using all the usual tools of logic. Computational means—that is, programs—seem harder to think of in abstract terms. They are usually, ultimately, transition systems given by combinatorial data. This is as it should be: we want a detailed model of the notion of combinatorial procedure so that we are sure to have captured the idea of effective computability.

But this leaves us wondering whether there is a level of abstraction at which to study computational processes which would reveal an underlying mathematical structure *in the process of computation itself*, rather than in the ends to which computation is put.

One might think that we have such an abstraction in the notion of an algorithm, but algorithms are either given at an intuitive level, or else they are specified by some sort of transition system. Indeed, [BDG09] attempt to show that this is necessary. Within the context of a very general model of computation, they ask whether there is an equivalence relation on programs which captures all of the ways in which two

programs might be said to follow the same algorithm. They then exhibit a sequence of programs such that, on their view, each consecutive pair ought to be related, but whose extremes should be distinct. Someone who did not object to their stipulations (or the formalization thereof) would conclude that there are no fixed entities called algorithms, so that to say that two programs implement the same algorithm is to communicate a subtly context-dependent idea. This would make the computer scientist's talk of algorithms a rather sophisticated affair. We would use that language to exchange ideas about proofs and programs, without ever needing to demand a concrete example; the currency of our science would be paper money.

A better way to proceed might be to look for abstractions of computation in an unbiased way. Hines' programme of "Machine Semantics" [Hin08] pursues this goal, by examining a lattice of deterministic machines which arise from a given machine by abstraction.

An alternative approach is to look for models of computation which are defined on the basis of clear conceptual ideas, and ask whether these leave a trace of abstract structure amid the combinatorial thitherings of their programs. In this thesis, we examine a model which stores information within some kind of space, in the hope of drawing out geometric structure.

Perhaps Turing machines were the first spatial model of computation. In [Tur36], the famous paper which introduced them, Turing justifies the model with respect to our everyday experience of carrying out calculations on paper. This involves spatial ideas because of the bounded area that the computer can inspect at one "glance" and because of the movement from square to square. The importance of spatial thinking is underscored when Turing offers a topological argument that any infinite collection of symbols, thought of as subsets of the unit square, contains symbols which are arbitrarily close.

An significant step is taken by Gandy in [Gan80]¹ in which he attempts to give a model of computation based on the restricted powers of physical machines, rather than human experiences. This involves two key principles:

¹The present author first encountered Gandy's ideas in the work of Sieg [Sieg08], whose perspective surely colours the present discussion in subtle ways.

- **Boundedness.** A finite region of space can have a finite number of physical configurations.
- **Local causation.** Every region has a causal neighbourhood such that its next state is determined by that of the previous state of its causal neighbourhood.

It is the idea of neighbourhoods here which gives this model an essentially spatial character. It is used to model the insight that in nature “changes” or “signals” propagate with a bounded velocity. The idea is that a change which starts far away from a region must reach it by passing into its causal neighbourhood. One wonders whether the system of causal neighbourhoods constitutes some sort of topological structure on our space; then the description of the motion of changes—whatever changes are—seems to be saying that they propagate *continuously*. Can we imagine that in our space is extended a body of changes, whose continuous motions constitute computation? Do the possible forms it can take, like those of our own body, arrange themselves into an abstract space which describes the kind of freedom it has to change configuration? Later writers discussing Gandy’s work, especially those who are interested in the physical significance of Gandy’s postulates,² have a habit of referring to “information” rather than “changes” or “signals”; we might re-phrase the principles above as follows.

- **Boundedness.** A finite region of space can only hold a finite amount of information.
- **Local causation.** Information passing into a region must first pass into its causal neighbourhood.

This is a fruitful change of perspective, because it suggests a more concrete object of study than changes or signals. It would be difficult to define a notion of a change in such a way that we can track the motion of a single change through space. Perhaps it could be done, just as we can track a wave through a medium, even though the atoms which make up the wave constantly change, but it is not obvious where to start. On the other hand, “information” suggests a connection to logic: although a piece of news may spread through different media, we can think of it as the same piece of news if the conclusions we would draw from receiving one of these transmissions are the same as from another. We are motivated to ask the following.

²For example, [DGR15, AD12, CS07, Fre13]

Question. Can one analyse the way information flows in a model of computation with an underlying base space, so as to arrive at an abstract space of information assignments whose paths encode computations?

This thesis can be seen as an attempt at answering this question for a particular model of computation. In Chapter 4, we define the model we intend to study, and our notion of information, and characterize those flows of information which can arise in the model. A slight relaxation of one of the conditions can be interpreted as a “continuity” condition, saying that what a region has come to know now, its neighbourhood must have known already. We then just barely sketch how this condition can be turned into a space, using the results of Chapter 3.

Viewing computations as paths through some space will, it is hoped, endow them with some sort of geometric structure. Algebraic topology³ is increasingly making in-roads into computer science. For example [HKR13] studies the question whether certain problems of distributed computing are solvable by expressing the problem as a complex whose homotopy type encodes the solution. In concurrency theory, the ideas of [FRG06] make great progress by using homotopy to ignore inessential details of concurrent computations. In such a computation, for example, there may be lots of logically independent activities which can happen in any order, and one can model the runs of the model which order these differently as different but homotopic paths. This requires a notion of *directed space* because computations have a sense of direction. For the same reason, we also need directed spaces in this thesis; we follow [Gra09], but what little we need for our present work is defined in Section 2.3.1.

The idea of replacing syntactic objects with more flexible analogues in order to study them also occurs, at least conceptually, in proof theory. It might be thought that proof nets in linear logic stand in this sort of relation to proofs (see the suggestive figures in [Str06]). More directly, the recent [DV16] models proofs in linear logic by two dimensional surfaces, such that suitable deformations of these do not change their logical content.⁴

³We give a brief definition of the notion of homotopy in Chapter 2; it is Definition 32; for a more complete introduction to algebraic topology, the reader is directed to [Spa81].

⁴Perhaps one could even use Homotopy Type Theory [Uni13] in order to study computations. There one certainly takes a spatial view of many things, which will include computations as a particular case. Indeed, if a particular computation is seen as a proof of an equality (perhaps of the form $f(x) = y$ where f is the function to be computed on input x), then computations correspond to paths. This is a very

A different, more philosophical motivation for viewing computations as movements of information is that it might help us to understand what is gained by computation, when the results are always logically implied by the inputs. Abramsky [Abr13] suggests this as one of two puzzles about computation. One wonders whether the way information moves in a spatial model of computation might help to clarify this. Imagine carrying out the following task: given a map with two marked cities, one must compute the shortest path between them. This might be done by progressively ruling out regions of the map through which this shortest path cannot pass. One could encode this information by crossing out such regions.⁵ This is interesting from the information point of view because if we suppose that someone is watching through the keyhole, they might eventually find out whether or not the shortest path goes through the bit of map they can see, because it will either get crossed out or not. At the end of computation, the once global question about where the shortest path goes has been completely localized: a “single glance”—to mimic the words of Turing—at any region of the map will reveal whether the path passes through it.⁶

To proceed to answer the question set above, we need to settle on a model of computation, a notion of space, and a definition of information. It is to these issues we now turn.

1.1.1 A Model of Computation

To begin an analysis of the flow of information in a model of computation, we need to choose a model to analyse. Although we are inspired by Gandy’s analysis of computation, Gandy’s model is too general for our needs. This is because we want an explicit notion of underlying space which is fixed throughout computation. One of Gandy’s key aims was *not* to explicitly separate out a spatial part of the model, and a crucial aspect of his work is that the underlying shape of a state of computation may change. This feature accounts for much of the complexity of the model.

different way of endowing computations with geometric structure to the one undertaken here.

⁵The thought experiment of the map colouring algorithm also occurs in the introduction to my MSc thesis [Raz12], which reports the results of a short research project on an idea which is conceptually related to the one presented here, but which is technically much simpler.

⁶See also [HMV94] which enquires about the knowledge which agents in a distributed system actually have available to act on, which inspired the present view.

Instead, we opt for a model with a fixed underlying space, but we continue to use discrete time and space. This essentially amounts to working with a version of the concept of cellular automaton, introduced by von Neumann [Neu66]. The usual ways of axiomatizing the notion of cellular automaton, however, are somewhat cumbersome for the present analysis; in Chapter 4, having briefly surveyed the usual formulations, we supply a convenient definition. Our definition is categorical in nature, making use of presheaves on a site whose objects we think of as regions of space. It is interesting that at the end of [Gan80], Gandy comments

The heavy use made of restrictions, and the complications involved in fitting them together [...] suggest that a treatment using concepts analogous to those of sheaf theory or topos theory might be worth developing.

The model presented here, of course, cannot be what Gandy had in mind, because of its clear separation of a fixed underlying space from the computation happening over it. Sieg, quoting the same passage of Gandy in [Sie02], states that the work of Herron [Her95] is a step towards a categorical model following Gandy’s ideas. The present author has been unable to locate a copy of Herron’s work, although it seems that, like other models proposed following Gandy’s lead (for example [AMW14, Obt14]), it is likely to be focussed precisely on the ability of the shape of space to change.

Before moving on, it is worth mentioning [Irw11] which also describes a model of computation which generalizes cellular automata, and in which computations are thought of as continuous paths in some sense. The focus of that work is very different to this thesis: it concerns an abstract notion of derivative, so that computations can be compared to differential equations. To model computations, one selects the allowed “derivatives” to be the step by step rules which constitute the program. Then any map which is locally one of these rules is a computation.

1.1.2 A Notion of Space

In order to give meaning to the question above, one needs a notion of “continuous path”. Although we have chosen a model in which time is discrete, we nevertheless choose to model paths as functions whose source is the real interval. This is partly because of the intuitive appeal of this definition of path. Another motivation is the hope that the present type of analysis might be extended to continuous systems in future. Finally, the interval has abstract properties which means that modelling paths as maps

out of the interval gives a well behaved notion.⁷

This creates a certain tension: we need a notion of space which encompasses the continuous world of the real interval, but also the discrete structures corresponding to computations. This problem is familiar to computer science in the realm of image processing. There one might want to study, for example, the homotopy type of a digital image, but it is not clear how to define this because a digital image is a discrete object. The study of discrete geometric objects by analogy with topology is called *digital topology*; it begins with [Ros79]. An interesting example⁸ is [Web97] which studies a category of objects with both a topological and graph theoretic structure. In this category there is a single object such that a map from this object into a graph is a path in the graph of arbitrary length.

Other work has sought to take the name “digital topology” more literally, and model digital objects by using topological spaces. One important instance of this, from the point of view of this thesis, is [MPCR11], which models paths through Alexandroff topological spaces, which correspond to preorders in a certain sense. Although general paths may be very wild, they show that every path is homotopic to one which can be understood as a finite sequence of steps through the preorder.

We need a notion of space somewhat more general than topological spaces. Indeed Smyth in [Smy95] suggests using *closure spaces* for digital topology. These are equivalent to the neighbourhood spaces which we describe in section 2.2. The paper [Gal00] shows how several structures proposed for the purposes of digital topology can be viewed as closure spaces.

Closure spaces, which we view here equivalently as neighbourhood spaces, constitute the right kind of space for the present work. In Chapter 3, an analogue of a result of [MPCR11] is proved, showing that maps from arbitrary simplicial complexes into Alexandroff neighbourhood spaces (which correspond to reflexive relations) are homotopic to well-behaved maps. This implies that paths are homotopic to finite sequences of steps in the relation underlying the Alexandroff space.

⁷For an important example, see [ES01] which shows that among topological spaces and sets, the real interval is the initial structure with a certain well-behaved notion of midpoint operation.

⁸In fact, the motivations and scope of [Web97] go beyond digital topology, but it is an important potential application.

At first many different kinds of space were considered. A notable example is the notion of “tetric space” [CLL⁺05], a generalization of metric spaces to allow values in an arbitrary monoid. It seemed for most of the duration of the present work that these spaces were not important for the project undertaken here. However the presence of monoid actions on spatial regions in Chapter 4 indicates that there might be more of a connection than first supposed.

1.1.3 A Definition of Information

Giving a definition of information suitable for our purposes is one of the main aims the present work. It is undertaken in Chapter 4. Here we briefly describe the conceptual underpinnings of the idea.

Recall that the reason we prefer “information” to “changes” or “signals” moving in space is that it seems that the same information may be propagated at different times and in different places by completely different signs. For example, suppose a little fish sees a terrible shark to its north. At first, the information “there is a shark to the north” is encoded in the shark itself. On seeing the shark, however, the fish might begin to swim frantically south, into the view of another fish. Now, if we suppose that the only cause of a frantic fish swimming south is a shark to the north, our second fish could learn that fact by means of this mediating sign. This might be useful to the fish if they do not swim very quickly. Though the first fish might be caught, the information about the location of the shark will spread through the shoal like the light of beacons, advancing so much faster than the shark that the fish as a whole seem to take prescient evasive manoeuvres.⁹

A similar view occurs in the Computer Science literature in a well known paper of Halpern and Fagin [HF89]. Their goal was to make sense of “knowledge based protocols” in distributed systems. For example, we might say “if agent X knows that agent Y has received the first part of the transmission, it begins to send the second part”. They argue that the knowledge of an agent in a distributed system could be defined as the equivalence class of global states of the system in which the agent’s local state has

⁹This example is inspired by the study of flocking birds. For example [BCG⁺14] studies the flow of information—in the statistician’s sense—in a flock of birds, using tools from thermodynamics; [FJK11] studies information flow in a flock empirically using video recordings, manipulating flock behaviour using robotic birds.

the observed value. The mathematical model is in terms of modal logic. It would be interesting to compare the model in this thesis to theirs, by defining a modal logic on the basis of our notion of information. It seems that the specific focus of the present work might provide interesting notions to study from this point of view, such as the idea that one agent could be considered a neighbourhood of another. This would require the present programme to be more advanced, however, since various strange features of our present model preclude a straightforward logical interpretation. The same can be said for the relation of this work to the work of Porter, who in [Por04, Por03] studies connections between categorical and geometric ideas and modal logic in the setting of multi-agent systems. Another model of information flow along logical lines, with a strikingly similar conceptual basis for its definition of information, is that of Barwise [BS97]. Indeed, compare the following description [BS97, p. 4] of information in everyday life with the analogy of the fish above.

Once one reflects on the idea of information flowing, it can be seen to flow everywhere [...] Clouds carry information about forthcoming storms; a scent on the breeze carries information to the predator about the location of prey; the rings of a tree carry information about its age; a line outside the gas station carries information about the national budget [...]

One can perhaps trace this type of thinking back to the nascent years of modern logic, to Lady Welby's programme of "Significs", intended as a science of meaning in the broadest sense. In [Wel96, p. 25], she writes

In one sense, the first thing which the living organism has to do,—beginning even with the plant—is to interpret an excitation and thus to discriminate between the appeals e.g. of food and danger. [...] From this point of view, therefore, the problem which every root as well as the tentacle and even the protozoic surface may be said to solve is that of 'meaning', which thus applies in unbroken gradation and in ever-rising scale of value, from the lowest moment of life to the highest moment of mind.

In this thesis, we pursue a "geometric" view of this idea, too. If we imagine the history of a possible world laid out before us, so that we can see every moment of time, then this history would look like a labelling of a higher-dimensional space by the physical state which obtained at each moment. We could do this for every world. We think of a region as knowing that it is one of the regions in this enormous space which

looks the same as it. We can think of the region as a small labelled space, and then this collection of possible places it could be is just the set of morphisms which embed it in the bigger space and preserve the labelling by states.

If our world were flat, and its sea a disc, then a history of the ocean might be viewed as a watery cylinder, populated by braided threads for the lives of fish and sharks, stretching from the beginning of time to its end. If we imagine that to every dawn of time, with its initial endowment of fishes and sharks, there corresponds such a cylinder which proceeds from it, then one imagines an enormous space, a disjoint union of very many ocean cylinders. It is to this picture that we imagine our little fish turns when it meets its frantic friend, terror in those familiar lineaments. It asks itself where in the history of all worlds it could be, given that such a sight is before it. Reflection reveals that wherever a situation like this can be found in history, there is a hungry shark to the north, and the fish joins in, therefore, with fleeing.

This language puts us in mind of the “Sheaf Semantics” of Goguen [Gog92]. His descriptions of observing the behaviour of a system at different times, and in different regions of space, are certainly similar to the view above, and to the technical definition of the history of a cellular automaton in Chapter 4. Indeed one may be able to define this object as an instance of Goguen’s idea, perhaps slightly technically generalized.¹⁰ However, the focus of Goguen’s work is to specify these systems of observations in a conceptual manner by using ideas from category theory to compose complex behaviours out of simple ones. Our focus is to analyse the information flow in specific such systems which correspond to computations over a space. As Goguen himself says “one person’s syntax is another person’s semantics”. Goguen goes on to study “information flow” in such systems, but in the sense of non-interference of parts which is important for the security of computer systems which store sensitive information. In this type of work (see for example [Den76, Smi07]), one wants to make sure that changes in “high security” program variables do not cause changes in “low security” ones.

Finally, a more prosaic motivation for the definition of information we give here is communication complexity [Yao79, KN97], in which one wants to put bounds on the

¹⁰We consider separated presheaves, rather than sheaves, and over slightly more general structures, although the basic examples in each case should find equivalents in the other.

amount of communication which agents in a distributed system must engage in to solve the problem they have been set. This acts as a heuristic motivation, in the sense that one wants an abstraction from computational processes which is still concrete enough that such analyses could be envisaged. An interesting study of the computational power of cellular automata by means of communication complexity is [GMRT11].

1.2 Outline of this Thesis

This thesis is an attempt to answer the question posed in the previous section for a specific model of computation. Chapter 3 deals with the tension between continuous paths and discrete computations, and Chapter 4 studies the way information flows in a model of computation. Chapter 4 ends with a brief sketch of how these two aspects might be linked to produce a space whose paths correspond to computations.

Chapter 2 provides background material for the other chapters. Section 2.1 concerns coverages and presheaves, and supports Chapter 4. The reader familiar with this material may still like to look at Proposition 3, which is a rephrasing of the separation condition in terms of the behaviour of hom-sets in the category of elements of the presheaf. This underlies Proposition 90 in Chapter 4, showing that an interesting feature of our notion of information follows purely from considerations about presheaves. In section 2.2, neighbourhood spaces are introduced and some basic propositions about them are stated. This material underlies Chapter 3, which studies paths and homotopies in Alexandroff neighbourhood spaces. These are shown, in Theorem 84 and Proposition 85, to be homotopic to ones of a simple combinatorial form. The main proof in Chapter 3 requires the machinery of simplicial complexes, which is covered by section 2.4. This is, for the most part, standard material from piecewise linear topology, although the reader may like to ensure that they believe Proposition 64 which we prove using a slight variation on the usual material. Chapter 2 concludes with section 2.3, which provides the material on paths necessary to understand Chapter 3. For the notion of directed paths, we follow Grandis [Gra09], though what we need for this thesis is described in Section 2.3.1.

Chapter 3 builds up to a proof that continuous maps from a simplicial complex to an Alexandroff space can be well approximated by combinatorial data. Theorem 84

generalizes [MPCR11, Theorem 2] (briefly described in the Introduction to that chapter) from the one-dimensional topological case to the case of neighbourhood spaces in all dimensions; Proposition 85 covers the directed, one-dimensional case.

Chapter 4 undertakes the main work of the thesis, studying the way information flows in cellular automata. Section 4.1.1 surveys the standard definitions of cellular automata, and section 4.2 gives our own definition in terms of separated presheaves. Our notion of information is defined in section 4.3. The properties of information flows which characterize those coming from cellular automata are given in section 4.3.1, and are then generalized slightly in section 4.3.2 to a notion of a continuous information flow. Propositions 99 and 100 then prove that there is computational content in this slight abstraction, by showing that a cellular automaton can be produced whose flow of information is strictly more informative than any given continuous information flow. Section 4.3.3 sketches a notion of space derived from the main work of Chapter 4, such that paths through the space correspond to computations. This requires Proposition 85, and thus begins to show how the ideas of Chapters 3 and 4 are related. The thesis concludes with some ideas for future work.

Chapter 2

Background

2.1 Category Theory

We assume the reader is familiar with the basics of category theory, up to, say, adjunctions. In this section we supply the additional background required to read this thesis. We give a brief introduction and some intuition here; for a full account, the reader is directed to [MM94] and [Joh02]. To appreciate a few remarks, a little more than what is given here may be necessary, but the reader may safely ignore them.

2.1.1 Presheaves, Sites, and Separation

Fix a category C , and think of its objects as “spaces” in some abstract sense. We are interested in describing the notion of “geometric data”, by which we mean a kind of data where a value is thought of as “varying over a space”. More concretely, if for some object R of C , one has a value v varying over R and a morphism $f : S \rightarrow R$, one expects to be able to “pull v back along f ” to obtain a piece of data $v \upharpoonright f$, the *restriction*¹ of v along f , varying over S . One also expects these restrictions to behave coherently, in the sense that restricting along the identity does nothing, and restricting along one map, then along another is the same as restricting along the composite. This amounts to saying that a particular type of geometric data corresponds to a functor $C^{op} \rightarrow \mathbf{Set}$ – that is, a *presheaf* on C . We think of restriction as the fundamental operation we want to perform on data: just as we might want to compute the successor of a natural number, or the left and right components of a pair, or evaluate a function at a point,

¹The term pullback would have been more enlightening in the general case, however, in many particular cases of interest, a data value will be a function of some sort, and restriction will be just that.

what we really want to do to a piece of geometric data is restrict it along a morphism.

Before we go on with the geometric discussion, we pause to note that there is a natural way to compare functors $F, G : D \rightarrow \mathbf{Set}$ (where, above, we have $D = C^{op}$) which is special to \mathbf{Set} . We say that F is a **subfunctor** of G when there is a natural transformation from F to G whose components are all inclusions. That means that for all objects R of D , we have $F(R) \subseteq G(R)$, and for all morphisms $f : R \rightarrow S$ of D , the diagram

$$\begin{array}{ccc} F(R) & \xrightarrow{F(f)} & F(S) \\ | \cap & & | \cap \\ G(R) & \xrightarrow{G(f)} & G(S) \end{array}$$

commutes, so $F(f)$ is the restriction and co-restriction of $G(f)$ to $F(R)$ and $F(S)$ respectively.

In the prototypical examples of presheaves, the base category C comes from topology, and has some extra geometric structure. The most exciting thing to do in topology is to study the transition from local phenomena to global phenomena, and so one relies on the extra structure to say when a certain kind of geometric data allows global conclusions to be drawn from local investigations. If we want to “do topology” in this sense over an arbitrary category C , we need to equip C with the right kind of extra structure. The crucial thing is to be able to say when a family of morphisms into an object R can be said to “cover” R . What we have in mind is that some sorts of data will be “well behaved” with respect to a notion of cover, in that knowing what is going on over the sources of morphisms in the cover allows us to completely understand the situation over the common target which they cover. What properties ought this collection to satisfy?

By “completely understand the situation” we really mean “compute a data value”. The idea is that if $\{f_x : U_x \rightarrow R \mid x \in X\}$ (for some indexing set X) covers R , then each data value over R is determined by its restrictions along each f_x , and conversely every consistent choice of a value over each U_x determines a value over R . At the moment, such a consistent collection of data values over a cover is like a code name

for a data value over R . If we know which data values are possible over each U_x , then we could count the number of possible data values over R by counting these code names. However, we don't just want to be able to count data values, we want to be able to *restrict them along morphisms*: we haven't really computed a data value if we can only name it uniquely but we don't know how to restrict it! What we want to demand, then, is that we can work out how to restrict a value over R given in terms of compatible values over each U_x along a morphism $g : S \rightarrow R$ by restricting the local piece over each U_x and glueing these restrictions together. But given a "local piece of a data value over R which is defined over U_x ", how can we compute a "local restriction along $g : S \rightarrow R$ " when we don't know *a priori* what the relationship between U_x and S is? The best we can hope to do is look at common factorizations of g and f_x , and restrict along these. If knowing some of these restrictions allows us to compute the value of the restriction to S along g , then we'll be ok. This means we want to demand

- $\{f_x : U_x \rightarrow R \mid x \in X\}$ covers R and $g : S \rightarrow R$ is a morphism of C then there is a cover $\{h_y : V_y \rightarrow S \mid y \in Y\}$ of S such that for all $y \in Y$ the map $g \cdot h_y$ factors through one of the maps f_x .

Now given "well behaved" data with respect to such a collection of covers, we will be able to study data values locally in the sense that we can name them by naming local pieces of them over each member of a cover, and we can compute their restrictions locally by looking at restrictions of their local pieces. Since we consider restrictions along morphisms to be the fundamental things about data values, it's clear that we can do anything we could possibly want to do with data values locally. We call such a collection of covering families a **coverage**.² A category equipped with a coverage is called a **site**.

Given a site, one often wants to construct a coverage on a related category. An example we need later in this thesis is to give a coverage on the slice category C/R whose objects are morphisms whose target is R and where an arrow from $f : S \rightarrow R$ to $g : T \rightarrow R$ is given by an arrow $h : S \rightarrow T$ such that the diagram

$$\begin{array}{ccc} S & \xrightarrow{h} & T \\ & \searrow f & \swarrow g \\ & & R \end{array}$$

²See [Joh02, Definition 2.1.1].

commutes.

To define a coverage on C/R , say that $\{h_x : f_x \rightarrow g \mid x \in X\}$, where $g : T \rightarrow C$ and $f_x : S_x \rightarrow T$ for all $x \in X$, covers g in C/R if and only if $\{h_x : S_x \rightarrow T \mid x \in X\}$ covers T in C .

To check that this gives a coverage, let $k : U \rightarrow T$ in C and $m : k \rightarrow g$ in C/R . We ought to be able to find a cover $\{p_y : n_y \rightarrow g \mid y \in Y\}$, where $n_y : N_y \rightarrow R$ in C , of k such that for all y the map $k \cdot p_y$ factors through some f_x . Since C is a site, there is a cover $p_y : N_y \rightarrow U$ of U satisfying the factorization condition. If we can find suitable morphisms $n_y : N_y \rightarrow R$ such that for all y the diagram

$$\begin{array}{ccc} N_y & \xrightarrow{p_y} & U \\ & \searrow n_y & \swarrow k \\ & & R \end{array}$$

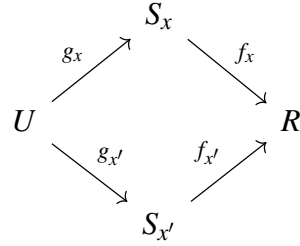
commutes, then we have finished. Of course, we can simply use $n_y = k \cdot p_y$, and so the condition above does define a coverage.

Having specified the aspect of the local-to-global transition which relies on having extra structure (a coverage) on our base category, we now want to specify what properties a presheaf should have in order to count as a kind of “well behaved” data. For the purposes of this thesis, we only need to assume that data assigned to large regions is determined by restricting it to a covering family of smaller regions. This is the separation property:

Definition 1. A presheaf F on a site C is **separated** if and only if for all objects $R \in C$ and all covers $\{f_x : S_x \rightarrow R \mid x \in X\}$ of R we have that for all $v, v' \in F(R)$ if for all f_x in the cover, $F(f_x)(v) = F(f_x)(v')$ then $v = v'$.

If, in addition, we wanted to be able to glue together consistent data values over a cover and produce a data value over the region covered, we would need to work with what is known as a collated presheaf.

Definition 2. A presheaf F on a site C is **collated** if and only if for all objects R of C and all covers $\{f_x : S_x \rightarrow R \mid x \in X\}$ of R whenever we have a collection $\{v_x \in F(S_x) \mid x \in X\}$ such that for all $x, x' \in X$ and commuting diagrams



we have $F(g_x)(v_x) = F(g_{x'})(v_{x'})$, there is a value $v \in F(R)$ such that for all $x \in x$ we have $F(f_x)(v) = v_x$.

A presheaf which is both separated and collated is call a **sheaf**. As mentioned above, in this thesis, collated presheaves, and hence sheaves, are never needed.

2.1.2 The Category of Elements of a Presheaf

Given a presheaf F , on a category C , we can construct a category $\int_C F$ with objects given by pairs (R, v) where R is an object of C and $v \in F(R)$. A morphism $g : (R, v) \rightarrow (S, w)$ is given by a morphism $g : R \rightarrow S$ of C such that $F(g)(w) = v$. We let $[-] : \int_C F \rightarrow C$ be the functor which takes the first co-ordinate, writing $[x]$ to mean $[-](x)$.

Given an object (S, w) of $\int_C F$ it is often clear what S should be from w , and so we often say that w is an object of $\int_C F$, leaving S implicit. This helps keep the notation uncluttered when the description of S would be complicated. Similarly, given $g : R \rightarrow S$ in C , we write $w \upharpoonright g$ to mean the object $(R, F(g)(w))$ of $\int_C F$. This restriction notation is convenient when dealing with a contravariant functor, because for g and h composable maps of C we have $w \upharpoonright g \cdot h = w \upharpoonright g \upharpoonright h$.

Note that this means that saying $g \in \int_C F[(R, v), (S, w)]$ means that $g : (R, v) \rightarrow (S, w)$ in $\int_C F$, which in turn amounts to $F(g)(w) = v$. In the sequel, we frequently make use of this equivalence in proofs. One way to think about this situation is that if we have objects (R, v) and (S, w) which we think of as “lying over” R and S , respectively, and a morphism $g : R \rightarrow S$, we wonder whether we can “lift” g to a morphism between (R, v) and (S, w) . We know this only occurs when $F(g)(w) = v$. This is a simple condition, but it can be useful to think of this as a simple instance of a more general geometric situation, where we ask whether a morphism of one category can be

lifted up to lie between objects of a related category.

At this point, it is tempting to wonder when being a “liftable” map is a local property. That is, given objects R and S of C , objects (R, ν) and (S, w) of $\int_C F$ over R and S respectively, a cover $\{f_x : U_x \rightarrow R \mid x \in X\}$, and a map $g : R \rightarrow S$, we can pre-compose g with each f_x to get “local” versions $g \circ f_x : U_x \rightarrow S$ of g . If all of these local versions of g lift to maps $g \circ f_x : (U_x, F(f_x)(\nu)) \rightarrow (S, w)$ of $\int_C F$, must g lift to a map $g : (R, \nu) \rightarrow (S, w)$; that is, must $\nu = F(g)(w)$?

It turns out that this happens precisely when F is separated. For suppose F is separated and all the local versions of g lift. Then for all $x \in X$ we have

$$\begin{aligned} F(f_x)(\nu) &= F(g \circ f_x)(w) \\ &= (F(f_x) \cdot F(g))(w) \text{ by contravariance of } F \\ &= F(f_x)(F(g)(w)). \end{aligned}$$

Then separation implies that $\nu = F(g)(w)$, which is what we wanted to show.

Conversely, suppose that all locally liftable maps can be lifted, and that for all $x \in X$, we have $F(f_x)(\nu) = F(f_i)(\nu')$. This latter assumption means that $1_R : R \rightarrow R$ is locally liftable to maps $1_R \circ f_i : (U_i, F(f_i)(\nu)) \rightarrow (R, \nu')$, hence we must have a global lift $1_R : (R, \nu) \rightarrow (R, \nu')$. But this just means that $F(1_R)(\nu) = \nu'$, which, by functoriality of F , implies $\nu = \nu'$ and so F must be separated.

