2011

The Differential Scheme and Quantum Computation

Robert J. Irwin
Syracuse University

The Differential Scheme and Quantum Computation

By

Robert J. Irwin

Abstract of Dissertation

August 2011

It is well-known that standard models of computation are representable as simple dynamical systems that evolve in discrete time, and that systems that evolve in continuous time are often representable by dynamical systems governed by ordinary differential equations. In many applications, e.g., molecular networks and hybrid Fermi-Pasta-Ulam systems, one must work with dynamical systems comprising both discrete and continuous components.

Reasoning about and verifying the properties of the evolving state of such systems is currently a piecemeal affair that depends on the nature of major components of a system: e.g., discrete vs. continuous components of state, discrete vs. continuous time, local vs. distributed clocks, classical vs. quantum states and state evolution.

We present the *Differential Scheme* as a unifying framework for reasoning about and verifying the properties of the evolving state of a system, whether the system in question evolves in discrete time, as for standard models of computation, or continuous time, or a combination of both. We show how instances of the differential scheme can accommodate classical computation. We also generalize a relatively new model of quantum computation, the quantum cellular automaton, with an eye towards extending the differential scheme to accommodate quantum computation and hybrid classical/quantum computation.

All the components of a specific instance of the differential scheme are *Convergence Spaces*. Convergence spaces generalize notions of continuity and convergence. The category of convergence spaces, **Conv**, subsumes both simple discrete structures (e.g., digraphs), and complex continuous structures (e.g., topological spaces, domains, and the standard fields of analysis: $\mathbb{R}$ and $\mathbb{C}$). We present novel uses for convergence spaces, and extend their theory by defining *differential calculi* on **Conv**. It is to the use of convergence spaces that the differential scheme owes its generality and flexibility.

Keywords: Differential Scheme, Convergence Spaces, Quantum Computation, Hybrid Computation, Dynamical Systems, Cellular Automata.

The Differential Scheme and Quantum Computation

By

Robert J. Irwin

B.S. Antioch College, 1973

M.S. Syracuse University, 1992

Dissertation

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer and Information Science

in the Graduate School of Syracuse University

August 2011

Contents

List of Tables

List of Figures

Preface

This thesis is part of a program of work, led by Howard A. Blair, the overaching objectives of which are to:

- show that continuous and discrete state evolution (subsuming computation) are special cases of the same thing.
- lay groundwork for the treatment of *hybrid* systems, combining both discrete and continuous components.

Central to the desired unification is the concept of the *differential scheme*, due to Blair [Bla00], which is introduced in chapter 1. The differential scheme provides a framework that:

- formalizes the general idea that state evolution takes place in a *space*, over *time*, and that all evolutionary (e.g., computational) activity occurs *locally*.
- unifies discrete and continuous *dynamical systems*.

The formalization is intended to be so broad as to encompass purely discrete and purely continuous systems as special cases, and to accommodate hybrid systems naturally. The complete specification of the differential scheme rests in turn on the *convergence space* concept, which is discussed in chapter 2.

The theory and vocabulary of dynamical systems provides much of the inspiration for the present work. The phrase "dynamical system" is interpreted liberally, to include the classical continuous systems as based on ordinary differential equations (ODEs) [HS74, HW91, HW95], iterated function systems (IFSs) [Bar93], and the

computations of abstract machines, viewed as evolutions of 1-dimensional cellular automata.

The broad notions of *continuity* and *differential* that are required to support such a wide variety of systems are adduced in chapter 2. Given this generality, the set of possible interpretations of systems of ODEs is extended to include spaces different from Euclidean spaces, even from manifolds (locally Euclidean spaces). In particular, ODEs can be meaningfully interpreted over discrete as well as continuous spaces.

The convergence space-based definition of continuity used herein is not unlike the continuity concept encountered in the theory of programming languages. However, the differentials presented live in a new and very general formulation of a *differential calculus*. These differentials are not constrained by linearity concepts nor is their definition tied to the properties of any particular domain of analysis such as the real or complex numbers.

Chapter 3 provides a series of examples demonstrating the scope of the new calculi, and chapter 4 casts some significant discrete and continuous dynamical systems as instances of the differential scheme.

After reviewing quantum mechanical time evolution, and describing some related spaces of interest in chapter 5, we proceed to the quantization of cellular automata in chapter 6. We present Watrous's construction of Quantum Cellular Automata (QCA) [Wat95], a formalism that requires a notion of "quiescent state." We then show how to construct QCA without quiescent states, by using shift-invariant Lesbegue measure on Cantor space.

Open problems and future directions are discussed in chapter 7.

Acknowledgements

Without the constant guidance, assistance, example and encouragement provided by my advisor, Prof. Howard A. Blair, this thesis would not have been completed. I remember warmly our innumerable chats on a widely disparate range of subjects as a wonderful entree into the life of the mind. To him, my deepest gratitude.

I would also like to thank Prof. James S. Royer for past support and many, many hours of illuminating conversation.

The Department of Electrical Engineering and Computer Science and its chairs both present (Prof. Chilukuri K. Mohan) and past (Prof. Carlos R.P. Hartmann), gave me an outstanding variety of support and opportunities for teaching and research. Despite many changes over the years, the department remains a great place to study and work. Thanks.

The moral support of my family was sorely needed, generously given, and is hereby most gratefully and inadequately acknowledged.

Chapter 1

Introduction:

The Differential Scheme

This thesis is a contribution to a larger program of work, initiated by Blair [Bla00], the overarching objectives of which are to:

- unify continuous and discrete state evolution.
- lay the groundwork for a seamless treatment of *hybrid* dynamical systems, combining both discrete and continuous components.

Importantly, discrete state evolution subsumes classical computation. Central to the desired unification is the concept of the *differential scheme*, which we sketch in this chapter. The differential scheme provides a framework that:

- formalizes the general idea that state evolution takes place in a *space*, over *time*, and that all evolutionary (e.g., computational) activity occurs *locally*.
- unifies discrete and continuous dynamical systems via a very general concept of what forms a *differential calculus*.

The theory and vocabulary of dynamical systems provide much of the inspiration for the present work. The hybrid differential calculus regards a system of first-order ordinary differential equations (ODEs) such as

$$\frac{dx_i}{dt} = f(x_1, \ \ldots, \ x_n, t) \quad i = 1, \ \ldots, \ n$$

as a system of axioms which may be interpreted over different universes with widely disparate continuity and differentiability properties. As will be shown, the category of *convergence spaces* [Ken64, Hec03], **Conv**, each of whose objects combine a carrier with a *convergence structure*, and each of whose morphisms is a continuous function between carriers with respect to their convergence structures, is sufficiently rich and general to provide the components required by the differential scheme framework. In particular, over the same carriers, as the convergence structure varies, interpretations of a fixed system of differential equations can range from those of conventional ODEs to those of discrete state transition rules. As a consequence, dynamical systems may comprise classical continuous systems based on ODEs [HS74, HW91, HW95], iterated function systems (IFSs) [Bar93], the computations of discrete abstract machines (viewed as evolutions of 1-dimensional cellular automata; see below) and other systems, such as logic programs.

Again, the idea behind the differential scheme is to capture the essence of the concept of a dynamical system as a collection of various elements whose states evolve according to "differential" constraints, broadly interpreted. The structure of an instance of the differential scheme depends upon the structure of its constituent convergence spaces. **Conv** is a Cartesian closed category that contains hierarchies of spaces obtained by beginning with discrete structures (represented as directed graphs) and topological spaces, and then combining these spaces into hybrid spaces through, for example, Cartesian products and exponentials. Differential calculi are obtained on and between convergence spaces [BJIR07]. These calculi include con-

servative extensions of familiar differential calculi on the usual spaces of interest in analysis, e.g., real and complex Hilbert spaces.

1.1 Informal Definition of the Differential Scheme

The differential scheme comprises four types of components. An instance of the differential scheme defines a particular dynamical system by specifying a particular component of each type.

Definition 1.1.1. *An **instance of the differential scheme** consists of the following components:*

1. *a **Computation Space** (Comp)*
2. *a **Time** space (Time)*
3. *a collection of **Local State Spaces** ($\mathcal{L}$; to each $x \in Comp$ there corresponds a local state space $\tau(x) \in \mathcal{L}$)*
4. *a **Differential Calculus** ($\mathcal{D}$)*

The differential scheme was initially advanced in [Bla00], though the key differential calculus concept was not fully developed at that time.

In any instance of the differential scheme, the computation space, time, and all local state spaces are particular convergence spaces, and the differential calculus is a category whose construction begins with a collection of particular convergence spaces. The rather involved formal specification of a differential calculus is deferred to chapter 2, which discusses both convergence spaces and differential calculi in detail.

Informally, one can think of a computation space as a collection of objects, each of which is associated, at any particular point in time, with a value taken from a

fixed local state space. Time may pass continuously or discretely, depending on the particular convergence space chosen for the *Time* component. The differential calculus provides candidate differentials that determine how the values associated with points of the computation space change over time, and allow us to interpret the crucial "g is a differential of f at x" concept for the associated spaces.

It may be helpful to regard an instance of the differential scheme as a kind of generalized cellular automaton (CA). Here, the computation space is the automaton's cell space. The values a particular cell may attain are given by the local state space corresponding to that cell (note that, in general, different cells may have different local state spaces). Time acts as itself, and the associated differential calculus provides CA update rules.

A CA of the most commonly studied type has a discrete computation space, each element of which is associated with the same discrete, typically finite, local state space, and evolves in discrete time steps. Table 1.1 [Bla00] shows how other well-known types of dynamical systems model the differential scheme.

Space	*Time*	*Local States*	*Model*
discrete	discrete	discrete	Logic Program
discrete	discrete	continuous	IFS
discrete	continuous	discrete	Asynch Neural Net
discrete	continuous	continuous	ODE
continuous	discrete	discrete	discrete evolution of spatial regions
continuous	discrete	continuous	discrete evolution of one space over ano.
continuous	continuous	discrete	continuous evolution of spatial regions
continuous	continuous	continuous	PDE

Table 1.1: Classes of instances of the differential scheme

1.2 Informal Example Instances of the Differential Scheme

We present here some informal example instances of the differential scheme. Complete formalizations of some examples will be given in Chapter 3 after the necessary preliminaries on convergence spaces and differential calculi are presented in Chapter 2. In particular, specifics of the differential calculi involved are deferred.

First, some terminology. For a given instance of the differential scheme, we define the **global state space** (GlSt) to be the product of the local state spaces:

$$\mathrm{GlSt} = \Pi_{x \in \mathrm{Comp}} \tau(x)$$

We then define a **computation** within an instance of the differential scheme to be a mapping from the convergence space representing time to the convergence space representing the global state space:

$$u : \mathrm{Time} \longrightarrow \mathrm{GlSt}$$

The progression of a computation over time is determined by the differential calculus $\mathcal{D}$ that provides candidate differentials to drive the evolution of the system.

1.2.1 Discrete Dynamical Systems

Classical Computation

It is well-known that the class of cellular automata [vN66, Tof77] subsumes that of Turing machines, so classical computation fits neatly into the differential scheme. In fact, any given TM can be emulated by a 1-dimensional, radius-1 cellular automaton (or (1d,1r)-CA, for short). Here,

1. Comp = $\mathbb{Z}$ (the set of CA cells indexed by integers corresponds to a two-way infinite tape)
2. Time = $\mathbb{N}$ (discrete time steps starting from 0)
3. $\mathcal{L}$ = original TM tape alphabet, augmented with some additional symbols
4. the differentials are just the CA update rules (finite functions, in this case)

A computation (as defined above) of this instance of the differential scheme traces the progression of instantaneous descriptions of the original TM. This example will be formally presented in §4.1.1.

1.2.2 Continuous Dynamical Systems

We consider continuous dynamical systems described by systems of ordinary differential equations (ODEs). Such systems progress from the simplest ones consisting of a single ODE, to finite systems of ODEs, to systems of infinitely many ODEs. The following examples show how to interpret such continuous dynamical systems as instances of the differential scheme.

A Simple ODE

Consider the single autonomous ODE:

$$\frac{dx}{dt} = f(x) \tag{1.1}$$

As an ODE over the real domain, a solution to (1.1) is a function

$$u : \text{Time} \longrightarrow \mathbb{R}$$

such that

$$\left.\frac{du}{dt}\right|_{t_0} = f(u(t_0)), \quad \text{for all } t_0 \in \mathbb{R}$$

In differential form, we have:

$$\begin{aligned} D_{t_0}u &= \lambda\Delta.[f(u(t_0))](\Delta) \\ &= [f(u(t_0))] \end{aligned}$$

As an instance of the differential scheme, we have:

1. Comp = $\{x\}$ (a singleton set)
2. $\mathcal{L} = \{\mathbb{R}\}$
3. Time = $\mathbb{R}$
4. differentials are the linear operators on $\mathbb{R}$, of classical analysis

Thus, a solution is just a *computation*:

$$u : \text{Time} \longrightarrow \text{GlSt}$$

which is, in this case:

$$u : \mathbb{R} \longrightarrow \mathbb{R}$$

1.3 Goals and Results

The advancement of the larger program discussed at the beginning of this chapter requires the full specification of the proposed framework for the unification of discrete

and continuous dynamical systems. Our first goal, therefore, is to:

1. complete the formalization of the differential scheme begun in [Bla00].

As previously mentioned, such formalization requires the use of very flexible structures for the components of the differential scheme, which have been found in the category of convergence spaces. Accordingly, our next goal is to:

2. detail the properties of convergence spaces that lend themselves to their new application to the differential scheme.

Differential calculi, constructed from convergence spaces, form the key components of instances of the differential scheme. The generality of these structures must be demonstrated, so a related goal is to:

3. provide specific examples of differential calculi, including the continuous differential calculus of classical analysis as well as differential calculi on discrete structures (digraphs).

To show the breadth of dynamical systems supported by the differential scheme, we must:

4. provide specific examples of dynamical systems as instances of the differential scheme, including classical computation — via classical cellular automata, as previously discussed.

Just as a theory of quantum computation complementary to classical computation has been developed ([Deu85], et al), Watrous has introduced a theory of quantum cellular automata (QCA) [Wat95]. In Watrous's quantized model, however, the set of possible global QCA states is restricted to those in which all but finitely many cells are in a "quiescent state." As our final goal, we:

5. show how to construct QCA based on Watrous's model, but without quiescent states.

In the sequel, Chapter 2 reviews the structure of convergence spaces, shows that **Conv** subsumes vast categories of both continuous and discrete structures, and details the abstract structure of the differential calculi at the heart of each instance of the differential scheme; importantly, the chain rule for our differential calculi is proved here.

Chapter 3 provides examples of our generalized differential calculi on both discrete and continuous structures.

Chapter 4 casts some significant families of discrete and continuous dynamical systems as instances of the full differential scheme. In particular, we show that cellular automata and ODEs are subsumed in the differential scheme framework.

After reviewing quantum computational time evolution, and describing some related spaces of interest in chapter 5, we proceed to the quantization of cellular automata in chapter 6. In the latter chapter, we present Watrous's construction of a Quantum Cellular Automaton (QCA) [Wat95] and our rendition of this model without quiescent states.

Open problems and future prospects for research are discussed in chapter 7.

Chapter 2

Convergence Spaces and Differential Calculi

As adumbrated in Chapter 1, the power of the differential scheme to accommodate both continuous and discrete dynamical systems depends upon the properties of its constituent *convergence spaces*. Here we present sufficient of the theory of convergence spaces to show that it provides a concept of continuity suitable for both discrete and continuous structures, and from which a new concept of differentiability is developed, one that allows the unified treatment of co-evolving ensembles of discrete and continuous variables.

In classical continuous dynamical systems, the trajectory of a variable is determined by the variable's value at a point in time and, as a function of the state of the whole system (and possibly time), the variable's derivative with respect to time. We adapt this viewpoint to more general settings by appropriating the forms of the well-established theory of continuous dynamical systems while generalizing their instantiations to encompass discrete and heterogeneous systems. Our generalization devolves upon the question of how to define "g is a differential of f at x," where $f, g : X \longrightarrow Y$ for convergence spaces X and Y, and $x \in X$. That is, we may unify

discrete and continuous dynamical systems by showing how to define *differential calculi* in the category of convergence spaces, **Conv**.

2.1 Generalized Continuity & Differentiability

2.1.1 Continuity

Ever since the idea of continuity was formalized for the continua of elementary analysis, mathematicians have sought to extend it to more general structures. Their efforts initially lead to the well-known and pervasive theory of *topological spaces* [Bou49, Kel55, Wil70]. Later, *pretopological spaces* [Cho47], several non-equivalent classes of structures each referred to as the class of *filter spaces*, and Čech's *closure spaces* [Čec66] were introduced, all extending the reach of the continuity concept beyond **Top**, the category of topological spaces.

Within computer science, several research communities sought a notion of continuity suitable for mappings between various classes of discrete spaces. For example, the AI/knowledge representation community's work on modelling time, space and motion in discrete domains lead to a thread of research concerned with continuity in discrete space and/or time [All84, Dav90, RCC92, Sto97]. On another track, digital imaging researchers developed "digital topology" as a means of placing digital image processing on a rigorous basis [KKM90, KR89, Kov89].

Commonalities among such research communities were eventually recognized and common frameworks for handling discrete spaces explored. In particular, Čech's closure spaces were advanced by Smyth [Smy95] and others [Gal03] as particularly well-suited for extending the continuity concept to discrete spaces. Note that the category of closure spaces is strictly larger than **Top**. We also note in passing that categories like those of Scott domains (**Dom**), directed complete partial orders

(**Dcpo**), etc., certainly provide for continuity between discrete spaces, but these are sub-categories of **Top** via the Scott topology, so none of them actually generalize the topological concept of continuity.

Conv was chosen as the basis of the differential scheme because it is the most general among those categories whose morphisms embody a useful continuity concept; this includes all the classes of structures mentioned above. **Conv** strictly subsumes the category of closure spaces, the category of pretopological spaces (**PreTop**), most formulations of "filter space" categories, and **Top**. (**Conv** is, in fact, identical to the class of filter spaces as formulated in [Hyl79].) Significantly for the handling of discrete spaces, and for computation in particular, **Conv** contains all reflexive directed graphs[1], both finite and infinite. Between digraph objects in **Conv**, morphisms turn out to be just the "edge-preserving" maps, i.e., the continuous maps between digraphs are ordinary digraph homomorphisms. Moreover, **Conv** is a Cartesian-closed category — unlike **Top**. Three immediate consequences of this fact together with the sub-categorical relationship between **Top** and **Conv** are:

1. convergence spaces broaden the notion of continuity, while preserving the topological concept of continuity for functions between convergence spaces that are already topological spaces

2. there is a uniform way of regarding all spaces of continuous functions as convergence spaces (this allows for hybrid functional analysis)

3. the composition operation on finite products of function spaces is continuous, as is the evaluation function (this is key to differential calculi)

[1] graph reflexivity — the presence of self-loops at all nodes — is a necessary technicality, which appears to pose no limitations in practice.

2.1.2 Differentiability

As compared with generalized continuity, there has been considerably less sustained ferment among researchers concerned with generalized differentiability. However, from time to time over the past sixty years or so, some authors have made efforts to extend the differentiability concept to more general spaces [Are46, AS68, Bin66, BK66, Fox45, FB66, Kel74, Kri83, Mar63, Mic38]. Most of these formulations of differentiability have depended heavily on the special nature of the spaces involved, typically treating abstractions of the domains of classical analysis (e.g., $\mathbb{R}^n$ and $\mathbb{C}^n$) such as topological (or near-topological) vector spaces.

None of these efforts treated classes of spaces as general as **Conv**; in fact, most treatments were restricted to subclasses of **Top**. Moreover, all the structures advanced in these investigations embodied linearity in one form or another. After all, in the classical definition, the differential of function f at point x is the best *linear* approximation of f near x. Our treatment does not presuppose linear structure.

2.2 Convergence Spaces Formally Defined

A convergence space is a set together with a *convergence structure* defined upon it. Convergence structures in turn are defined with respect to the well-known *filter* concept [Car37, Bou49, Cho47, Kel55].

Recall that a **filter** on a set X is a collection of subsets of X closed under finite intersection and reverse inclusion (i.e., if A belongs to a filter on X and $A \subseteq B \subseteq X$, then B also belongs to that filter). A filter which does not include the empty set as a member is called a **proper filter**. We let $\Phi(X)$ denote the set of all filters on X, and use calligraphic letters like $\mathcal{F}$, $\mathcal{G}$ and $\mathcal{H}$ to represent filters. For $A \subseteq X$, let $[A] = \{\, B \,|\, A \subseteq B \subseteq X \,\}$. Clearly, $[A] \in \Phi(X)$; $[A]$ is called the **principal filter** at

A. For $x \in X$, we abbreviate $[\{x\}]$ by $[x]$, and call it the **point filter** at x.

Definition 2.2.1. [Ken64, Hec03] *A **convergence structure** on X is a relation $\downarrow$ (read "converges to") on $\Phi(X) \times X$ such that for each $x \in X$:*

1. *$[x] \downarrow x$, and*
2. *if $\mathcal{F} \downarrow x$ and $\mathcal{F} \subseteq \mathcal{G} \in \Phi(X)$, then $\mathcal{G} \downarrow x$.* ∎

Definition 2.2.2. [Ken64, Hec03] *A pair $(X, \downarrow)$ consisting of a set X and a convergence structure $\downarrow$ on X is called a **convergence space**.* ∎

We may refer to a convergence space $(X, \downarrow)$ by its carrier X when the particular convergence structure is understood or immaterial. Similarly, we use "$\downarrow$" to represent different convergence structures in the context of different convergence spaces, when no confusion should result.

Note that any function $f : X \longrightarrow Y$, where X and Y are sets, induces a function between powersets $\hat{f} : 2^X \longrightarrow 2^Y$ via $A \mapsto \{f(a) \mid a \in A\}$; we call the last set the **f-image** of A. Such an f also induces a function between filter collections $\hat{\hat{f}} : \Phi(X) \longrightarrow \Phi(Y)$ via $\mathcal{F} \mapsto \bigcup_{A \in \mathcal{F}} [\hat{f}(A)]$.[2] We overload notation and omit decorations $\hat{}$ and $\hat{\hat{}}$ when typing considerations make the intended interpretation unambiguous.

Definition 2.2.3. [Ken64, Hec03] *Let $f : X \longrightarrow Y$, where X and Y are convergence spaces, and let $x_0 \in X$. We say f is **continuous at** x_0 iff for each $\mathcal{F} \in \Phi(X)$, if $\mathcal{F} \downarrow x_0$ in X, then $f(\mathcal{F}) \downarrow f(x_0)$ in Y.*[3] *We say f is **continuous** iff f is continuous at x, $\forall x \in X$.* ∎

The category of convergence spaces, **Conv**, has the collection of all convergence spaces for its objects, and the collection of all continuous functions between con-

[2] Note that, while the collection of the f-images of the sets in a filter do not necessarily form a filter, the union of the upward closures of each member of this collection must be a filter.

[3] Recall that, by our overloading convention, $f(\mathcal{F})$ is shorthand for $\hat{\hat{f}}(\mathcal{F})$.

vergence spaces for its morphisms. Continuity can also be characterized in terms of filter members, which play a role analogous to that of neighborhoods in topological spaces.[4] We will rigorously define a concept of neighborhood in a convergence space shortly.