Thinking in terms of whole hom-sets yields the following corollary of the above discussion.

Proposition 3. *Given a separated presheaf F on a site C , objects R and S of C , objects (R, ν) and (S, w) of $\int_C F$, and a cover $\{f_i : U_i \rightarrow R \mid x \in X\}$ of R , we have*

$$\left[\int_C F[(R, \nu), (S, w)] \right] = \bigcap_{x \in X} C[f_i, S]^{\leftarrow} \left[\int_C F[(U_i, F(f_i)(\nu)), (S, w)] \right].$$

Recall from above that for a morphism f of $\int_C F$ we write $\lfloor f \rfloor$ to mean the underlying morphism of C . Then the left hand side of the equality in Proposition 3 is just the set of liftable morphisms as discussed above. The right hand side is the set of “locally liftable” morphisms: the intersection of the set of morphisms whose pre-composition with an element of the cover is liftable.

2.2 Neighbourhood Spaces

In this thesis, our ultimate aim is to view computations by a model of computation as continuous paths through an abstract space of distributions of information over a base space. By a continuous path, we mean a continuous function whose domain is the real interval. Points of the real interval have “neighbourhoods” around them which must be entered eventually by anything which approaches the point. This system of neighbourhoods allows us to define what we mean by “continuous path”, and such systems of neighbourhoods are the subject matter of this chapter.

A key source of tension in the idea of computations as continuous paths is that the notion of computation we work with is discrete; a computation is a sequence of steps. Additionally, the underlying space will also be a combinatorial object, made of indivisible parts in contact with each other. An example we will refer to later is an infinite grid of square cells. Now, a cell in such a grid can also be said to have neighbourhoods consisting of those sets of cells which completely surround it. In particular, it has a *smallest neighbourhood* of cells which touch it. We might imagine that if we wanted to send a signal to a cell from far away, through the network of cells, it would have to pass into this smallest neighbourhood on its way to its destination.

Neighbourhood spaces constitute the right kind of geometric structure for encoding the notion of continuity which we need in this thesis, one which includes as examples both the real interval and the more combinatorial structures we need for modelling computations.

The propositions of this section are straightforward. We sketch a proof of the first one, Proposition 6, for the completely new reader, since it supports many of the arguments in this thesis. The reader is referred to [Cec66], or [HLCS91], [SS]³, or [KM02] for a fuller account.

Definition 4. A *neighbourhood space*⁴ is an underlying set S of points, together with, for each point $s \in S$, a collection of subsets of S called *neighbourhoods of s* such that

- S is a neighbourhood of s ;

³This reference is supplementary material for the paper [SS02].

⁴*Caveat lector*: some authors say “Cech space”, “pretopological space”, “preclosure space”, or even “closure space”. Yet other authors use the last of these to mean something else!

- if X is a neighbourhood of s , then $s \in X$;
- if X is a neighbourhood of s and $X \subseteq Y$, then Y is a neighbourhood of s ;
- if X and Y are neighbourhoods of s then $X \cap Y$ is a neighbourhood of s .

We sometimes say that one set is a neighbourhood of another if it is a neighbourhood of every point of the other. These axioms can be motivated by our intuition that X is a neighbourhood of s exactly when in order to approach s we must eventually end up in X . Since S is the whole space, we must spend all our time in S , and so it is a neighbourhood of every point. Since if I am already at s , I can approach it by sitting where I am, every neighbourhood of s must contain s . If I must eventually pass into X and $X \subseteq Y$, then when I do so I am also passing into Y . Finally, if I must eventually end up in X , and I must eventually end up in Y , then at some point I will find myself within both of them, never to leave. We think of the points of a neighbourhood of s as being somehow “close” to s , and we think of the smaller neighbourhoods as being “closer” to s than the larger ones.

The notion of morphism we require between two neighbourhood spaces is that of a *continuous function*. The idea is that a function f is continuous if and only if to get closer to $f(s)$ in the output, it is enough to supply an input sufficiently close to s .

Definition 5. Let S and T be neighbourhood spaces and $f : S \rightarrow T$ be a function from the underlying set of S to that of T . We say that f is **continuous** if and only if for all $s \in S$ and every neighbourhood $N_{f(s)}$ of $f(s)$ in T there is a neighbourhood N_s of s in S such that $f[N_s] \subseteq N_{f(s)}$.

We sometimes call a continuous function a **continuous map** or sometimes just a **map**. The word “function” by itself will always mean an arbitrary, not necessarily continuous function between underlying sets. We can somewhat simplify this condition to arrive at a more frequently used formulation.

Proposition 6. Let S and T be neighbourhood spaces and $f : S \rightarrow T$ be a function. Then f is continuous if and only if for all $s \in S$ and neighbourhoods $N_{f(s)}$ of $f(s)$ in T , the set $f^{-1}[N_{f(s)}]$ contains a neighbourhood of s .

Proof. This is just the (adjoint) relationship between inverse and direct images: if s is an arbitrary point of S and $N_{f(s)}$ a neighbourhood of its image, then f is continuous if and only if there is a neighbourhood N_s of s and $f[N_s] \subseteq N_{f(s)}$; this is equivalent to $N_s \subseteq f^{-1}[N_{f(s)}]$, which is precisely what we stipulated. \square

Note that in the statement of the proposition above, we could have demanded that $f^{-1}[N_{f(s)}]$ be a neighbourhood of s ; a set is a neighbourhood of a point if and only if it contains such a neighbourhood. By stating the proposition as we did we avoided the need for this fact, and the utility of the proposition is precisely that this is a common situation.

Bases and Subbases

It is often inconvenient, when defining a neighbourhood space, to give all neighbourhoods of a point. Often, there are some “obvious” neighbourhoods, and all the others are given by closing off to arrive at a collection satisfying the definition. The key thing we require is that these data are sufficient to detect whether a function is continuous or not.

Definition 7. *Let S be a neighbourhood space. A **basis** for S is given by specifying for each point $s \in S$ a collection of neighbourhoods of s called **basic neighbourhoods** such that for every neighbourhood N_s of s there is a basic neighbourhood B_s of s such that $B_s \subseteq N_s$.*

Proposition 8. *Let S and T be neighbourhood spaces and $f : S \rightarrow T$ be an arbitrary function. Then f is continuous if and only if for all $s \in S$ and basic neighbourhoods $B_{f(s)}$ of $f(s)$, the set $f^{-1}[B_{f(s)}]$ contains a basic neighbourhood of s .*

Now we know that specifying a basis for a space is sufficient to test for continuity of functions, we wonder when a system of subsets is a basis for some space.

Proposition 9. *Suppose that for a set S we specify, for each $s \in S$, a collection of given subsets of S , such that*

- *there is at least one given subset for s ,*
- *if X is a given subset for s then $s \in X$,*
- *if X and Y are given subset for s , then there is a given subset for s which is a subset of $X \cap Y$.*

Then S can be given the structure of a neighbourhood space in such a way that the given subsets form a basis of S .

This is quite convenient, but we can do a little better.

Definition 10. Let S be a set. A **subbasis** for a neighbourhood space on S is given by specifying for each point $s \in S$ a collection of neighbourhoods of s called **subbasic neighbourhoods** such that

- there is at least one subbasic neighbourhood of s , and
- if N_s is a subbasic neighbourhood of s then $s \in N_s$.

Given a subbasis on a set S , if we assign to each point $s \in S$ the collection of finite intersections of subbasic sets, then this will be a basis. By construction, it contains the intersection of any two elements (and hence a subset of the intersection). Since we can take a one-element intersection, every point has at least one basic neighbourhood. Finally, since s is an element of all subbasic neighbourhoods of s , it is also contained in any finite intersection of them. It turns out that a subbasis is sufficient to check whether a function is continuous.

Proposition 11. Let S and T be neighbourhood spaces with T specified by a subbasis, and $f : S \rightarrow T$ be a function. Then f is continuous if and only if for all $s \in S$ and subbasic neighbourhood $C_{f(s)}$ of $f(s)$, the set $f^{-1}[C_{f(s)}]$ contains a (basic) neighbourhood of s .

We are often lucky enough that the inverse image of a subbasic neighbourhood is a subbasic neighbourhood in the source. When this happens, we know by the above proposition that the function in question is continuous.

The Interior Operator

We could take a different perspective on the structure we have in a neighbourhood space. Rather than asking which sets are neighbourhoods of a point, we could ask instead which points consider a given set to be a neighbourhood. This will give us an operation on the powerset of our space S as follows.

Definition 12. Given a neighbourhood space S , the interior⁵ operator $\circ : \mathcal{P}S \rightarrow \mathcal{P}S$ is the function such that for all $X \subseteq S$ we have

$$X^\circ = \{s \in S \mid X \text{ is a neighbourhood of } s\}.$$

⁵Some authors prefer to say “preinterior”, reserving “interior” for the case where the operator we have defined is idempotent.

We can characterize which operations $\text{int} : \mathcal{P}S \rightarrow \mathcal{P}S$ are the interior operator of some neighbourhood space.

Proposition 13. *Let S be a set and $\text{int} : \mathcal{P}S \rightarrow \mathcal{P}S$. Then int is the interior operator for a neighbourhood space on S if and only if*

- we have $S^\circ = S$;
- it is decreasing: for all $X \subseteq S$ we have $X^\circ \subseteq X$;
- it is monotone: for all $X \subseteq Y \subseteq S$ we have $X^\circ \subseteq Y^\circ$; and
- for all $X, Y \subseteq S$ we have $(X \cap Y)^\circ = X^\circ \cap Y^\circ$.

We can think of X° as the set of points which can be reached by a small motion only from within X . Intuitively, a function is continuous if and only if a small motion in the input causes a small motion in the output. This will be true if and only if whenever I want the output to be reachable by a small motion only from within X , I must choose an input which can be reached by a small motion only from input points whose output values lie in X . This leads us to the following characterization of continuous maps.

Proposition 14. *Let S and T be neighbourhood space and $f : B \rightarrow T$ be a function. Then f is continuous if and only if for all $X \subseteq S$ we have $f^{\leftarrow}[X^\circ] \subseteq f^{\leftarrow}[X]^\circ$*

2.2.1 Open Sets and Topological Spaces

We call a subset of a neighbourhood space **open** if and only if it is a neighbourhood of each of its points. They are exactly the fixed points of the interior operator.

Proposition 15. *Let S be a neighbourhood space and $X \subseteq S$. Then X is open if and only if $X^\circ = X$.*

In any neighbourhood space, the open sets are closed under certain set theoretic operations.

Proposition 16. *Let S be a neighbourhood space. The collection of open subsets of S is closed under finite intersections and arbitrary unions.*

Open sets are well behaved with respect to continuous maps.

Proposition 17. *Let S and T be neighbourhood spaces, $X \subseteq S$ be an open set and $f : S \rightarrow T$ be a map. Then $f^{\leftarrow}[X]$ is open.*

Open sets have many uses. For example, one says that a set is **disconnected** if and only if it is the union of two disjoint, non-empty open sets. This corresponds to our intuitive idea that the set is separated into two pieces, and we cannot approach a point of one while remaining in the other. Roughly following the intuition that an open set is one whose members can be observed to be so (because any point “close” to a member is also one), we might want to know when the geometry of our space is observable, in the sense that if something approaches a point, we can observe it getting closer and closer. One might first think that every neighbourhood of a point should be open, but this is much too strong because any superset of a neighbourhood is itself a neighbourhood. If we just want to observe that our moving object approaches our given point, we need only demand that every neighbourhood of the latter *contains* an open neighbourhood, which patience will see the former enter eventually. So that is the property to demand: we call a space **topological** if and only if it has a basis of open neighbourhoods.

This can be rephrased in terms of the interior operator.

Proposition 18. *A neighbourhood space is topological precisely when the interior operator is idempotent.*

A corollary is that the interior of every set is open, since the image of an idempotent operator is its set of fixed points.

We can characterize topological spaces from a more purely geometric point of view (put nicely in [Ish02, p. 32] whose intuitions we follow here): a space is topological precisely when every neighbourhood of a point is also a neighbourhood of all sufficiently close points.

Proposition 19. *A neighbourhood space S is topological if and only if for every point $s \in S$ and every neighbourhood N_s of s , there is a neighbourhood $U_s \subseteq N_s$ of s such that for all $u \in U_s$ N_s is also a neighbourhood of u .*

We can interpret this as a uniformity condition on the space. Without it, there is little relationship between the neighbourhoods of nearby points, and so we could specify these to be wildly different if we wished. However, it might be natural to suppose that as points get nearer and nearer, more and more points are near them both.

We can characterize continuity of functions into topological spaces in terms of open sets.

Proposition 20. *Let S be a neighbourhood space, T be a topological space, and $f : S \rightarrow T$ be a function. Then f is continuous if and only if for all open sets $U \subseteq T$ the preimage $f^{-1}[U]$ is open in S .*

Similarly, we can produce a topological space from any collection of open sets with the right closure properties.

Proposition 21. *Let S be a set and suppose that \mathcal{O} is a collection of subsets of S closed under arbitrary unions and finite intersections. Then there is a unique neighbourhood structure on S which is topological whose open sets are exactly those contained in \mathcal{O} .*

The functor which includes the category of topological spaces as a full subcategory of neighbourhood spaces has a left adjoint. A neighbourhood space has a collection of open sets, which by Proposition 16 satisfy the conditions of Proposition 21 and so define a topological space. Observe that this space preserves the underlying set of points. The unit of this adjunction has as components maps whose underlying function is the identity. The counit is the identity functor of the inclusion of topological spaces into neighbourhood spaces.

2.2.2 Cospecialization and Alexandroff Spaces

An important relation for investigating the fine structure of a neighbourhood space is the **cospecialization** relation.

Definition 22. *Let S be a neighbourhood space and $s, s' \in S$. We set $s \rightsquigarrow s'$ if and only if s is an element of every neighbourhood of s' . We say that s **cospecializes** s' .⁶*

If s cospecializes s' , then we imagine that we can approach s' just by sitting at s . In other words, s is pressed up arbitrarily close to s' already: no mote of space exists between them; no sliver could intervene to make them part. This relation can in general be quite arbitrary, but we do know one rather unexciting fact about it.

Proposition 23. *The cospecialization relation of any neighbourhood space is reflexive.*

Now since a continuous map, though it may distort space considerably, cannot rip it apart, the distorted images of two points which were in contact with each other must remain in contact.

⁶One might say s' specializes s , or that s specializes to s' ; it is common to define the reverse of this relation, hence the strange name.

Proposition 24. *Let S and T be neighbourhood spaces, $s, s' \in S$ with $s \rightsquigarrow s'$, and let $f : S \rightarrow T$ be a continuous map. Then $f(s) \rightsquigarrow f(s')$.*

If we think about the infinite grid of squares where our cellular automata live, we notice that every cell is surrounded by a neighbourhood of cells which touch it. Indeed, this neighbourhood is made up of all the points which cospecialize s . This motivates us to think more generally about such spaces.

Definition 25. *A neighbourhood space S is called **Alexandroff** if and only if for every point $s \in S$ the set $\downarrow s = \{s' \in S \mid s' \rightsquigarrow s\}$ is a neighbourhood of s .*

Since every point of $\downarrow s$ cospecializes s , it is contained in every neighbourhood of s . This means that $\downarrow s \subseteq N_s$ for every neighbourhood N_s of s . Indeed, we could equivalently have stipulated that every point have a smallest neighbourhood. A further consequence is that the assignment of $\{\downarrow s\}$ to s gives a subbasis of S .

We can characterize Alexandroff spaces in terms of the interior operator.

Proposition 26. *A neighbourhood space S is Alexandroff if and only if for every collection \mathcal{U} of subsets of S we have*

$$\left(\bigcap \mathcal{U}\right)^\circ = \bigcap_{U \in \mathcal{U}} U^\circ.$$

We can characterize continuity of functions out of Alexandroff spaces in terms of the cospecialization relation.

Proposition 27. *Let S be an Alexandroff space, T be an Alexandroff space, and $f : S \rightarrow T$ be a function. Then f is continuous if and only if for all $s, s' \in S$ such that $s \rightsquigarrow s'$ we have $f(s) \rightsquigarrow f(s')$.*

Every reflexive relation gives rise to a neighbourhood space with that relation as the cospecialization relation.

Proposition 28. *Let S be a set and let R be a reflexive relation on S . Then S there is a unique neighbourhood structure on S which is Alexandroff and whose cospecialization relation is R .*

The inclusion functor from the full subcategory of Alexandroff spaces to the category of neighbourhood spaces has a right adjoint. This sends any space to the one

defined by its cospecialization relation, via Propositions 23 and 28. The unit is the identity natural transformation on the inclusion. The components of the counits have identities as underlying functions.

2.2.3 Coarse and Fine Structures, Limits and Colimits

A set S can be equipped with many different neighbourhood space structures. There is a natural way to compare two such structures S_0 and S_1 by asking whether the identity function $1_S : S_0 \rightarrow S_1$ is continuous. When it is we say that S_0 is **finer** than S_1 and S_1 is **coarser** than S_0 . The definition of continuity means that this happens if and only if every neighbourhood of s in S_1 is a neighbourhood of s in S_0 . This justifies our idea that S_0 is a “finer” space because if something approaches s in S_0 then it must approach in S_1 —in other words, S_0 can “see” all details in the geometry of S_1 which might get in the way of approaches to points.

Proposition 29. *Let S be a set. The set of neighbourhood space structures on S ordered by the fineness relation form a complete lattice.*

The finest space on a set S is called the **discrete space** on S and it the space in which every set containing s is one of its neighbourhoods; conversely, the coarsest space, the **indiscrete space** is the one in which the only neighbourhood of any point is the whole space. Any function out of a discrete space is continuous because the inverse image of a neighbourhood is surely a set, hence a neighbourhood; any function into an indiscrete space is continuous because the inverse image of the whole space is the whole of the source space. The functor taking a set to the discrete space on that set gives a left adjoint to the forgetful functor which sends a space to its set of points; the functor which takes a set to the indiscrete space is a right adjoint.

Given a collection $\{S_x \mid x \in X\}$ of neighbourhood spaces, with underlying sets $\{U_x \mid x \in X\}$, a set U and functions $\{f_x : U \rightarrow U_x \mid x \in X\}$ there is always a finest space S such that for all $x \in X$ the function f_x is a continuous map $f_x : S \rightarrow S_x$. Dually, given $\{g_x : U_x \rightarrow U \mid x \in X\}$ there is a coarsest space making them continuous.

This property implies that the category of neighbourhood spaces has all limits and colimits. The discrete and indiscrete space adjunctions mean that the underlying sets of points of these limits are the same as the corresponding limits of the sets of points of the given spaces. Similarly, the inclusion of the category of topological spaces must

preserve limits, and the inclusion of Alexandroff spaces must preserve colimits.

It happens that the coproduct of topological spaces, taken in the category of neighbourhood spaces, is again a topological space. This is because a neighbourhood of a point in the coproduct is a set which contains (the image under an inclusion of) one of the existing neighbourhoods of the point. If the arguments are topological spaces then these existing neighbourhoods contain open neighbourhoods and so the coproduct is again a topological space.

Not all colimits are preserved, however. This can be seen by considering the Alexandroff space on the set $\{a, b, c, d\}$ given by the relation indicated below (where we do not draw the reflexivity edges).

$$a \longrightarrow b \qquad c \longrightarrow d$$

This is a topological space. Consider the equivalence relation which identifies b with c . The quotient of the space by this equivalence in the category of neighbourhood spaces is simply

$$[a] \longrightarrow [b] \longrightarrow [d],$$

but the quotient as a topological space is the transitive closure

$$\begin{array}{ccc} & [b] & \\ & \nearrow & \searrow \\ [a] & \longrightarrow & [d]. \end{array}$$

In this thesis, the only quotient we ever need to take is in the background section on simplicial complexes, in order to describe the notion of the *join* in Section 2.4.2. Unfortunately, what we need is the quotient of a topological space *taken in the category of topological spaces*, a point we emphasise again at the appropriate moment.

Similarly, it happens that finite products of Alexandroff spaces are again Alexandroff. This fails for countable products.

In [LCS93], it is proved that the category of neighbourhood spaces is not cartesian closed. Indeed, the exponentiable objects are exactly the Alexandroff spaces (there called finitely generated spaces). This does not cause any trouble for the work of this thesis, but it may for future work, where it may be desirable to construct, for example, path spaces in neighbourhood spaces. It may be possible to use cartesian closed supercategories of the category of neighbourhood spaces for this purpose.

2.3 Paths, Homotopies, and Directed Spaces

We are aiming for a description of computation in terms of paths through spaces. The spaces in question are neighbourhood spaces, and so we need a notion of a path through a neighbourhood space. The usual notion of path in topology is a map whose source is the real interval $\mathbb{I} = \{i \in \mathbb{R} \mid 0 \leq i \leq 1\}$. Since all topological spaces are neighbourhood spaces, the interval is a neighbourhood space, and so we can use this familiar definition of path for neighbourhood spaces. A path needs a beginning and an end point, and it is profitable to think about points of a space as maps out of the one point space \bullet . The interval has a pair of very conspicuous endpoints $0, 1 : \bullet \rightarrow \mathbb{I}$ given by the constant functions with value 0 and 1 respectively. Given a space S , if we pick desired start and end points $s, t : \bullet \rightarrow S$ for a path, we the path to be a map $p : \mathbb{I} \rightarrow S$ such that $p \cdot 0 = s$ and $p \cdot 1 = t$.

Definition 30. *Given a neighbourhood space S and points $s, t \in S$, a **path from s to t** in S is a map $p : \mathbb{I} \rightarrow S$ such that $p \cdot 0 = s$ and $p \cdot 1 = t$.*

One interesting class of paths are those given by the cospecialization relation.

Proposition 31. *Let S be a neighbourhood space and $s, s' \in S$ be such that $s \rightsquigarrow s'$. Then there is a path from s to s' .*

Proof. Consider the function which takes the value s on the interval $[0, \frac{1}{2})$ and s' afterwards. We want to show that this is continuous. Let $i \in \mathbb{I}$ and consider $f(i)$. If $i \leq \frac{1}{2}$ then $f(i) = s$ and every neighbourhood of $f(i)$ contains s by the definition of neighbourhood space; if $f(i) = s'$ then every neighbourhood of $f(i)$ contains s' and also s because $s \rightsquigarrow s'$. In the former case, the inverse image of s is just $[0, \frac{1}{2})$ which is a neighbourhood of i ; in the latter the inverse image is all of \mathbb{I} . \square

Given a space S , points $a, b, c \in S$ and paths p from a to b and q from b to c , We define the **concatenation** of p and q to be the path $p \frown q : \mathbb{I} \rightarrow S$ such that for all $i \in \mathbb{I}$ we have

$$(p \frown q)(i) = \begin{cases} p(2i) & \text{if } i \leq \frac{1}{2} \\ q(2(i - \frac{1}{2})) & \text{if } i \geq \frac{1}{2}. \end{cases}$$

Note that this makes sense when $i = \frac{1}{2}$ since $(p \frown q)(\frac{1}{2}) = p(1) = b = q(0)$.

We could think of a path as a way of continuously deforming one point into another. By seeing points as maps out of the one-point space, we might be motivated to consider

a way of deforming maps whose source is an arbitrary space. This idea is to stretch out the source space so that we have one copy of it for each moment of time.

Definition 32. *Given neighbourhood spaces S and T and maps $s, t : S \rightarrow T$, a **homotopy from s to t** is a map $h : S \times \mathbb{I} \rightarrow T$ such that $h \cdot 1_S \times 0 = s$ and $h \cdot 1_S \times 1 = t$.*

It might happen that our two maps f and g agree on some subspace $U \subseteq S$, and we want to make sure that the homotopy preserves this property. One might imagine a tablecloth in the wind, which is fixed where it touches the table, but which might change shape otherwise. A position of the tablecloth is a continuous map from the disc into three dimensional space, and a deformation from one position to another is a homotopy of maps, but every intermediate position must also fix the part which lies over the table. A more serious reason to want to fix a subspace is to preserve some important structural information about the map. For example, a path is not just a map out of \mathbb{I} , because we care about where it comes from and where it goes to.

Definition 33. *Given neighbourhood spaces S and T , a subspace $U \subseteq S$ and maps $s, t : S \rightarrow T$ such that $s \upharpoonright U = t \upharpoonright U$, a **homotopy relative to U from s to t** is a map $h : S \times \mathbb{I} \rightarrow T$ such that $h \cdot 1_S \times 0 = s$ and $h \cdot 1_S \times 1 = t$, and such that $h \upharpoonright U \times \mathbb{I} = s \upharpoonright U \times 1_{\mathbb{I}}$.*

One often sees the shortened form **homotopy rel U** , and if $U = \{0, 1\} \subseteq \mathbb{I}$ then one says the homotopy is relative to “endpoints”.

2.3.1 Directed Spaces

We are aiming to give a kind of space whose paths correspond to computations. Computation can involve irreversible operations, like forgetting information. Thus, one needs a notion of space which incorporates a notion of direction on its paths. We follow [Gra09].

Definition 34. *A **directed neighbourhood space** is a neighbourhood space S together with a collection of selected paths in S which is closed under*

- *concatenation of paths, and*
- *precomposition with any continuous map $\mathbb{I} \rightarrow \mathbb{I}$ which is monotone with respect to the usual order on \mathbb{I} .*

For example, if we take the interval \mathbb{I} and equip it with the collection of all monotone paths, then this gives the **directed interval** \mathbb{I}_{\leq} . An example which does not come from an ordering is the circle with its collection of clockwise paths.

Definition 35. *If S and T are directed neighbourhood spaces and $f : S \rightarrow T$ is a map, then f is **directed** if and only if the image of a selected path in S is selected in T .*

For example, the selected paths in a space S are exactly the directed maps $\mathbb{I}_{\leq} \rightarrow S$. This gives a category of directed spaces. We can embed the category of neighbourhood spaces by selecting all paths in an object. In our example \mathbb{I}_{\leq} , the directed structure is based on a relation on the underlying undirected space. We are interested in general instances of this phenomenon.

Definition 36. *A directed space D is **given by a relation** R if and only if*

- *a path $\mathbb{I} \rightarrow D$ is directed if and only if for all $i \in \mathbb{I}$ there is a neighbourhood N_i such that for all $i_0, i_1 \in N_i$ such that $i_0 \leq i_1$ we have $f(i_0) R f(i_1)$, and*
- *for all $d \in D$ there is a neighbourhood N_d such that for all $d_0, d_1 \in N_d$ we have $d_0 R d_1$ if and only if there is a directed path from d_0 to d_1 .*

We have seen that one example is the space \mathbb{I}_{\leq} above. Another is the circle, where the idea is to take the relation “ p is no more than a half-turn clockwise from p' ”, or any like it. An important class of examples is given by equipping any space with the directed paths given by the cospecialization relation. We specify the directed paths to make the first stipulation true; the second is then Proposition 31 where we use the neighbourhood $\downarrow p$ for every point p . We get a nice characterization of the directed maps between two spaces given by relations.

Proposition 37. *Let S and T be directed spaces given by the relations Q and R respectively. If $f : S \rightarrow T$ is a continuous map, then f is directed if and only if for every point $s \in S$ there is a neighbourhood N_s such that for all $s_0, s_1 \in N_s$ if $s_0 Q s_1$ then $f(s_0) R f(s_1)$.*

Proof. We first do the “if” direction. Let $p : \mathbb{I}_{\leq} \rightarrow S$ be a directed path in S . We want to show that $f \cdot p : \mathbb{I}_{\leq} \rightarrow T$ is directed. Let $i \in \mathbb{I}$. Since p is directed there is a neighbourhood N_i such that for all $i_0, i_1 \in N_i$ with $i_0 \leq i_1$ we have $p(i_0) Q p(i_1)$. By our assumption about f , we have a neighbourhood $N_{p(i)}$ such that for all $s_0, s_1 \in N_{p(i)}$ if $s_0 Q s_1$ then $f(s_0) R f(s_1)$. Since $p^{\leftarrow}[N_{p(i)}]$ is a neighbourhood of i , so is $N_i \cap p^{\leftarrow}[N_{p(i)}]$.

Now let $i_0, i_1 \in N_i \cap p^{\leftarrow}[N_{p(i)}]$ and $i_0 \leq i_1$. Then $p(i_0), p(i_1) \in N_{p(i)}$ and $p(i_0) \mathcal{Q} p(i_1)$, and hence $f(p(i_0)) \mathcal{R} f(p(i_1))$ as required.

We now do the “only if” direction. Suppose that f is directed and let $s \in S$. Since S is given by \mathcal{Q} , let N_s be a neighbourhood of s such that for all $s_0, s_1 \in N_s$ we have $s_0 \mathcal{Q} s_1$ if and only if there is a directed path from s_0 to s_1 . Since T is given by \mathcal{R} , let $N_{f(s)}$ be a neighbourhood of $f(s)$ such that for all $t_0, t_1 \in N_{f(s)}$ we have $t_0 \mathcal{R} t_1$ if and only if there is a directed path from t_0 to t_1 . Since f is continuous, $f^{\leftarrow}[N_{f(s)}]$ is a neighbourhood of s ; so is $f^{\leftarrow}[N_{f(s)}] \cap N_s$. Let $s_0, s_1 \in f^{\leftarrow}[N_{f(s)}] \cap N_s$ and $s_0 \mathcal{Q} s_1$. Then there is a directed path p from s_0 to s_1 . Since f is directed, $f \cdot p$ is a path from $p(s_0)$ to $p(s_1)$. Since $p(s_0), p(s_1) \in N_{f(s)}$ we have $p(s_0) \mathcal{R} p(s_1)$ as required. \square

2.4 Simplicies and Simplicial Complexes

2.4.1 Motivation

In a later chapter, Chapter 3, we want to reason about paths (which are maps out of the unit interval), paths between paths (maps out of the unit cube) and so on in all dimensions. When proving that these paths have the properties we need, we need to split up the unit hypercubes into easily managed pieces. The tools we use for this job are *simplicial complexes* – that is, we triangulate our spaces and work with the triangulations. The utility of this method derives from the special structure of the simplex.

In this section, we always work with *concrete simplicial complexes*: special subsets of \mathbb{R}^n . We assume the latter comes with a specified set of co-ordinates: we identify points with their position vectors, and add and scale them with abandon. This is really a convenience, and what is needed in subsequent chapters are the properties of simplicial complexes given here in as Propositions. The reader should think of these properties as axioms for a notion of “simplicial complex”; the fact that we can prove them in our model of subsets of \mathbb{R}^n shows that they are applicable to the hypercubes we ultimately care about. Occasionally, we state a property which is only needed within this section in order to prove something about the concrete situation. These are designated Observations, and readers who believe all the Propositions for their preferred idea of a simplicial complex should feel free to ignore them.

The material in this section is, for the most part, standard fare in piecewise linear topology; we generally follow Hudson [Hud69]. Where a reference covers exactly the property needed (in most cases, with a proof), the reader is referred there; otherwise, we give a proof from the other properties.