Proposition 2.2.4. *Let $f : X \longrightarrow Y$, where X and Y are convergence spaces, and let x_0 be a point of X. f is continuous at x_0 iff, for every filter $\mathcal{F}$ converging to x_0 in X, there is a filter $\mathcal{G}$ converging to $f(x_0)$ in Y such that $(\forall V \in \mathcal{G})(\exists U \in \mathcal{F})[f(U) \subseteq V]$.*

□

Definition 2.2.5. [Ken64] *A **homeomorphism** between two convergence spaces is a continuous bijection whose inverse is continuous.* ∎

Definition 2.2.6. *A **subspace** of a convergence space X is a convergence space W such that*

1. *the carrier of W is a subset of the carrier of X, and*
2. *For each point $w \in W$ and each filter $\mathcal{F} \in \Phi(W)$, $\mathcal{F} \downarrow w$ in W if and only if $\mathcal{F} = \{ A \cap W \mid A \in \mathcal{G} \}$ for some filter $\mathcal{G} \in \Phi(X)$ where $\mathcal{G} \downarrow w$ in X.* ∎

Let $\{ X_\alpha \mid \alpha \in \Gamma \}$ be an indexed family of convergence spaces. For each index α, let π_α be the projection function from the product set $\prod_{\alpha\in\Gamma} X_\alpha$ onto X_α, which maps each tuple $(x_\alpha)_{\alpha\in\Gamma}$ to its α^{th} component x_α.

Definition 2.2.7. *The **product convergence structure** on $\prod_{\alpha\in\Gamma} X_\alpha$ is defined as follows: a filter $\mathcal{F}$ on the product set converges to a point $(x_\alpha)_{\alpha\in\Gamma}$ if and only if for each index α, $\pi_\alpha(\mathcal{F}) = \{ \pi_\alpha(A) \mid A \in \mathcal{F} \}$ converges to x_α in X_α* ∎

We now define the concept of *neighborhood* in a convergence space:

[4] In the sense that a set N is a topological neighborhood of a point x iff $x \in U \subseteq N$ for some open set U.

Definition 2.2.8. *Let x be a point of a convergence space X, and let U be a subset of X. We call U a **neighborhood** of x iff U belongs to every filter converging to x.* ∎

It is easily seen that the collection of all neighborhoods of a given point is a filter.

2.3 Continuous Spaces as Convergence Spaces

In the following, we show how the category of topological spaces, **Top**, and the related category of pretopological spaces, **PreTop**, are subcategories of **Conv**. We thus establish that **Conv** subsumes all classical continuity structures, and much more besides.

2.3.1 Topological Spaces

If X is a topological space, $\mathcal{F} \in \Phi(X)$, and $x \in X$, then $\mathcal{F}$ is said to converge to x iff the collection of all open sets containing x is a sub-collection of $\mathcal{F}$. It is easily seen that $\{\,(\mathcal{F}, x) \,|\, \mathcal{F} \in \Phi(X), x \in X,$ and $\mathcal{F}$ converges to $x\,\}$ is a convergence structure on X. It is also readily verified that, for topological spaces, the convergence space notions of neighborhood, continuous function, and homeomorphism coincide with the corresponding topological notions [Cho47, Bou49, Kel55].

2.3.2 Pretopological Spaces

Definition 2.3.1. [Cho47] *A convergence space $(X, \downarrow)$ is called a **pretopological space** if and only if $\downarrow$ is a **pretopology**, i.e., for each $x \in X$, the collection of all neighborhoods of x converges to x.* ∎

Proposition 2.3.2. *Let $f : X \longrightarrow Y$, where X and Y are pretopological spaces, and*

let $x_0 \in X$. Then f is continuous at x_0 iff for every neighborhood V of $f(x_0)$, there is a neighborhood U of x_0 such that $f(U) \subseteq V$. □

The category of pretopological spaces, **PreTop**, has the collection of all pretopological spaces for its objects, and the collection of all continuous functions between pretopological spaces for its morphisms. Note that an arbitrary convergence structure $\downarrow$ on a set X induces a pretopological structure $\Downarrow$ on X in the obvious way: given a filter $\mathcal{F}$ and a point x, let $\mathcal{F} \Downarrow x$ iff the collection of all $\downarrow$-neighborhoods of x is a sub-collection of $\mathcal{F}$. This construction is idempotent: $\Downarrow$ coincides with $\downarrow$ iff $\downarrow$ is pretopological.

It is easily seen that the convergence structure of every topological space is pretopological. Furthermore, the collection of all open sets of a topological space X can be recovered from the convergence structure by letting a set be open iff it is a neighborhood of each of its members. In short, equipping each topological space with its convergence structure embeds **Top** as a full subcategory of **PreTop**. In turn, **PreTop** is a full subcategory of **Conv**.

Moreover, **Top** is embedded as a full, reflective subcategory of **PreTop**, which in turn is a full, reflective subcategory of **Conv** [BHL91, HLCS91, Hec03]. That is, if $\mathcal{A}$ is the subcategory in question and $\mathcal{B}$ is the larger category, then there is a functor $F : \mathcal{B} \longrightarrow \mathcal{A}$ such that for each object B of $\mathcal{B}$, there is an $\eta_B : B \longrightarrow FB$ in $\mathcal{B}$ such that, for each object A of $\mathcal{A}$ and each $f : B \longrightarrow A$ in $\mathcal{B}$, there is a unique $\overline{f} : FB \longrightarrow A$ in A such that $f = \overline{f} \circ \eta_B$. [5]

[5] An immediate corollary is that for all objects A and B of $\mathcal{B}$ and all arrows $f : B \longrightarrow A$ in $\mathcal{B}$, we have $\eta_A \circ f = Ff \circ \eta_B$.

2.4 Discrete Structures as Convergence Spaces

In section 2.3 we showed how topological spaces are special cases of convergence spaces. As the standard metric (and topological vector) spaces of analysis are all specialized topological spaces, it is plain that the convergence space concept encompasses the categories of ordinary continuous mathematics. To demonstrate that the convergence space concept also incorporates discrete structures, we will show how directed graphs ("digraphs") — discrete structures par excellence — are also convergence spaces.

Formally, by **digraph** we mean an ordered pair (V, E) such that V is a set, and E is a binary relation on V. For a technical reason to be discussed shortly, we limit our attention to the category **ReflDiGr** of **reflexive digraphs** — digraphs in which every vertex is adjacent to itself — and the edge-preserving maps between pairs of them. Little generality is lost by stipulating self-loops. **ReflDiGr** can be embedded, in several different ways, as a full subcategory of **Conv**. In this section, we describe two of these embeddings.

Definition 2.4.1. *The* ***convergence structure on a reflexive digraph*** (V, E) *is that obtained by letting a proper filter* $\mathcal{F}$ *on* V *converge to a vertex* x *iff* $\mathcal{F} = [y]$ *for some vertex* y *with an edge in* E *from* x *to* y. ∎

Note that, as the point filter $[x]$ must converge to x in any convergence structure, Definition 2.4.1 requires there be a edge from x to x. This is why we consider only reflexive digraphs.

Definition 2.4.2. [BJIR07]*A convergence space* X *is called* ***postdiscrete*** *if and only if every convergent proper filter is a point filter.* ∎

The category **PostDisc** comprises the postdiscrete convergence spaces as objects and the continuous functions between them as morphisms.

Proposition 2.4.3. *The postdiscrete pretopological spaces are precisely the discrete topological spaces (i.e., topological spaces in which every singleton is open).* □

It is easily seen that reflexive digraphs with the convergence structures of Definition 2.4.1 are already postdiscrete spaces. Similarly, it is easily verified that if (V_1, E_1) and (V_2, E_2) are reflexive digraphs, then a function $f : V_1 \longrightarrow V_2$ is continuous (with respect to the induced convergence structures on V_1 and V_2) iff it is a digraph homomorphism; i.e., for all edges (x, y) in E_1, the edge $(f(x), f(y))$ is present in E_2. Furthermore, given any reflexive digraph (V, E), the edge set E can be recovered from the induced convergence structure on V by drawing an edge from x to y iff $[y]$ converges to x.

Proposition 2.4.4. *The construction in Definition 2.4.1 is an isomorphism between* ***ReflDiGr*** *and* ***PostDisc****. In turn,* ***PostDisc*** *is a full subcategory of* ***Conv****.* □

We note that **PostDisc** is a full, co-reflective subcategory of **Conv**. That is, there is a functor F: **Conv** $\longrightarrow$ **PostDisc** such that for each object B of **Conv**, there is an $\eta_B : FB \longrightarrow B$ in $\mathcal{B}$ such that, for each object A of **PostDisc** and each $f : A \longrightarrow B$ in **Conv**, there is a unique $\hat{f} : A \longrightarrow FB$ in **PostDisc** such that $f = \eta_B \circ \hat{f}$. [6]

Alternatively, **ReflDiGr** can be embedded as a full subcategory of **PreTop** by letting a filter $\mathcal{F}$ converge to a vertex x of a digraph (V, E) iff $\{\, y \,|\, (x, y) \in E \,\}$ is a member of $\mathcal{F}$ [SSWF01, RS03, Ale37].

The edge set of a digraph can be recovered from the induced pretopology in exactly the same way that it can be recovered from the induced postdiscrete structure on the digraph. (Cf. *specialization order* [Ale37, GD71, GHK+80, Joh82].)

In general, the induced pretopology on a reflexive digraph is weaker than the induced postdiscrete structure on the digraph.

[6] An immediate corollary is that for all objects A and B of **Conv** and all arrows $f : A \longrightarrow B$ in **Conv**, we have $f \circ \eta_A = \eta_B \circ Ff$. See [AM75], [Sch01], [AHS90], [Her68], or [Mac71].

Proposition 2.4.5. *For each reflexive digraph (V, E), the pretopological structure on V induced by E coincides with the pretopological structure induced by the post-discrete structure induced by E.* □

The reflexive digraphs whose induced pretopologies are topological are precisely the preordered sets; i.e., those digraphs in which the underlying binary relation is transitive as well as reflexive. [SSWF01, RS03].

The reflexive digraphs which are T_0 spaces (topological spaces in which every non-empty indiscrete subspace is a singleton) with respect to their induced pretopologies are precisely the posets; i.e., those digraphs in which the underlying binary relation is a partial order (cf. [Ale37]). The induced pretopology on a poset (V, E) is the pretopology induced by the well-known *Alexandroff topology* on V [Ale37, GD71, GHK+80, Joh82].

The following table summarizes the convergence space-related categories treated so far:

Table 2.1: **Conv**-related categories

Conv	the category of convergence spaces and continuous functions
PostDisc	the category of postdiscrete spaces and continuous functions
PreTop	the category of pretopological spaces and continuous functions
ReflDiGr	the category of reflexive digraphs and digraph homomorphisms
Top	the category of topological spaces and continuous functions

2.5 Function Spaces

Unlike **Top** and **PreTop**, **Conv** is a Cartesian closed category ([Mac71, AM75, AHS90, Sch01, Kat65]). The relevant supporting definitions and propositions follow.

Definition 2.5.1. *[Kat65] Let X and Y be convergence spaces. The **function space** Y^X is the set of all continuous functions from X to Y, equipped with the*

convergence structure $\downarrow$ *defined as follows: for each* $\mathcal{H} \in \Phi(Y^X)$ *and each* $f_0 \in Y^X$, *let* $\mathcal{H} \downarrow f_0$ *iff for each* $x_0 \in X$ *and each* $\mathcal{F} \downarrow x_0$, $\{\bigcup_{f \in H} f(F) \,|\, H \in \mathcal{H}, F \in \mathcal{F}\}$ *is a base for a filter which converges to* $f_0(x_0)$ *in* Y. ∎

Definition 2.5.1 is essentially the same as the definition of function graphs in [IK00]. This is not surprising as **ReflDiGr** is a Cartesian closed category (cf. [AHS90]).

Proposition 2.5.2. *[Bin66, Bin75] Let* X *and* Y *be convergence spaces. Then the evaluation function from* $Y^X \times X$ *to* Y *is continuous.* □

The following is a very useful consequence of Proposition 2.5.2 (cf. [Mac71, AM75, AHS90]):

Corollary 2.5.3. *Let* X, Y, *and* Z *be convergence spaces. Then the composition function from* $Z^Y \times Y^X$ *to* Z^X *is continuous.* □

2.6 Translation Groups and Homogeneous Convergence Spaces

Definition 2.6.1. *An* ***automorphism*** *of a convergence space* X *is a homeomorphism* $f : X \longrightarrow X$.

Definition 2.6.2. *A* ***translation group*** *on a convergence space* X *is a group* T *of automorphisms of* X *such that, for each pair of points* p *and* q *of* X, *there is* at most *one member of* T *which maps* p *to* q. *In general, we will denote this unique member of* T *(if it exists) by* $(q - p)$. ∎

Notation: The group operation (namely, composition) of a translation group T on a convergence space X will be written additively, whether or not T is Abelian. Furthermore, for all $\tau \in T$ and all $x \in X$, we will write $\tau(x)$ as $x + \tau$. In this notation,

the requirement that the translation $(q-p)$ (if it exists) maps p to q becomes the familiar requirement that if $(q-p)$ exists, then $p+(q-p)=q$

Definition 2.6.3. *A **full translation group** on a convergence space X is a translation group on X which contains a translation $(q-p)$ for each pair of points p and q.* ∎

Definition 2.6.4. *A convergence space X is **homogeneous** iff for each pair of points p and q of X, there is an automorphism of X which maps p to q* ∎

Note that a convergence space which has a full translation group must be homogeneous. Moreover, a full translation group on a nonempty convergence space X must have the same cardinality as X.

Proposition 2.6.5. [BJ]

i. *Every convergence space X can be embedded as a subspace of a convergence space HX which has a full translation group.*

ii. *X and HX have the same cardinality if and only if the cardinality of X is either infinite or zero.*

iii. *The embedding of X into HX is onto HX if and only if X is empty.*

iv. *If X and Y are convergence spaces, then every continuous function $f: X \longrightarrow Y$ can be be extended to a continuous function $Hf: HX \longrightarrow HY$.*

v. *If f is a homeomorphism, then so is Hf.* □

An immediate consequence of (ii) above is that, if X is a non-empty finite convergence space with (or without) a full translation group, then HX cannot be homeomorphic to X. It is important to note that, in general, a continuous extension Hf of f need not be unique.

2.7 Differential Calculi on Convergence Spaces

In the classical differential calculus, differentials of a finite set of basic functions are obtained *ab initio*; e.g., the identity function is its own differential. Differentials of functions generated from the basic functions by generalized composition are obtained via the *chain rule*: differentiation distributes over composition.[7] Thus, the familiar rule for differentiation of products follows from the chain rule once we know the differential of the multiplication operation.

Of course, the chain rule is not the whole story in differential calculi. Many familiar results in advanced calculus are barely touched by its consequences. However, such results are consequences of the details of both the convergence structures, the algebraic structures, and the relationship between these structures on the particular spaces under consideration. The functions chosen to serve as differentials in setting up a particular differential calculus, *together* with the convergence structures on the spaces involved in the calculus, jointly determine the additional properties of the calculus. The chain rule, however, holds in all of them.

Definition 2.7.1. *A **differential calculus** is a category $\mathcal{D}$ in which*

i. every object of $\mathcal{D}$ is a triple $\mathcal{X} = (X, 0, T)$, where X is a convergence space, 0 is a point of X (called the origin *of $\mathcal{X}$), and T is a full translation group on X.*

ii. every arrow in $\mathcal{D}$ from an object $(X, 0_X, T_X)$ to an object $(Y, 0_Y, T_Y)$ is a continuous function from X to Y which maps 0_X to 0_Y

iii. composition of arrows in $\mathcal{D}$ is function composition.

iv. for every object $\mathcal{X} = (X, 0_X, T_X)$, the identity function on X is an arrow in $\mathcal{D}$ from $\mathcal{X}$ to $\mathcal{X}$

[7] The usefulness of Cartesian-closure for obtaining a robust chain-rule was pointed out in [Kri83].

v. for each pair of objects $\mathcal{X} = (X, 0_X, T_X)$ *and* $\mathcal{Y} = (Y, 0_Y, T_Y)$*, the constant function mapping every point of* X *to* 0_Y *is an arrow in* $\mathcal{D}$ *from* $\mathcal{X}$ *to* $\mathcal{Y}$

The arrows of a differential calculus $\mathcal{D}$ *are called* $\mathcal{D}$-***differentials***. ∎

By Proposition 2.6.5, the requirement that each object have a full translation group is not unduly restrictive.

2.8 Differentiability

Let $a \in A \subseteq X$ and let $B \subseteq Y$, where $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus $\mathcal{D}$. Let $f : A \longrightarrow B$ be an arbitrary function.

Let $L \in \mathcal{D}(\mathcal{X}, \mathcal{Y})$, where $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ is the set of all $\mathcal{D}$-differentials from $\mathcal{X}$ to $\mathcal{Y}$, equipped with the subspace convergence structure inherited from the function space Y^X in **Conv**.

Definition 2.8.1. *L is a* ***differential*** *of f at a iff*
for every $\mathcal{F} \downarrow a$ in A, there is some $\mathcal{H} \downarrow L$ in $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that

i. $\mathcal{H} \subseteq [L]$*, and*

ii. for every $H \in \mathcal{H}$, there is some $F \in \mathcal{F}$ such that
for every point $x \in F$, there is at least one $\mathcal{D}$-differential $t \in H$ such that

$$t(x - a) = f(x) - f(a)$$

∎

With respect to the preceding definition, $(f(a) - 0_Y) \circ t \circ (0_X - a)$ is called an **extrapolant** of f through $(a, f(a))$ and $(x, f(x))$.

Definition 2.8.2. *A function from A to B is* ***differentiable*** *(respectively,* ***uniquely differentiable****) at a point a iff it has* at least *one (respectively,* precisely *one) differential at a.*

A function from A to B is ***differentiable*** *(respectively,* ***uniquely differentiable****) iff it is differentiable (respectively, uniquely differentiable) at each point of A.* ∎

2.9 The Chain Rule

As previously mentioned, the chain rule plays a central role in our differential calculi. In elementary analysis, for example, the product rule follows from the chain rule after obtaining the differential of the multiplication operation. Expressed in terms of differentials, the product rule for real-valued functions of a real variable reduces to matrix multiplication (i.e., composition of linear functions):

$$\begin{aligned} D_x(\text{mult} \circ (f,g)) &= (D_{(f,g)(x)}\text{mult}) \circ (D_x(f,g)) \\ &= (D_{(f(x),g(x))}\text{mult}) \circ (D_x f, D_x g) \\ &= [g(x) \quad f(x)] \begin{bmatrix} D_x f \\ D_x g \end{bmatrix} \\ &= g(x)D_x f + f(x)D_x g \end{aligned}$$

Returning to our more general setting, let $a \in A \subseteq X$, let $B \subseteq Y$, and let $C \subseteq Z$, where $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ are objects of a differential calculus $\mathcal{D}$. Let $f : A \longrightarrow B$ and $g : B \longrightarrow C$ be arbitrary functions. Finally, let $K : \mathcal{X} \longrightarrow \mathcal{Y}$ and $L : \mathcal{Y} \longrightarrow \mathcal{Z}$ be $\mathcal{D}$-differentials.

Theorem 2.9.1. *(****Chain Rule****)*

Suppose that f is continuous at a. Also suppose that K is a differential of f at a, and L is a differential of g at $f(a)$.

Then $L \circ K$ is a differential of $g \circ f$ at a.

Proof: Let $\mathcal{F}$ be a filter converging to a in X. Since K is a differential of f at a, there is some $\mathcal{G} \downarrow K$ in $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that $\mathcal{G} \subseteq [K]$ and, for every $G \in \mathcal{G}$, there is some $F_{1,G} \in \mathcal{F}$ such that for each point $x \in F_{1,G}$ there is some $\mathcal{D}$-differential $s_{G,x} \in G$ such that

$$s_{G,x}(x - a) = f(x) - f(a) \tag{2.1}$$

On the other hand, since f is continuous at a, we have $f(\mathcal{F}) \downarrow f(a)$ in B. Since L is a differential of g at $f(a)$, there is some filter $\mathcal{H} \downarrow L$ in $\mathcal{D}(\mathcal{Y}, \mathcal{Z})$ such that $\mathcal{H} \subseteq [L]$ and, for every $H \in \mathcal{H}$, there is some $N_H \in f(\mathcal{F})$ such that for each point $y \in N_H$, there is some $\mathcal{D}$-differential $t_{H,y} \in H$ such that

$$t_{H,y}(y - f(a) = g(f(x)) - g(f(a)) \tag{2.2}$$

Consider such a set N_H. By definition, $N_H \in f(\mathcal{F})$, i.e. there is some $F_{2,H} \in \mathcal{F}$ such that

$$f(F_{2,H}) \subseteq N_H$$

By (2.2), for each point $x \in F_{2,H}$, we have

$$t_{H,f(x)}(f(x) - f(a)) = g(f(x)) - g(f(a)) \tag{2.3}$$

Next, note that $\{ \{ h_2 \circ h_1 \,|\, h_1 \in G, h_2 \in H \} \,|\, G \in \mathcal{G}, H \in \mathcal{H} \}$ is a basis for a filter $\mathcal{J}$ on $\mathcal{D}(\mathcal{X}, \mathcal{Z})$, and that $\mathcal{J} \subseteq [L \circ K]$.

By joint continuity of composition (Corollary 2.5.3), $\mathcal{J} \downarrow L \circ L$ in $\mathcal{D}(\mathcal{X}, \mathcal{Z})$.

Let J be an arbitrary member of $\mathcal{J}$. There exist $G \in \mathcal{G}$ and $H \in \mathcal{H}$ such that

$$\{ h_2 \circ h_1 \,|\, h_1 \in G, h_2 \in H \} \subseteq J$$

Let $F = F_{1,G} \cap F_{2,H}$. Then $F \in \mathcal{F}$. For each point $x \in F$, we have $s_{G,x} \in G$ and $t_{H,f(x)} \in H$, and therefore $t_{H,f(x)} \circ s_{G,x} \in J$. Furthermore, by (2.1) and (2.3),

$$t_{H,f(x)}(s_{G,x}(x-a)) = t_{H,f(x)}(f(x) - f(a)) = g(f(x)) - g(f(a)) \tag{2.4}$$

Thus, $(g(f(a)) - 0_Z) \circ t_{H,f(x)} \circ s_{G,x} \circ (0_X - a)$ is the required extrapolant of $g \circ f$ through $(a, g(f(a))$ and $(x, g(f(x))$.

Since the selection of x is arbitrary (once G and H have been chosen), $L \circ K$ is indeed a differential of $g \circ f$ at a. □

2.10 Differential Calculi With Non-homogeneous Objects

In the preceding specification of differential calculi (see Definition 2.7.1), the convergence spaces acting as carriers of objects of a differential calculus must come equipped with full translation groups. We can generalize the concept of differential calculus so that *all* convergence spaces may serve as carriers of objects.

Observation 2.10.1. *Let T be a translation group on a convergence space X. For each point x of X, let $[x]_T$ be the T-* orbit of x, *i.e. $[x]_T = \{ x + \tau \,|\, \tau \in T \}$.*

If X is nonempty, then the set of all T orbits partitions X into homogeneous subspaces. For each T-orbit $[x]_T$, the restrictions of the members of T to $[x]_T$ form a full translation group $T_{[x]}$ on $[x]_T$.