2.4.2 The Structure of Simplices

A set of points in \mathbb{R}^n is called *independent* if and only if none lies on the hyperplane containing the others. In terms of their position vectors, this means that if we pick one and look at all the difference vectors between it and the other points, these difference vectors are linearly independent. An ***n*-simplex** is the convex hull of $n + 1$ independent points. More explicitly, given $n + 1$ independent points $\{p_0, \dots, p_n\}$ the *n*-simplex

spanned by $\{p_0, \dots, p_n\}$ is the set of points of the form

$$\sum_{x=0}^n \lambda_x p_x$$

where the λ_x are non-negative real numbers which sum to 1. The subset in which the λ_x are all non-zero is called the **simplicial interior** of the simplex. Given an n -simplex D , we write $intD$ for the simplicial interior of D . Take care not to confuse the *simplicial* interior with the *topological* interior of D : they coincide for an n -simplex in \mathbb{R}^n , but not for an n -simplex in \mathbb{R}^k when $k \geq n$. In the references, “interior” usually means the simplicial interior.

Any subset of a set of independent points is also independent. If D is the simplex spanned by $\{p_0, \dots, p_n\}$, then any subset give a simplex whose points are a subset of D . We call such simplices *faces* of D , and say ‘proper face’ for a proper subset. Observe that the proper faces of D are all disjoint from $intD$ because the corresponding coefficients are all zero. We say that a simplex defined by $n + 1$ independent points is n -dimensional; it follows that the dimension of a face of D is strictly less than the dimension of D . Note that we *do not* require faces to have dimension *exactly* one less than the dimension of D .

The faces are an important part of the structure of a simplex in their own right, and they appear frequently in this and subsequent sections. They also crucially parametrize a family of “join structures” corresponding to pairs of opposite faces. We call two faces of a simplex *opposite* if they are generated by complementary subsets of vertices. This implies that every proper face has an opposite (spanned by the complement of its set of vertices). Consider opposite faces D_0 and D_1 of D . Let \sim be the equivalence relation on $D_0 \times D_1 \times I$ such that $(d_0, d_1, 1) \sim (d'_0, d_1, 1)$ for all $d_0, d'_0 \in D_0$ and $(d_0, d_1, 0) \sim (d_0, d'_1, 0)$ for all $d_1, d'_1 \in D_1$. We consider the quotient of $D_0 \times D_1 \times I$ by the relation \sim *in the category of topological spaces* (see the discussion of quotients on page 36). The space $D_0 \times D_1 \times I / \sim$ can be pictured in the following way: ‘on the bottom’, when the I co-ordinate is 0, the space looks like D_0 ; ‘on the top’, when the I co-ordinate is 1, the space looks like D_1 . In between, it consists of the union of all lines which ‘join a point of D_0 to a point of D_1 ’. This space is called the **join** of D_0 and D_1 .

Then D is homeomorphic to the join of D_0 and D_1 via a pair of maps

$$\begin{array}{ccc}
 & \xrightarrow{\phi_{D,D_0}} & \\
 D & \curvearrowright & D_0 \times D_1 \times I / \sim \\
 & \xleftarrow{\phi_{D,D_0}^{\leftarrow}} &
 \end{array}$$

We examine these maps in this section, ultimately in order to prove Proposition 64. Our study of them here will conclude in giving two observations about a family of related maps, defined later in this subsection, which are what the proof of the just-mentioned Proposition actually requires.

Of the two maps comprising the homeomorphism above, the one we really need is ϕ_{D,D_0} . But then $\phi_{D,D_0}^{\leftarrow}$ is easier to describe, so we give $\phi_{D,D_0}^{\leftarrow}$ and deduce the properties of ϕ_{D,D_0} which we need from it.

Let $\phi_{D,D_0}^{\leftarrow} : D_0 \times D_1 \times I \rightarrow D$ be given by

$$\phi_{D,D_0}^{\leftarrow}(d_0, d_1, i) = (1-i)d_0 + id_1.$$

The textbook [RSF⁺13, Section 8.1] gives this map and states that it forms part of a homeomorphism.

Let us examine some important properties of ϕ_{D,D_0} . First, observe that the output of $\phi_{D,D_0}^{\leftarrow}$ is a point of D_0 if and only if the I co-ordinate is 0, and of D_1 if and only if it is 1. This means that the image of D_0 under ϕ_{D,D_0} is the set of points of the join with I co-ordinate 0, and the image of D_1 is all points with I co-ordinate 1.

Note that a point $[(d_0, d_1, 0)]$ is sent by $\phi_{D,D_0}^{\leftarrow}$ to d_0 , and is therefore the value of $\phi_{D,D_0}(d_0)$ (It may look as though we need to choose a value for d_1 , but the quotient means that all choices are equivalent).

Finally, suppose that E is a face of D such that $E \cap D_0 \neq \emptyset$ and $E \cap D_1 \neq \emptyset$; these intersections must be opposite faces of E . The points mapped to E by $\phi_{D,D_0}^{\leftarrow}$ (that is, $\phi_{D,D_0}[E]$) are exactly those whose D_0 and D_1 co-ordinates are contained in $E \cap D_0$ and

$E \cap D_1$ respectively, since E is the convex hull of its opposite faces. Then by evaluating at a point of $\phi_{D,D_0}[E]$, the reader will verify that

$$\phi_{E,E \cap D_0}^{\leftarrow} = \phi_{D,D_0}^{\leftarrow} \upharpoonright \phi_{D,D_0}[E],$$

and hence

$$\phi_{E,E \cap D_0} = \phi_{D,D_0} \upharpoonright E.$$

Now, on the subspace $D_0 \times D_1 \times [0, 1) / \sim$ of the join, the map π_0 which takes a point to its D_0 co-ordinate is well defined. Our observations imply that $\pi_0 \circ \phi_{D,D_0} \upharpoonright D_0 = 1_{D_0}$.

We can use what we have just learned to define, for each simplex D and face D_0 of D with opposite face D_1 , a map ∇_{D,D_0} which is a retraction from $D - D_1$ to D_0 . If D_0 is a proper face, we set $\nabla_{D,D_0} = \pi_0 \circ (\phi_{D,D_0} \upharpoonright D - D_1)$, otherwise $\nabla_{D,D_0} = \nabla_{D,D} = 1_D$.

We are interested in how this family of retractions interacts with faces. Later on, we combine these retractions on various simplices to produce a larger retraction, and we want to know that this is consistent. That means that if one of the simplices is a face of another, we make compatible assignments.

Observation 38. *Let D be a simplex and D_0 a face of D . If E is a face of D such that $E \cap D_0 \neq \emptyset$, then $\nabla_{E,E \cap D_0} = \nabla_{D,D_0} \upharpoonright E$.*

Proof. Let E be a face of D which has a non-empty intersection with D_0 . If $E \subseteq D_0$ (so $D_0 \cap E = E$) then

$$\begin{aligned} \nabla_{E,E \cap D_0} &= 1_E \\ &= 1_{D_0} \upharpoonright E \\ &= \nabla_{D,D_0} \upharpoonright D_0 \upharpoonright E \\ &= \nabla_{D,D_0} \upharpoonright D_0 \cap E \\ &= \nabla_{D,D_0} \upharpoonright E. \end{aligned}$$

Otherwise $E \cap D_0$ is a proper face of E and $E \cap D_1 \neq \emptyset$, so we have

$$\begin{aligned}\nabla_{E, E \cap D_0} &= \pi_0 \circ \phi_{E, E_0} \\ &= \pi_0 \circ \phi_{D, D_0} \upharpoonright E \text{ by the discussion above,} \\ &= \nabla_{D, D_0} \upharpoonright E.\end{aligned}$$

Hence in both cases $\nabla_{E, E \cap D_0} = \nabla_{D, D_0} \upharpoonright E$. \square

We also want to know that these retractions preserve simplicial interiors. We prove the contrapositive, that every point whose image is in a proper face of D_0 must have come from a proper face of D .

Observation 39. *Let D be a simplex, and D_0 and D_1 be opposite faces of D or $D_1 = \emptyset$ and $D_0 = D$. For all $p \in D$ such that $\nabla_{D, D_0}(p)$ is an element of a proper face of D_0 , there is a proper face of D which contains p .*

Proof. If $D_0 = D$ then this is clear since ∇_{D, D_0} is the identity. Otherwise

$$\begin{aligned}\nabla_{D, D_0}(p) &= \pi_0 \cdot (\phi_{D, D_0} \upharpoonright D - D_1)(p) \\ &= (\pi_0 \cdot \phi_{D, D_0})(p).\end{aligned}$$

Now if E is a proper face of D_0 and $\nabla_{D, D_0}(p) \in E$, then by the above calculation $(\pi_0 \cdot \phi_{D, D_0})(p) \in E$. That means that the d_0 co-ordinate of ϕ_{D, D_0} is a point of E . Hence, we have

$$p = (1 - i)d_0 + i(d_1)$$

for some $0 \leq i \leq 1$, $d_1 \in D_1$ and $d_0 \in E$.

Now let D be spanned by the points $\{p_0, \dots, p_n\}$, ordered in such a way that for some $0 \leq m \leq n$, D_0 is spanned by $\{p_0, \dots, p_m\}$ and D_1 by $\{p_{m+1}, \dots, p_n\}$. Then there are non-negative real numbers $\{\alpha_i, \dots, \alpha_n\}$ such that

$$d_0 = \sum_{x=0}^m \alpha_x p_x$$

and

$$d_1 = \sum_{x=m+1}^n \alpha_x p_x,$$

and hence

$$p = (1 - i) \sum_{x=0}^m \alpha_x p_x + i \sum_{x=m+1}^n \alpha_x p_x.$$

So we can express p in the form

$$p = \sum_{x=0}^n \lambda_x p_x$$

where for all $0 \leq x \leq m$ we have $\lambda_x = (1 - i)\alpha_x$ and for $m + 1 \leq x \leq n$ we have $\lambda_x = i\alpha_x$.

Now since E is a proper face of D_0 and d_0 lies in E , there exists $0 \leq x \leq m$ such that $\alpha_x = 0$. But then $\lambda_x = (1 - i)(\alpha_x) = 0$, and so p lies on a proper face of D . \square

The reader who suffers through the proofs of these observations is entitled to ask: why do we have to prove them? It might be thought that the piecewise linear topologists should have taken care of this in the 1960s. The answer is that the maps ∇_{D,D_0} are certainly not linear when restricted to the simplicial interiors of simplices.⁷ To see why, let D be a triangle and D_0 its base. The map ∇_{D,D_0} sends all points of an edge other than the base to the point where this edge intersects the base. If this map were linear, then it could be extended to a map defined on the whole triangle (recall that it is really defined on the triangle *without* the point opposite the base). But this clearly cannot be done in a continuous manner. Nevertheless, the idea is similar to the standard notion of collapse used in piecewise linear topology. Our deviation from standard constructions here serves to ensure that Observations 38 and 39 are relatively simple to prove.

It is this collection of maps ∇_{D,D_0} which is important in the sequel; the reader can now forget ϕ . Indeed, even ∇ is only used in this section, to define a way of extending continuous maps under good circumstances (in the proof of Proposition 64), and this is the useful consequence of the preceding discussion for later chapters. Before we can say what we mean by “good circumstances”, however, we need aspects of the theory of simplicial complexes, to which we now turn.

⁷Indeed, this seems to be an instance of the infamous “standard mistake” mentioned by [RS12, Bry01, Zee63]. See in particular the example in the last reference.

2.4.3 Simplicial Complexes and Subcomplexes

Having studied single simplices in isolation, we proceed to study well-behaved subsets of \mathbb{R}^n which are the unions of nice collections of simplices.

Definition 40. A simplicial complex S is a finite set of simplices in \mathbb{R}^n satisfying the following conditions:

- If $D \in S$ and E is a face of D , then $E \in S$, and
- If $D, E \in S$ then $D \cap E$ is either empty or a common face of D and E .

A collection of simplices satisfying the second condition can always be completed to give a simplicial complex by adding the faces of all simplices it contains. The *dimension* of a simplicial complex is the maximum dimension of any of its simplices. Observe that since the dimension of a proper face is always strictly less than that of the simplex it is a face of, if we complete a collection of simplices to be a simplicial complex, the simplices we need to add are of dimension strictly lower than the maximum appearing in the collection.

Proposition 41. Let D and D' be simplices of a simplicial complex S . If $D \cap \text{int} D' \neq \emptyset$, then $D' \subseteq D$.

Proof. Let S be a simplicial complex and $D, D' \in S$ such that $D \cap D' \neq \emptyset$. Since $D \cap D'$ is non-empty, it must be a face of both D and D' . Since the intersection contains a point of the simplicial interior of D' , it cannot be a proper face of the latter, and hence $D \cap D' = D$ which implies the result. \square

Given a simplicial complex S , a subcomplex $S' \subseteq S$ is any subset which is closed under faces. The following fact is immediate from the definition, but we name it for future use.

Proposition 42. If S is a simplicial complex and S_0 and S_1 are subcomplexes of S , then so are $S_0 \cup S_1$ and $S_0 \cap S_1$.

By the remarks above, any subset of a complex can be completed to a subcomplex by adding simplices of low dimension. This leads to the following observation.

Proposition 43. *If S is a simplicial complex and S_0 and S_1 are subcomplexes such that $S = S_0 \cup S_1$ then there exist subcomplexes $S'_0 \subseteq S_0$ and $S'_1 \subseteq S_1$ such that $S = S'_0 \cup S'_1$ and $S'_0 \cap S'_1$ is a subcomplex of S of dimension strictly lower than that of S .*

Proof. We let $S'_0 = S_0$ and S'_1 be the completion of $S_1 - S_0$ to a subcomplex. Then

$$\begin{aligned} S'_0 \cup S'_1 &\supseteq S_0 \cup (S_1 - S_0) \\ &= S_0 \cup S_1 \\ &= S, \end{aligned}$$

and since it is clear that $S'_0 \cup S'_1 \subseteq S$ since these are subcomplexes, we have $S'_0 \cup S'_1 = S$ as required. Proposition 42 implies that $S'_0 \cap S'_1$ is a subcomplex of S . Furthermore, since any simplex contained in $S'_0 \cap S'_1$ must have been added in order to complete S'_1 to a complex, these must all have dimension strictly less than that of S . \square

2.4.4 The Underlying Space of a Complex

Given a simplicial complex S , we denote the union of S by $|S|$: this is the *underlying space* of the complex.

Proposition 44. *Let $p \in |S|$, then there is a unique $D \in S$ such that $p \in \text{int} D$*

Proof. [Adh16, Proposition 6.2.18]. \square

For our purposes in this document, we need the fact that all cubes are the underlying space of certain complexes.

Proposition 45. *For all $n \in \mathbb{N}$ there is a simplicial complex S such that $|S| = I^n$.*

Proof. This is clear for $n = 0, 1$ since we can take a single point, or the interval itself together with its endpoints. For the step case of an induction, it suffices to show that the product of (the underlying spaces of) simplicial complexes is again a (space underlying a) simplicial complex. This follows from two principles in Hudson which we do not need to use again in this document. First, the products of the simplices are polyhedral cells (see [Hud69, p. 2]), and so we can easily find a more general complex of these whose union is the product. Then, any such complex can be subdivided into simplices; see [Hud69, Lemma 1.4]. \square

We give it the topology inherited from \mathbb{R}^n as a subspace. This lets us work with the topology simplex by simplex, in a sense made clear by the following fact.

Proposition 46. *A subset of $|S|$ is closed if and only if its intersection with every simplex of S is closed.*

Proof. [Adh16, Proposition 6.2.18]. □

It is instructive to look at a straightforward, but useful, consequence.

Proposition 47. *Let S' be a subcomplex of S . Then in $|S|$ we have $|S'|$ is closed.*

Proof. Since any simplex is closed in itself, and, by Definition 40, the intersection with any other simplex gives a face, which is closed, any simplex is a closed set in $|S|$. Then, since there are a finite number of simplices in S , there are a finite number in S' , and a finite union of closed sets are closed. □

This lets us get at the topology of $|S|$ in terms of the simplices in a rudimentary way, but we really want to describe open and closed *neighbourhoods* of points and subcomplexes in terms of the combinatorial structure of the complex. To that end we make the following definitions.

Definition 48. *Let $p \in |S|$ then*

$$\overline{\star}p = \{D \in S \mid \text{there exists } E \in S \text{ such that } D \subseteq E \text{ and } p \in E\}.$$

Proposition 49. *Let S be a simplicial complex and D and F be simplices of S with $F \subseteq D$. Suppose $d \in \text{int}D$ and $f \in F$. Then $\overline{\star}d \subseteq \overline{\star}f$.*

Proof. Suppose $G \in S$ and $G \in \overline{\star}d$. Then there exists $E \in S$ such that $G \subseteq E$ and $d \in E$. Now $E \cap D$ must be a common face, and since $d \in \text{int}D$, this must in fact be all of D , and so $D \subseteq E$. But then $f \in F \subseteq D \subseteq E$ and so $G \in \overline{\star}f$. □

Definition 50. *Let S' be a subcomplex of S . Then*

$$\overline{\text{str}}S' = \{D \in S \mid \text{there exists } E \in S \text{ such that } E \subseteq D \text{ and } E \in S'\}.$$

Proposition 51. *Let $p \in |S|$; let S' be a subcomplex of S . Then $\overline{\star}p$ and $\overline{\text{str}}S'$ are both subcomplexes of S .*

Proof. Suppose a simplex $D \in \overline{\star}p$ then there exists $E \in S$ such that $D \subseteq E$ and $p \in E$. Now for all faces D' of D , we have $D' \subseteq D \subseteq E$ and hence $D' \in \overline{\star}p$ as required. The proof for $\overline{\text{str}}S'$ is the same, *mutatis mutandis* (in particular, the condition that $p \in E$ must be exchanged with $E \in S'$). □

Definition 52. Let $p \in |S|$ then

$$\star p = \bigcup_{\substack{D \in S \\ x \in D}} \text{int} D$$

Definition 53. Let S' be a subcomplex of S . Then

$$\text{str} S' = \bigcup_{\substack{D \in S \\ D \cap |S'| \neq \emptyset}} \text{int} D$$

Proposition 54. Let $p \in |S|$ then $\star p$ is an open neighbourhood of p .

Proof. First, we check that $p \in \star p$. By Proposition 44, we have a unique $D \in S$ such that $p \in \text{int} D$. Since $\text{int} D \subseteq D$, it follows that $p \in D$ and hence $D \in \overline{\star p}$. Then $p \in \text{int} D \subseteq \star p$ by Definition 52.

Now, we must check that $\star p$ is open. We will show that its complement is closed by showing that it is the underlying space of a subcomplex. Suppose that p' is not contained in the simplicial interior of a simplex which contains p ; then p' is contained in a simplex which does not contain p . Conversely, a simplex which does not contain p does not contain p in its interior. Hence, the complement of $\star p$ is the union of the simplices which do not contain p . This is a subcomplex because if p is not in a simplex, then it is not in any subset, in particular, it is not in any face. Then Proposition 47 implies that this is closed, and so $\star p$ is open. \square

Proposition 55. Let S' be a subcomplex of S . Then

$$\text{str} S' = \bigcup_{p \in |S'|} \star p.$$

Proof. If $p \in |S'|$ and $p \in D$ for some simplex D . Then $D \cap |S'| \neq \emptyset$ and so $\star p \subseteq \text{str} S'$. Conversely, if $D \cap |S'| \neq \emptyset$ then there exists $p \in |S'|$ such that $p \in D$, meaning that $\text{str} S' \subseteq \star p$. The required result follows. \square

The above lead to the following corollary.

Proposition 56. Let S' be a subcomplex of S and $p \in |S|$. Then

- $\star p$ is an open neighbourhood of p ,
- $|\overline{\star p}|$ is a closed neighbourhood of p ;

- $strS'$ is an open neighbourhood of S' , and
- $|\overline{strS'}|$ is a closed neighbourhood of S' .

Proof. The first claim is exactly Proposition 54; then Proposition 55 yields the third because a union of open sets is open. By Proposition 51 $\overline{\star}p$ and $\overline{strS'}$ are both closed; it suffices to show that they are neighbourhoods. We will show that they contain $\star p$ and $strS'$ respectively.

In the former case, suppose $p' \in \star p$. Then p' is in the interior of a simplex which contains p . This simplex will then be in $\overline{\star}p$. The case for S' is similar. \square

2.4.5 Subdivisions

A crucial aspect of simplicial complexes is that they can be subdivided so as to approximate a space more and more finely.

Definition 57. Let S_0 be a simplicial complex. We say that a complex S_1 is a **subdivision** of S_0 if and only if

- every $D \in S_0$ is the union of a finite collection of simplices D_1, \dots, D_n of simplices of S_1 ,
- every $D \in S_1$ is a subset of some simplex of S_0 , and
- $|S'| = |S|$.

There is of course some redundancy in this definition (see [Hud69, Lemma 1.2]), but it is convenient to state all the properties which we rely on. The following are key properties of subdivisions.

Proposition 58. Let S_0 be a simplicial complex and S_1 be a subdivision of S_0 . Then for all $p \in |S_0|$ we have that $|\overline{\star}p|$ taken in S_1 is a subset of $|\overline{\star}p|$ taken in S_0 .

Proof. Suppose that in S_1 we have $D_1 \in \overline{\star}p$. Then there exists $E_1 \in S_1$ such that $p \in E_1$ and $D_1 \subseteq E_1$. Now since S_1 is a subdivision of S_0 , there is a simplex $E_0 \in S_0$ with $E_1 \subseteq E_0$ and hence $D_1 \subseteq E_1 \subseteq E_0$. Now, since $p \in E_1$, we have $p \in E_0$ and hence $E_0 \in \overline{\star}p$ in S_0 . Hence every simplex of $\overline{\star}p$ in S_1 is contained in some simplex of $\overline{\star}p$ in S_0 and since the underlying space of a simplicial complex is the union of the simplices, this establishes the claim. \square

Proposition 59. *If S'_0 is a subcomplex of S_0 and S_1 is a subdivision of S_0 , then the collection of simplices of S_1 which are contained in $|S'_0|$ form a subdivision of S_0 .*

Proof. [Hud69, Lemma 1.3]. □

This is sometimes called the *induced subdivision* of S'_0 .

Proposition 60. *If S'_0 is a subcomplex of S_0 and S'_1 is a subdivision of S'_0 , then there is a subdivision of S_0 such that the induced subdivision on S'_0 is S'_1 .*

Proof. [Hud69, Lemma 1.3]. □

This is also called the *induced subdivision*; context always adjudicates between the two meanings of this expression, according to whether the complex subdivided is a subcomplex of the other, or *vice versa*.

Proposition 61. *If S_0 is a simplicial complex and S_1, \dots, S_n are subdivisions of S_0 then there exists a subdivision S which is a common subdivision of all the subdivisions S_1, \dots, S_n .*

Proof. The result proved by Hudson [Hud69, Corollary 1.6] is much stronger; the result we state here is sufficient for our purposes, and remains true for more abstract notions of simplicial complex. □

Remark. The reader should note that the simplicial complexes for which Proposition 61 provides a common subdivision are all subdivisions of a given complex. It is important not to believe that simplicial complexes with homeomorphic underlying spaces always have a common subdivision: that is the famously false *Hauptvermutung* of combinatorial topology; see [RCS⁺13].

A crucial fact about simplicial complexes is that we can always subdivide them so as to have a better and better combinatorial model of the underlying space. In particular, we can do this in a way which allows a wonderful degree of control.

Proposition 62. *Given a selection of a point in the simplicial interior of each simplex of S_0 , we can produce a subdivision S_1 of S_0 with the property that an n -simplex of S_0 is given by a sequence p_0, p_1, \dots, p_n of selected points, with p_i being the selected point of an i -simplex of S_0 which contains the simplices corresponding to the previous points as faces.*

Proof. [Hud69, p. 7-9]. □

The subdivision described above is called the *stellar subdivision* for the selection of points.

One consequence of this which we can state immediately allows us to model a particular neighbourhood of a point by using a star as defined in the previous section.

Proposition 63. *Let S_0 be a simplicial complex, $p \in |S_0|$ and N be a neighbourhood of p in $|S_0|$. Then there exists a subdivision S of S_0 such that in S we have $|\overline{\star}p| \subseteq N$.*

Proof. Any neighbourhood in S_0 is the intersection of a neighbourhood from the ambient space \mathbb{R}^n intersected with $|S_0|$. Since \mathbb{R}^n has a convex basis, it suffices to examine the case where the neighbourhood intersected with is convex. Let us call this convex neighbourhood of p in \mathbb{R}^n $N_{\mathbb{R}}$.

First, we perform a stellar subdivision with p as a specified point (p must be in the interior of some simplex by Proposition 44). We obtain a subdivision S_1 with $p \in S_1$, that is, with p a 0-simplex of S_1 .

Now we make another selection of points, this time taking care that if the simplicial interior of a simplex of S_1 has non-empty intersection with N , we choose a point in this intersection. Now we take the stellar subdivision of S_1 given this selection of points; call this subdivision S_2 .

Now, if $D_2 \in S_2$ has the property that $p \in D_2$, then there must be a simplex $D_1 \in S_1$ such that $p \in D_1$ and $D_2 \subseteq D_1$, because S_2 is a subdivision of S_1 . Hence by Proposition 62, the vertices of D_2 must be selected points of D_1 . Since $p \in D_1$, the intersection of D_1 with N is not empty. Therefore, all vertices of D_2 must be in N . (If a vertex was the selected one in the simplicial interior of D_1 , this is immediate. Otherwise, it must lie on a face of D_1 , and we can repeat the argument with that face.) Now, since all of the vertices lie within N , they all lie within $N_{\mathbb{R}}$. Since $N_{\mathbb{R}}$ is convex, we have $D_2 \subseteq N_{\mathbb{R}}$; since $D_2 \in S_2$ which is a subdivision of S_0 , we have $D_2 \subseteq |S_0|$. Hence $D_2 \subseteq N = |S_0| \cap N_{\mathbb{R}}$. At the start of this paragraph, all we assumed about D_2 was that p lay inside it, so this conclusion holds for all simplices of S_2 containing p ; hence we can take $S = S_2$ and we obtain that in S we have $|\overline{\star}p| \subseteq N$. \square

2.4.6 Full Subcomplexes and Extending Maps

A subcomplex S' of S is called **full** if and only if the intersection of any simplex in S with $|S'|$ is a simplex of S . This may fail to occur, for example, if we let S be a single triangle, and take S' to be two of its sides (together with their vertices). In some sense,

being full in S means not being coiled up too tightly around the simplices of S .

Full subcomplexes are extremely useful because they allow us to extend continuous maps into a neighbourhood around them.

Proposition 64. *Let S' be a full subcomplex of S , A be any neighbourhood space, and let $f : |S'| \rightarrow A$. Then there is a map $f_\star : strS' \rightarrow A$ such that $f_\star \upharpoonright |S'| = f$ and for all simplices D of S we have $f_\star[D \cap strS'] \subseteq f[D \cap |S'|]$ and $f_\star[intD \cap strS'] \subseteq f[int(D \cap |S'|)]$.*

Proof. We first construct a map $\star_{S'} : strS' \rightarrow S'$ such that for all $D \in S$ we have $\star_{S'}[D \cap strS'] \subseteq D \cap |S'|$ and $\star_{S'}[intD \cap strS'] \subseteq intD \cap |S'|$. Once this is done we can simply let $f_\star = f \circ \star_{S'}$ and the properties we require of f_\star will follow from those of $\star_{S'}$.

We now turn to defining $\star_{S'}(p)$ for $p \in strS'$. First note that since $p \in strS'$ we know that p is in the simplicial interior of a simplex F such that $F \cap |S'| \neq \emptyset$. Then $F \cap |S'|$ must be a face of F because S' is full. Our idea is to build $\star_{S'}$ piecemeal by defining its restriction to F . If $F \cap S' = F$ then we can simply take the identity function. Otherwise let $F_0 = F \cap |S'|$ and F_1 be the opposite face. Then p , being in the interior of F , is an element of $F - F_1$, and we define $\star_{S'} \upharpoonright F - F_1 = \nabla_{F, F \cap |S'|}$.

Now we ought to check that this is a consistent definition. Suppose two such simplices F and F' intersect in a common face E . Then $E \cap |S'| = F \cap F' \cap |S'|$, and so Observation 38 implies that the function we have given as the restriction of $\star_{S'}$ to E agrees with the restrictions from F and F' .

Now we turn to proving the two properties we require of $\star_{S'}$.

Let $D \in S$; we want to show that $\star_{S'}[D \cap strS'] \subseteq D \cap |S'|$. Suppose $p \in D \cap strS'$. Then there is a simplex $D' \in S$ such that $p \in intD'$ and $D' \cap |S'| \neq \emptyset$. Since S' is full, $D' \cap |S'|$ is a face of S' .

Now since $p \in D$ and $p \in \text{int}D'$, we must have $D' \subseteq D$ by Proposition 41. Then

$$\begin{aligned} \star_{S'}(p) &= \nabla_{D', D' \cap |S'|}(p) \\ &\in D' \cap |S'| \text{ by the definition of } \nabla_{D', D' \cap |S'|} \\ &\subseteq D \cap |S'| \end{aligned}$$

as required.

Finally, we need to check that for all simplices $D \in S$ we have

$$\star_{S'}[\text{int}D \cap \text{str}S'] \subseteq \text{int}D \cap |S'|.$$

This is a consequence of Observation 39. With this, thanks to the considerations of the first paragraph, we conclude the proof. \square

This means we can extend any map $f : |S'| \rightarrow A$, where A is an arbitrary neighbourhood space, to a map $f_\star : \text{str}S' \rightarrow A$ whose restriction to $|S'|$ is f by setting $f_\star = f \circ \star$.

This is a nice property, but one might worry that full subcomplexes might be rare. If we are prepared to subdivide, this is not so.

Proposition 65. *Let S'_0 be a subcomplex of S_0 , S_1 be a stellar subdivision of S_0 and S'_1 be the induced subcomplex of S_1 by S'_0 . Then S'_1 is full in S_1 .*

Proof. [Hud69, Lemma 2.5] \square

Chapter 3

Paths and Homotopies in Alexandroff Spaces

3.1 Introduction

In this chapter, we ask whether Alexandroff neighbourhood spaces are a reasonable choice for the task we need them to do. We hope that a certain space will have the property that paths through it correspond to sequences of steps of computations. Amongst neighbourhood spaces, the Alexandroff spaces encode “combinatorial” information, since they are determined by their underlying cospecialization relation. In this chapter, we prove that a combinatorial path can be extracted from a continuous one, and that higher dimensional homotopies are also essentially given by combinatorial information.

Thinking of an Alexandroff space as a relation, we sometimes picture the situation by imagining that its points are colours, and so a map from a unit hypercube into an Alexandroff space is a colouring of the hypercube. The relation then encodes “rules of good taste” for these colourings: if a region of one colour touches another, then the colour of the first must be related to that of the second.

Because we are interested in paths through Alexandroff spaces, and homotopies between them, we wonder what constraints continuity puts on maps from the unit interval, square, etc. into an Alexandroff space.

The indiscrete spaces are Alexandroff, and we know that all functions into an indiscrete space are continuous; this includes “infinite” colourings, where some point of the domain has no constant neighbourhood, like fractals, hyperbolic designs, and monsters indescribable but for Choice. However, indiscrete spaces are also “small” in the sense that all points are arbitrarily close to each other, and so all of these maps can be approximated “arbitrarily well” by a constant colouring, or a finite approximation which preserves a specified amount of detail.