Each $T_{[x]}$ is a quotient group of T. □

Definition 2.10.2. *A **system of origins** for a convergence space X with respect to a translation group T is a set of representatives of the T-orbits, i.e., a subset $\mathcal{O}$ of X containing precisely one member of each T-orbit.*

For each point x of X, let 0_x be the unique member of $\mathcal{O}$ belonging to the same T-orbit as x. ∎

Definition 2.10.3. *Let $f : X \longrightarrow Y$ be a function between convergence spaces. Let $\mathcal{O}_X$ ($\mathcal{O}_Y$, respectively) be a system of origins for X with respect to a translation group S (for Y with respect to a translation group T, respectively).*

*i. f will be said to **respect orbits** iff, for each pair of points p and q of X, if p and q lie in the same S-orbit, then $f(p)$ and $f(q)$ lie in the same T-orbit.*

*ii. f will be said to be **preserve origins** iff $f(\mathcal{O}_X) \subseteq \mathcal{O}_Y$.*

Definition 2.10.4. *A **generalized differential calculus** is a category $\mathcal{D}$ in which*

i. every object of $\mathcal{D}$ is a triple $\mathcal{X} = (X, T, \mathcal{O})$ such that X is a convergence space, T is a translation group on X, and $\mathcal{O}$ is a system of origins for X with respect to T.

ii. every arrow in $\mathcal{D}$ from an object $(X, S, \mathcal{O}_X)$ to an object $(Y, T, \mathcal{O}_Y)$ is a continuous, orbit-respecting, origin-preserving function from X to Y.

iii. composition of arrows in $\mathcal{D}$ is function composition.

iv. for every object $\mathcal{X} = (X, T, \mathcal{O})$, the identity function on X is an arrow in $\mathcal{D}$ from $\mathcal{X}$ to $\mathcal{X}$

v. for each pair of objects $\mathcal{X} = (X, S, \mathcal{O}_X)$ *and* $\mathcal{Y} = (Y, T, \mathcal{O}_Y)$ *and each* ζ *in* $\mathcal{O}_Y$*, the constant function mapping every point of* X *to* ζ *is an arrow in* $\mathcal{D}$ *from* $\mathcal{X}$ *to* $\mathcal{Y}$ ∎

A differential calculus, in the sense of Definition 2.7.1 is essentially a generalized differential calculus in which the translation group of every object is a full translation group.

At the opposite extreme, there are generalized differential calculi in which the translation group of every object is trivial (i.e., all orbits are singletons).

Example 2.10.5. ***CONV as a generalized differential calculus***

The objects of the trivial generalized differential calculus *are* all *convergence spaces, equipped with trivial translation groups. The arrows from an object* X *to an object* Y *are* all *continuous functions from* X *to* Y*. (Since all orbits are singletons,* every *function is orbit-respecting and origin-preserving.)*

Now, let $a \in X$ and let $b \in Y$, where $\mathcal{X} = (X, S, \mathcal{O}_X)$ and $\mathcal{Y} = (Y, T, \mathcal{O}_Y)$ are objects of a generalized differential calculus $\mathcal{D}$. Let $f : A \longrightarrow B$ be an arbitrary function. Let $L \in \mathcal{D}(\mathcal{X}, \mathcal{Y})$, where, again, $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ is the set of all arrows in $\mathcal{D}$ from $\mathcal{X}$ to $\mathcal{Y}$, equipped with the subspace convergence structure inherited from the function space Y^X in **Conv**.

Definition 2.10.6. *L is a* ***differential*** *of f at a iff*
for every $\mathcal{F} \downarrow a$ in X, there is some $\mathcal{H} \downarrow L$ in $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that

i. $\mathcal{H} \subseteq [\{L\}]$*, and*

ii. for every $H \in \mathcal{H}$*, there is some* $F \in \mathcal{F}$ *such that*
for every point $x \in F$*, there is at least one function* $t \in H$ *such that*

$$t(x + (0_x - a)) = f(x) + (0_{f(x)} - f(a))$$

∎

Differentiability and unique differentiability are defined precisely as in Definition 2.8.2, though, of course, with respect to the immediately preceding definition of differential.

2.11 The Chain Rule for Generalized Differential Calculi

Let $a \in X$, , where $\mathcal{X} = (X, R, \mathcal{O}_X)$, $\mathcal{Y} = (Y, S, \mathcal{O}_Y)$, and $\mathcal{Z} = (Z, T, \mathcal{O}_Z)$ are objects of a generalized differential calculus $\mathcal{D}$. Let $f : X \longrightarrow Y$ and $g : Y \longrightarrow Z$ be arbitrary functions. Let $K : \mathcal{X} \longrightarrow \mathcal{Y}$ and $L : \mathcal{Y} \longrightarrow \mathcal{Z}$ be arrows of $\mathcal{D}$.

Theorem 2.11.1. ***(Chain Rule)***

Suppose that f is continuous at a. Also suppose that K is a differential of f at a, and L is a differential of g at $f(a)$.

Then $L \circ K$ is a differential of $g \circ f$ at a.

Proof: The proof is like that for Theorem 2.9.1, *mutatis mutandis*. □

Chapter 3

Examples of Differential Calculi

3.1 Classical Differential Calculi

Let $\mathbb{R}$ be the real line, equipped with its usual Euclidean topology, and let $\mathbb{N}$ be the set of all natural numbers.

3.1.1 The classical differential calculus of real variables

The carriers of the objects of this differential calculus are the sets $\mathbb{R}^n$ ($n \in \mathbb{N}$), equipped with their respective Euclidean topologies, with their respective zero vectors as origins, and with their usual translation groups, which are full as required of a differential calculus (cf. Definition 2.7.1).

The arrows of this calculus are just the $\mathbb{R}$-linear functions. This comports with the remaining requirements for a differential calculus: the $\mathbb{R}$-linear functions are continuous with respect to the convergence structures induced by the Euclidean topologies (cf. §2.3.1), compositions of $\mathbb{R}$-linear functions are also $\mathbb{R}$-linear, and the identity map and all zero-constant maps are also $\mathbb{R}$-linear functions.

In this calculus, our choice of arrows guarantees unique differentiability, and a function f has a differential at a point p (according to Definition 2.8.1) iff f is differentiable (under the classical definition) at p.

3.1.2 The directional calculus of real variables

The objects of this calculus are the sets $\mathbb{R}^n$ ($n \in \mathbb{N}$), with their respective zero vectors as origins, and with their usual translation groups. Here, $\mathbb{R}^1$ is equipped with the Euclidean topology, but for $n > 1$, the convergence structure imposed on $\mathbb{R}^n$ is stronger than the Euclidean product structure of the previous example.

In the directional calculus of real variables, a filter $\mathcal{F}$ will be said to converge to a point p iff there is some unit vector q such that $\{ p + \alpha q \,|\, \alpha \in \mathbb{R}, |\alpha| < \epsilon \} \in \mathcal{F}$ for every real number $\epsilon > 0$.

The differentials of this calculus are the $\mathbb{R}$-homogeneous functions of degree one. (A function f between modules over a ring R is said to be *R-homogeneous of degree n* iff

$$f(\alpha x) = \alpha^n f(x)$$

for all scalars α and all vectors x.)

In this calculus, differentiability and unique differentiability are again equivalent, but a function f has a differential at a point p iff f has directional derivatives in all directions at p.

3.1.3 The classical affine differential calculus of real variables

The objects of this generalized differential calculus (cf. Definition 2.10.4) are the Euclidean spaces, equipped with trivial translation groups. The arrows from $\mathbb{R}^m$ to

$\mathbb{R}^n$ are all $\mathbb{R}$-affine functions from $\mathbb{R}^m$ to $\mathbb{R}^n$.

As in the classical linear differential calculus, a function f has a differential at a point p iff f is differentiable (in the usual sense) at p.

Let $A_p f$ be the differential of f at p in the classical affine differential calculus of real variables. That is, $A_p(f)$ is the affine function which best approximates f in arbitrarily small neighhborhoods of p.

Then the differential of f at p in the classical linear differential calculus is the unique linear function which can be obtained from $A_p f$ by composing it on both sides with translations (in the usual sense), i.e., the function which maps each point q to $f(p) + (A_p f)(q - p)$

3.2 Differential Calculi on Digraphs

We specify here several different, pairwise nonequivalent differential calculi on reflexive digraphs. Specialized to reflexive digraphs, our general definition of a differential calculus (cf. Definition 2.7.1) becomes:

Definition 3.2.1. *A **differential calculus on reflexive digraphs** is a category $\mathcal{D}$ in which*

*i. every object of $\mathcal{D}$ is a triple $\mathcal{X} = (X, 0, T)$ such that X is a reflexive digraph, 0 is a vertex of X (called the **origin of $\mathcal{X}$**), and T is a group of digraph homomorphisms (functions which preserve directed edges) which acts regularly[1] on the vertices of X. T is called the **translation group** of $\mathcal{X}$.*

ii. each arrow from an object $(X, 0_X, T_X)$ to an object $(Y, 0_Y, T_Y)$ is an origin-preserving digraph homomorphism from X to Y.

[1] that is, for any two x, y in X there exists precisely one τ in T such that $\tau(x) = y$.

iii. composition of arrows is function composition

iv. for every object $\mathcal{X} = (X, 0_X, T_X)$*, the identity function on* X *is an arrow in* $\mathcal{D}$ *from* $\mathcal{X}$ *to* $\mathcal{X}$

v. for each pair of objects $\mathcal{X}$ *and* $\mathcal{Y}$*, the constant function mapping every vertex of* $\mathcal{X}$ *to the origin of* $\mathcal{Y}$ *is an arrow in* $\mathcal{D}$ *from* $\mathcal{X}$ *to* $\mathcal{Y}$*.*

The arrows of a differential calculus $\mathcal{D}$ *are called* $\mathcal{D}$-***differentials***.

In this setting, the differences in the calculi to be defined must appear among the functions chosen as $\mathcal{D}$-differentials.

Note that having self-loops at each vertex (reflexivity) ensures that every constant function between vertex sets is a digraph homomorphism. Of course, the convergence structures induced on digraphs (cf. Definition 2.4.1), which guarantee that the continuous functions between digraphs are just ordinary graph homomorphisms, already require self-loops.

When only one differential calculus $\mathcal{D}$ is involved, we will refer to a $\mathcal{D}$-differential simply as a differential.

For each vertex v of X, where $\mathcal{X} = (X, 0, T)$ is an object of a differential calculus, let τ_v be the unique member of T mapping the origin to v. Note that, since X has at least one vertex (namely, the origin), the mapping $v \mapsto \tau_v$ is a bijection between X and T.

Observation 3.2.2. *Let* $\mathcal{X} = (X, 0, T)$ *be an object of a differential calculus. Then*

1. τ_0 *is the identity map on* X
2. *for each vertex* v *of* X*,* $\tau_v(0) = v$
3. *for each vertex* v *of* X*,* ${\tau_v}^{-1} = \tau_{-v}$*, where* $-v = {\tau_v}^{-1}(0)$

4. for all vertices u and v of X, $\tau_v \circ \tau_u = \tau_{u+v}$, where $u + v = \tau_v(u)$ □

Equivalently, therefore, by identifying the carrier of the translation group of an object with the set of vertices of the underlying digraph, we could define a differential calculus on reflexive digraphs as a concrete category $\mathcal{D}$ in which

1. every object of $\mathcal{D}$ is a set X, equipped with both a group structure (written additively), and a reflexive binary relation on X, such that, for each a in X, right translation by a preserves the binary relation.

2. each $\mathcal{D}$-morphism from an object X to an object Y is a digraph homorphism from X to Y which maps the origin of X to the origin of Y, where, by the *origin* of an object, we mean the identity element of the object's group operation.

3. for every object X, the identity function on X is a $\mathcal{D}$-morphism from X to X.

4. for each pair of objects X and Y, the constant function mapping every member of X to the origin of Y is a $\mathcal{D}$-morphism from X to Y.

We will shift back and forth between these two viewpoints as is convenient. In particular, if a and x are vertices of the underlying digraph of an object of a differential calculus on reflexive digraphs, we will almost always write ${\tau_a}^{-1}(x)$ as $x - a$.

Note that, although (by analogy with the usual translation groups of Euclidean spaces) we use additive notation for the group operation, the translation group need not be Abelian.

Example 3.2.3. ***The calculus of complete finite Boolean digraphs***

Let $\boldsymbol{B}$ be a set consisting of two members, say F and T, which we will identify with 0 *and* 1*, respectively.*

The objects of the calculus of complete finite Boolean digraphs are the complete finite digraphs $\boldsymbol{B}^n$ *whose vertices are all bit strings of length* n*; i.e., all the bit vectors of* $\boldsymbol{B}^n$*.*

The group generated by the flips $h_1, h_2, \ldots, h_n$ *is taken as the translation group of* $\boldsymbol{B}^n$*, where (as one would expect)* $h_k(\vec{b})$ *is obtained from bit vector* $\vec{b}$ *by changing the* k^{th} *bit of* $\vec{b}$ *(and leaving every other bit unchanged).*

For each m *and* n*, define the differentials from* $\boldsymbol{B}^m$ *to* $\boldsymbol{B}^n$ *to be all origin-preserving functions from* $\boldsymbol{B}^m$ *to* $\boldsymbol{B}^n$*.* □

We will return to this example in the sequel.

3.2.1 Graph differentials

Notation:

If X and Y are digraphs, let $\mathrm{Hom}(X, Y)$ be the *exponential digraph* [Hel79, Rib83] whose vertices are the digraph homomorphisms from X to Y. By definition, the edge (L, M) is present in $\mathrm{Hom}(X, Y)$ iff, for each edge (u, v) of X, the edge $(L(u), M(v))$ is present in Y.

Note that if either X or Y is reflexive, then so is $\mathrm{Hom}(X, Y)$.

Notation: If $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus $\mathcal{D}$ on reflexive digraphs, let $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ be the full subdigraph of $\mathrm{Hom}(X, Y)$ whose vertices are the $\mathcal{D}$-differentials from $\mathcal{X}$ to $\mathcal{Y}$.

Proposition 3.2.4. *Let* (K_1, K_2) *be an edge of* $\mathrm{Hom}(X, Y)$ *and let* (L_1, L_2) *be an edge of* $\mathrm{Hom}(Y, Z)$*, where* X*,* Y*, and* Z *are digraphs.*

Then $(L_1 \circ K_1,\ L_2 \circ K_2)$ *is an edge of* $\mathrm{Hom}(X, Z)$*.*

Proof: It is not hard to see that the category of digraphs and digraph homomor-

phisms is Cartesian closed, and that exponential digraphs are exponential objects in this category [Hel79, Shr88]. The proposition to be proved is an immediate corollary [Mac71].

For a direct proof, let (u, v) be an arbitrary edge of X. Since (K_1, K_2) is an edge of $\mathrm{Hom}(X, Y)$, $(K_1(u), K_2(v))$ is an edge of Y. But (L_1, L_2) is an edge of $\mathrm{Hom}(Y, Z)$, so $(L_1(K_1(u)),\ L_2(K_2(v)))$ is an edge of Z.

But the edge (u, v) was arbitrary, so $(L_1 \circ K_1, L_2 \circ K_2)$ is an edge of $\mathrm{Hom}(X, Z)$ □

Corollary 3.2.5. *Let (K_1, K_2) and (L_1, L_2) be edges of $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ and $\mathcal{D}(\mathcal{Y}, \mathcal{Z})$, respectively, where $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ are objects of a differential calculus $\mathcal{D}$ on reflexive digraphs.*

Then $(L_1 \circ K_1, L_2 \circ K_2)$ is an edge of $\mathcal{D}(\mathcal{X}, \mathcal{Z})$ □

In the following, let A and B be full subdigraphs of X and Y, respectively where $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus $\mathcal{D}$ on digraphs. Let a be a vertex of A, and f be an arbitrary function from the vertex set of A to the vertex set of B.

Let $L : \mathcal{X} \longrightarrow \mathcal{Y}$ be a differential.

Definition 3.2.6. *We say that L is a **graph differential** of f at a iff for every vertex x of A such that there is an edge from a to x, there is a differential $M : \mathcal{X} \longrightarrow \mathcal{Y}$ such that*

1. *the edge (L, M) is present in $\mathcal{D}(X, Y)$, and*
2. *$M(x - a) = f(x) - f(a)$*

Explanation: Condition (2) in Definition 3.2.6 can be rewritten as

$$M({\tau_a}^{-1}(x)) = {\tau_{f(a)}}^{-1}(f(x))$$

or, equivalently, as

$$\tau_{f(a)}(M({\tau_a}^{-1}(x))) = f(x)$$

Furthermore, since differentials preserve origins,

$$\tau_{f(a)}(M({\tau_a}^{-1}(a))) = \tau_{f(a)}(M(0_X)) = \tau_{f(a)}(0_Y) = f(a)$$

In other words, $\tau_{f(a)} \circ M \circ {\tau_a}^{-1}$ is a translated differential extrapolant of $f|_{\{a,x\}}$.

Definition 3.2.7. *A function from A to B is* ***graph-differentiable*** *(respectively,* ***uniquely graph-differentiable****) at a vertex a iff it has* at least *one (respectively,* precisely *one) graph differential at a.*

Lemma 3.2.8. *If $f : A \longrightarrow B$ is graph-differentiable at a vertex a, then f takes all out-edges of a in A to out-edges of $f(a)$ in B.*

Proof: Let $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ be objects of a differential calculus on reflexive digraphs. Suppose that $L : \mathcal{X} \longrightarrow \mathcal{Y}$ is a graph differential of $f : A \longrightarrow B$ at a.

Let x be a vertex such that the edge (a, x) is present in A. Then

1. there exists a differential M as in Definition 3.2.6, and
2. since $\tau_a : X \longrightarrow X$ is a digraph isomorphism, the edge

$$(0_X, x - a) = ({\tau_a}^{-1}(a), {\tau_a}^{-1}(x))$$

   is present in X.

Since $M : \mathcal{X} \longrightarrow \mathcal{Y}$ is a graph differential of f at a, and therefore is an origin-preserving digraph homomorphism, the edge

$$(\tau_{f(a)}{}^{-1}(f(a)), \tau_{f(a)}{}^{-1}(f(x))) = (0_Y, f(x) - f(a)) = (M(0_X), M(x - a))$$

is present in Y. Since $\tau_{f(a)} : Y \longrightarrow Y$ is a digraph isomorphism, the edge

$$(f(a), f(x))$$

is present in Y, and therefore in the full subdigraph B. □

Definition 3.2.9. *A function from A to B is **graph-differentiable** (respectively, **uniquely graph-differentiable**) iff it is graph-differentiable (respectively, uniquely graph-differentiable) at each vertex of A.*

In the following, let $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ be objects of a differential calculus $\mathcal{D}$ on reflexive digraphs, and let A, B, and C be full subdigraphs of X, Y, and Z, respectively.

Theorem 3.2.10. *(**Chain Rule for Graph Differentials**)*

Let $K : X \longrightarrow Y$ be a graph-differential of $f : A \longrightarrow B$ at a, and let $L : Y \longrightarrow Z$ be a graph-differential of $g : B \longrightarrow C$ at $f(a)$.

Then $L \circ K$ is a graph-differential of $g \circ f$ at a.

Proof: Let x be a vertex such that the edge (a, x) is present in A. Then there is a differential $M_1 : X \longrightarrow Y$ such that

1. the edge (K, M_1) is present in $\mathcal{D}(X, Y)$, and
2. $M_1(x - a) = f(x) - f(a)$

By Lemma 3.2.8, the edge $(f(a), f(x))$ is present in B, and, therefore, there is a differential $M_2 : Y \longrightarrow Z$ such that

1. the edge (L, M_2) is present in $\mathcal{D}(Y, Z)$, and

2. $M_2(f(x) - f(a)) = f(f(x)) - g(f(a))$

By Corollary 3.2.5, the edge

$$(L \circ K, M_2 \circ M_1)$$

is present in $\mathcal{D}(X, Z)$. Furthermore,

$$M_2(M_1(x - a))) = M_2(f(x) - f(a)) = g(f(x)) - g(f(a))$$

□

Of course, we need not provide a chain rule specifically for graph differentials, nor for the other types of differentials on digraphs to be specified later in this section. The general chain rule (cf. Theorem 2.9.1) covers all these cases.

Example 3.2.11. *Let $\mathcal{B}$ be the calculus of complete finite Boolean digraphs (Example 3.2.3). It is easily verified that, for each m and n,* $\mathrm{Hom}(\boldsymbol{B}^m, \boldsymbol{B}^n)$ *is a complete digraph. Since $\mathcal{B}(\boldsymbol{B}^m, \boldsymbol{B}^n)$ is a full subdigraph of* $\mathrm{Hom}(\boldsymbol{B}^m, \boldsymbol{B}^n)$*, $\mathcal{B}(\boldsymbol{B}^m, \boldsymbol{B}^n)$ is also a complete digraph.*

But $\boldsymbol{B}^n$ is also a complete digraph. Therefore, in the category $\mathcal{B}$, every *differential from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ is a graph differential of* every *function from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ at* every *point of $\boldsymbol{B}^m$.*

3.2.2 Discrete differentials

Again, let A and B be full subdigraphs of X and Y, respectively where $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus on digraphs. Let a be a vertex of A, and f be an arbitrary function from the vertex set of A to the vertex set of B.

Let $L : \mathcal{X} \longrightarrow \mathcal{Y}$ be a differential.

Definition 3.2.12. *We say that L is a* ***discrete differential*** *of f at a iff for every vertex x of A such that there is an edge from a to x,*

$$L(x - a) = f(x) - f(a) \tag{3.1}$$

Observation 3.2.13. *If L is a discrete differential of f at a, then L is a graph-differential of f at a.*

Proof: If X and Y are reflexive digraphs, then so is the exponential digraph $\mathrm{Hom}(X, Y)$. □

Definition 3.2.14. *A function from A to B is* ***discretely differentiable*** *(respectively,* ***uniquely discretely differentiable****) at a vertex a iff it has* at least *one (respectively,* precisely *one) discrete differential at a.*

Observation 3.2.15. *If $f : A \longrightarrow B$ is discretely differentiable at a vertex a, then f takes all out-edges of a in A to out-edges of $f(a)$ in B.*

Proof: Immediate from Observation 3.2.13 and Lemma 3.2.8. □

Definition 3.2.16. *A function from A to B is* ***discretely differentiable*** *(respectively,* ***uniquely discretely differentiable****) iff it is discretely differentiable (respectively, uniquely discretely differentiable) at each vertex of A.*

In the following, let $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ be objects of a differential calculus on reflexive digraphs, and let A, B, and C be full subdigraphs of X, Y, and Z, respectively.

Theorem 3.2.17. *(**Chain Rule for Discrete Differentials**)*

Let $K : X \longrightarrow Y$ be a discrete differential of $f : A \longrightarrow B$ at a, and let $L : Y \longrightarrow Z$ be a discrete differential of $g : B \longrightarrow C$ at $f(a)$.

Then $L \circ K$ is a discrete differential of $g \circ f$ at a.

proof: Let x be a vertex of A such that there is an edge from a to x. Then

$$L(K(x-a))) = L(f(x)-f(a)) = g(f(x))-g(f(a))$$

□

Example 3.2.18. *In the calculus of complete finite Boolean digraphs (Example 3.2.3), every differential from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ is a* graph *differential of every function from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ at every vertex of $\boldsymbol{B}^m$ (Example 3.2.11).*

On the other hand, it is not hard to verify that, in this calculus, every function from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ has a unique discrete *differential at every vertex.*

To illustrate this, we exhibit the discrete differentials of the unary Boolean operators, and the discrete differentials of the binary Boolean connectives, in Tables 3.1 and 3.2, respectively.