All Alexandroff spaces are locally “small” in a similar way: each point has a neighbourhood of points which are each arbitrarily close to it. By continuity, this means that each point of the domain has a neighbourhood around it in which only colours close to its colour are used. Therefore, if there is any “infinite” behaviour at some point, though no neighbourhood is constant, there must be a neighbourhood which is close to being constant. By the compactness of all the unit hypercubes, only a finite number of such neighbourhoods suffice to cover the domain.

A picture of a general path, then, is something which passes through finitely many small neighbourhoods, but with infinite scintillations, like the motion of a star over the night. Can we glue together the constant maps to produce an underlying finite path? What about in higher dimensions?

The problem is this: suppose in our cover by special neighbourhoods, two overlap. The image of each of these is close to a constant map, but these two constant colours may not be close to each other in the target! Instead, the original map might have used a mediating point, close to both, a “direction” by which to exit the neighbourhood it was in, and it may have vacillated infinitely about which direction to take.

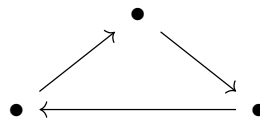
If we can choose a border between our two neighbourhoods, we can look at the original map along that border to work out how to get from one colour to the other. If this colouring of the border is infinite, we can repeat the process inductively, getting a finite colouring of the border. If we can control the process enough that the colourings on either side of the border are close to the colouring we get for the boundary, then we can glue the two regions together.

But how can we choose a well-behaved border, or control how it is re-coloured,

if our neighbourhoods are arbitrarily shaped? It is better to put some structure on the space which allows us to control the shapes of the regions we have to deal with. The right tools for the task are simplicial complexes, described in Section 2.4.

In the one-dimensional case for certain topological Alexandroff spaces, this question is resolved by [MPCR11, Theorem 2]. The proof we present here can be viewed as a generalization of their technique, which we roughly sketch here.

The topological spaces must satisfy a condition such that the cospecialization relation is a partial order¹, not merely a preorder. In the context of topology, this is not a significant demand: there is always map from a space failing to have this property to one which does which preserves the relevant information about which paths are homotopic to which [McC66, Theorem 4]. This is no longer true for neighbourhood spaces. Indeed the space



has two homotopically distinct paths from the point on the left back to itself, but these are not distinguishable any longer in any image of it in a space given by a poset.

Given a map from the interval into a space given by a preorder, the strategy in [MPCR11] is to cover the source with open intervals whose images are contained in the neighbourhood of a single point. This gives an open cover of the interval, so using compactness yields a finite sub-cover. It can be arranged that no more than two of these covering intervals overlap at any given point. One then selects a point in each of the overlaps, and replaces the value of the original function on the overlap by its value on the selected point, and in between the overlaps by its value on the points whose neighbourhoods are part of the finite sub-cover. The resulting map is pointwise comparable to the original one, and so there is a homotopy between them [MPCR11, Corollary 2].

The proof given here for the general case of Alexandroff neighbourhood spaces in all dimensions is a generalization of the one in [MPCR11] in the sense that one finds an open cover of the interval by well behaved sets whose images are contained in the

¹This is the separation property T_0 .

smallest neighbourhood of points, applies compactness, and looks at the value of the function on the overlaps to work out how to glue constant functions together.

We generalize in two ways. First, we cannot make use of pointwise comparability of functions, since it no longer plays the same role. This is why we use a slightly subtler notion, defined in the sequel, which we call relaxation. Second since we work in all dimensions, we require a more general notion of well behaved regions which overlap in a more controlled manner. This is why we use simplicial complexes. The glueing procedure for this higher-dimensional data is more involved than that required to glue two functions defined on intervals based on a point where they overlap.

3.2 Tools from Simplicial Complexes

In order to carry out our programme above, whereby we show every path and homotopy is well approximated by a simple one, we need a way of singling out when a map is simple. The idea is to consider those maps which are constant on the simplicial interior of a simplex, so that if we think of them as colourings, we have essentially one colour per simplex.

Definition 66. *If S is a simplicial complex, X is a subspace of $|S|$ and B is a neighbourhood space, we say that a function $f : |X| \rightarrow B$ is **S -constant** if and only if for all $D \in S$, the restriction of f to $\text{int}D \cap X$ is constant.*

The main result we are aiming towards at the end of this chapter, Theorem 84, is proved by taking finer and finer subdivisions of a simplicial complex. Therefore, we are interested in how any property we define behaves with respect to subdivisions.

Proposition 67. *Let S_0 be a simplicial complex, X a subspace of $|S_0|$, B be a neighbourhood space, and $f : X \rightarrow B$ be an S_0 -constant map. Then if S_1 is a subdivision of S , we have that f is S_1 -constant.*

Proof. Let $D \in S_1$. Since every simplex of S_0 is a union of simplices of S_1 , there is a simplex $E \in S_0$ such that $\text{int}D \subseteq \text{int}E$, and hence f is constant on the interior of D . \square

We want to use the simplicial complex structure to glue S -constant maps together, and to do so we often need to extend them onto the star of a full subcomplex. We ought to check, therefore, that this operation preserves S -constancy.

Proposition 68. *Let X be a full subcomplex of S , B be a neighbourhood space and $f : |X| \rightarrow B$ be an S -constant map. Then f_\star is S -constant.*

Proof. Let $D \in S$. Then $f_\star(\text{int}D \cap \text{str}X) \subseteq f(\text{int}(D \cap |X|))$ by Proposition 64. Now since X is full, $D \cap |X|$ is a simplex of X , and f is constant on its interior. That means that $f(\text{int}(D \cap |X|))$ is a singleton and hence $f_\star(\text{int}D \cap \text{str}X)$, since it cannot be empty, is a singleton too: f_\star is constant on $\text{int}D \cap \text{str}X$. \square

We are interested in ways in which one map can be “close” to another. The weakest sensible notion of closeness might be to consider two maps close if they have the same image—in terms of the metaphor of colourings, if they use the same colours. This, however, does not demand very much about *where* the original map put those colours. Since our source spaces are simplicial complexes, we can demand a more refined version of this, where each simplex can only be coloured using colours which it contained before. By using finer and finer subdivisions, one can make the “close” approximation to a map closer and closer.

Definition 69. *Given a simplicial complex S , a neighbourhood space B , a subspace X of $|S|$ and maps $f : |S| \rightarrow B$ and $g : X \rightarrow B$, say that g S -**accommodates**² f if and only if for every $D \in S$ we have $g[D \cap X] \subseteq f[D]$.*

We check to see how this behaves with respect to subdivisions.

Proposition 70. *Let S_0 be a simplicial complex, S_1 be a subdivision of S_0 , B be a neighbourhood space, X be a subspace of $|S_0|$ and $f : |S_0| \rightarrow B$ and $g : X \rightarrow B$ be maps. Suppose g S_1 -accommodates f . Then g S_0 -accommodates f .*

Proof. Let $D \in S_0$. Then D is the union of finitely many simplices D_0, \dots, D_n of S_1 . We have

²All versions of the verb “to accommodate” refer to this concept; we often want a slightly different form for readability. For example we often say that something is a map S -accommodating another map.

$$\begin{aligned}
g[D \cap X] &= g \left[\left(\bigcup_{0 \leq i \leq n} D_i \right) \cap X \right] \\
&= g \left[\bigcup_{0 \leq i \leq n} (D_i \cap X) \right] \\
&= \bigcup_{0 \leq i \leq n} g[D_i \cap X] \\
&\subseteq \bigcup_{0 \leq i \leq n} f[D_i] && \text{since } g \text{ } S_1\text{-accommodates } f \\
&= f \left[\bigcup_{0 \leq i \leq n} D_i \right] \\
&= f[D]
\end{aligned}$$

as required. □

We now check an apparently obvious property, namely that a map which accommodates another cannot move “new” colours into a subcomplex which the map it accommodates did not already put there.

Proposition 71. *Given a simplicial complex S , a subspace X and a subcomplex Y , a neighbourhood space B and maps $f : |S| \rightarrow B$ and $h : X \rightarrow B$ such that h S -accommodates f , then $h[|Y| \cap X] \subseteq f[|Y|]$.*

Proof. We have

$$\begin{aligned}
h[|Y| \cap X] &= \bigcup_{D \in Y} h[D \cap X] \\
&\subseteq \bigcup_{D \in Y} f[D] \\
&= f[|Y|]
\end{aligned}$$

as required. □

It is worth pausing to reflect that, although the proof above is straightforward, it depends on Y being a subcomplex more strongly than might be supposed. Indeed, suppose we try to use $\star p$ for some point $p \in \bigcup X$. Then it may fail to be the case that

$h[\star p \cap X] \subseteq f[\star p]$ because h is free to use colours from the boundary of $\star p$.

As above, since we are interested in extending maps to the stars of full subcomplexes, we check that this procedure behaves well with respect to accommodation.

Proposition 72. *Given a simplicial complex S_0 , a subdivision S_1 of S_0 , a neighbourhood space B , a full subcomplex X of S_1 and maps $f : |S| \rightarrow B$ and $g : |X| \rightarrow B$ such that g S_0 -accommodates f , we have g_\star (taken in S_1) S_0 -accommodates f .*

Proof. Let $D \in S_0$. Then D is the union of finitely many simplices D_0, \dots, D_n of S_1 . We have

$$\begin{aligned}
 g_\star[D \cap \text{str}X] &= g_\star \left[\left(\bigcup_{0 \leq i \leq n} D_i \right) \cap \text{str}X \right] \\
 &= g_\star \left[\left(\bigcup_{0 \leq i \leq n} D_i \cap \text{str}X \right) \right] \\
 &= \bigcup_{0 \leq i \leq n} g_\star[D_i \cap \text{str}X] \\
 &\subseteq \bigcup_{0 \leq i \leq n} g[D_i \cap |X|] && \text{by Proposition 64} \\
 &= g \left[\left(\bigcup_{0 \leq i \leq n} D_i \right) \cap |X| \right] \\
 &= g[D \cap |X|] \\
 &\subseteq f[D] && \text{since } g \text{ } S_0\text{-accommodates } f.
 \end{aligned}$$

This is what we wanted to show. □

Note that, despite their similar proofs, we cannot make use of Proposition 70 in proving Proposition 72.

3.2.1 Alexandroff Spaces

We now turn to the fact that the maps we want to compare have an Alexandroff space as their target. What notion of closeness of maps is natural in this setting? In an Alexandroff space, the relation underlying the space tells us when one point is arbitrarily close to another. This notion of closeness interacts well with continuity: if x is

close to y , then a map which is constantly x on some open set and constantly y everywhere else is continuous. Thinking of the points of the Alexandroff space as colours, this means that if x is close to y , then a region of colour x can touch a region of colour y .

We would like to extend this idea to arbitrary colourings, and say that if a colouring g is close to a colouring f , then we can replace g by f on some open set.

At first, one might have thought that a pointwise version of closeness would work, but this is not enough. In general, we need to examine the *local behaviour* of the functions at a point, not just their values.

Definition 73. Given spaces A and B , subspaces W and X of A and maps $f : W \rightarrow B$ and $g : X \rightarrow B$, say that g **relaxes** f if and only if for every point $x \in X$ there is a neighbourhood R_x of x in A such that $f[R_x \cap W] \subseteq \downarrow g(x)$.

It is helpful to have some notation to state the special property of a map which relaxes another.

Definition 74. Let A and B be spaces, $f : A \rightarrow B$, X a subspace of A , and $g : X \rightarrow B$. Then

$$if(X, g; f)(a) = \begin{cases} g(a) & \text{if } a \in X \\ f(a) & \text{otherwise} \end{cases}$$

is a function (which is not necessarily continuous).

Observe that for any subset S of A , we have $if(X, g; f)[S] = f[S - X] \cup g[S \cap X]$.

Proposition 75. Consider a neighbourhood space A and an Alexandroff neighbourhood space B , an open subspace X of A , and maps $f : A \rightarrow B$ and $g : X \rightarrow B$ such that f relaxes g . Then the function $if(X, g; f)$ is continuous.

Proof. Let $a \in A$. We wish to show that $if(X, g; f)^{\leftarrow}(\downarrow if(X, g; f)(a))$ is a neighbourhood of a ; it suffices to show that it contains such a neighbourhood. By monotonicity of inverse image, we need only find a neighbourhood whose image is contained in $\downarrow if(X, g; f)(a)$.

Suppose $a \in X$. By continuity of g , we have a neighbourhood G_a of a such that $g[G_a] \subseteq \downarrow g(a)$. Since X is open, it is also a neighbourhood of a , and hence $X \cap G_a$

is a neighbourhood of a as well. Since $if(X, g; f)$ behaves like g on X , and hence on $X \cap G_a$, we have

$$\begin{aligned} if(X, g; f)[X \cap G_a] &= g[X \cap G_a] \\ &\subseteq g[G_a] \\ &\subseteq \downarrow g(a) \\ &= \downarrow if(X, g; f)(a) \end{aligned}$$

as required. Note that so far we have not needed the assumption that f relaxes g at all.

Otherwise, $a \in A - X$. By continuity of f , we have a neighbourhood F_a of a such that $f[F_a \cap X] \subseteq \downarrow f(a)$. Now, however, there may be no neighbourhood of a on which $if(X, g; f)$ behaves like f , because a might be on the boundary of X . We must do something different to the previous case. By assumption, f relaxes g , so there is a neighbourhood R_a of a such that $g[R_a \cap X] \subseteq \downarrow f(a)$. Hence $F_a \cap R_a$ is a neighbourhood of a and we have:

$$\begin{aligned} if(X, g; f)[F_a \cap R_a] &= f[(F_a \cap R_a) - X] \cup g[(F_a \cap R_a) \cap X] \\ &\subseteq f[F_a] \quad \cup g[R_a \cap X] \\ &\subseteq \downarrow f(a) \quad \cup \downarrow f(a) \\ &= \downarrow f(a) \\ &= \downarrow if(X, g; f)(a) \end{aligned}$$

as required. □

The map $if(X, g; f)$ of the previous proposition will again relax g .

Proposition 76. *Consider a neighbourhood space A and an Alexandroff space B , an open subspace X of A , and maps $g : A \rightarrow B$ and $f : X \rightarrow B$ such that g relaxes f . Then $if(X, f; g)$ relaxes f*

Proof. Consider a point $a \in A$. Since g relaxes f we have a neighbourhood R_x such that $f[R_x \cap X] \subseteq \downarrow g(x)$. By continuity of g , there is a neighbourhood G_x of x such that $g[G_x] \subseteq \downarrow g(x)$. Then $R_x \cap G_x$ is a neighbourhood of x and we have

$$\begin{aligned}
if(X, f; g)(R_x \cap G_x) &= g[(R_x \cap G_x) - X] \cup f[(R_x \cap G_x) \cap X] \\
&\subseteq g[G_x] \quad \cup f[R_x \cap X] \\
&\subseteq \downarrow g(p) \quad \cup \downarrow g(p) \\
&= \downarrow g(p)
\end{aligned}$$

as required. □

3.3 Constancy, Accommodation, and Relaxation

We can now begin to think about the relationship between relaxation and S -constancy, the property of functions defined at the end of Section 3.2. First, we observe that the operation $if(X, -, -)$ preserves constancy whenever X is the star of a subcomplex.

Proposition 77. *Let S be a simplicial complex, S' a subcomplex of S , $X = strS'$, A an Alexandroff space, $f : |S| \rightarrow A$ an S -constant map and $g : X \rightarrow A$ an S -constant map such that f relaxes g . Then $if(X, g; f)$ is an S -constant map.*

Proof. It is a map by proposition 75, as X is open because it is $strS'$. To see that $if(X, g; f)$ is S -constant, consider the simplicial interior of a simplex of S . Either it is completely contained in X or it is completely outside X since, by Definition 53, $X = strS'$ is the union of the simplicial interiors of all simplices with which it has non-empty intersection. In the first case, its value is that given by g , in the second, by f . In either case it is a constant. □

It turns out that we need to strengthen the notion of relaxation to make it interact better with the simplicial complex. In particular, we want to be able to glue maps together.

Definition 78. *Given a simplicial complex S and a neighbourhood space B , subcomplexes W and X of S and maps $f : W \rightarrow B$ and $g : X \rightarrow B$, say that g **S -relaxes** f if and only if for every point $x \in X$ $f[|\overline{\star}x| \cap W] \subseteq \downarrow g(x)$.*

We ought to check how S -relaxation interacts with subdivision.

Proposition 79. *Let S_0 be a simplicial complex, B be a neighbourhood space, W and X be subcomplexes of S_0 , and $f : W \rightarrow B$ and $g : X \rightarrow B$ such that g S -relaxes f . Let S_1 be a refinement of S_0 , then g S_1 -relaxes f .*

Proof. Since S_1 is a subdivision of S_0 , we have $|\overline{\star}p|$ in S_1 is a subset of $|\overline{\star}p|$ in S_0 . Then $f[|\overline{\star}x| \cap W]$ taken in S_1 is a subset of $f[|\overline{\star}x| \cap W]$ in S_0 , which is a subset of $\downarrow g(x)$ since g S -relaxes f . \square

We check that an analogue of Proposition 76 holds.

Proposition 80. *Let S be a simplicial complex, W, X and Y be subspaces of $|S|$, B be a neighbourhood space, and $f : W \rightarrow B$, $h : X \rightarrow B$ and $g : Y \rightarrow B$ such that g and h both S -relax f . Then $if(X, h; g) : X \cup Y \rightarrow B$ S -relaxes f .*

Proof. Let $x \in X \cup Y$. Suppose $x \in X$. Then $if(X, h; g)(x) = h(x)$ and then since h S -relaxes f we have $f[|\overline{\star}x| \cap W] \subseteq \downarrow h(x) = \downarrow if(X, h; g)(x)$. The case when $x \in Y$ is analogous. \square

We also check that this is well behaved with respect to the star operation.

Proposition 81. *Let S be a simplicial complex, X a full subcomplex of S , B be a neighbourhood space, and $f : |S| \rightarrow B$ and $g : |X| \rightarrow B$ be maps such that g S -relaxes f . Then g_\star S -relaxes f .*

Proof. Let $x \in strX$. We wish to show that $f[|\overline{\star}x|] \subseteq \downarrow g_\star(x)$. By Definition 53 there is some simplex D such that $x \in intD$ and $D \cap |X| \neq \emptyset$. Since X is full, $D \cap |X|$ is a simplex of X . By Proposition 64 there is a point $y \in D \cap |X|$ such that $g_\star(x) = g(y)$. Now Proposition 49 implies that $|\overline{\star}x| \subseteq |\overline{\star}y|$ and since g S -relaxes f we have

$$\begin{aligned} f[|\overline{\star}x|] &\subseteq f[|\overline{\star}y|] \\ &\subseteq \downarrow g(y) \\ &= \downarrow g_\star(x) \end{aligned}$$

as required. \square

The notion of accommodation interacts well with the operation of replacing one function by another on a subset of their domains.

Proposition 82. *Let S be a simplicial complex, B be a neighbourhood space, $f : |S| \rightarrow B$, $g : |S| \rightarrow B$, X be a subspace of $|S|$, and $h : X \rightarrow B$. Suppose that both g and h S -accommodate f . Then $if(X, h; g)$ S -accommodates f .*

Proof. Suppose $D \in S$. Then

$$\begin{aligned}
if(X, h; g)[D] &= g[D - X] \cup h[D \cap X] \\
&\subseteq g[D] \cup h[D] \\
&\subseteq f[D]
\end{aligned}$$

as required. □

Accommodation also works well with respect to relaxation, in the following way.

Proposition 83. *Consider a simplicial complex S , an Alexandroff space B , a map $f : S \rightarrow B$, a map $g : S \rightarrow B$ S -relaxing f , X a subspace of S , and a map $h : X \rightarrow B$ S -accommodating f . Then g S -relaxes h .*

Proof. Consider a point $p \in |X|$. We want to prove that $h[\star p \cap X] \subseteq \downarrow g(p)$

Since h S -accommodates f , we make use of proposition 71 to see that $h[\overline{\star p} \cap X] \subseteq f[\overline{\star p}] \subseteq \downarrow g(p)$ as required. □

3.4 Combinatorial Approximation of a Map

We now have all the necessary tools in place to make the idea from the first section rigorous.

Theorem 84. *Given a simplicial complex S_0 , a subcomplex X_0 of S_0 whose underlying space is compact, an Alexandroff space A and a map $f : |S_0| \rightarrow A$, there is a subdivision S_* of S_0 inducing a subdivision X_* of X_0 and a map $f_* : |X_*| \rightarrow A$, such that f_* S_* -relaxes f , S -accommodates f , and is X_* -constant.*

Proof. Before we begin, the reader may wish to take careful note of the fact that accommodation is treated differently from the other properties in the statement of the lemma. The map f_* we obtain at the end has *weak* properties, in the sense that we have two simplicial complexes, S_0 and its subcomplex S_* , and we get properties of f_* stated with respect to one or the other of these complexes in such a way that Propositions 67, 70 and 79 do not tell us anything with respect to the other complex. Since Proposition 70, dealing with accommodation, works in the opposite direction to the other two with respect to subdivisions, accommodation will constantly be handled differently from the other two properties throughout this proof.

The proof is by induction on the maximum dimension of X_0 . If X_0 is a point, then the result holds trivially since f already has the required properties with respect to the given simplicial complex. Otherwise, we proceed in two parts, where the induction hypothesis will be used only in the second part.

First, by continuity of f , each point x in $|X_0|$ has a neighbourhood N_x in $|S|$ such that $f[N_x] \subseteq \downarrow f(x)$. By Proposition 63 we have a subdivision S_x of S_0 such that the underlying space of $\overline{\star}x$ in S_x is contained in N_x . We intersect this with X_0 to obtain a subcomplex $M_0(x)$ with this property. Note that the map f_x defined on $|M_0(x)|$ which is constantly $f(x)$ is $M_0(x)$ -constant, S_x -relaxes f , and S_x -accommodates f . By Proposition 70, we know that f_x S_0 -accommodates f .

Proposition 56 implies that for all x as above, $|M_0(x)| = |\overline{\star}x \cap X_0|$ is a neighbourhood of x in X_0 . The (topological!) interiors of the $M_0(x)$ then form an open cover of X_0 , and since X_0 is compact, let J be a finite collection of points of $|X_0|$ such that the interiors of $\{|M_0(j)| \mid j \in J\}$ cover $|X_0|$. Making use of Proposition 61, let S_1 be the common subdivision of this finite subcollection of the S_x . This induces subdivisions $M_1(j)$ of $M_0(j)$ for each $j \in J$ and a subdivision X_1 of X_0 because of Proposition 59. For $j \in J$, Propositions 67 and 79 tell us that f_j is S_1 -constant and S_1 -relaxes f ; recall from above that all of these maps S -accommodate f . This completes the first part of the step case of our induction, in which the induction hypothesis was not used.

We now find ourselves in the following position. We have a subdivision S_k of S_0 (that is, S_1 , however we allow ourselves some flexibility for reasons which will become clear shortly) inducing a subdivision X_k of X_0 , and a finite subset J of $|X|$ together with subcomplexes $M_k(j)$ for each $j \in J$ whose underlying spaces cover that of X_k . Additionally, every $M_k(j)$ comes with a map $f_j : |M_k(j)| \rightarrow B$ which S_0 -accommodates f , S_k -relaxes f and is $M_k(j)$ -constant. It looks as though the job is done: all we need to do is “glue together” the maps f_j to get a map $|X_k| \rightarrow B$ which S_0 -accommodates f , S_k -relaxes f and is X_k -constant. Unfortunately, these maps do not necessarily agree with each other where they overlap, so we can’t glue them together! However, we will be able to glue them together “up to homotopy” in some sense. Each time we do this, we will have one subcomplex $M_k(j)$ fewer to deal with, but our subdivision S_k will be replaced by a sequence of finer and finer subdivisions. Since J is finite, we need only

do this finitely many times and we will have achieved our aim.³

Let W'_k and Z'_k be the two subcomplexes we are trying to glue together, and f'_W and f'_Z their respective maps. By Proposition 43 we can find subcomplexes $W \subseteq W'$ and $Z \subseteq Z'$ such that $W \cap Z$ is a subcomplex of dimension strictly less than that of X_0 . That means that we can use the induction hypothesis.

But take care! The induction hypothesis applies to *all* simplicial complexes with a subcomplex of this dimension. In particular, we will use S_k , not S_0 , as the simplicial complex which we apply the induction hypothesis to, and $W \cap Z$, rather than X_0 , as its subcomplex. We will, however, use the original function f . This gives us a subdivision S_{k+1} of S_k inducing subdivisions X_{k+1} of X_k and $M_{k+1}(j)$ for all $j \in J$. In particular we will have induced subdivisions W_{k+1} and Z_{k+1} of W_k and Z_k respectively, and a map $f^\cap : |W_{k+1} \cap Z_{k+1}| \rightarrow A$ which is S_{k+1} -constant, and which S_{k+1} -relaxes and S_k -accommodates f . Observe that Propositions 67 and 79 mean that f_W and f_Z are both S_{k+1} -constant and both S_{k+1} -relax f .

In applying the induction hypothesis above, we have performed the following intuitive procedure. We wanted to glue together our two colourings of W and Z , but they might disagree on the (thin, by construction) subcomplex $W \cap Z$. We know that the original function f has some way of getting between the colours used by W on one side, and those used by Z on the other, and we look up what those colours are. Now this original colouring of $W \cap Z$ might be some monstrous, infinite thing, but the induction hypothesis smooths it out. By asking for this smoothing to give us a map which S_k -accommodates f , we ensure that in doing it we do not move the colours used by f so far that they can no longer serve as bridges between the colours used in the simplices of S_k on either side. Our strategy now is to extend this new colouring a little way into W and Z so that we can glue all this data together, and we must check that we have successfully produced data which *can* be glued together.

Perform stellar subdivision so as to obtain a subdivision S_{k+2} of S_{k+1} inducing subdivisions X_{k+2} , $M_{k+2}(j)$, W_{k+2} , and Z_{k+2} as usual. In S_{k+2} , Proposition 65 implies that $W_{k+2} \cap Z_{k+2}$ is full, and so the map $f_\star^\cap : str(W_{k+2} \cap Z_{k+2}) \rightarrow B$ is well defined. Proposition 72 means that this S_{k+1} -accommodates f . Note the fact that this is relative to

³Within the step case of our main induction, the reader will notice that this part of the proof is really performing another induction, one on the size of J . This is straightforward.

S_{k+1} and not S_{k+2} ; this really is what Proposition 72 tells us. Moreover, Proposition 68 means that it is S_{k+2} -constant, and Proposition 81 means that it S_{k+2} -relaxes f .

Now, since $f_{\star}^{\cap} S_{k+1}$ -accommodates f and $f_W S_{k+1}$ -relaxes f , Proposition 83 means that $f_W S_{k+1}$ -relaxes f_{\star}^{\cap} . Then, since Proposition 56 means that $\text{str}(W_{k+2} \cap Z_{k+2})$ is open, by Proposition 75 the function $\text{if}(\text{str}(W_{k+2} \cap Z_{k+2}), f_{\star}^{\cap}; f_W)$ is continuous. Moreover, by Propositions 77 and 80, we know that it is S_{k+2} -constant and that it S_{k+2} -relaxes f . Now since $f_{\star}^{\cap} S_{k+1}$ -accommodates f , Proposition 70 allows us to weaken this and say that $f_{\star}^{\cap} S_0$ -accommodates f . Now since f_W also S_0 -accommodates f , by Proposition 82 we know that $\text{if}(\text{str}(W_{k+2} \cap Z_{k+2}), f_{\star}^{\cap}; f_W) S_0$ accommodates f . By an argument analogous to the preceding paragraph, we find $\text{if}(\text{str}(W_{k+2} \cap Z_{k+2}), f_{\star}^{\cap}; f_W)$ to have the same properties, and since they now agree on their intersection, the open set $\text{str}(W_{k+2} \cap Z_{k+2})$, they can be glued together to form a map from $|W \cup Z| \rightarrow A$ with those properties.

In the course of all this work, the other subcomplexes $M_k(i)$ have been repeatedly subdivided to become $M_{k+2}(i)$, subcomplexes of S_{k+2} . All of our maps f_i S_0 -accommodate f . Propositions 67 and 79 mean that f_i is $M_{k+2}(i)$ -constant and S_{k+2} -relaxes f . We are now back in the position we were in before, but with S_{k+2} instead of S_k . Moreover, we can replace the two subcomplexes which we selected from among the $M_k(i)$ (that is, W' and Z') into one, thus reducing the number of subcomplexes we need to consider. We will have to supply a map to go with it, and we take the one just constructed. After performing the gluing operation a finite number of times, we will have a subdivision S_* of S , and only one subcomplex left, which must cover the whole of X_0 . We can take the unique map associated with it, and the argument above establishes that this map S_0 -accommodates f , S_* -relaxes f and is X_* -constant. Thus, it satisfies the conditions stipulated in the statement we have been proving. \square

Note that this immediately gives us a homotopy from f to f_* . We must define $H : |S_0| \times I \rightarrow A$ in such a way that restricted to the 0 level of the I co-ordinate we recover f , while on the 1 level, we get f_* . Consider the map $\text{if}(|S_0| \times [1, \frac{1}{2}), f \times \mathbb{I}; f_* \times \mathbb{I})$ which manifestly has the right endpoints. Since $|S_0| \times [1, \frac{1}{2})$ is open and f_* relaxes f , Proposition 75 implies that $\text{if}(|S_0| \times [1, \frac{1}{2}), f \times \mathbb{I}; f_* \times \mathbb{I})$ is continuous.

If our original map f was already a homotopy, then we would like to get an analogue of the result but where we *fix the endpoints*. Of course, if the endpoints are

themselves infinite paths and we fix them, then the resulting homotopy f_* will not be S -constant for any simplicial complex S . We can, of course, limit the damage by extending the original values of f just a small way into f_* . The way we do this is to use f_\star with respect to the subdivision B_* of the boundary of the cube in S_* (where we assume that S_* has been sufficiently subdivided so as to ensure that B_* is full). This extends the values which f used on the boundary into $\text{str}B_*$ in S_* . Since this is an open set and f_* relaxes f , we may appeal again to Proposition 75 to argue that $\text{if}(\text{str}B_*, f_\star; f_*)$ is continuous. This proves that if there exists a homotopy with the same endpoints as f , then there exists one which is finite except near the boundary. Because $\text{if}(\text{str}B_*, f_\star; f_*)$ may fail to relax f , we do not get a homotopy between them via the procedure above.⁴ This can be remedied if f the boundary of the cube was already S_0 -constant. In that case it will be S_* -constant by Proposition 67, and then Proposition 75 implies that one can take value of the new map on the interior of the cube, and the original value on the boundary, and the function obtained by so doing will be continuous and S -constant. It will relax f by Proposition 76, and then the argument above implies that it is homotopic to f .

The meaning of all this is that if one wants to study homotopy classes of paths, homotopies between paths, homotopies between homotopies etc. in an Alexandroff space, then one can study finite-homotopy classes of their finite versions. Each of these objects is homotopic to a finite version, and any homotopy between the originals is homotopic to one between the finite versions of its endpoints.

One Dimensional Directed Case

We assume the Alexandroff space is directed where the directed structure is given by the cospecialization relation.