Function	**Differential at** 0	**Differential at** 1
constant with value 0	constant with value 0	constant with value 0
constant with value 1	constant with value 0	constant with value 0
identity	identity	identity
¬	identity	identity

Table 3.1: Unary Boolean operators and their discrete differentials in the calculus of complete finite Boolean digraphs

3.2.3 Convergence of filters of differentials

Definition 3.2.19. *Let $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ be objects of a differential calculus $\mathcal{D}$ on reflexive digraphs, let L be a member of $\mathcal{D}(\mathcal{X}, \mathcal{Y})$, and let $\mathcal{F}$ be a filter on $\mathcal{D}(\mathcal{X}, \mathcal{Y})$.*

Function	**Differential at**			
	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
const_0	const_0	const_0	const_0	const_0
const_1	const_0	const_0	const_0	const_0
proj_1	proj_1	proj_1	proj_1	proj_1
proj_2	proj_2	proj_2	proj_2	proj_2
$\neg \circ \text{proj}_1$	proj_1	proj_1	proj_1	proj_1
$\neg \circ \text{proj}_2$	proj_2	proj_2	proj_2	proj_2
$\wedge$	$\wedge$	$\not\longrightarrow$	$\not\longleftarrow$	$\vee$
$\vee$	$\vee$	$\not\longleftarrow$	$\not\longrightarrow$	$\wedge$
NAND	$\wedge$	$\not\longrightarrow$	$\not\longleftarrow$	$\vee$
NOR	$\vee$	$\not\longleftarrow$	$\not\longrightarrow$	$\wedge$
$\longleftarrow$	$\not\longleftarrow$	$\vee$	$\wedge$	$\not\longrightarrow$
$\longrightarrow$	$\not\longrightarrow$	$\wedge$	$\vee$	$\not\longleftarrow$
$\not\longleftarrow$	$\not\longleftarrow$	$\vee$	$\wedge$	$\not\longrightarrow$
$\not\longrightarrow$	$\not\longrightarrow$	$\wedge$	$\vee$	$\not\longleftarrow$
$\longleftrightarrow$	XOR	XOR	XOR	XOR
XOR	XOR	XOR	XOR	XOR

Table 3.2: Boolean connectives and their discrete differentials in the calculus of complete finite Boolean digraphs

1. *We will say that $\mathcal{F}$ **strongly converges** to L iff for each edge (u,v) of X, there exist $F_{u,v} \in \mathcal{F}$ and a vertex $w_{u,v}$ of Y such that*

   (a) the edge $(L(u), w_{u,v})$ is present in Y, and

   (b) every member of $F_{u,v}$ maps v to $w_{u,v}$.

2. *We will say that $\mathcal{F}$ **weakly converges** to L iff for each vertex u of X, there is some $F_u \in \mathcal{F}$ such that for each M in F_u and each vertex v such that the edge (u,v) is present in X, the edge*

$$(L(u), M(v))$$

   is present in Y.

Observation 3.2.20. *Let $\mathcal{X}$ and $\mathcal{Y}$ be as above, let L be a member of $\mathcal{D}(\mathcal{X},\mathcal{Y})$, and let $\mathcal{F}$ be a filter on $\mathcal{D}(\mathcal{X},\mathcal{Y})$.*

If $\mathcal{F}$ converges strongly to L and every vertex of X has finite outdegree, then $\mathcal{F}$ converges weakly to L

Proof:

Suppose that $\mathcal{F}$ converges strongly to L, and every vertex of X has finite outdegree.

Let u be a vertex of X. Then for each v such that the edge (u, v) is present in X, there exist $F_{u,v}$ and $w_{u,v}$ as in the definition of strong convergence.

Let

$$F_u = \bigcap \{ F_{u,v} \mid \text{the edge } (u, v) \text{ is present in } X \}$$

Since $\mathcal{F}$ is a filter, and u has finite outdegree,

F_u is a member of $\mathcal{F}$

Let M be a member of F_u, and let V be a vertex such that the edge (u, v) is present in X. Since M belongs to $F_{u,v}$,

1. the edge $(L(u), w_{u,v})$ is present in Y, and
2. $M(v) = w_{u,v}$

In short, the edge

$$(L(u), M(v))$$

is present in Y. □

Observation 3.2.21. *Let $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ be objects of a differential calculus. Let L and M be vertices of $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that the edge (L, M) is present. Let $[M]$ be the point filter at M, i.e.*

$$[M] = \{ F \subseteq \mathcal{D}(\mathcal{X}, \mathcal{Y}) \mid M \in F \}$$

Then $[M]$ *strongly converges to* L.

Proof:

Let (u, v) be an arbitrary edge of X. Let $F_{u,v}$ be the singleton $\{ M \}$, and let $w_{u,v} = M(v)$. □

The preceding observation has a partial converse.

Observation 3.2.22. *Let* $\mathcal{X} = (X, 0_X, T_X)$ *and* $\mathcal{Y} = (Y, 0_Y, T_Y)$ *be as above. Let* L *be a member of* $\mathcal{D}(\mathcal{X}, \mathcal{Y})$. *Let* $g : X \longrightarrow Y$ *be a graph homomorphism. Let* $\mathcal{F}$ *be a filter on* $D(\mathcal{X}, \mathcal{Y})$.

If $\mathcal{F}$ *strongly converges to* L, *and* X *is finite, then*

$$\mathcal{F} = [M]$$

for some M *such that* (L, M) *is an edge of* $\mathcal{D}(X, Y)$.

Proof: Suppose that X is finite, and let $\mathcal{F}$ be a filter which strongly converges to L.

For each edge (u, v) of X, there exist $F_{u,v}$ in $\mathcal{F}$ and a vertex $w_{u,v}$ of Y as in the definition of strong convergence.

Let

$$F = \bigcap \{ F_{u,v} \mid (u, v) \text{ is an edge of } X \}$$

Since each $F_{u,v}$ belongs to the filter $\mathcal{F}$, and the edge set of X is finite, F also belongs to $\mathcal{F}$.

Let u be an arbitrary vertex of X. Since X is reflexive, it has a loop at u, and therefore

1. the edge $(L(u), w_{u,u})$ is present in Y, and

2. every member of F maps u to $w_{u,u}$.

All members of F agree at u. But u is arbitrary, and F is independent of u. Therefore, all members of F agree everywhere, i.e., F has at most one member. But F is nonempty, and therefore must be a singleton $\{ M \}$. Furthermore, for every vertex u of X,

1. the edge $(L(u), w_{u,u})$ is present in Y, and

2. M maps u to $w_{u,u}$.

That is to say, the edge (L, M) is present in $\mathcal{D}(X, Y)$. □

Observation 3.2.23. *Let $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ be objects of a differential calculus. Let L be a vertex of $\mathcal{D}(\mathcal{X}, \mathcal{Y})$.*

Let $N(L)$ be the outward graph neighborhood of L in $\mathcal{D}(X, Y)$, i.e.

$$N(L) = \{ M \mid \text{the edge } (L, M) \text{ is present in } \mathcal{D}(X, Y) \}$$

Let F_0 be an arbitrary nonempty subset of $N(L)$, and let $[F_0]$ be the principal filter at F_0, i.e.

$$[F_0] = \{ F \subseteq D(\mathcal{X}, \mathcal{Y}) \mid F_0 \subseteq F \}$$

Then $[F_0]$ weakly converges to L.

Proof: Let u be an arbitrary vertex of X.

Let M be an arbitrary member of F_0, and let v be an arbitrary vertex such that the edge

$$(u, v)$$

is present in X. Since $F_0 \subseteq N(L)$, the edge

$$(L, M)$$

is present in $|mathcalD(X, Y)$, and therefore the edge

$$(L(u), M(v))$$

is present in Y □

This too has a partial converse:

Observation 3.2.24. *Let $N(L)$ be the outward graph neighborhood of L in $\mathcal{D}(X, Y)$. If $\mathcal{F}$ weakly converges to L, and X is finite, then*

$$N(L) \in \mathcal{F}$$

Proof: Suppose that X is finite.

For each vertex u of X, there is some $F_u \in \mathcal{F}$ as in the definition of weak convergence.

Let

$$F = \bigcap \{ F_u \mid u \text{ is a vertex of } X \}$$

Since each F_u belongs to the filter $\mathcal{F}$, and the vertex set of X is finite, F also belongs to $\mathcal{F}$.

Let $K \in F$. Then for each edge (u, v) of X, K belongs to F_u, and therefore the edge

$$(L(u), K(v))$$

is present in Y

In short, the edge

$$(L, K)$$

is present in $\mathcal{D}(X, Y)$. But K was an arbitrary member of F. In other words,

$$F \subseteq N(L)$$

Since F is a member of $\mathcal{F}$, and $\mathcal{F}$ is a filter, $N(L)$ also belongs to $\mathcal{F}$ □

Notation It is readily verified that if $\mathcal{F}$ and $\mathcal{G}$ are filters on $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ and $\mathcal{D}(\mathcal{Y}, \mathcal{Z})$ respectively, where $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ are objects of a differential calculus $\mathcal{D}$ on reflexive digraphs, then

$$\{\, \{\, g \circ f \,|\, g \in G,\ f \in F \,\} \,|\, G \in \mathcal{G},\ F \in \mathcal{F} \,\}$$

is a base for a filter on $\mathcal{D}(\mathcal{X}, \mathcal{Z})$. Denote this filter by $\mathcal{G} \cdot \mathcal{F}$

Lemma 3.2.25. *Let K and L be members of $D(\mathcal{X}, \mathcal{Y})$ and $D(\mathcal{Y}, \mathcal{Z})$ respectively, where $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ are objects of a differential calculus.*

1. *If $\mathcal{F}$ strongly converges to K, and $\mathcal{G}$ strongly converges to L, then $\mathcal{G} \cdot \mathcal{F}$ strongly converges to $L \circ K$.*

2. *If $\mathcal{F}$ weakly converges to K, and $\mathcal{G}$ weakly converges to L, then $\mathcal{G} \cdot \mathcal{F}$ weakly converges to $L \circ K$.*

Proof:

1. Suppose that $\mathcal{F}$ strongly converges to L and $\mathcal{G}$ strongly converges to M.

   Let (u, v) be an edge of X. Then there exist F in $\mathcal{F}$ and a vertex w of Y such that

   (a) the edge $(K(u), w)$ is present in Y, and

(b) every member of F maps v to w.

In turn, since $(K(u), w)$ is an edge of Y, there exist G in $\mathcal{G}$ and a vertex z of Z such that

(a) the edge $(L(K(u)), z)$ is present in Z, and

(b) every member of G maps w to z.

Let $GF = \{\, g \circ f \,|\, g \in G, f \in F \,\}$. Then GF belongs to $\mathcal{G} \cdot \mathcal{F}$, and every member of GF maps v to z.

2. Suppose that $\mathcal{F}$ weakly converges to K and $\mathcal{G}$ weakly converges to L.

Let u be a vertex of X. Then there is some F in $\mathcal{F}$ such that for each M in F and each vertex v such that the edge (u, v) is present in X,

the edge $(K(u), M(v))$ is present in Y.

In turn, there is some G in $\mathcal{G}$ such that for each N in G and each vertex w such that the edge $(K(u), w)$ is present in Y,

the edge $(L(K(u)), N(w))$ is present in Z.

Let $GF = \{\, g \circ f \,|\, g \in G, f \in F \,\}$. Then GF is a member of $\mathcal{G} \cdot \mathcal{F}$. Furthermore, for each $N \circ M$ in GF ($N \in G$, $M \in F$), and each vertex v such that the edge (u, v) is present in X, the edge $(K(u), M(v))$ is present in Y, and therefore the edge $(L(K(u)), N(M(v))$ is present in Z.

□

3.2.4 Postdiscrete differentials

Let A and B be full subdigraphs of X and Y, respectively where $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus $\mathcal{D}$ on digraphs. Let a be a

vertex of A, and f be an arbitrary function from the vertex set of A to the vertex set of B.

Let L be a member of $\mathcal{D}(\mathcal{X}, \mathcal{Y})$.

Definition 3.2.26. *We say that L is a* postdiscrete differential *of f at a iff for every vertex x of A such that there is an edge from a to x, there is a filter $\mathcal{H}$ strongly converging to L in $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ such that*

1. *L belongs to every member of $\mathcal{H}$, and*
2. *for every $H \in \mathcal{H}$, there is at least one M in H such that*

$$M(x - a) = f(x) - f(a)$$

Observation 3.2.27.

1. *If L is a discrete differential of f at a, then L is a postdiscrete differential of f at a.*
2. *If L is a postdiscrete differential of f at a, and X is finite, then L is a discrete differential of f at a.*

Proof:

1. Suppose that L is a discrete differential of f at a. Let x be a vertex of A such that the edge (a, x) is present. Then

$$L(x - a) = f(x) - f(a)$$

Since both X and Y are reflexive, so is $\mathrm{Hom}(X, Y)$. In particular, $\mathrm{Hom}(X, Y)$ has a loop at L, and therefore $[L]$ strongly converges to L. Furthermore, L belongs to every member of $[L]$.