Proposition 85. *Given an Alexandroff space A and a path $p : \mathbb{I} \rightarrow A$, there exists a subdivision \mathbb{I}_* of \mathbb{I} and a path $p^* : \mathbb{I}_* \rightarrow A$ such that p^* is \mathbb{I}_* -constant and relaxes p .*

Proof. Around each point of the interval, we choose, by Proposition 37, a small neighbourhood such that values to the left of the point cospecialize its value, and the value of the point cospecializes and is cospecialized by values of points to the right. The interiors of these neighbourhoods cover \mathbb{I} , so by compactness a finite number of them do so. As in the proof of Theorem 84, Proposition 56 and Proposition 61 give a subdivision of

⁴There is one in this situation, but it requires a more complicated procedure involving a continuous deformation.

\mathbb{I} such that each of these intervals contains a subcomplex (necessarily a subdivision of a closed interval) of the subdivision. On each of the intervals, we choose the constant function whose value is that of the given point.

We want to glue these functions together to give a function defined on the whole interval. We proceed to glue intervals together “from left to right”—that is, in the direction in which the usual order increases.

Call the rightmost point of the intersection of the leftmost two intervals p . We know that the value of the constant function to the left specializes to $f(p)$, so replacing the value of this point with the original value of $f(p)$ there yields a continuous, directed map. The relationship of $f(p)$ with the value selected to the right depends on the relationship between p and the point x whose neighbourhood is the one on the right.

If $p \leq x$ then $f(p) \rightsquigarrow f(x)$, and so if we replace the constant function whose value is $f(x)$ on a small half-open interval starting at p with the constant function with value $f(p)$, then this function is directed and continuous. If we choose this half-open interval to be small enough, then the resulting map will still relax f , and also we won't affect the function we are considering close to its rightmost point of definition, so this argument can be repeated.

In the other case, we have both $f(p) \rightsquigarrow f(x)$ and $f(x) \rightsquigarrow f(p)$ and so the idea used in the other case will still work.

In the course of this, we may subdivide \mathbb{I} several times, so that the small intervals used above correspond to subcomplexes of the final subdivision, which will be the subdivision \mathbb{I}_* referred to in the statement. \square

3.4.1 Future Work

Higher Dimensional Directed Case

It might seem, at first, that the idea for the proof in the one dimensional, directed case can be combined with that in the undirected case, to yield one which works in all dimensions in the undirected case. This meets with two difficulties. The first is that the star operation on a simplicial complex does not preserve directedness, because one might have a line which slopes such that its points are mutually incomparable, and hence a directed function defined on that line is free to do as it will, but when we apply the star operation, which thickens it, order relations will obtain between points of the thickening. This problem can be solved by moving to a more rigid notion of complex, perhaps some form of cubical complex where we demand that we subdivide the unit

hypercube into products of intervals. A more fundamental problem is that in higher dimensions, we only get information about what happens close to each point *for points which are comparable to it*. In one dimension, all points are comparable with each other, but in higher dimensions we do not have this property. This makes the glueing idea used in the directed case problematic, because we do not know the relationships between the values of these points. The present author feels that it is likely than an analogue of the preceding results is true, although the second of these obstacles is reason for pause.

Proof via the Work of McCord

In [MPCR11], whose proof we generalize here, a major concern is to show that the fundamental group⁵ of an Alexandroff space can be computed by combinatorial reasoning using finite paths. They point out that [BM07] establishes the same fact by a different method.

McCord [McC66] establishes that Alexandroff spaces can be reasoned about using simplicial complexes. To each Alexandroff space A , he associates a complex S and a map $S \rightarrow A$ which is a weak equivalence. One can then use the edge path groupoid of [Spa81] to reason about paths in the original space by combinatorial means.

It seems that this construction is ill-suited to the present situation, because it relies on the behaviour of chains in posets, and the reduction of the general case to that of posets. Furthermore, McCord proves a key technical lemma about weak equivalences, which would need to be re-proved in the setting of neighbourhood spaces.⁶

In [LT04,Lar06], a discrete version of homotopy theory is developed within the category of reflexive relations, in order to study certain constraint satisfaction problems by computing the homotopy groups of related reflexive relations. A discrete notion of homotopy group is developed, and this is proved to be the same as the topological one in the case that the relation is a poset. Then they show that there is an isomorphism of their homotopy groups for any relation and a poset model which they define. They

⁵In this section we use some terms from homotopy theory not defined in this thesis. The interested reader is directed to [Spa81].

⁶McCord's original paper is quite terse; the present author relied on [Cla09a,Cla09b], which proves an interesting converse to do with approximating spaces by the limits of better and better Alexandroff models, as a reference for these ideas.

combine these results with the result of McCord to provide a simplicial model of a reflexive relation.

The result we prove above ought to imply that the discrete homotopy groups of [LT04, Lar06] are in fact equivalent to the usual homotopy groups obtained by viewing the reflexive relation as an Alexandroff neighbourhood space. If this is true, then it might be possible to use their construction as a replacement for the constructions of McCord.

The hope of replacing the present proof with one along the lines of McCord is that one might be able to replace the combinatorial reasoning above with abstract conceptual reasoning. One should take care, however, not to lose some of the advantages of the concrete approach. For example, the homotopy we construct above takes place entirely within the image of the original map.

In our intended application, wherein we want to interpret points of a space as distributions of information, this is important because it shows that no extraneous information distributions need to be added to obtain a finite path. The reason that this is important is that we care about a “finite density of information” constraint. If we think of an algorithm as a family of paths, one for each possible input, then if we extract a finite path from each of these, we might worry that lots of different distributions are being added by the extraction, in a way which may violate the bounded information density principle. The feature of the result above that only points already used are required reassures us that this does not occur.

Chapter 4

Information Flow in a Model of Computation

4.1 Introduction

In this chapter, we resume the main theme of the thesis: to analyse the flow of information in a spatial model of computation. The idea is to give a definition of a flow of information and characterize those information flows which correspond to computations. Moreover, we want to isolate a crucial property of such information flows which can be interpreted as a “continuity” condition, and argue that there is a space of some kind (in the end, in this treatment, rather combinatorially defined) which captures this notion of continuity.

First, we need to fix on a model of computation to study. Although the point of view we take here is inspired by Gandy’s analysis of computation, the model of computation which Gandy arrives at by this analysis is, as he intended, very general. In particular, the geometry of the underlying space may change as the computation progresses. Although one would ultimately wish to extend the analysis presented here to such general computations, this feature presents two problems. First, taking the changing shape of space into account increases the complexity of Gandy’s model considerably, and we would like a simple model so that the notion of information can be described easily. Second, and more fundamentally, we hope to describe an abstract space of distributions of information over some base space, such that paths in this abstract space give rise to our information flows. While this could perhaps be envisaged

in the case where the geometry of space changes, it would clearly add significant additional complexity; if (as in Gandy’s model) the contents of space can affect the way in which its geometry evolves, then the definition of information used in this thesis would need to be radically revised. For similar reasons, we do not attempt to be *more* general than Gandy. For instance, we keep to a linear, discrete, notion of time which would not be sufficient to describe computations involving randomness or lacking a strong notion of simultaneity; we also assume that space is composed of discrete pieces.

These restrictions essentially imply that we are studying a form of *cellular automaton*. Indeed, Gandy [Gan80, §2.5] reassuringly states that “an example which played an important part” in his development is that of “the crystalline [i.e. cellular] automata of Von Neumann” (parenthesis mine). Although our focus is on cellular automata as simple spatial computation systems, we briefly review the history of the usual study of them, with a particular emphasis on their mathematical formalization. We then describe the formalization to be used here, and finally move on to questions of information flow.

4.1.1 Standard Formalizations of Cellular Automata

Cellular automata were introduced by von Neumann in an unfinished manuscript, [Neu66]. This was edited and completed for publication by Burks, on whose historical notes we rely. Von Neumann was interested in the information processing and reproductive capacities of biological systems. In the course of his investigations, he set out to describe a machine which, set to work in a vast reservoir of resources, could assemble any conceivable device—including a copy of itself. He thought of this *construction universality* as a generalization of the computational universality of Turing machines. Initially, he had hoped to design a realistic machine, capable in principle of operating in the real world. Ulam¹ suggested that a sensible starting point would be to work in a simplified model of the world consisting of a two dimensional grid of square cells. Each cell has a collection of neighbours. For von Neumann, these were the four cells above, below, left and right of it. In this thesis, we will additionally use the cells diagonally adjacent, a choice often called the *Moore neighbourhood*. It is important that the neighbourhoods of each cell are translations of each other, so that all cells are

¹In [Neu66, §1.1.2.3], Burks lists some early papers of Ulam on similar mathematical structures.

spatially related to their neighbours in the same way.

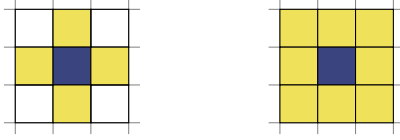


Figure 4.1: The von Neumann (left) and Moore (right) neighbourhoods of a cell.

At any time, a given cell is in a particular state. The states of all cells are drawn from a fixed finite set, which, for the purposes of this thesis, we think of as an *alphabet of symbols* written down in the course of computation. At each moment in time, a cell updates its state by applying some function of the states of itself and its neighbours. It may in general use the relative locations of these neighbours. For example it may replace its own state with the one of the cell directly above.

Let us be a little more explicit. We have a set of cells, which we might as well identify with \mathbb{Z}^2 , and an alphabet A of states. Each cell $z \in \mathbb{Z}^2$ has a neighbourhood $N_z \in (\mathbb{Z}^2)^9$, which we could describe using the additive group structure on \mathbb{Z}^2 via

$$N_z = \left(\begin{array}{ccc} z + (-1, -1) & , & z + (0, -1) & , & z + (1, -1), \\ z + (-1, 0) & , & z & , & z + (1, 0), \\ z + (-1, 1) & , & z + (0, 1) & , & z + (1, 1) \end{array} \right)$$

The dynamics of the discrete world are given by a local update function $L : A^9 \rightarrow A$. Now, a global configuration is a function in $\mathbb{Z}^2 \rightarrow A$, and can be evaluated at a point via the function $e : \mathbb{Z}^2 \times (\mathbb{Z}^2 \rightarrow A) \rightarrow A$. This can be extended co-ordinate wise to a function $e^9 : (\mathbb{Z}^2)^9 \times (\mathbb{Z}^2 \rightarrow A) \rightarrow A^9$. Then the global update function $U : (\mathbb{Z}^2 \rightarrow A) \rightarrow (\mathbb{Z}^2 \rightarrow A)$ is given by

$$U(f)(z) = (L \cdot e^9(f))(N_z).$$

Von Neumann gave a particular 29 state automaton and described how an initial assignment of states to cells could produce a structure which would construct other structures described to it by a tape of instructions. Burks [Neu66, §5.1.3] shows explicitly how von Neumann's constructions can be used to simulate a Turing machine.

In von Neumann's model, an important role is played by "quiescent" or "unexcited" states², which remain the same as long as all their neighbours are also quiescent. For von Neumann, this represented the empty space (or perhaps the sea of formless resources) in which his structures worked. In the initial configuration, only a finite number of cells were in a non-quiescent state. This is clearly an effectiveness requirement: we want to imagine actually *building* the device encoded in this initial configuration. Burks [Neu66, §1.3.3.5] remarks that "[t]his state plays a fundamental role with respect to the information content of the cellular structure" and goes on to point out that "the unexcited state of von Neumann's system is analogous to a blank square on the tape of a Turing machine". The fact that a Turing machine's tape is initially blank but for a finite number of squares is important for effectiveness in the usual sense. For instance, it prevents us from supplying, along with the input, an infinite look-up table containing the answers to an undecidable problem. We have paused here to belabour this point because for the most part the present analysis will *ignore* it: we assume that the user supplies their inputs following some sensible encoding, and examine the way the information they have provided flows within the model.

Work of the original type, where the pattern of connections between cells is given in a more-or-less ad-hoc manner, and structures for a fixed update rule are studied, continues to this day. It is clear that the general idea can be extended to interconnection patterns very different from the grid of squares. For example, [IHPS12] and [HSW05] study self-reproducing structures in a two-state cellular automaton whose underlying pattern of cells is a Penrose tiling. In this thesis, we keep to the two dimensional grid of squares, but these more exotic situations play a heuristic role: nothing we do ought to fail for them.

An early study of the computational powers of cellular automata (in a one or two dimensional grid) was undertaken by Smith [Smi72, Smi71, Smi76]. In this work, it is the space itself which is thought of as the model; the particular alphabet of states and local update rule is the *program* which is varied with the problem considered. This is more or less the point of view which we take here. His studies include which languages could be decided by cellular automata under extreme time and space constraints [Smi72], and what trade-offs had to be made between neighbourhood size and number

²In [Neu66], these two terms have subtly different technical meanings, but the distinction does not concern us here.

of local states [Smi71]. In [Smi76] the most natural encoding of Turing machines as cellular automata is given. For us, an important aspect of [Smi72] is that at the end of the computation, a single cell's state determines whether or not the original input is in the language or not. This is interesting because the language usually encodes a global property of the input, such as being a palindrome. At the start of computation, we would need to see the whole initial configuration to tell whether the input was a palindrome, but at the end, we could just observe a single cell. This influences our own definition of deciding.

A separate point of view is provided by work on cellular automata as dynamical systems, beginning with Hedlund [Hed69]. A modern treatment can be found in [CSC10]. We noted above that in a cellular automaton, it is important that each cell's neighbourhood be the same shape as any other. This allows us to write a single local update rule and apply it everywhere. To give meaning to the notion of "same shape", we made use of the group structure on \mathbb{Z}^2 : we could have specified the neighbourhood of any point, and then the neighbourhood of any other could be found by translating by the difference between the two points. In the dynamical systems point of view, the space is given by a free, transitive action τ of a group G on a set of cells C . We consider the group action—telling us which regions are the same shape as each other—to be fixed, but we allow the neighbourhood to vary arbitrarily, as long as it is finite. This action induces a related action $\tau' : G \times (C \rightarrow A) \rightarrow (C \rightarrow A)$ on global configurations. Now we can specify a cellular automaton by giving the neighbourhood N_c of any cell $z \in C$ and the local update function $L : (N_z \rightarrow A) \rightarrow A$ there. We then stipulate that the global update $U : (C \rightarrow A) \rightarrow (C \rightarrow A)$ be such that $U(c)(z) = L(c \upharpoonright N_z)$ and that it commute with $\tau'(g)$ for all $g \in G$.

The point of all this is to be able to detect when a given $U : (C \rightarrow A) \rightarrow (C \rightarrow A)$ is a cellular automaton. Now we need only say that there exists a suitable N_z and L . This can be reinterpreted as a demand for *finite causes*: for each cell $z \in C$ there is some finite neighbourhood N_z such that for all $c \in (C \rightarrow A)$ the value $U(c)(z)$ is determined by the restriction of c to N_z . We don't need to demand that these neighbourhoods are all the same shape: that will be taken care of by the group action. So we have split the property of being a cellular automaton into two conceptual parts: commuting with translations, so that a cell may not look up its "co-ordinates" to decide what to do, and finite causation, so that a cell need only ask for the values of a finite number of other

cells to update its own value³.

This property of finite causes can be viewed topologically. Put on $C \rightarrow A$ the *pro-discrete* topology such that a basic set is one containing all configurations which agree on a finite set which we call its support⁴. This is the coarsest topology in which the evaluation function $e(z)$ (which returns the state of the cell z in the configuration given as an argument) is continuous with respect to the discrete topology on A . We demand that $U : (C \rightarrow A) \rightarrow (C \rightarrow A)$ be continuous in this topology. Now for each $a \in A$, $\{a\}$ is an open set in the discrete topology, and so for any cell $z \in C$, the set of configurations $c' \in (C \rightarrow A)$ such that $U(c')(z) = a$ is open in $(C \rightarrow A)$ and hence contains a basic set. This is supported by a finite set $N_{z,a}$, and if the restriction of a configuration c to $N_{z,a}$ matches that of any of the configurations c' , then $U(c)(z) = a$. We can repeat this argument for all possible $a \in A$, and take the union of the finite sets $N_{z,a}$. This will be a finite set whose value in a configuration determines the updated value of z .

This much improves our definition of cellular automaton. Given a finite alphabet A and a set of cells C with a free, transitive action of a group, we put a topology and group action on the set of configurations $C \rightarrow A$ as above. A cellular automaton is then a continuous map $(C \rightarrow A) \rightarrow (C \rightarrow A)$ which commutes with the action. Little more than undergraduate algebra and topology is required to understand a proof, for example, that the inverse of a globally bijective cellular automaton is actually given by another cellular automaton⁵! This result really makes use of the freedom to choose an arbitrary neighbourhood: Kari [Kar90] shows that the problem of deciding whether a two dimensional cellular automaton has an inverse is undecidable; he concludes that one cannot compute a bound on the size of the neighbourhood of the inverse automaton from the size of the original neighbourhood and the alphabet (or any computable property of the automaton given).

³It is interesting to reflect that neither of these properties by itself guarantees effectiveness. Fix an undecidable predicate P on the natural numbers. Let $C = G = \mathbb{Z}$ with either translation action of the group on itself, $A = \{0, 1\}$. Consider the update rule U which given a configuration with n cells labelled 1 turns every cell to 1 if and only $P(n)$; this commutes with translations. On the other hand consider a U where the cell $z \geq 0$ examines the z cells before it. If the extreme cells are both 0 and in the others the number of 1s satisfies P , it changes its state to 1, otherwise to 0; this has finite causes.

⁴This is a basis since configurations in the intersection of two such sets are in the basic set supported by the union of the supports.

⁵See [CSC10, Theorem 1.10.1]. One needs to recall that a finite discrete space is compact and Hausdorff, that these properties are preserved under arbitrary products, and that a continuous bijection from a compact to a Hausdorff space has a continuous inverse.

The definition of cellular automata which we use below is inspired by the dynamical systems point of view in the sense that we regard shape-preserving translations of regions as an important part of the structure of our space. On the other hand, we view the neighbourhoods as an even more intrinsic aspect of the space, and so we do not allow them to vary. We also interpret the idea of “shape preserving translation” somewhat more broadly. To demand a free, transitive group action would rule out lots of ways of connecting cells together. For example, it would clearly violate our heuristic that everything we do ought to work for automata on a Penrose tiling, which can support no such action⁶.

We now proceed to define cellular automata in a way which is convenient for the purposes of this thesis. We then define a notion of “information” contained in a region, and examine which properties of the spatio-temporal behaviour of information so defined characterize cellular automata.

4.2 Cellular Automata on a Category

In the standard formalizations of cellular automata, an important role is played by the ability to translate one region into another. This allows us to say when two regions look alike, so that we can demand that the update rule of the cellular automaton only uses the way the world looks around a cell, rather than incidental features of our model. In many of the usual situations, these translations can be taken to be translations of the whole space, but we saw that people do work on spaces which are not homogeneous enough for that. Instead, we take a more local point of view. We model space as a category, whose objects we think of as regions, and where a morphism from one to the other is a way of fitting the source into the target.

For example, we can think of the usual definition of cellular automaton as taking place over a category **Plane** whose objects are square patches of square tiles, and where a morphism from a smaller square into a larger one is a copy of the former within the latter. We conventionally call an n -by- n square simply n , and we write (x,y) to mean a morphism which locates a small square in a larger one at a horizontal offset of x and a

⁶That said, see the interesting recent [AMN12] which transports some of the theory sketched above to arbitrary bounded-degree graphs by viewing them as generalized Cayley graphs for arbitrary quotients of formal languages.

vertical one of y . There are four ways to fit a two-by-two square into a three-by-three one:

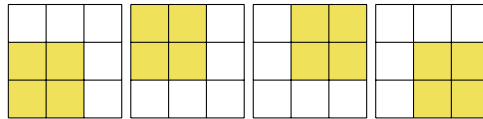


Figure 4.2: Depiction of the morphisms from a two-by-two square to a three-by-three square..

In this section, we specify what properties a category needs in order that it can function as an underlying space for the cellular automata which we define. In subsequent sections, we go on to specifying how we represent labellings of the regions by an alphabet, and describe what we mean by a cellular automaton in this context. Our definition is local in the sense that one must specify how the automaton should update a region of an arbitrary shape. Finally, we define information, and ask what characterizes the way information flows which correspond to cellular automata.

To serve as an underlying space, a category needs to come with a coverage to tell us when a collection of morphisms (i.e. objects located within another object) cover their target. We also require some objects to play the role of finite regions, so that we can demand that the alphabet only assign these a finite number of labellings. Finally, we need a notion of the interior of an object. Although the motivation for this notion comes from neighbourhood spaces, we have very few requirements on the behaviour of this notion of interior, and so we use a different name: erosion⁷.

We let a **tissue** be a small site C with a specified set of objects which we call **bounded objects**, and an action

$$\ominus : \mathcal{T} \times C \rightarrow C$$

of some monoid \mathcal{T} of times, which we call **erosion**. We write the time argument as a subscript so that if $t \in \mathcal{T}$ and R is an object of C then $\ominus_t(R)$ is the value of \ominus on (t, R) . We often call the objects of a tissue *regions* and morphisms *regions located in their target*.

⁷This name is based on terminology familiar in image processing, see for example [CNS09], which takes a categorical perspective.

In our example category **Plane**, the notion of cover is the intuitive one, in which a family of maps covers its target if every map from the one-by-one square into the target factors through a member of the family. Note that we have not yet described the erosion on this tissue, and we do not need to describe coverings: this is covering “in the set-theoretic sense”. For example, the maps

$$\begin{aligned} &\{(3,0) : 1 \rightarrow 3, \\ &\quad (0,3) : 1 \rightarrow 3, \\ &\quad (0,0) : 2 \rightarrow 3, \\ &\quad (1,1) : 2 \rightarrow 3\} \end{aligned}$$

cover 3 as indicated by the figure below.

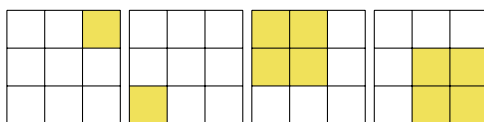


Figure 4.3: A cover of the three by three square.

The notion of erosion is useful because in the sequel we demand that an automaton should be able to update the value of an arbitrary region. When it does so, it will not be able to make use of any information about what is happening “outside” the given region, and so it will only be able to produce an updated labelling of the interior of the region. The erosion of a region specifies what we mean by “interior”. One way in which we allow more general phenomena than the interior of a neighbourhood space is that erosion is allowed to take a time argument $t \in \mathcal{T}$, so that we can specify how much of a region the automaton should be able to update, given its current labelling, t steps of time from the present.

In the standard notion of cellular automaton, $\mathcal{T} = \mathbb{N}$ and so we can simply specify $\ominus : \mathbf{Plane} \rightarrow \mathbf{Plane}$ and obtain an action of \mathbb{N} by iteration, letting $\ominus_t = \ominus^t$. In **Plane**, modelling the Moore neighbourhood, the action of erosion is to reduce the size of a square by two in both dimensions.⁸ This corresponds to our intuition of removing a border all the way around a square patch of tiles. If we do this both to the source

⁸The erosion on **Plane** means that we require an “empty square” so that every square smaller than three-by-three erodes to it. It is best to model this as an initial object.

and target of a morphism, then the eroded version of the source will still fit in the eroded target “at the same location”, in the sense that the bit of material which must be removed from the source to fit it into the eroded target is exactly what its erosion removes.

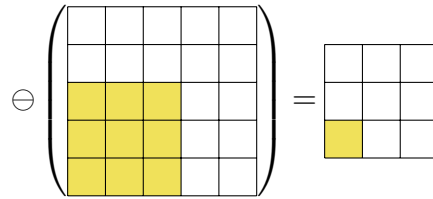


Figure 4.4: The action of erosion on a morphism.

The notion of tissue as a model for the underlying space of an automaton is very abstract. The reader should not think that every instance is really a meaningful space, nor that the automata on them defined in the next section always have computational content. The purpose of this thesis is not to define a model of computation, but to conduct an analysis of computation in certain models. The analysis in most of what follows will apply to any model of computation which can be thought of as a cellular automaton on a tissue, in the sense of the next section.

At this level of abstraction, we can already define the notion of labelling of a region and automaton, define our notion of information and characterize those flows of information which come from automata. However, the purpose of this analysis is supposed to be to investigate the role of *continuity* of the flow of information in a spatial model of computation. In order to do this, we need to weaken the properties which characterize cellular automata, because the flows of information which we come to call continuous are slightly more general than those which underlie cellular automata. To show that these continuous information flows still have computational content, we describe how to produce automata which are strictly more informative than them, in a sense which will become clear once our notion of information is defined. To perform this construction, we need to assume that the tissue we are working over is much closer to our example **Plane**. We call a tissue **cellular** if and only if

- our monoid \mathcal{T} of times is actually \mathbb{N} so that its actions can be specified by iteration. In particular, we have a functor $\ominus : C \rightarrow C$ which gives the required erosion

action in the sense that we let $\ominus_t(R) = \ominus^t(R)$;

- there is a collection of objects, which we call **cells** such that all maps which are part of a cover of a cell are isomorphisms;
- every object R of C comes with a specified cover $Cells(R)$ by cells such that
 - if $\chi : K \rightarrow R$ is an element of $Cells(R)$ and $f : R \rightarrow S$, then $f \cdot \chi \in Cells(S)$,
 - if $f : R \rightarrow S$ then

$$Cells(R) = \{\chi_x : K_x \rightarrow R \mid x \in X \text{ and } f \cdot \chi_x \in Cells(S)\}, \text{ and}$$

- if $\{f_x : R_x \rightarrow S \mid x \in X\}$ covers S , then

$$Cells(S) = \{\chi_x \cdot f_x \mid x \in X \text{ and } \chi_x \in Cells(R_x)\};$$

- a region R is bounded if and only if $Cells(R)$ is finite; and
- to every $\chi : K \rightarrow \ominus_u R$ where K is a cell (which we sometimes call a “cell of $\ominus_u R$ ”) we assign a map $N_\chi : R \rightarrow K$ such that this assignment satisfies:
 - $\ominus(N_\chi) = \chi$,
 - $f \cdot N_\chi = N_{\ominus f \cdot \chi}$.

Time being the natural numbers and the existence of cells are allied constraints: their function is to provide basic, indecomposable units which certain constructions require. In **Plane**, of course, every object comes with an obvious notion of a cover by cells.⁹

The mysterious assignment N tells us that each cell in the interior of a region has a smallest neighbourhood: something which is small enough that if we have a cell $\ominus(f) \cdot \chi$ which factors through the erosion of a map f , then neighbourhood $N_{\ominus(f) \cdot \chi}$ itself factors through f , meaning that in some sense it had an analogue in the source.

In **Plane**, whenever we find a cell in the erosion of a square of cells, we know that there must have been a map from the three-by-three square into the un-eroded region

⁹The empty square is covered by the empty set of cells.

whose erosion is the cell we started with. That is, a cell is in the erosion of a region only if its smallest neighbourhood fits inside the region to begin with.

This assignment can be thought of as coming from an “un-erosion” functor on **Plane** which is a right adjoint to erosion, but this is a very special property of **Plane**. The idea of a global “un-erosion” does not apply to other natural examples like automata on a Penrose tiling. It might be possible to think of the demands on the N assignment as some sort of local version of this adjunction—but to own the truth, they are simply what was needed to make the constructions go through. Understanding what they mean should be thought of as work in progress.

At this point, we could go on to define labellings and automata, which we do in the next section. However, we pause for a moment to discuss two more issues which the present author finds mysterious.

There are two features of **Plane** which we have not demanded analogues of in our general set-up, and which are worthy of remark. The first is that in **Plane** we need to assume the existence of an object corresponding to an infinite grid of squares, because when we come to describe computations with automata, we encode inputs by global labellings. In **Plane** this object is fixed by erosion (if I know what everything looks like now, then I know what everything will look like next), and this makes the notion of erosion more similar to the interior operator on a space, which fixes the “whole space”. That we do not need to demand this indicates that one could imagine a notion of cellular automaton which receives its inputs on little scraps, possibly of different shapes and sizes, and these scraps may even be worn away by the effort of computation.

The second piece of structure on **Plane** which we do not demand for all tissues is that there is a natural transformation from \ominus to the identity functor in **Plane** which tells us where the eroded copy of a square is located within the original. To model the Moore neighbourhood we want this to be “in the middle”.

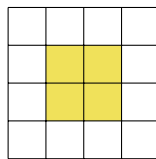


Figure 4.5: Interior of a four-by-four region.

This matches the property of the interior operator that it is *decreasing*. It also has a tempting interpretation for general tissues. One might think that our monoid \mathcal{T} of times should be an *ordered monoid*. This is a simple example of a 2-category, and so erosion might be thought of as a higher dimensional functor which sends increments of time to erosions of spatial regions, and order relations between times to natural transformations between these erosions, telling us where the eroded copy is located.¹⁰ That we do not demand this structure might indicate that the present analysis can be applied to models which change the shape of space in dramatic ways during computation, although because we define erosion once and for all on spatial regions, this modification of space must be independent of the labelling.

4.2.1 Alphabets, Encodings, and Automata

Now that we have a notion of an underlying space, we move on to defining what a cellular automaton on a tissue should be. A cellular automaton does not act on space alone, but on labelled versions of space. Our definition of automaton here is a *local* one, and so we want to specify a set of labellings for every regions. This ought to have the property that labellings can be restricted to sub-regions¹¹ along morphisms, and that if a region is covered by some located regions, then a labelling of the whole region ought to be determined by its restrictions to the cover. This means we think of labellings as some sort of syntax written on our regions.

Therefore, associated to our tissue C we demand a separated presheaf \mathfrak{K} called *the alphabet*, such that for every bounded object X , $\mathfrak{K}(X)$ is a finite set. We call an object R' of $\int_C \mathfrak{K}$ such that $\lfloor R' \rfloor = R$ for some object R of C a **labelling** of R .¹²

We now define a **cellular automaton** to be an action \mathcal{A} of \mathcal{T} on $\int_C \mathfrak{K}$ such that for all $t \in \mathcal{T}$ the diagram

¹⁰This picture starts to come apart somewhat when we define automata, for there we would want to throw away the order and ask for an action of the underlying monoid.

¹¹Sub-region will always be synonymous with morphism here; we do not isolate any special class, even monomorphisms, to which this title is exclusive. In **Plane**, every arrow is epic and monic, and this is likely to be a feature of all tissues which model sensible spaces.

¹²Recall from Section 2.1.2 that we write $\lfloor - \rfloor : \int_C F \rightarrow C$ for the functor from a category of elements of a presheaf F on a category C which sends an object to its underlying object in C .

$$\begin{array}{ccc}
\int_C \mathfrak{K} & \xrightarrow{\mathcal{A}(t)} & \int_C \mathfrak{K} \\
\downarrow [_] & & \downarrow [_] \\
C & \xrightarrow{\Theta_t} & C
\end{array}$$

commutes.

This corresponds to our requirement that an automaton must be able to update the value of an arbitrary region, but the updated value will be a labelling of the *eroded* version of the input, because other parts of the original region may have their values affected by a broader context which they could be located in.

In practice, the important implication of this is that for any morphism p , we have $\mathcal{A}(t)(p) = \Theta_t p$. This leads to the following observation.

Proposition 86. *Let \mathcal{A} be a cellular automaton on a tissue C . For all objects R and S of $\int_C \mathfrak{K}$, and $p : R \rightarrow S$, we have $\mathcal{A}(t)(R \upharpoonright p) = \mathcal{A}(t)(R) \upharpoonright \Theta_t p$.*

Proof. Working in $\int_C \mathfrak{K}$ we have, by the definition of that category, $p : R \upharpoonright p \rightarrow R$. Since $\mathcal{A}(t)$ is a functor, we have $\mathcal{A}(t)(p) : \mathcal{A}(t)(R \upharpoonright p) \rightarrow \mathcal{A}(t)(R)$ hence

$$\Theta_t p : \mathcal{A}(t)(R \upharpoonright p) \rightarrow \mathcal{A}(t)(R),$$

and so $\mathcal{A}(t)(R \upharpoonright p) = \mathcal{A}(t)(R) \upharpoonright \Theta_t p$ as required. \square

Since we are interested in computing, fix a set J of **inputs** and an **encoding**: a function $\varepsilon : J \rightarrow \int_C \mathfrak{K}$ assigning to each possible input a labelled object from which to begin computation. We will focus on *decision problems*, that is, for subsets $D \subseteq J$, we wonder whether a given $j \in J$ is also an element of D .

Now we want to know what properties of elements of J our automaton is good for deciding. For this, we ask what properties of an input we could learn by observing a region R which we find after t steps of time having started computing from some input j . The property of being a plausible input given our observation is just the property of being an input j' such that after computing from $\varepsilon(j')$ for some time we find a located copy of R (that is, a morphism into $\Theta_t [\varepsilon(j)]$) which is labelled just like the copy of R we actually observe. To that end, given a morphism $f : R \rightarrow \Theta_t [\varepsilon(j)]$ of C_+ we let

$$\mathfrak{W}_{\mathcal{A}}(j, t, f) = \{j' \in J \mid \text{there exists } t' \in \mathcal{T}, f' : R \rightarrow \ominus_{t'} \epsilon(j') \\ \text{such that } \mathcal{A}(t')(\epsilon(j')) \upharpoonright f' = \mathcal{A}(t)(\epsilon(j)) \upharpoonright f\}.$$

Note that we always have $j \in \mathfrak{W}_{\mathcal{A}}(j, t, f)$.

We can think of $\mathfrak{W}_{\mathcal{A}}(j, t, p)$ as being a coarse model of what the region (j, t, p) knows: it models only what it knows *about inputs* by containing all the inputs which (j, t, p) can't rule out on the basis of its labelling. We could view this as the extension of the property of inputs which we learn by observing a region labelled in the way (j, t, p) is labelled by \mathcal{A} . This gives a kind of logical interpretation of labellings. In order to continue to use “knowledge” and “information” informally, we call this the *wit* of (j, t, p) .

Given $P \subseteq J$ we say that \mathcal{A} *decides* D if and only if there is a bounded object R of C such that for all $j \in J$ there exists $t \in \mathcal{T}$ and $f : R \rightarrow \ominus_t \epsilon(j)$ such that either $\mathfrak{W}_{\mathcal{A}}(j, t, f) \subseteq D$ or $\mathfrak{W}_{\mathcal{A}}(j, t, f) \subseteq J/D$. The intuition is that since R is bounded, it has a finite number of labellings. First, we memorize which ones imply D and which ones imply its negation under the logical interpretation of labellings given by \mathfrak{W} . Then, we run our automaton from $\epsilon(j)$ and search for a labelled copy of R located via a morphism f in the result of some computation step, such that the labelling of R determines whether $j \in D$ or not. We are guaranteed to find such an f eventually by the definition of deciding.

4.3 Local Information

We now begin laying the foundations of our point of view that information in a labelled region is the set of places where the region embeds into history. The picture to have in mind is that history is a labelling of an enormous region, and we intend our notion of information in a region to be the set morphisms from the given region into history such that the labelling is preserved. This is a measure of the uncertainty we have about where in history we might be if we have only observed the contents of that particular region.

We first need to extend our category of regions to include a huge region which represents the shape of history. We could picture this as something made of slices, with one slice for each moment of time and each input. For $t \in \mathcal{T}$ and $j \in J$, the slice corresponding to t and j is the shape of the whole computation space if we had started from the encoding of j and computed for t steps of computation.

Given an encoding ε we form a category $C_{+,\varepsilon}$ in the following way. The objects of $C_{+,\varepsilon}$ are the objects of C , together with a new object Σ_ε . The maps between objects are the same as those in C if both their source and target are in C . In addition to its identity morphism, the only maps into Σ_ε are triples of the form $(j, t, f) : R \rightarrow \Sigma_\varepsilon$ where $j \in J$, $t \in \mathcal{T}$ and $f : R \rightarrow \ominus_t[\varepsilon(j)]$, with composition with a map $g : S \rightarrow R$ given by $(j, t, f) \cdot g = (j, t, f \cdot g)$. There are no non-trivial maps with Σ_ε as the source. Observe that, because of the way we have defined composition, maps into Σ_ε form a presheaf with precomposition as the action on morphisms. This means our $C_{+,\varepsilon}$ is a subcategory of the category of presheaves on C . Indeed Σ_ε corresponds to the coproduct of the (representable functors given by) objects of the form $\ominus_t[\varepsilon(j)]$. We do not take the *full* subcategory containing this coproduct, however, because we have no interest in morphisms whose source is Σ_ε . This simplifies some of the following discussion, but probably obscures its categorical meaning.

We can think of Σ_ε as the “shape of possible history”¹³ in that if we imagine the objects of C as regions with points, then Σ_ε would contain a point for every point in space, for every time, for every possible input. The input $i \in J$ will be encoded on a region of shape $[\varepsilon(j)]$; this is what the universe of our computation will look like at the dawn of time. After $t \in \mathcal{T}$ steps of computation, the universe will be of shape $\ominus_t[\varepsilon(j)]$. We collect all of these possibilities into one big space Σ_ε . We adopt the convention of dropping ε when it is clear from context, to speak only of C_+ and Σ ; we apply this convention without further comment to other constructions relating to C_+ which depend on ε . We use an additional convention that the inclusion of $\ominus_t[\varepsilon(j)]$ into Σ at input j and time t (obtained by using its identity morphism) may be written $\iota_{j,t}$. Observe that $\iota_{j,t} \cdot f = (j, t, f)$ for any morphism f of C . The reader will see the need for this notation by writing out the cumbersome name for the inclusion in the triple notation for maps into Σ given above.

¹³Or, perhaps more vividly, the “shape of all possible spacetimes” — although of course we have none of the complexity of physical spacetime, or the causal structure of a distributed system here; for instance, we have a trivial notion of simultaneity.

Now, we want C_+ to be a site, so we need to use the coverage on C to give a coverage on C_+ . For objects which are already in C , we take the covering families they already have there. Given an indexing set X , we say that a family $\{(j_x, t_x, f_x) : R_x \rightarrow \Sigma \mid x \in X, j_x \in J, t_x \in \mathcal{T}\}$ covers Σ if and only if for all $j \in J$ and $t \in \mathcal{T}$, the family $\{f_x : R_x \rightarrow \Theta_t[\epsilon(j)] \mid x \in X\}$ covers $\Theta_t[\epsilon(j)]$.

We check this defines a coverage. A morphism from R into Σ amounts to a map $f : R \rightarrow \Theta_t[\epsilon(j)]$ for some $j \in J$ and $t \in \mathcal{T}$. Since C is a site, there is a cover of R such that every composite of a map g in the cover with f factors through the induced cover of $f : R \rightarrow \Theta_t[\epsilon(j)]$. But then $(j, t, g \cdot f)$ is in our original cover of Σ as required.

If C is cellular, we set

$$\text{Cells}(\Sigma) = \{\iota_{j,t} \cdot \chi \mid j \in J, t \in \mathcal{T} \text{ and } \chi \in \text{Cells}(\Theta_t[\epsilon(j)])\}.$$

We can also extend the alphabet \aleph to a presheaf $\aleph_{+, \epsilon}$ on $C_{+, \epsilon}$ which agrees with \aleph on the objects already present in C . We need only define this on Σ and (non-identity) maps into Σ . We set

$$\aleph_+(\Sigma) = \prod_{\substack{j \in J \\ t \in \mathcal{T}}} \aleph(\Theta_t[\epsilon(j)]).$$

Write $\pi_{j,t}$ for the projections of this product. Given $(j, t, f) : R \rightarrow \Sigma$ we set

$$\aleph_+(j, t, f) = \aleph(f) \cdot \pi_{j,t}.$$

The reader familiar with the Yoneda embedding will see that the above construction is just the one given by the universal property of the category of presheaves as the co-completion of C , viewing the presheaf \aleph as a functor to \mathbf{Set}^{op} , which is co-complete.

We ought to check that \aleph_+ is a separated presheaf.

Proposition 87. *Let C be a tissue, \aleph be a separated presheaf on C , and $\epsilon : J \rightarrow \int_C \aleph$ be an encoding for a set of inputs J . Then $\aleph_+ : C_+^{op} \rightarrow \mathbf{Set}$ is a separated presheaf.*

Proof. Functoriality follows by functoriality of \aleph since for all $R, S \in C$, $g : R \rightarrow S$ and

$(j, t, p) : R \rightarrow \Sigma$ in C_+ we have

$$\begin{aligned} \mathfrak{K}_+((j, t, f) \cdot g) &= \mathfrak{K}_+(j, t, f \cdot g) \\ &= \mathfrak{K}(f \cdot g) \cdot \pi_{j,t} \\ &= \mathfrak{K}(g) \cdot \mathfrak{K}(f) \cdot \pi_{j,t} \\ &= \mathfrak{K}_+(g) \cdot \mathfrak{K}_+(j, t, f). \end{aligned}$$

We turn to showing that \mathfrak{K}_+ is separated.

Let X be an indexing set and $\{(j_x, t_x, f_x) : R_x \rightarrow \Sigma \mid x \in X\}$ be a cover of Σ . Let $v, v' \in \mathfrak{K}_+(\Sigma)$ and suppose that $\mathfrak{K}_+(j_x, t_x, f_x)(v) = \mathfrak{K}_+(j_x, t_x, f_x)(v')$ for all $x \in X$. By the definition of \mathfrak{K}_+ , we know that $\pi_{j,t}(v)$ and $\pi_{j,t}(v')$ are elements of $\mathfrak{K}(\Theta_t[\epsilon(j)])$. The cover of Σ gives us a cover $\{f_x \mid x \in X, f_x : R_x \rightarrow \Theta_t[\epsilon(j)]\}$ of $\Theta_t[\epsilon(j)]$. We have

$$\mathfrak{K}(f_x)(\pi_{j,t}(v)) = \mathfrak{K}(f_x)(\pi_{j,t}(v'))$$

by the definition of \mathfrak{K}_+ , and so, since \mathfrak{K} is separated, we deduce $\pi_{j,t}(v) = \pi_{j,t}(v')$. This follows for all $j \in J$ and $t \in \mathcal{T}$, so $v = v'$ as required. \square

Having described the shape of history, and extended our notion of labelling to include it, we want to describe history itself. We picture Σ as being made of slices, one for each input and moment of time, and now we want to label those slices so that their contents are what we would actually observe if we encoded the input and computed from its encoding for the given amount of time.

Given an automaton and an encoding, we produce an object $\widehat{\mathcal{A}}_\epsilon$ of $f_{C_+} \mathfrak{K}_+$ such that $\lfloor \widehat{\mathcal{A}}_\epsilon \rfloor = \Sigma_\epsilon$ by setting

$$\widehat{\mathcal{A}}_\epsilon \upharpoonright_{j,t} = \mathcal{A}(t)[\epsilon(j)].$$

We check that $\widehat{\mathcal{A}}$ (dropping the subscript ϵ as per our convention) matches the behaviour of \mathcal{A} :

Proposition 88. *Given a cellular automaton \mathcal{A} on a tissue C , an element $u \in \mathcal{T}$ and $(j, t, p) : R \rightarrow \Sigma$ in C_+/Σ , we have $\widehat{\mathcal{A}} \upharpoonright (j, t + u, \Theta_u p) = \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright (j, t, p))$.*

Proof. We calculate

$$\begin{aligned}
 \widehat{\mathcal{A}} \upharpoonright (j, t + u, \ominus_u p) &= \widehat{\mathcal{A}} \upharpoonright \mathbf{1}_{j, t + u} \cdot \ominus_u p && \text{by definition} \\
 &= \widehat{\mathcal{A}} \upharpoonright \mathbf{1}_{j, t + u} \upharpoonright \ominus_u p && \text{by functoriality of } \mathfrak{K} \\
 &= \mathcal{A}(t + u)(\varepsilon j) \upharpoonright \ominus_u p && \text{by the definition of } \widehat{\mathcal{A}} \\
 &= \mathcal{A}(u)(\mathcal{A}(t)(\varepsilon j)) \upharpoonright \ominus_u p && \text{by functoriality of } \mathcal{A} \\
 &= \mathcal{A}(u)(\mathcal{A}(t)(\varepsilon j) \upharpoonright p) && \text{by Proposition 86} \\
 &= \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright \mathbf{1}_{j, t} \upharpoonright p) \\
 &= \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright \mathbf{1}_{j, t} \cdot p) \\
 &= \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright (j, t, p))
 \end{aligned}$$

which is all we wanted. \square

This object $\widehat{\mathcal{A}}$ is then a record of the *history of computation* as it proceeds from any possible input.

This sets us up for a first attempt to define what we mean by the information in a region. We could view the problem in the following way. We imagine ourselves watching the progress of the automaton from some input. After a while, a friend comes over and asks for the contents of some particular located region. We might reply with its syntactic contents, but they protest that they don't care about the implementation details of the automaton; they have no interest in anything which requires the encoding to make sense. They only care about inputs, time, and space—things which all automata must deal with. Had they only cared about inputs, we would have had to reply with the wit of the region, defined in the previous section, but because they also care about time and space, we can do much better: we can tell them all of the possible places in the history of the computation where the region might have occurred, given its contents.

Given a morphism $(j, t, p) : R \rightarrow \Sigma$ of C_+ we let

$$\mathfrak{K}_{\mathcal{A}, \varepsilon}^{(1)}(j, t, p) = \left[\int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j, t, p), \widehat{\mathcal{A}}] \right].$$

Note that $\mathfrak{W}_{\mathcal{A}} = \pi_1 \cdot \mathfrak{K}_{\mathcal{A}}^{(1)}$.

Let $dom : C_+/\Sigma \rightarrow C_+$ be the domain functor on the slice category. We have the

following.

Proposition 89. *For all cellular automata \mathcal{A} on a tissue C , $\mathfrak{K}_{\mathcal{A}}^{(1)}$ is a subfunctor of the functor $C_+[dom(-), \Sigma]$.*

Proof. It follows easily from the definition of $\mathfrak{K}_{\mathcal{A}}^{(1)}$ that for all $(j, t, p) : R \rightarrow \Sigma$ of C_+ we have $\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p) \subseteq C_+[dom(j, t, p), \Sigma]$; what we really want to check is that whenever $f : (j, t, p) \rightarrow (k, u, q)$ in C_+/Σ and $(k', u', q') \in \mathfrak{K}_{\mathcal{A}}^{(1)}(k, u, q)$ then $(k', u', q') \cdot f \in \mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p)$.

Supposing that $(k', u', q') \in \mathfrak{K}_{\mathcal{A}}^{(1)}(k, u, q)$, we must have $(k', u', q') : \widehat{\mathcal{A}} \upharpoonright (k, u, q) \rightarrow \widehat{\mathcal{A}}$ in $\int_{C_+} \mathfrak{K}_+$ and so by the definition of $\int_{C_+} \mathfrak{K}_+$ we have $\widehat{\mathcal{A}} \upharpoonright (k', u', q') = \widehat{\mathcal{A}} \upharpoonright (k, u, q)$. But then

$$\begin{aligned} \widehat{\mathcal{A}} \upharpoonright (k', u', q') \cdot f &= \widehat{\mathcal{A}} \upharpoonright (k', u', q') \upharpoonright f \\ &= \widehat{\mathcal{A}} \upharpoonright (k, u, q) \upharpoonright f \\ &= \widehat{\mathcal{A}} \upharpoonright (k, u, q) \cdot f \\ &= \widehat{\mathcal{A}} \upharpoonright (j, t, p), \end{aligned}$$

which means that $(k', u', q') \cdot f : \widehat{\mathcal{A}} \upharpoonright (j, t, p) \rightarrow \widehat{\mathcal{A}}$ in $\int_{C_+} \mathfrak{K}_+$. □

An important property of $\mathfrak{K}_{\mathcal{A}}^{(1)}$ is that it allows us to glue together its values on small regions to obtain its value on a region which they cover. The reader may like to recall the coverage on a slice category given on page 23.

Proposition 90. *If $\{\chi_x : (j, t, q_x) \rightarrow (j, t, p) | x \in X\}$ is a cover of (j, t, p) in C_+/Σ , then*

$$\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p) = \bigcap_{x \in X} C_+[\chi_x, \Sigma]^{\leftarrow} \left[\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, q_x) \right].$$

Proof. Since $\{\chi_x : (j, t, q_x) \rightarrow (j, t, p) | x \in X\}$ is a cover of (j, t, p) , we have $\{\chi_x : \widehat{\mathcal{A}} \upharpoonright (j, t, q_x) \rightarrow \widehat{\mathcal{A}} \upharpoonright (j, t, p) | x \in X\}$ is a cover of $\widehat{\mathcal{A}} \upharpoonright (j, t, p)$ in $\int_{C_+} \mathfrak{K}_+$. Then recalling that

\mathfrak{K}_+ is required to be separated, we have

$$\begin{aligned} \mathfrak{K}_{\mathcal{A},\varepsilon}^{(1)}(j,t,p) &= \left[\int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j,t,p), \widehat{\mathcal{A}}] \right] \\ &= \bigcap_{x \in X} C_+[\chi_x, \Sigma]^\leftarrow \left[\int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j,t,q_x), \Sigma] \right] \text{ by Proposition 3} \\ &= \bigcap_{x \in X} C_+[\chi_x, \Sigma]^\leftarrow \left[\mathfrak{K}_{\mathcal{A}}^{(1)}(j,t,q_x) \right] \end{aligned}$$

as required. \square

This $\mathfrak{K}_{\mathcal{A}}^{(1)}$ is almost what we want to model our informal notion of what a region knows. We might think of abandoning the accidents of syntax at this point, to study $\mathfrak{K}_{\mathcal{A}}^{(1)}$ instead of \mathcal{A} . However, this point of view has the inconvenient property that knowing $\mathfrak{K}_{\mathcal{A}}^{(1)}$ of a large region and the position of a small region within the large one, we can't restrict to compute $\mathfrak{K}_{\mathcal{A}}^{(1)}$ of the small region. This is a shame because the glueing property just proved leads us to think of $\mathfrak{K}_{\mathcal{A}}^{(1)}(j,t,p)$ as itself a piece of geometric data on (j,t,p) , coming from a set of possible such data as \mathcal{A} varies, and thus something we ought to be allowed to restrict. We solve this problem by the power of abstract nonsense, and the reader who prefers generalities can skip the next paragraph. For the reader who likes a story, however, a story can be told.

Our error in forgetting everything about a region but $\mathfrak{K}_{\mathcal{A}}^{(1)}$ of it is that it is not quite all we can say in the language of time, space, and inputs about the significance of its contents. While we have the labelled region in front of us, our friend who abhors syntax can ask us for the value of $\mathfrak{K}_{\mathcal{A}}^{(1)}$ on any of its sub-regions, and we can oblige them. Thus, our real model for the contents of a region ought to be the capability to answer such queries. Its value on a region ought to be something which can be evaluated at any sub-region to discover where in the history of computation that sub-region might have been.

Let the functor $\mathfrak{K}_{\mathcal{A},\varepsilon} : C_+/\Sigma \rightarrow \mathbf{Cat}/\mathbf{Set}$ be as follows. Given $f : S \rightarrow R$ in C_+ , we set

$$\begin{aligned} \mathfrak{K}_{\mathcal{A},\varepsilon}(j,t,p) : C_+/dom(j,t,p) &\rightarrow \mathbf{Set} \\ f &\mapsto \mathfrak{K}_{\mathcal{A},\varepsilon}^{(1)}((j,t,p) \cdot f) \end{aligned}$$

We send an arrow α in C_+/Σ to the functor C_+/α (“postcomposition with α ”).

This gives an arrow of **Cat/Set** because

$$\begin{aligned} (\widehat{\mathfrak{K}}_{\mathfrak{A}}(k, u, q) \cdot C_+/\alpha)(-) &= \widehat{\mathfrak{K}}_{\mathfrak{A}}(k, u, q)(\alpha \cdot -) \\ &= \widehat{\mathfrak{K}}_{\mathfrak{A}}((k, u, q) \cdot \alpha)(-). \end{aligned}$$

Hence if $\alpha : (j, t, p) \rightarrow (k, u, q)$ then $(j, t, p) = (k, u, q) \cdot \alpha$, and thus

$$\begin{aligned} \widehat{\mathfrak{K}}_{\mathfrak{A}}(k, u, q) \cdot C_+/\alpha &= \widehat{\mathfrak{K}}_{\mathfrak{A}}((k, u, q) \cdot \alpha) \\ &= \widehat{\mathfrak{K}}_{\mathfrak{A}}(j, t, p) \text{ as required.} \end{aligned}$$

Functoriality follows by functoriality of $C_+/-$ (because on arrows $\widehat{\mathfrak{K}}_{\mathfrak{A}}$ does the same thing as $C_+/-$).

Note that we can recover $\widehat{\mathfrak{K}}_{\mathfrak{A}}^{(1)}$ from $\widehat{\mathfrak{K}}_{\mathfrak{A}}$ because

$$\widehat{\mathfrak{K}}_{\mathfrak{A}}^{(1)}(j, t, p) = \widehat{\mathfrak{K}}_{\mathfrak{A}}(j, t, p)(1_{\text{dom}(j, t, p)}).$$

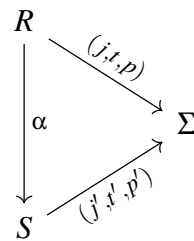
How should we understand this functor? If we think about the functor $\text{dom} : \mathbf{Cat/Set} \rightarrow \mathbf{Cat}$, we see that $\text{dom} \cdot \widehat{\mathfrak{K}}_{\mathfrak{A}} = C/\text{dom}(-)$. We define another such functor $\mathfrak{C}_{\Sigma} : C_+/\Sigma \rightarrow \mathbf{Cat/Set}$ as follows.

$$\mathfrak{C}_{\Sigma}(j, t, p) : C_+/\text{dom}(j, t, p) \rightarrow \mathbf{Set} = C_+[\text{dom}(-), \Sigma].$$

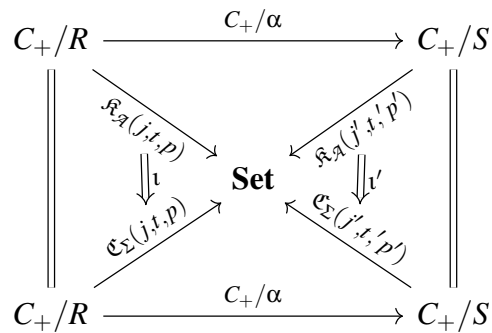
Then Proposition 89 implies that $\widehat{\mathfrak{K}}_{\mathfrak{A}}$ is a pointwise subfunctor of \mathfrak{C}_{Σ} , in the sense that for the same input, the output of $\widehat{\mathfrak{K}}_{\mathfrak{A}}$ is a subfunctor of the output of \mathfrak{C}_{Σ} . We call a functor $\mathfrak{A} : C_+/\Sigma \rightarrow \mathbf{Cat/Set}$ an **information flow** when $\text{dom} \cdot \mathfrak{A} = C_+(\text{dom}(-))$ and \mathfrak{A} is a pointwise subfunctor of \mathfrak{C}_{Σ} in the sense just given. Practically speaking, what we will really use is the fact that for all $p : R \rightarrow \Sigma$ and all $g : Q \rightarrow R$ we have $\mathfrak{A}(p)(g) = \mathfrak{A}(p \cdot g)$. This is a consequence of the condition on the domain of the functor which we proved just above in the case of $\widehat{\mathfrak{K}}_{\mathfrak{A}}$. We might as well reprise the argument. We have

$$\begin{aligned}
 (\mathfrak{A}(p) \cdot C_+/g)(-) &= \mathfrak{A}(p)(g \cdot -) \\
 &= \mathfrak{A}((p) \cdot g)(-).
 \end{aligned}$$

Before moving on, let us explore the properties of $\mathfrak{K}_{\mathfrak{A}}$ a little more. Let

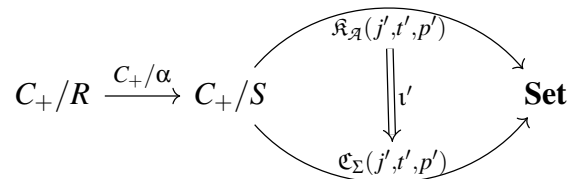


commute in C_+ so that $\alpha : (j, t, p) \rightarrow (j', t', p')$ in C_+/Σ . We know that $\mathfrak{K}_{\mathfrak{A}}(j, t, p)$ is a subfunctor of $\mathfrak{C}_{\Sigma}(j, t, p)$ so there is a natural transformation \mathfrak{t} from the former to the latter whose components are all set inclusions. Similarly, we have $\mathfrak{t}' : \mathfrak{K}_{\mathfrak{A}}(j', t', p') \rightarrow \mathfrak{C}_{\Sigma}(j', t', p')$ with the same property. Then in **Cat** we get the diagram



in which the empty faces commute.

Now $\mathfrak{K}_{\mathfrak{A}}(j', t', p') \cdot C_+/\alpha = \mathfrak{K}_{\mathfrak{A}}(j, t, p)$ and $\mathfrak{C}_{\Sigma}(j', t', p') \cdot C_+/\alpha = \mathfrak{C}_{\Sigma}(j, t, p)$ and the composite



is equal to

$$\begin{array}{ccc}
 & \xrightarrow{\mathfrak{K}_{\mathcal{A}}(j,t,p)} & \\
 C_+/R & \xrightarrow{\quad} & \mathbf{Set} \\
 & \xleftarrow{\mathfrak{C}_{\Sigma}(j,t,p)} & \\
 & \Downarrow \mathfrak{l} & \\
 & \xrightarrow{\quad} &
 \end{array}$$

It is not clear what the significance of all this is, but it is suggestive; perhaps some sort of higher-order analogue of subfunctoriality is involved. Note that if we did not have $dom \cdot \mathfrak{K}_{\mathcal{A}} = dom \cdot \mathfrak{C}_{\Sigma}$, then the required composites would not make sense, and so this is quite a special situation.

Remark. A different way of viewing $\mathfrak{K}_{\mathcal{A}}$, one we do not use, would be as follows. Consider $dom : C_+/\Sigma \rightarrow C_+$ and the category $\mathbf{Arrows}(C_+)$ of arrows of C_+ with the codomain functor $cod : \mathbf{Arrows}(C_+) \rightarrow C_+$. Let P be the pullback of dom and cod . This means that P is the category of composable pairs of arrows in C_+ such that the target of the second arrow is Σ . We call the functor which takes the domain of the first arrow $dom : P \rightarrow C_+$. Now we could define $\mathcal{K} : P \rightarrow \mathbf{Set}$ via $\mathcal{K}(f, (j,t,p)) = \mathfrak{K}_{\mathcal{A}}^{(1)}((j,t,p) \cdot f)$. (“The same formula” will work for arrows: an arrow in P is a pair of arrows making a commuting square and a commuting triangle with a common edge. Pasting them together along this edge produces an arrow of C_+/Σ to which we can apply $\mathfrak{K}_{\mathcal{A}}^{(1)}$.) Then this would be a subfunctor of $C_+[dom(-), \Sigma] : P \rightarrow \mathbf{Set}$. We do not do this, because our intention is that the knowledge in a region (j,t,p) will be represented by the functor $\mathfrak{K}_{\mathcal{A}}(j,t,p)$, which earlier we called the capability to answer queries about the contents of sub-regions. It is this functor which we consider to be of primary importance; its values at various objects are just particular aspects.

We still want to continue to use “knowledge” and “information” informally, so we call $\mathfrak{K}_{\mathcal{A}}(j,t,p)$ the *ken* of (j,t,p) . We occasionally use “ken” as a verb, saying that (j,t,p) *kens* $\mathfrak{K}_{\mathcal{A}}(j,t,p)$, or say that one region kens more than another when its value under $\mathfrak{K}_{\mathcal{A}}$ is a *subset* of the other’s; when it rules out more possibilities than its counterpart.

Now observe: by definition we have $(j,t,p) : \widehat{\mathcal{A}} \upharpoonright (j,t,p) \rightarrow \widehat{\mathcal{A}}$ in $\int_{C_+} \mathfrak{K}_+$, so $(j,t,p) \in \mathfrak{K}_{\mathcal{A}}^{(1)}(j,t,p)$. This leads us to see that we can think of $\mathfrak{K}_{\mathcal{A}}$ (and $\mathfrak{K}_{\mathcal{A}}^{(1)}$) as a convoluted way of representing labellings.

Lemma 91. For all objects (j, t, p) and (j', t', p') of C_+/Σ , we have $\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p) = \mathfrak{K}_{\mathcal{A}}^{(1)}(j', t', p')$ if and only if $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}(j', t', p')$ if and only if $\widehat{\mathcal{A}} \upharpoonright (j, t, p) = \widehat{\mathcal{A}} \upharpoonright (j', t', p')$.

Proof. The “if” direction is a trivial matter of substituting equals for equals. For suppose $\widehat{\mathcal{A}} \upharpoonright (j, t, p) = \widehat{\mathcal{A}} \upharpoonright (j', t', p')$, then

$$\begin{aligned} \mathfrak{K}_{\mathcal{A}, \varepsilon}(j, t, p)(-) &= \left[\int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j, t, p) \cdot -, \widehat{\mathcal{A}}] \right] \\ &= \left[\int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j', t', p') \cdot -, \widehat{\mathcal{A}}] \right] \\ &= \mathfrak{K}_{\mathcal{A}, \varepsilon}(j', t', p')(-) \end{aligned}$$

as required. Then, since

$$\text{dom}(j, t, p) = [\widehat{\mathcal{A}} \upharpoonright (j, t, p)] = [\widehat{\mathcal{A}} \upharpoonright (j', t', p')] = \text{dom}(j', t', p'),$$

we have

$$\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p) = \mathfrak{K}_{\mathcal{A}}(j, t, p)(1_{\text{dom}(j, t, p)}) = \mathfrak{K}_{\mathcal{A}}(j', t', p')(1_{\text{dom}(j', t', p')}) = \mathfrak{K}_{\mathcal{A}}^{(1)}(j', t', p').$$

For the “only if” direction, suppose $\mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p) = \mathfrak{K}_{\mathcal{A}}^{(1)}(j', t', p')$. Then since, as observed above, $(j, t, p) \in \mathfrak{K}_{\mathcal{A}}^{(1)}(j, t, p)$, we have $(j, t, p) \in \mathfrak{K}_{\mathcal{A}}^{(1)}(j', t', p')$. Then by the definition of $\mathfrak{K}_{\mathcal{A}}^{(1)}$, we have $(j, t, p) \in \int_{C_+} \mathfrak{K}_+[\widehat{\mathcal{A}} \upharpoonright (j', t', p'), \widehat{\mathcal{A}}]$. This means that $\widehat{\mathcal{A}} \upharpoonright (j, t, p) = \widehat{\mathcal{A}} \upharpoonright (j', t', p')$. Now by using what we have already proved in the “if” direction, we see that $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}(j', t', p')$, which is what we wanted to show. \square

This means we expect to be able to recover \mathcal{A} from $\mathfrak{K}_{\mathcal{A}}$, and indeed we expect to be able to extract an automaton from any information flow which is “like $\mathfrak{K}_{\mathcal{A}}$ ”. But what is $\mathfrak{K}_{\mathcal{A}}$ like?

4.3.1 Properties which Characterize Ken

Ken has the following properties:

- **Equivalence.** For all $(j, t, p), (k, u, q) : R \rightarrow \Sigma$ and $f : S \rightarrow R$ in C_+ , we have $(j, t, p) \in \mathfrak{K}_{\mathcal{A}}(k, u, q)(f)$ if and only if $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}((k, u, q) \cdot f)$.

Proof. For the “if” direction, note that $(j, t, p) \in (\mathfrak{K}_{\mathcal{A}}(j, t, p))$, hence

$$(j, t, p) \in \mathfrak{K}_{\mathcal{A}}^{(1)}((k, u, q) \cdot f).$$

But this is just the definition of $\mathfrak{K}_{\mathcal{A}}(k, u, q)(f)$. For the “only if” direction, suppose $(j, t, p) \in \mathfrak{K}_{\mathcal{A}}(k, u, q)(f) = \mathfrak{K}_{\mathcal{A}}^{(1)}((k, u, q) \cdot f)$; then, as in the proof of Lemma 91, we must have $\widehat{\mathcal{A}} \upharpoonright (j, t, p) = \widehat{\mathcal{A}} \upharpoonright (k, u, q) \cdot f$. Now Lemma 91 itself implies that $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}((k, u, q) \cdot f)$ as required. \square

- **Grounding.** If $r : R \rightarrow \Sigma$, and $\{\chi_x : q_x \rightarrow p \mid x \in X\}$ is a cover of p in C_+/R , then

$$\mathfrak{K}_{\mathcal{A}}(r)(p) = \bigcap_{x \in X} C_+[\chi_x, \Sigma]^{\leftarrow} [\mathfrak{K}_{\mathcal{A}}(r)(q_x)]$$

Proof. By Proposition 90. \square

- **Finite Variation.** Given a bounded region R of C_+ , the set

$$\{\mathfrak{K}_{\mathcal{A}}(j, t, p) \mid j \in J, t \in \mathcal{T}, p : R \rightarrow \Sigma\}$$

is finite.

Proof. By Lemma 91, the set listed above is in bijective correspondence with the set of colourings of R , which is finite since R is bounded. \square

- **Local Determinism.** For all $(j, t, p), (j', t', p') : R \rightarrow \Sigma$ and $p : S \rightarrow R$ in C_+ , and $u \in \mathcal{T}$, if $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}(j', t', p')$ then $\mathfrak{K}_{\mathcal{A}}(j, t + u, \ominus_u p) = \mathfrak{K}_{\mathcal{A}}(j', t' + u, \ominus_u p')$.

Proof. Since $\mathfrak{K}_{\mathcal{A}}(j, t, p) = \mathfrak{K}_{\mathcal{A}}(j', t', p')$, Lemma 91 tells us that

$$\widehat{\mathcal{A}} \upharpoonright (j, t, p) = \widehat{\mathcal{A}} \upharpoonright (j', t', p').$$

Then, making use of Proposition 88, we find

$$\begin{aligned} \widehat{\mathcal{A}} \upharpoonright (j, t + u, \ominus_u p) &= \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright (j, t, p)) \\ &= \mathcal{A}(u)(\widehat{\mathcal{A}} \upharpoonright (j', t', p')) \\ &= \widehat{\mathcal{A}} \upharpoonright (j', t' + u, \ominus_u p'), \end{aligned}$$

so indeed $\mathfrak{K}_{\mathcal{A}}(j, t + u, \ominus_u p) = \mathfrak{K}_{\mathcal{A}}(j', t' + u, \ominus_u p')$, by Lemma 91 again. \square

We call an information flow \mathfrak{A} which satisfies the properties above a **deterministic information flow**. We pause to reflect on the meaning of these conditions.

Equivalence says that values of $\mathfrak{R}_{\mathfrak{A}}^{(1)}$ are equivalence classes of historically located regions. We think of this as a strong hint that an information flow comes from a syntactic object.

Grounding tells us that the information in large regions is determined by the information in regions which cover them, together with the way in which these regions are arranged. In some sense this tells us that there is no special holistic knowledge which large regions are privy to, beyond knowing the arrangement of their parts. While metaphysically unexciting, this is clearly an important property of syntactic objects. More generally, one might think of it as a “compositionality” property, if we take this talk of syntax seriously and consider $\mathfrak{R}_{\mathfrak{A}}$ a kind of semantics.

Local determinism tells us that the information presently contained in a region is enough to determine the information which will be in an eroded copy of it some time later. This is another way in which the information in regions mimics the behaviour of labellings. These labellings would be interpreted as physical states if we were to think of the automaton as a model of the dynamics of some world. In locally deterministic worlds, the flow of information in the sense of this thesis is also locally deterministic.

Finally, finite variation is the “finite information density” constraint which we expect to be complementary to restrictions on the flow of information. Over all possible courses of computation governed by a fixed algorithm, it is little surprise that a bounded region can only be asked to remember one of finitely many propositions at one time.

Given a deterministic information flow \mathfrak{A} , we want to extract an automaton from \mathfrak{A} which proves that \mathfrak{A} is in the image of \mathfrak{R}_- . We need to find an alphabet \mathfrak{A}' , and encoding $\mathfrak{E}'_{\mathfrak{A}}$ and an automaton \mathfrak{A}' such that $\mathfrak{R}_{\mathfrak{A}'} = \mathfrak{A}$.

Thanks to our care in defining $\mathfrak{R}_{\mathfrak{A}}$, we can simply set

$$\mathfrak{A}'(R) = \{\mathfrak{R}_{\mathfrak{A}}(j, t, p) \mid j \in J, t \in \mathcal{T}, p : R \rightarrow \Sigma\};$$

we have taken care to give this set restrictions (to restrict along f , compose with C_+/f),

grounding gives us separation, and finite variation is exactly the constraint that the alphabet must be finite on bounded regions. We must set $\varepsilon'_{\mathfrak{A}}(j)$ to be a labelling of $[\varepsilon j]$.

To define an encoding, we must set $\varepsilon'_{\mathfrak{A}}(j)$ to be a labelling of $[\varepsilon j]$. This is straightforward: we let

$$\varepsilon'_{\mathfrak{A}}(j) = \mathfrak{A}(\mathfrak{t}_{j,0}).$$

Now we have to define $\underline{\mathfrak{A}}'$. We let

$$\underline{\mathfrak{A}}'(u)(\mathfrak{A}(j, t, p)) = \mathfrak{A}(j, t + u, \ominus_u p);$$

this is well defined by local determinism. It clearly satisfies the constraint that its domain is $\ominus_u p$.

We need to check functoriality.

Proposition 92. *Let \mathfrak{A} be a deterministic information flow on a tissue C . Then $\underline{\mathfrak{A}}' : \mathcal{T} \times \int_C \mathfrak{X}'_{\mathfrak{A}} \rightarrow \int_C \mathfrak{X}'_{\mathfrak{A}}$ is functorial in both arguments.*

Proof. In the first argument, we compute

$$\begin{aligned} \underline{\mathfrak{A}}'(v)(\underline{\mathfrak{A}}'(u)(\mathfrak{A}(j, t, p))) &= \underline{\mathfrak{A}}'(v)(\mathfrak{A}(j, t + u, \ominus_u p)) \\ &= \mathfrak{A}(j, t + u + v, \ominus_v(\ominus_u p)) \\ &= \mathfrak{A}(j, t + u + v, \ominus_{u+v} p) \text{ by functoriality of } \ominus \\ &= \underline{\mathfrak{A}}'(u + v)(\mathfrak{A}(j, t, p)) \text{ as required.} \end{aligned}$$

Now we check functoriality in the second argument. Suppose

$$f : \mathfrak{A}(j, t, p) \rightarrow \mathfrak{A}(j', t', p'),$$

that is $\mathfrak{A}(j, t, p) = \mathfrak{A}(j', t', p') \cdot C_+ / f$. We want to check that

$$\ominus_u f : \underline{\mathfrak{A}}'(u)(\mathfrak{A}(j, t, p)) \rightarrow \underline{\mathfrak{A}}'(u)(\mathfrak{A}(j', t', p'))$$

which means

$$\underline{\mathfrak{A}}'(u)(\mathfrak{A}(j, t, p)) = \underline{\mathfrak{A}}'(u)(\mathfrak{A}(j', t', p')) \cdot C_+ / (\ominus_u f).$$

By the definition of $\underline{\mathfrak{A}}'$, this amounts to checking that

$$\mathfrak{A}(j, t + u, \ominus_u p) = \mathfrak{A}(j', t' + u, \ominus_u p') \cdot C_+ / (\ominus_u f).$$

By functoriality of \ominus , we have $\ominus_u f : \ominus_u p \rightarrow \ominus_u p'$. Then since $\mathfrak{A}(j', t', p') \cdot C_+ / f = \mathfrak{A}(j', t', p' \cdot f)$ we have

$$\mathfrak{A}(j, t, p) = \mathfrak{A}(j', t', p' \cdot f).$$

Then local determinism implies

$$\begin{aligned} \mathfrak{A}(j, t + u, \ominus_u(p)) &= \mathfrak{A}(j', t' + u, \ominus_u(p' \cdot f)) \\ &= \mathfrak{A}(j', t' + u, \ominus_u p' \cdot \ominus_u f) \\ &= \mathfrak{A}(j', t' + u, \ominus_u p') \cdot C_+ / (\ominus_u f) \end{aligned}$$

as required. \square

We now turn to checking that the constructed automaton has the correct value for ken. First, we prove a helpful technical proposition about the history of the automaton we have produced.

Proposition 93. *Given a deterministic information flow \mathfrak{A} on a tissue C , for all objects (j, t, p) of C_+ / Σ , we have $\widehat{\underline{\mathfrak{A}}}' \upharpoonright (j, t, p) = \mathfrak{A}(j, t, p)$.*

Proof. We have

$$\begin{aligned} \widehat{\underline{\mathfrak{A}}}' \upharpoonright (j, t, p) &= \widehat{\underline{\mathfrak{A}}}' \upharpoonright (\mathbf{1}_{j,t}) \cdot p \\ &= \widehat{\underline{\mathfrak{A}}}' \upharpoonright (\mathbf{1}_{j,t}) \upharpoonright p \\ &= (\underline{\mathfrak{A}}'(t)(\mathfrak{E}'_{\mathfrak{A}}(j))) \upharpoonright p \\ &= \underline{\mathfrak{A}}'(t)(\mathfrak{A}(\mathbf{1}_{j,0})) \upharpoonright p \\ &= \mathfrak{A}(\mathbf{1}_{j,t}) \upharpoonright p \\ &= \mathfrak{A}(j, t, p) \end{aligned}$$

as required. \square

Finally, we check $\mathfrak{K}_{\underline{\mathfrak{A}}}'$.

Proposition 94. *Let C be a tissue. For all deterministic information flows \mathfrak{A} on C , we have $\mathfrak{K}_{\underline{\mathfrak{A}}}' = \mathfrak{A}$*

Proof. Given $(j, t, p) : R \rightarrow \Sigma$ in C_+ we have

$$\begin{aligned} \mathfrak{K}_{\mathfrak{A}'}(j, t, p)(f) &= \{(j', t', p') \mid \widehat{\mathfrak{A}'} \upharpoonright (j', t', p') = \widehat{\mathfrak{A}'} \upharpoonright (j, t, p) \cdot f\} \\ &= \{(j', t', p') \mid \mathfrak{A}(j, t, p) = \mathfrak{A}((j, t, p) \cdot f)\} \text{ by Proposition 93} \\ &= \mathfrak{A}(j, t, p)(f) \text{ by equivalence} \end{aligned}$$

as required. □

Remark. If we start with an automaton \mathcal{A} and look at $\mathfrak{K}_{\mathcal{A}'}$, then if the alphabet is efficient in the sense that every object of $\int_C \mathfrak{K}$ admits a morphism to one of the form $\mathcal{A}(t)\varepsilon(j)$ for some $t \in \mathcal{T}$ and $j \in J$, Lemma 91 ought to imply that there is a natural isomorphism $\eta : \mathfrak{K} \cong \mathfrak{K}'_{\mathcal{A}}$ such that for all objects R of $\int_C \mathfrak{K}$ we have $\eta^{\leftarrow}(\mathfrak{K}'_{\mathcal{A}}(\eta(R))) \cong \mathcal{A}(R)$ and vice versa. This efficiency condition means that the only labellings of an object are those which actually occur in history. It is a desirable property of alphabets, which explains why we demand only separation, not collation: historical events cannot usually be glued along common geographical parts¹⁴!

Having established that $\mathfrak{K}_{\mathcal{A}}$ is just a complicated way of representing \mathcal{A} , the reader might wonder what we have gained. A first observation is that the most superficial differences between automata have disappeared: if two automata differ only by having different alphabets (but where there is a natural bijection between the alphabets which preserves the action of the automata), $\mathfrak{K}_{\mathcal{A}}$ of any region will be *equal* for the two automata, agreeing with us that there is no content in such an “isomorphism”. More substantially, we can now ask whether axiomatizing cellular automata in terms of ken might suggest directions in which to generalize which were invisible before.

4.3.2 Continuity and Computation

From the point of view of computation as a continuous flow of information over a space, some of the axioms above look too strong. In particular, equivalence, finite variation and local determinism put constraints on $\mathfrak{K}_{\mathcal{A}}$ which mention its values at different times. This means that one must either check these properties for all times or else fail to check them at all. This doesn’t look like a notion of “continuity with respect to time”, because continuity can be meaningfully said to hold locally, and checking

¹⁴The efficiency condition could also have a computational meaning if only inputs of a certain form are allowed. This would prevent someone giving an automaton from having to specify nonsense rules for updating impossible labellings.

continuity locally is sufficient to determine whether it holds globally. This is not much of a problem for finite variation, which philosophically is supposed to be the additional, complementary constraint to continuity, but equivalence and local determinism must go. We replace them with weaker properties, but in order to do so we must make an auxiliary definition. Given a pointwise subfunctor \mathfrak{A} of \mathfrak{C}_Σ , and an element $u \in \mathcal{T}$, we define another such functor, $[+u]\mathfrak{A}$ via $[+u]\mathfrak{A}(f) = \{(j, t + u, \ominus_u p) \mid (j, t, p) \in \mathfrak{A}(f)\}$. Now equivalence and local determinism imply the following.

- **Initial equivalence.** For all $(j, 0, p), (j', 0', p') : R \rightarrow \Sigma$ and $f : S \rightarrow R$ in C_+ , we have $(j, 0, p \circ f) \in \mathfrak{K}_{\mathfrak{A}}(j', 0, p')(f)$ if and only if $\mathfrak{K}_{\mathfrak{A}}(j, 0, p) = \mathfrak{K}_{\mathfrak{A}}((j', 0, p') \cdot f)$

Proof. Straightforward application of equivalence. \square

- **Continuity.** For all $(j, t, p) : R \rightarrow \Sigma$ and $f : S \rightarrow R$ in C_+ , and $u \in \mathcal{T}$, we have $[+u]\mathfrak{K}_{\mathfrak{A}}(j, t, p)(f) \subseteq \mathfrak{K}_{\mathfrak{A}}(j, t + u, \ominus_u p)(\ominus_u f)$

Proof. Suppose $(k, w, q) \in [+u]\mathfrak{K}_{\mathfrak{A}}(j, t, p)(f)$ then there is $w' \in \mathcal{T}$ and $q' \in C_+/\Sigma$ such that $w = w' + u$, $\ominus_u q' = q$, and $(k, w', q') \in \mathfrak{K}_{\mathfrak{A}}(j, t, p)(f)$. Then equivalence tells us that $\mathfrak{K}_{\mathfrak{A}}(j, t, p \cdot f) = \mathfrak{K}_{\mathfrak{A}}(k, w', q')$. Hence by local determinism we have

$$\begin{aligned} \mathfrak{K}_{\mathfrak{A}}(j, t + u, \ominus_u(p \cdot f)) &= \mathfrak{K}_{\mathfrak{A}}(k, w' + u, \ominus_u q') \\ &= \mathfrak{K}_{\mathfrak{A}}(k, w, q). \end{aligned}$$

Now a consequence of equivalence is that (k, w, q) is an element of $\mathfrak{K}_{\mathfrak{A}}^{(1)}(k, w, q)$, which equals $\mathfrak{K}_{\mathfrak{A}}^{(1)}(j, t + u, \ominus_u(p \cdot f))$ by the above calculation. But then (k, w, q) must be an element of $\mathfrak{K}_{\mathfrak{A}}(j, t + u, \ominus_u p)(\ominus_u f)$ since we have

$$\begin{aligned} \mathfrak{K}_{\mathfrak{A}}(j, t + u, \ominus_u p)(\ominus_u f) &= \mathfrak{K}_{\mathfrak{A}}^{(1)}((j, t + u, \ominus_u p) \cdot \ominus_u f) \\ &= \mathfrak{K}_{\mathfrak{A}}^{(1)}(j, t + u, \ominus_u(p \cdot f)), \end{aligned}$$

and that settles the matter. \square

We call an information flow which satisfies initial equivalence, grounding, finite variation, and continuity a **continuous information flow**.

Note that in order to state the continuity condition, we make use of the fact that values of $\mathfrak{K}_{\mathcal{A}}$ can be compared using subset inclusions, and shifted in time. These operations are not available on labellings; even though we saw above that in some sense $\mathfrak{K}_{\mathcal{A}}$ is equivalent to the syntactic presentation of the automaton, it has extra structure which the definition of continuity makes use of.

Initial equivalence, like equivalence, can be seen as a kind of consistency condition, saying that *initially* $\mathfrak{K}_{\mathcal{A}}$ can be seen assigning a historically located region to an equivalence class. We are demanding that the information flow *start* from an essentially syntactic object, although we now allow more general behaviour to occur later. This is reasonable given that an encoding will produce a syntactic object, and so a continuous information flow may proceed from any sensible notion of encoded data. Its use in what follows is to be a kind of consistency condition. A consequence of initial equivalence is that at the start of computation, a region's true location must be amongst those it considers possible.

Continuity can be seen as saying that information propagates with a bounded velocity: the information in a region at time t is at least as informative as that in its erosion by u at time $t + u$. Note that we can't quite take this literally (as a simple subset relation) because the timing information has to be dealt with by shifting the knowledge of the large region forward in time, and the difference in shape of the regions has to be accounted for by taking erosions. If the underlying tissue is cellular, then continuity simplifies to

- **Continuity.** We have $[+1]\mathfrak{K}_{\mathcal{A}}(j, t, p)(f) \subseteq \mathfrak{K}_{\mathcal{A}}(j, t + 1, \ominus p)(\ominus f)$,

which says that at time $t + 1$, a region knows at most what was known at time t by a region which erodes to it. Thinking of the latter as the smallest neighbourhood of the former, we can see this as demanding that information can enter a region only by passing through its smallest neighbourhood. This is why we think of such data as describing a continuous motion of information through the underlying space.

We want to extract an automaton from a continuous information flow \mathfrak{A} such that the ken of the extracted automaton is always at least as informative. We think of this as a certificate of effectiveness: any information which \mathfrak{A} claims to be able to put in a certain place at a certain time really can be put there by a computation in our chosen model of cellular automata. It turns out that to do this, we need to assume that

our tissue C is cellular. The strategy is the standard method for determinizing a non-deterministic computation. We replace values with sets of possible values, and check that this can be made to work.

We need to proceed carefully to define the new alphabet $\mathfrak{K}_{\mathfrak{A}}$. We want this to be a non-deterministic version of $\mathfrak{K}'_{\mathfrak{A}}$ which on a region R has the value $\mathfrak{K}'_{\mathfrak{A}}(R) = \{\mathfrak{A}(j, t, p) \mid j \in J, t \in \mathcal{T}, p : R \rightarrow \Sigma\}$. Were we simply to set $\mathfrak{K}_{\mathfrak{A}} = \mathcal{P}\mathfrak{K}'_{\mathfrak{A}}$ it would be difficult to give a notion of restriction of values which results in a separated presheaf. This is because if we restrict a lot of values of $\mathfrak{K}'_{\mathfrak{A}}$ and then glue together the restrictions in all possible ways (using the fact that \mathfrak{A} satisfies grounding), we are likely to produce values which were not in the original set. One might think of solving this problem by demanding that the values be saturated in some sense under this kind of restriction and glueing. But it is difficult to prove that this makes sense, and results in a finite number of possibilities over a bounded object, for a general tissue. For instance, one might imagine a tissue whose regions are two-dimensional subsets of the Euclidean plane, so that every region has an infinite number of arbitrarily small sub-regions. This produces an excessive number of degrees of freedom which must be accounted for.¹⁵ To prevent this problem from occurring, we ask that a bounded region is made up of a finite number of irreducible regions in a well-behaved way; that is, we ask for a cellular tissue.

We make use of the fact that our tissue is cellular by letting $\mathfrak{K}_{\mathfrak{A}}(R)$ be the set of functions whose source is the specified cover of R by cells and which maps each cell $\chi : K \rightarrow R$ to a subset of $\mathcal{P}\mathfrak{K}_{\mathfrak{A}}(K)$. These are really elements of

$$\prod_{\substack{\chi : K \rightarrow R \\ \chi \in \text{Cells}(R)}} \mathcal{P}\mathfrak{K}_{\mathfrak{A}}(K),$$

but we continue to use the language and notation of functions, writing evaluation at χ for the corresponding projection.

¹⁵In fact, if one only wanted to define a sensible alphabet, a strategy like this could be made to work. The real problem seems to come later, when one wants to define the action of the automaton. Roughly speaking, we are forced to consider the possibility of updating each sub-region independently. It is the requirement that the resulting infinite collection of data can be integrated into a datum for the whole region, while our alphabet remains separated, which causes the alphabet to blow up to infinite size. The restriction to natural numbered time is similar. One might otherwise forget less information about the past by going $t + u$ steps into the future than by first going t and then u steps. To solve this, one ensures that the least informative path is taken. Over the natural numbers, this means specifying explicitly how to do a single step, and letting general advances in time be given by iteration.

We need to specify how to restrict this data along a morphism $f : S \rightarrow R$. This is essentially restriction of functions, though we have to take a little care. Given a labelling R' of R and $\chi \in \text{Cells}(S)$, we let

$$(R' \upharpoonright p)(\chi) = R'(p \cdot \chi).$$

Note that the fact that $p \cdot \chi \in \text{Cells}(R)$ corresponds exactly to one of our demands about the selected covers by cells.

Proposition 95. *Given a continuous information flow \mathfrak{A} , the alphabet presheaf $\mathfrak{K}'_{\mathfrak{A}}$ is separated, and has a finite set of elements on a bounded object.*

Proof. We prove the finiteness condition first, then separation.

By construction, $\mathfrak{K}'_{\mathfrak{A}}$ takes on a finite number of values for a bounded region R , since in a cellular tissue a bounded region has a finite number of cells in its specified cover, and then boundedness of \mathfrak{A} means that for each of these cells K the set \mathfrak{K}'_K , and hence $\mathcal{P}\mathfrak{K}'_K$, is finite. Hence the set

$$\mathfrak{K}'_R = \prod_{\substack{\chi: K \rightarrow R \\ \chi \in \text{Cells}(R)}} \mathcal{P}\mathfrak{K}'_{\mathfrak{A}}(K)$$

is finite.

Separation holds because if we have a cover of R , then each region of the cover comes with its specified cover by cells. Composing these specified covers by cells with the morphisms in the cover of R yields $\text{Cells}(R)$ by the definition of cellular tissue. Now, if two values of \mathfrak{K}'_A on a large region are the same when restricted to all elements of the cover, then they agree on all cells, and so they are equal. \square

We think of this alphabet in the following way. If an element of $\mathfrak{K}'_{\mathfrak{A}}(C)$ is a cell trying to guess where it is in the history of computation, then an element of $\mathcal{P}\mathfrak{K}'_{\mathfrak{A}}(C)$ is a cell conjecturing about the guess which \mathfrak{A} would have had it make; it represents the frontier of possible values which our determinization of \mathfrak{A} must take into account. In a large region we will be interested in accumulating these conjectures made by the

cells. Given $R' \in \int_{C_+} (\mathfrak{K}_{\mathfrak{A}})$ with $\lfloor R' \rfloor = R$, let

$$R' * = \{ \mathfrak{A}(j, t, p) \mid \mathfrak{A}(j, t, p \cdot \chi) \in R'(\chi) \text{ for all } \chi \in \text{Cells}(R) \}.$$

One thinks of this as the set of conjectures R must make on the basis of those made by its cells. It tries to glue together every combination of the conjectures of its cells, though it may rule out those sets obtained by such a gluing procedure if they are not really values of $\mathfrak{A}(R)$.

We turn to defining an encoding $\varepsilon_{\mathfrak{A}}$; this is a simple affair. For a cell χ of Σ , we let

$$\varepsilon_{\mathfrak{A}}(j)(\chi) = \{ \mathfrak{A}(j, 0, \chi) \}.$$

Now we move on to extracting an automaton, $\underline{\mathfrak{A}}$. Since we are in the cellular case, modelling time by the natural numbers, we need only specify $\underline{\mathfrak{A}}(1)$: all other values will be given by iteration.

Given a labelling R' of a region R , how should we update it in such a way as to match \mathfrak{A} ? The labelling R' represents a set of conjectures about the value which \mathfrak{A} would have assigned to some historically located region of shape R . This set is given by $R' *$. We need to use this information to give a set of conjectures about what the value of \mathfrak{A} on $\ominus(R)$ might be at the next moment of time.

We might think of the following scheme: each conjecture in $R' *$ is something of the form $\mathfrak{A}(j, t, p)$ for $(j, t, p) : R \rightarrow \Sigma$ in C_+ . That means that from the set of conjectures about the value of \mathfrak{A} on a region, we can get a set of places where this region might be—namely, the set of morphisms $(j, t, p) : R \rightarrow \Sigma$ just considered, where $\mathfrak{A}(j, t, p) \in R' *$. Now we have an idea where R might be located in history, we can compute a set of places where its erosion might be at the next moment in time, for these will be the morphisms $(j, t + 1, \ominus(p)) : \ominus(R) \rightarrow \Sigma$ such that (j, t, p) is a possible location of R . Now we might speculate that the updated value of \mathfrak{A} on $\ominus(R)$ must be $\mathfrak{A}(j, t + 1, \ominus(p))$ for one of the possible positions of $\ominus(R)$.

However, to give a value of $\mathfrak{K}_{\mathfrak{A}(\ominus R)}$, we must supply a value *for each of its cells*. This ensures that large regions do not contain mysterious extra information coming from their broader view of the world at some previous time, which cannot be inferred

from the information presently contained in their parts. This is related to the demand that $\mathfrak{K}_{\mathfrak{A}}$ be separated, which in turn is required if we want to interpret \mathfrak{A} as a cellular automaton.

Nevertheless, the idea of the previous paragraph carries over. We use the information contained in a larger context of the cell we wish to update to work out where that cell might be located in history. Then we take the values which \mathfrak{A} takes on the set of these possible locations, but at the next moment in time.

There remains, however, the problem of how much global context to use. Suppose we have a large region R with labelling R' and a sub-region $p : S \rightarrow R$ of R . It is clear that if we are updating the value of a cell $\chi : K \rightarrow \ominus(R)$, which factors through $\ominus(p) : \ominus(S) \rightarrow \ominus(R)$, we cannot use all of the information in R' to update it. The reason is we have $p : R' \upharpoonright p \rightarrow R'$ in $\int_C \mathfrak{K}_{\mathfrak{A}}$. Then we must have $\ominus(p) : \mathfrak{A}(1)(R' \upharpoonright p) \rightarrow \mathfrak{A}(1)(R')$ in $\int_C \mathfrak{K}_{\mathfrak{A}}$. In order for this to be possible, the updated value of χ could only have used information available in $R' \upharpoonright p$. To use more of the global context available in R' would violate functoriality in the spatial argument of the “automaton”, and so it would be no automaton at all.

Now we may worry how small a neighbourhood of our cell we need to take. We need a region so small that whenever a cell factors through an erosion of a map, the chosen neighbourhood has an analogue in the smaller region. This amounts to a desire for a *smallest neighbourhood* of the cell in some sense, and this is precisely what the assignment N in the definition of cellular tissue provides us with.

Therefore, given a labelling R' of R and a cell $\chi : K \rightarrow \ominus(R)$ we let

$$\mathfrak{A}(1)(R)(\chi) = \{\mathfrak{A}(j, t + 1, \ominus p) \mid \mathfrak{A}(j, t, p) \in (R' \upharpoonright N_\chi)^*\}$$

We ought to check that this is really functorial.

Proposition 96. *Given a continuous information flow \mathfrak{A} on a tissue C , we have $\mathfrak{A}(1) : \int_C \mathfrak{K}_{\mathfrak{A}} \rightarrow \int_C \mathfrak{K}_{\mathfrak{A}}$ is a functor.*

Proof. Suppose $f : S \rightarrow R$ in $\int_C \mathfrak{K}_{\mathfrak{A}}$. Let $\chi : K \rightarrow \ominus[S]$ be a cell of $\ominus[S]$, so that $\ominus f \cdot \chi$

is a cell of $\ominus [R]$. Then we have

$$\begin{aligned} R \upharpoonright N_{\ominus f \cdot \chi} &= R \upharpoonright f \cdot N_{\chi} && \text{by the definition of cellular tissue} \\ &= R \upharpoonright f \upharpoonright N_{\chi}, \end{aligned}$$

And so

$$\underline{\mathfrak{A}}(1)(R \upharpoonright f) = \underline{\mathfrak{A}}(1)(R) \upharpoonright \ominus f,$$

which suffices to show functoriality by the definition of $\int_C \mathfrak{A}$. \square

We have now defined *an* automaton from the data of a continuous information flow. What we hope is that the ken of this automaton is always at least as informative as the data \mathfrak{A} would have supplied us with itself. We split this into two complementary parts.

First, with any determinization procedure, we want to check that the actual value of \mathfrak{A} is amongst the alternatives in the frontier of possibilities considered by our extracted automaton.

Proposition 97. *Given a continuous information flow \mathfrak{A} on a cellular tissue C , and $(j, t, p) : R \rightarrow \Sigma$ in C_+ , we have $\mathfrak{A}(j, t, p) \in (\widehat{\underline{\mathfrak{A}}} \upharpoonright (j, t, p))^*$.*

Proof. It suffices to show that for all cells $\chi \in \text{Cells}(R)$ we have

$$\mathfrak{A}(j, t, p \cdot \chi) \in ((\widehat{\underline{\mathfrak{A}}} \upharpoonright (j, t, p))(\chi)).$$

Recalling that for a cellular tissue, the monoid of times is \mathbb{N} , we do this by induction on t .

When $t = 0$ we have $(\widehat{\underline{\mathfrak{A}}} \upharpoonright (j, t, p)) = \varepsilon_{\mathfrak{A}}(j) \upharpoonright p$. Then for all cells χ of R we have

$$\begin{aligned} \mathfrak{A}(j, 0, p \cdot \chi) &\in \{\mathfrak{A}(j, 0, p \cdot \chi)\} \\ &= (\varepsilon_{\mathfrak{A}}(j) \upharpoonright p)(\chi) \end{aligned}$$

as required.

Otherwise assume by the induction hypothesis that

$$\mathfrak{A}(j, t, N_p \cdot \chi) \in (\widehat{\underline{\mathfrak{A}}} \upharpoonright (j, t, N_p \cdot \chi))^*.$$

Then we have

$$\begin{aligned}
\mathfrak{A}(j, t+1, p \cdot \chi) &= \mathfrak{A}(j, t+1, \ominus(N_{p \cdot \chi})) \\
&\in \mathfrak{A}(1)[\widehat{\mathfrak{A}} \upharpoonright (j, t, N_{p \cdot \chi})] && \text{by the definition of } \mathfrak{A} \\
&= \widehat{\mathfrak{A}} \upharpoonright (j, t+1, \ominus(N_{p \cdot \chi})) && \text{by Proposition 88} \\
&= \widehat{\mathfrak{A}} \upharpoonright (j, t+1, p \cdot \chi) && \text{by the definition of cellular tissue,}
\end{aligned}$$

which is all we wanted. \square

Second, we show a kind of consistency condition holds: every conjecture we make about the value of \mathfrak{A} on a historical region $(j, t, p) : R \rightarrow \Sigma$ (for some object R of C) is a guess about where in history (j, t, p) might be, and we want to check that it is always considered possible for (j, t, p) to be located at (j, t, p) !

Proposition 98. *Given a continuous information flow \mathfrak{A} on a cellular tissue C , and $(j, t, p) : R \rightarrow \Sigma$ in C_+ , for all $\epsilon \in (\widehat{\mathfrak{A}} \upharpoonright (j, t, p))^*$ we have $(j, t, p) \in \epsilon(1_R)$.*

Proof. By induction on t . For the base case, let $t = 0$. For all cells χ of R we have $(\epsilon_{\mathfrak{A}}(j) \upharpoonright p)(\chi) = \{\mathfrak{A}(j, 0, p \cdot \chi)\}$. By initial equivalence, $(j, 0, p \cdot \chi) \in \mathfrak{A}(j, 0, p \cdot \chi)$. Now let $\epsilon \in (\widehat{\mathfrak{A}} \upharpoonright (j, 0, p))^*$. For all cells χ of R , we have $(j, 0, p \cdot \chi) \in \epsilon(\chi)$ and so by grounding $(j, 0, p) \in \epsilon(1_R)$ as required.

We proceed to the step case. Let $\chi : K \rightarrow R$ be an element of $Cells(R)$ so $p \cdot \chi$ is a cell of $\left[\widehat{\mathfrak{A}} \upharpoonright \mathfrak{v}_{j, t+1} \right]$, and let $\mathfrak{f} \in (\widehat{\mathfrak{A}} \upharpoonright (j, t+1, p))(\chi)$. Note that we have both

$$(\widehat{\mathfrak{A}} \upharpoonright (j, t+1, p))(\chi) = (\widehat{\mathfrak{A}} \upharpoonright (j, t+1, p \cdot \chi))(1_K),$$

and, by Proposition 88,

$$\begin{aligned}
(\widehat{\mathfrak{A}} \upharpoonright (j, t+1, p))(\chi) &= (\mathfrak{A}(1)(\widehat{\mathfrak{A}} \upharpoonright \mathfrak{v}_{j, t} \upharpoonright p))(\chi) \\
&= (\mathfrak{A}(1)(\widehat{\mathfrak{A}} \upharpoonright \mathfrak{v}_{j, t}))(p \cdot \chi).
\end{aligned}$$

Then the definition of \mathfrak{A} means that there exists an object (k, u, q) of C_+/Σ such that $\mathfrak{f} = \mathfrak{A}(k, u+1, \ominus q)$, and

$$\begin{aligned}
\mathfrak{A}(k, u, q) &\in (\widehat{\mathfrak{A}} \upharpoonright \mathfrak{v}_{j, t} \upharpoonright N_{p \cdot \chi})^* \\
&= (\widehat{\mathfrak{A}} \upharpoonright (j, t, N_{p \cdot \chi}))^*.
\end{aligned}$$

Now by the induction hypothesis $(j, t, N_{p \cdot \chi}) \in \mathfrak{A}(k, u, q)(1_{\text{dom}q})$. So by continuity we have

$$\begin{aligned} (j, t+1, p \cdot \chi) &= (j, t+1, \ominus N_{p \cdot \chi}) \\ &\in \mathfrak{A}(k, u+1, \ominus q)(1_K) \\ &= \mathfrak{f}(1_K). \end{aligned}$$

Finally, given $\epsilon \in (\widehat{\mathfrak{A}} \upharpoonright (j, t+1, p))^*$, the above calculation implies that for all cells χ of R we have $(j, t+1, p \cdot \chi) \in \epsilon(\chi)$, and hence by grounding $(j, t, p) \in \epsilon(1_R)$ as required. \square

Now these combine to show that the ken of \mathfrak{A} is a good approximation to \mathfrak{A} .

Proposition 99. *Given a continuous information flow \mathfrak{A} on a cellular tissue C , and $(j, t, p) : R \rightarrow \Sigma$ in C_+ , we have $\mathfrak{K}_{\mathfrak{A}}(j, t, p) \subseteq \mathfrak{A}(j, t, p)$.*

Proof. Suppose $(k, u, q) \in \mathfrak{K}_{\mathfrak{A}}(j, t, p)(f)$. Then by equivalence (since \mathfrak{A} is a *bona fide* automaton) we have $\widehat{\mathfrak{A}} \upharpoonright (j, t, p \cdot f) = \widehat{\mathfrak{A}} \upharpoonright (k, u, q)$. Then by Proposition 97 we have $\mathfrak{A}(j, t, p \cdot f) \in (\widehat{\mathfrak{A}} \upharpoonright (j, t, p \cdot f))^*$ and hence $\mathfrak{A}(j, t, p \cdot f) \in (\widehat{\mathfrak{A}} \upharpoonright (k, u, q))^*$. But then by Proposition 98, we have $(k, u, q) \in \mathfrak{A}(j, t, p \cdot f)(1_{\text{dom}q}) = \mathfrak{A}(j, t, p)(f)$, just as we had hoped. \square

The above proposition means that for every functor like \mathfrak{A} , there is a cellular automaton which kens more than \mathfrak{A} at all times. This is what we wanted to show, but we might wonder whether this is achieved trivially, by putting lots of extra information into the initial configuration. We now show that this is not the case, by showing that any extra information put into the initial configuration of \mathfrak{A} amounts to allowing it to know that it is at the initial configuration.

Proposition 100. *Given a continuous information flow \mathfrak{A} on a cellular tissue C , a morphism $(j, 0, p) : R \rightarrow \Sigma$ in C_+ , and $f : S \rightarrow R$ in C , for all $(j', 0, p') \in \mathfrak{A}(j, 0, p)(f)$ we have $(j', 0, p') \in \mathfrak{K}_{\mathfrak{A}}(j, 0, p)(f)$.*

Proof. Suppose $(j', 0, p') \in \mathfrak{A}(j, 0, p)(f)$. By initial equivalence, we have

$$\mathfrak{A}(j', 0, p') = \mathfrak{A}(j, 0, p \cdot f).$$

Then for all $\chi \in \text{Cells}(S)$, by the definition of $\varepsilon_{\mathfrak{A}}$ we have

$$\begin{aligned} (\widehat{\mathfrak{A}} \upharpoonright (j', 0, p'))(\chi) &= \{\mathfrak{A}(j', 0, p' \cdot \chi)\} \\ &= \{\mathfrak{A}(j, 0, p \cdot f \cdot \chi)\} \\ &= (\widehat{\mathfrak{A}} \upharpoonright (j, 0, p \cdot f))(\chi), \end{aligned}$$

so $\widehat{\mathfrak{A}} \upharpoonright (j', 0, p') = \widehat{\mathfrak{A}} \upharpoonright (j, 0, p \cdot f)$, and then by Lemma 91 we have $\mathfrak{K}_{\widehat{\mathfrak{A}}}^{(1)}(j', 0, p') = \mathfrak{K}_{\widehat{\mathfrak{A}}}^{(1)}(j, 0, p \cdot f)$. Now by equivalence for $\mathfrak{K}_{\mathfrak{A}}$, we have

$$\begin{aligned} (j', 0, p') &\in \mathfrak{K}_{\mathfrak{A}}(j', 0, p')(1_S) \\ &= \mathfrak{K}_{\mathfrak{A}}^{(1)}(j', 0, p') \\ &= \mathfrak{K}_{\mathfrak{A}}^{(1)}(j, 0, p \cdot f) \\ &= \mathfrak{K}_{\mathfrak{A}}(j, 0, p)(f) \end{aligned}$$

as required. □

Remark. Given an automaton \mathcal{A} , one might wonder what the relationship is between $\mathfrak{K}_{\mathcal{A}'}$ and $\mathfrak{K}_{\mathcal{A}}$. In fact we always have $\widehat{\mathfrak{K}}_{\mathcal{A}'} \upharpoonright (j, t, p) = \{\mathfrak{K}_{\mathcal{A}} \upharpoonright (j, t, p)\}$ because $\mathfrak{K}_{\mathcal{A}}$ is a deterministic information flow. This means that when $\mathfrak{K}_{\mathcal{A}}$ computes the updated value for a cell, there will be a unique choice. Then by grounding, there is a unique updated value for any region.

4.3.3 Towards a Spatial View of Computations

In this section we sketch a way in which the notion of continuity of the last section can be re-interpreted as continuity of a path in an abstract space. The idea is that a point should be a distribution of information over the tissue we are given, so that a collection of sequences of these, one for each input, will be something of the same type as ken. The continuity condition of the previous section can be interpreted as giving a relation on these distributions. This relation will generally not be reflexive, but we may consider its reflexive closure, which corresponds to the fact that in passing from a discrete to a continuous notion of time, we must allow a process to delay for a while, doing nothing; in the discrete world, we only model what happens when the clock ticks, but in the continuous one we watch events unfold between the ticks. To do this, we need to assume that there is a fixed “whole space” ∞ which is the underlying space of the encoding of any input, and which is fixed by the interior functor. This is the case in

our paradigm example, where we would take the whole plane to be ∞ .

To that end, we let an **information distribution** be a functor from C/∞ to **Cat/Set** which is a pointwise subfunctor of the functor which sends a spatially located region $r : R \rightarrow \infty$ to the functor from C/R to **Set** which sends a region $S \rightarrow R$ located in R to the set of morphisms $S \rightarrow \Sigma$ in C_+ , and which satisfies the grounding condition.

We define a relation on the set of information distributions using continuity in the following way. An information distribution \mathfrak{X} precedes another \mathfrak{Y} if and only if for all $r : R \rightarrow \infty$ we have $[+1]\mathfrak{X}(r) \subseteq \mathfrak{Y}(\ominus(R))$. Now the reflexive closure of this relation can be thought of as an Alexandroff neighbourhood space, and Proposition 85 implies that a path in this space corresponds to a finite sequence of information assignments, each related to its successor so as to make the sequence obey the “continuity” condition above. To get something of the same type as ken, we need an infinite such sequence for each input. We can do this by going from the last point of the sequence to the point which knows nothing about where it is in the history of computation. This point is related to itself, and so we can extend to an infinite sequence by repeating it. If we have one of these paths for each input, and additionally this collection of paths begins from a point satisfying initial equivalence and altogether satisfies finite variation, then the procedure of the previous section will extract an automaton from it.

We could instead think of the space given by taking the product of one copy of the space just described for each possible input. A path through this space is a family of paths through the space just given, each of these being homotopic to a finite path. We could think of points of this space satisfying the initial equivalence condition as point where paths are allowed to begin, and then select regions which satisfy the finite variation property. We could define a notion of a point deciding a property if the property is localized at that point. Then the above would tell us that a path from a point where we are allowed to start, passing through a bounded region and finishing at a point which decides a property P corresponds to an encoding of inputs and an automaton which decides P from than encoding.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we set ourselves the question whether by studying the flow of information in a spatial model of computation, one could arrive at a space of distributions of information whose paths correspond to computations by the model. Selecting cellular automata as a model, we answer tentatively in the positive. From the sketch given at the end of the previous chapter, one could use the results of this thesis to define such a space. However, it is not yet clear whether the way in which this is done is fruitful.

One wants to know whether the space defined produces any interesting information about computations by cellular automata. For example, one ought to study whether the notion of homotopy given by the space gives an interesting equivalence relation on cellular automata.

One also wonders whether the combinatorial definition of the space is the only one possible, or whether the same space of information distributions can be described by applying some geometric construction to the base space in which the computation occurs. This would mean that viewing the model of computation as a space would provide a different perspective from which to study it, and would perhaps suggest generalizations to continuous space and time.

Finally, the definition of information used is very close to the syntax one starts with. It has the advantage that all computations by cellular automata are paths in a single, fixed space. This is done in such a way that computations which start from the

same data are represented by paths which start from the same point, even if they do different jobs—something which is difficult to imagine from a syntactic point of view. However, one would hope for a definition with more inherent logical content.

In the next section, we outline how these three issues might be addressed.

5.2 Future Work

5.2.1 First Investigations of Homotopy in the Space

It seems that a combinatorial argument is likely to show that homotopies between cellular automata are extremely rare in the space described in the previous chapter. This is because Theorem 84 implies that the image of a homotopy is made of triangles in the cospecialization relation. If a path traverses the triangle around two sides, rather than one, then the continuity condition implies very strange relationships hold for the information in certain regions: essentially, they will not be able to tell whether one tick of the clock has happened or two. This is an extremely odd property for an information flow to have. It is not certain, however, that these long paths through the underlying relation *must* be traversed by an intermediate path of a homotopy, nor that these mysterious effects cannot be later removed. It is still a disconcerting phenomenon.

This problem seems to be caused by the temporal information contained in *ken*. We could try to encourage homotopies by fiat by declaring, for example, that one need not take a step of time to forget information, so we would add edges to the relation corresponding to region-wise implications. Then, however, we would have a continuous conjunction operation which produced an output which implied its two inputs, and applying this to any two paths, we would find a path between them connected to them with sufficiently many triangles to construct a homotopy from one to the other: all paths would be homotopic. Without the extra edges, we still have a continuous conjunction operation, but we only get a series of rectangles between our paths, precisely because one must use time to forget information.

Another strategy one might think of would be to study the undirected homotopy of the space as an approximation to the directed homotopy. However, since there is a point which knows nothing, every path is homotopic to the undirected path which forgets everything it knew, and then un-forgets everything it wishes to know! In other words, the undirected homotopy of the space is trivial.

5.2.2 A Better Spatial Description

The reader may well object that what is produced in the previous chapter is not a space, but only a relation. It may well still be interesting, since we have a single relation once and for all, such that a good path through the relation gives an encoding and an automaton for deciding some property. However, it may be possible to give a more geometric construction of the space, starting from the base neighbourhood space. As a step on the way to this aim, it is worth just describing a hyperspace—a space whose points are regions of the given space—which describes how information may flow in the given space, without keeping track of the complicated details of this information. One possibility for this is to take the space where a sub-basic neighbourhood of a region is given by first selecting a neighbourhood of each of its points, and then taking all regions which are contained in the union of these neighbourhoods and which touch all of them. Given a hyperspace, it is natural to consider the universal family of regions over it given by taking the product of it with the base space, and then selecting the subspace which is the graph of the membership relation. This has a projection back onto the hyperspace. We want to know which families of regions of the base space can be obtained by pulling this universal family back along continuous maps. These pullbacks will certainly be open maps, this describes the “continuity” of the flow of information. They will also have some sort of density property, which the present author believes will turn out to imply a lifting condition—such as a path lifting condition, or something which allows local sections to be extended to global ones—which ought to be interpreted as saying that information cannot appear out of nowhere. This construction would make sense over any space, and the remaining difficulties can probably be overcome so that when given, for example, the infinite planar grid of squares, we would recover the space defined combinatorially in the previous chapter. This would allow us to ask what happens if we put in \mathbb{R}^2 as the base space, but the present analysis, with its ad-hoc definition of bounded information, will produce bad answers about the notion of computation over something like \mathbb{R}^2 : the present author expects that the present notions of continuity and bounded information will conspire to demand that automata over \mathbb{R}^2 be constant functions! This is because if we include timing information, then even a simple signal, propagating continuously through real space, must take on a continuum of positions, at a continuum of times, and distinguishing all these times from each other will quickly exceed our finite allowed amount of information. What is needed is an example of a sensible model of computation over \mathbb{R}^2 so that the right analogue of the bounded information condition can be formulated.

5.2.3 A Better Notion of Information

At the outset of this project, it was hoped that ken would be only the first notion of information studied—it turned out to be much more complicated than expected. One sees in the previous subsections that the presence of timing information causes various unpleasant issues, and it was hoped that a notion of information which did not contain timing information (like wit, or perhaps something which remembered where a region might be, as well as which inputs are plausible) might be amenable to a similar analysis. Indeed, for wit one gets suggestive relations such as

$$\mathfrak{W}(j,t,p)/\mathfrak{W}(j,t+1,p) = \mathfrak{W}(j,t,\partial p)/\mathfrak{W}(j,t+1,\partial p),$$

where if p is a subset of squares of the grid, ∂p is the “boundary” computed by taking p/p° . This looks like it is a “differential” property, saying that the information gained at an instant by a region is gained by its boundary. Wit does not have the grounding property, but it was hoped that a property like the one just given might make up for it, in some sense saying that differences of information across time behave sensibly, even if the initial endowment of information may have strange properties. However, it is not possible for the hoped for approach to work, because without the timing information, one could always accumulate information, remembering whatever you had previously learned. This is because for a bounded region, we would have a finite number of possible propositions to remember, by (an analogue of) finite variation. The set of conjunctions of these propositions would also be finite, so we could use these instead and never have to forget anything. Now given a single cell, this set would have some finite size N , and after taking N steps, we would have gone through all possibilities: we would never learn anything new, and so computation could be halted. But this N , though it would depend on the algorithm, would not depend on the input: everything which could be computed could be computed in constant time! This absurdity means that the approach taken in this thesis cannot be extended to simplifications of ken which completely forget the timing information without adding something else of a similar complexity. It might be better, rather than extracting an automaton from the flow of information, to prove that the outcome is computable by some abstract means: it is not presently known how to make progress on such a programme.

Bibliography

- [Abr13] Samson Abramsky. Two puzzles about computation. In S. Barry Cooper and Jan van Leeuwen, editors, *Alan Turing, His Work and Impact*, pages 53–57. Elsevier, 2013.
- [AD12] Pablo Arrighi and Gilles Dowek. The physical church-turing thesis and the principles of quantum theory. *International Journal of Foundations of Computer Science*, 23(05):1131–1145, 2012.
- [Adh16] M.R. Adhikari. *Basic Algebraic Topology and its Applications*. Springer India, 2016.
- [AMN12] Pablo Arrighi, Simon Martiel, and Vincent Nesme. Generalized cayley graphs and cellular automata over them. *CoRR*, abs/1212.0027, 2012.
- [AMW14] Pablo Arrighi, Simon Martiel, and Zizhu Wang. Causal dynamics of discrete surfaces. In *DCM 2013*, pages 30–40, 2014.
- [BCG⁺14] William Bialek, Andrea Cavagna, Irene Giardina, Thierry Mora, Oliver Pohl, Edmondo Silvestri, Massimiliano Viale, and Aleksandra M Walczak. Social interactions dominate speed control in poisoning natural flocks near criticality. *Proceedings of the National Academy of Sciences*, 111(20):7212–7217, 2014.
- [BDG09] Andreas Blass, Nachum Dershowitz, and Yuri Gurevich. When are two algorithms the same? *Bulletin of Symbolic Logic*, 15(2):145–168, 2009.
- [BM07] Jonathan Ariel Barmak and Elias Gabriel Minian. Minimal finite models. *Journal of Homotopy and Related Structures*, 2(1):127–140, 2007.
- [Bry01] J. L. Bryant. Handbook of geometric topology. In R.B. Sher and R.J. Daverman, editors, *Handbook of Geometric Topology*, chapter 5, pages 219–260. Elsevier Science, 2001.

- [BS97] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.
- [Cec66] Eduard Cech. *Topological spaces*. John Wiley and Sons, 1966.
- [Cla09a] Emily Clader. Homotopy theory of finite topological spaces, 2009.
- [Cla09b] Emily Clader. Inverse limits of finite topological spaces. *Homology Homotopy Appl.*, 11(2):223–227, 2009.
- [CLL⁺05] A. Cataldo, Edward A. Lee, X. Liu, E. Matsikoudis, and H. Zheng. Discrete-event systems generalizing metric spaces and fixed-point semantics. Technical Report UCB/ERL M05/12, EECS Department, University of California, Berkeley, Apr 2005.
- [CNS09] Jean Cousty, Laurent Najman, and Jean Serra. *Some Morphological Operators in Graph Spaces*, pages 149–160. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [CS07] B Jack Copeland and Oron Shagrir. Physical computation: how general are gandy’s principles for mechanisms? *Minds and Machines*, 17(2):217–231, 2007.
- [CSC10] T. Ceccherini-Silberstein and M. Coornaert. *Cellular Automata and Groups*. Springer monographs in mathematics. Springer-Verlag, 2010.
- [Den76] Dorothy E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243, May 1976.
- [DGR15] G. Dowek, P. Guillot, and M. Roman. *Computation, Proof, Machine: Mathematics Enters a New Age*. Cambridge University Press, 2015.
- [DV16] Lawrence Dunn and Jamie Vicary. Surface proofs for linear logic. *arXiv preprint arXiv:1601.05372*, 2016.
- [ES01] Martín H. Escardó and Alex K. Simpson. A universal characterization of the closed euclidean interval. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, LICS ’01*, pages 115–, Washington, DC, USA, 2001. IEEE Computer Society.

- [FJK11] Esteban Fernández-Juricic and Victor Kowalski. Where does a flock end from an information perspective? a comparative experiment with live and robotic birds. *Behavioral ecology*, page arr132, 2011.
- [Fre13] N. Fresco. *Physical Computation and Cognitive Science*. Studies in Applied Philosophy, Epistemology and Rational Ethics. Springer Berlin Heidelberg, 2013.
- [FRG06] Lisbeth Fajstrup, Martin Rauen, and Eric Goubault. Algebraic topology and concurrency. *Theoretical Computer Science*, 357(1):241 – 278, 2006.
- [Gal00] Antony Galton. Continuous motion in discrete space. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000)*. Citeseer, 2000.
- [Gan80] R. Gandy. Church’s Thesis and Principles for Mechanisms. *The Kleene Symposium*, 1980.
- [GMRT11] Eric Goles, P-E Meunier, Ivan Rapaport, and Guillaume Theyssier. Communication complexity and intrinsic universality in cellular automata. *Theoretical Computer Science*, 412(1):2–21, 2011.
- [Gog92] Joseph A Goguen. Sheaf semantics for concurrent interacting objects. *Mathematical Structures in Computer Science*, 2(02):159–191, 1992.
- [Gra09] Marco Grandis. *Directed algebraic topology: models of non-reversible worlds*. New Mathematical Monographs. Cambridge Univ. Press, Leiden, 2009.
- [Hed69] G. A. Hedlund. Endomorphisms and automorphisms of the shift dynamical systems. *Mathematical Systems Theory*, 3(4):320–375, 1969.
- [Her95] T Herron. An alternative definition of pushout diagrams and their use in characterizing k-graph machines, 1995.
- [HF89] Joseph Y. Halpern and Ronald Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, 1989.
- [Hin08] Peter Hines. Machine semantics. *Theor. Comput. Sci.*, 409(1):1–23, December 2008.

- [HKR13] M. Herlihy, D. Kozlov, and S. Rajsbaum. *Distributed computing through combinatorial topology*. Elsevier Science, 2013.
- [HLCS91] Horst Herrlich, Eva Lowen-Colebunders, and Friedhelm Schwarz. Improving top: Prtop and pstop. In *Category Theory at Work*, pages 21–34. Heldermann Verlag, 1991.
- [HMY94] Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. Algorithmic knowledge. In *Proceedings of the 5th Conference on Theoretical Aspects of Reasoning About Knowledge*, TARK '94, pages 255–266, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [HSW05] Margaret Hill, Susan Stepney, and Francis Wan. Penrose life: ash and oscillators. In *Proceedings of the 8th European conference on Advances in Artificial Life*, ECAL'05, pages 471–480, Berlin, Heidelberg, 2005. Springer-Verlag.
- [Hud69] J.F.P. Hudson. *Piecewise linear topology*. Mathematics lecture note series. W. A. Benjamin, 1969.
- [IHPS12] Katsunobu Imai, Takahiro Hatsuda, Victor Poupet, and Kota Sato. A universal semi-totalistic cellular automaton on kite and dart penrose tilings. In Enrico Formenti, editor, *DCM*, volume 90 of *EPTCS*, pages 267–278, 2012.
- [Irw11] Robert J. Irwin. *The Differential Scheme and Quantum Computation*. PhD thesis, L.C. Smith College of Engineering and Computer Science, Syracuse University, 2011.
- [Ish02] C. J. Isham. *Modern Differential Geometry for Physicists*. Allied Publ., 2002.
- [Joh02] P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*. Number v. 2 in Oxford Logic Guides. Clarendon Press, 2002.
- [Kar90] Jarkko Kari. Reversibility of 2d cellular automata is undecidable. *Physica D: Nonlinear Phenomena*, 45(1):379 – 385, 1990.
- [KM02] DC Kent and Won Keun Min. Neighborhood spaces. *International Journal of Mathematics and Mathematical Sciences*, 32(7):387–399, 2002.

- [KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, New York, NY, USA, 1997.
- [Lar06] Benoit Larose. Taylor operations on finite reflexive structures. *International Journal of Mathematics and Computer Science*, 1(1):1–26, 2006.
- [LCS93] Eva Lowen-Colebunders and G. Sonck. Exponential objects and cartesian closedness in the construct prtop. *Applied Categorical Structures*, 1(4):345–360, 1993.
- [LT04] Benoit Larose and Claude Tardif. A discrete homotopy theory for binary reflexive structures. *Advances in Mathematics*, 189(2):268 – 300, 2004.
- [McC66] Michael C McCord. Singular homology groups and homotopy groups of finite topological spaces. *Duke Math. J*, 33(3):465–474, 1966.
- [MM94] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Universitext. Springer New York, 1994.
- [MPCR11] Loc Mazo, Nicolas Passat, Michel Couprie, and Christian Ronse. Paths, homotopy and reduction in digital images. *Acta Applicandae Mathematicae*, 113(2):167–193, 2011.
- [Neu66] John Von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign, IL, USA, 1966.
- [Obt14] Adam Obtulowicz. *Causal Nets for Geometrical Gandy–Păun–Rozenberg Machines*, pages 355–372. Springer International Publishing, Cham, 2014.
- [Por03] Timothy Porter. Geometric aspects of multiagent systems. *Electronic Notes in Theoretical Computer Science*, 81:73 – 98, 2003.
- [Por04] Timothy Porter. Interpreted systems and kripke models for multiagent systems from a categorical perspective. *Theoretical Computer Science*, 323(1):235 – 266, 2004.
- [Raz12] Joseph A. B. Razavi. *Local knoweldge in models of computation*, 2012.
- [RCS⁺13] A. Ranicki, A. J. Casson, D. P. Sullivan, M. A. Armstrong, C. P. Rourke, and G. E. Cooke. *The Hauptvermutung Book: A Collection of Papers*

- on the Topology of Manifolds*. K-Monographs in Mathematics. Springer Netherlands, 2013.
- [Ros79] Azriel Rosenfeld. Digital topology. *American Mathematical Monthly*, pages 621–630, 1979.
- [RS12] C. Rourke and B. Sanderson. *Introduction to Piecewise-Linear Topology*. Springer Study Edition. Springer Berlin Heidelberg, 2012.
- [RSF⁺13] V.A. Rokhlin, C. Shaddock, D.B. Fuchs, O.Y. Viro, and S.P. Novikov. *Topology II: Homotopy and Homology. Classical Manifolds*. Encyclopaedia of Mathematical Sciences. Springer Berlin Heidelberg, 2013.
- [Sie02] Wilfried Sieg. Calculations by man & machine: Mathematical presentation. In *Proceedings of the Cracow International Congress of Logic, Methodology and Philosophy of Science*, Synthese, pages 245–260. Kluwer Academic Publishers, 2002.
- [Sie08] Wilfried Sieg. Church without dogma: Axioms for computability. In S.Barry Cooper, Benedikt Lwe, and Andrea Sorbi, editors, *New Computational Paradigms*, pages 139–152. Springer New York, 2008.
- [Smi71] Alvy Ray Smith. Cellular automata complexity trade-offs. *Information and Control*, 18(5):466–482, 1971.
- [Smi72] Alvy Ray Smith. Real-time language recognition by one-dimensional cellular automata. *Journal of Computer and System Sciences*, 6(3):233–253, 1972.
- [Smi76] Alvy Ray Smith. Introduction to and survey of cellular automata or polyautomata theory. *Automata, Language, Development*, pages 405–422, 1976.
- [Smi07] Geoffrey Smith. *Principles of Secure Information Flow Analysis*, pages 291–307. Springer US, Boston, MA, 2007.
- [Smy95] M.B. Smyth. Semi-metrics, closure spaces and digital topology. *Theoretical Computer Science*, 151(1):257 – 276, 1995. Selected Papers of the Workshop on Topology and Completion in Semantics.
- [Spa81] E.H. Spanier. *Algebraic Topology*. Springer, New York, 1981.

- [SS] Bärbel MR Stadler and Peter F Stadler. Basic properties of closure spaces. Supplementary material to [SS02], available at <https://www.bioinf.uni-leipzig.de/~studla/Publications/PREPRINTS/01-pfs-007-sub12.pdf>.
- [SS02] Bärbel MR Stadler and Peter F Stadler. Generalized topological spaces in evolutionary theory and combinatorial chemistry. *Journal of chemical information and computer sciences*, 42(3):577–585, 2002.
- [Str06] Lutz Straßburger. Proof nets and the identity of proofs. *CoRR*, abs/cs/0610123, 2006.
- [Tur36] Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936.
- [Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. <https://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
- [Web97] Julian Hilary Michael Webster. *Topology and measure theory in the digital setting: on the approximation of spaces by inverse sequences of graphs*. PhD thesis, 1997.
- [Wel96] Victoria Welby. Sense, meaning and interpretation.(i). *Mind*, (1):24–37, 1896.
- [Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM.
- [Zee63] E.C. Zeeman. *Seminar on combinatorial topology*. Number v. 1-2 in Seminar on Combinatorial Topology. and Institut des hautes études scientifiques (Paris, France), 1963.