2. Suppose that X is finite, and L is a postdiscrete differential of f at a.

   Let x be a vertex of A such that there is an edge from a to x. Then there is some filter $\mathcal{H}$ as in Definition 3.2.26.

   By the partial converse to Observation 3.2.21, $\mathcal{H}$ must be the principal filter $[M]$ at some M such that there is an edge from L to M in $\mathrm{Hom}(X, Y)$.

   But, since $\mathcal{H}$ was chosen as in the definition of postdiscrete differentials, every member of H must contain L. As $[L]$ is the only such principal filter, we have

   $$\mathcal{H} = [L]$$

   That is, $\{ L \}$ is a member of $\mathcal{H}$. But $\mathcal{H}$ was chosen as in the definition of postdiscrete differentials, and thus

   $$L(x - a) = f(x) - f(a)$$

   so L is a discrete differential of f at a.

□

Example 3.2.28. *Returning to Example 3.2.3, the preceding observation tells us that, in the calculus of complete finite Boolean digraphs, every function from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ has a unique postdiscrete differential at each vertex, namely, its discrete differential at that vertex.*

Definition 3.2.29. *A function from A to B is **postdiscretely differentiable** (respectively, **uniquely postdiscretely differentiable**) at a vertex a iff it has **at least** one (respectively,* precisely *one) postdiscrete differential at a.*

Lemma 3.2.30. *If $f : A \longrightarrow B$ is postdiscretely differentiable at a vertex a, then f takes all out-edges of a in A to out-edges of $f(a)$ in B.*

Proof:

We prove this lemma for the special case in which a and $f(a)$ are the origins of $\mathcal{X}$ and $\mathcal{Y}$, respectively. A general proof can be obtained from the proof of the special case by applying ${\tau_a}^{-1}$ and $\tau_{f(a)}$ to the vertices of X and the vertices of Y, respectively, exactly as in the proof of Lemma 3.2.8.

Suppose that $L : \mathcal{X} \longrightarrow \mathcal{Y}$ is a postdiscrete differential of $f : A \longrightarrow B$ at 0_X, and suppose that f maps 0_X to 0_Y.

Let x be a vertex of A such that the edge $(0_X, x)$ is present. Then there is some filter $\mathcal{H}$ as in Definition 3.2.26.

Each H in $\mathcal{H}$ contains some M_H such that

$$M_H(x) = f(x)$$

Furthermore, $\mathcal{H}$ strongly converges to L, and the edge $(0_X, x)$ is present in A (and therefore in X). Therefore, there exist H in $\mathcal{H}$ and a vertex w of Y such that

1. the edge $(f(0_X), w) = (0_Y, w) = (L(0_X), w)$ is present in Y, and
2. every member of H maps x to w.

and therefore

$$f(x) = M_H(x) = w$$

□

Definition 3.2.31. *A function from A to B is **postdiscretely differentiable** (respectively, **uniquely postdiscretely differentiable**) iff it is postdiscretely differentiable (respectively, uniquely postdiscretely differentiable) at each vertex of A.*

As before, let $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ be objects of a differential calculus on reflexive digraphs, and let A, B, and C be full subdigraphs of X, Y, and Z, respectively.

Theorem 3.2.32. *(**Chain Rule for Postdiscrete Differentials**)*

Let $K : \mathcal{X} \longrightarrow \mathcal{Y}$ be a postdiscrete differential of $f : A \longrightarrow B$ at a, and let $L : \mathcal{Y} \longrightarrow \mathcal{Z}$ be a postdiscrete differential of $g : B \longrightarrow C$ at $f(a)$.

Then $L \circ K$ is a postdiscrete differential of $g \circ f$ at a.

Proof:

Let x be a vertex of A such that the edge (a, x) is present. Then there is some filter $\mathcal{F}$ strongly converging to K in $D(\mathcal{X}, \mathcal{Y})$ such that

1. K belongs to every member of $\mathcal{F}$, and
2. for every $F \in \mathcal{F}$, there is at least one M in F such that

$$M(x - a) = f(x) - f(a)$$

By lemma 3.2.30, the edge $(f(a), f(x))$ is present in B, and therefore there is some $\mathcal{G}$ strongly converging to L in $D(\mathcal{Y}, \mathcal{Z})$ such that

1. L belongs to every member of $\mathcal{G}$, and
2. for every $G \in \mathcal{G}$, there is at least one N in G such that

$$N(f(x) - f(a)) = g(f(x)) - g(f(a))$$

By Lemma 3.2.25, $\mathcal{G} \cdot \mathcal{F}$ strongly converges to $L \circ K$. Furthermore,

1. $L \circ K$ belongs to every member of $\mathcal{G} \cdot \mathcal{F}$, and

2. for every $G \circ F$ in $\mathcal{G} \cdot \mathcal{F}$ ($G \in \mathcal{G}$, $F \in \mathcal{F}$), there exist at least one M in F and at least one N in G such that

   (a) $M(x - a) = f(x) - f(a)$, and

   (b) $N(M(x - a)) = N(f(x) - f(a)) = g(f(x)) - g(f(a))$

In short, $N \circ M$ is a postdiscrete differential of $g \circ f$ at a. □

3.2.5 Pretopological differentials

Let A and B be full subdigraphs of X and Y, respectively, where $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y} = (Y, 0_Y, T_Y)$ are objects of a differential calculus on digraphs. Let a be a vertex of A, and f be an arbitrary function from the vertex set of A to the vertex set of B.

Let L be a member of $D(\mathcal{X}, \mathcal{Y})$.

Definition 3.2.33. *We say that L is a **pretopological differential** of f at a iff for every vertex x of A such that there is an edge from a to x, there is some $\mathcal{H}$ weakly converging to L in $D(\mathcal{X}, \mathcal{Y})$ such that*

i. L belongs to every member of $\mathcal{H}$, and

ii. for every $H \in \mathcal{H}$, there is at least one M in H such that

$$M(x - a) = f(x) - f(a)$$

Observation 3.2.34.

1. *If L is a graph differential of f at a, then L is a pretopological differential of f at a.*

2. *If L is a pretopological differential of f at a, and X is finite, then L is a graph differential of f at a.*

Proof:

1. Suppose that L is a graph differential of f at a.

   Let x be a vertex of A such that there is an edge from a to x. Then there is some M as in Definition 3.2.6.

   Let $\mathcal{H} = N(M)$, i.e. let $\mathcal{H}$ be the outward graph neighborhood of L in $\mathcal{D}(X, Y)$. By the partial converse to Observation 3.2.23, the point filter $[M]$ weakly converges to L in $\mathcal{D}(X, Y)$, and thus L is a pretopological differential of f at a.

2. Conversely, suppose that L is a pretopological differential of f at a, and X is finite.

   Let x be a vertex of A such that there is an edge from a to x. Then there is some $\mathcal{H}$ as in Definition 3.2.33.

   Again, let $N(L)$ be the outward neightborhood of L in $\mathcal{D}(X, Y)$. By the partial converse to Observation obs:NbhdWeak, $N(L)$ is a member of $\mathcal{H}$

   Therefore, there is at least one M in $N(L)$ such that

   $$M(x - a) = f(x) - f(a)$$

   i.e. L is a graph differential of f at a. □

Example 3.2.35. *Returning again to Example 3.2.3, the preceding observation, together with Example 3.2.11, tells us that, in the calculus of complete finite Boolean digraphs, every differential from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ is a pretopological differential of every function from $\boldsymbol{B}^m$ to $\boldsymbol{B}^n$ at each vertex of $\boldsymbol{B}^m$.*

Definition 3.2.36. *A function from A to B is* ***pretopologically differentiable*** *(respectively,* ***uniquely pretopologically differentiable****) at a vertex a iff it has* ***at least*** *one (respectively,* precisely *one) pretopological differential at a.*

Lemma 3.2.37. *If $f : A \longrightarrow B$ is pretopologically differentiable at a vertex a, then f takes all out-edges of a in A to out-edges of $f(a)$ in B.*

Proof: As in the proof of Lemma 3.2.30, we prove this lemma for the special case in which a and $f(a)$ are the origins of $\mathcal{X}$ and $\mathcal{Y}$, respectively. Again, a general proof can be obtained by applying ${\tau_a}^{-1}$ and $\tau_{f(a)}$ to the vertices of X and the vertices of Y, respectively.

Suppose that $L : \mathcal{X} \longrightarrow \mathcal{Y}$ is a pretopological differential of $f : A \longrightarrow B$ at the origin of X, and suppose that f maps 0_X to 0_Y.

Let x be a vertex of A such that the edge $(0, x)$ is present. Then there is some filter $\mathcal{H}$ as in Definition 3.2.33.

There is some M in H such that

$$M(x) = f(x)$$

Since $\mathcal{H}$ weakly converges to L, there exists H in $\mathcal{H}$ such that for each M in H and each vertex v such that the edge $(0, v)$ is present in X, the edge

$$(f(0_X), M(v)) = (0_Y, M(v)) = (L(0_X), M(v))$$

is present in Y.

In particular, the edge $(0_X, x)$ is present in A, and therefore in X, so the edge

$$f(0_X, f(x)) = (0_Y, f(x)) = (0_Y, M(x))$$

is present in Y, and therefore in the full subdigraph B. □

Definition 3.2.38. *A function from A to B is **pretopologically differentiable** (respectively, **pretopologically differentiable**) iff it is pretopologically differentiable (respectively, uniquely pretopologically differentiable) at each vertex of A.*

Again, let $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y} = (Y, 0_Y, T_Y)$, and $\mathcal{Z} = (Z, 0_Z, T_Z)$ be objects of a differential calculus on reflexive digraphs, and let A, B, and C be full subdigraphs of X, Y, and Z, respectively.

Theorem 3.2.39. *(**Chain Rule for Pretopological Differentials**)*

Let $K : \mathcal{X} \longrightarrow \mathcal{Y}$ be a pretopological differential of $f : A \longrightarrow B$ at a, and let $L : \mathcal{Y} \longrightarrow \mathcal{Z}$ be a pretopological differential of $g : B \longrightarrow C$ at $f(a)$.

Then $L \circ K$ is a pretopological differential of $g \circ f$ at a.

Proof: The proof is exactly the same as the proof of Theorem 3.2.32, except that weak convergence is used instead of strong convergence, and Lemma 3.2.37 is used instead of Lemma 3.2.30. □

3.2.6 Differential calculi With Kronecker products

Definition 3.2.40. *A differential calculus $\mathcal{D}$ on reflexive digraphs (cf. Definition 3.2.1) will be said to have **Kronecker products** iff the following three additional axioms hold:*

vi. for each natural number n, if $\mathcal{X}_1 = (X_1, 0_1, T_1)$, $\mathcal{X}_2 = (X_2, 0_2, T_2)$, ..., $\mathcal{X}_n = (X_n, 0_n, T_n)$ are objects of $\mathcal{D}$, then so is

$$\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n = (X_1 \times X_2 \times \ldots \times X_n,\ (0_1, 0_2, \ldots, 0_n),\ T_1 \oplus T_2 \oplus \ldots \oplus T_n)$$

where $X_1 \times X_2 \times \ldots \times X_n$ is the categorical product (i.e., cross product) of the digraphs X_1, X_2, ..., X_n [Wei62, HW67, IK00], and each member of $T_1 \oplus T_2 \oplus \ldots \oplus T_n$ is an automorphism on $X_1 \times X_2 \times \ldots \times X_n$ obtained by componentwise application:

$$(h_1 \times h_2 \times \ldots \times h_n)(x_1, x_2, \ldots x_n) = (h_1(x_1), h_2(x_2), \ldots, h_n(x_n))$$

(Side remark: Since the digraphs in question are reflexive, their cross product coincides with their weak product.)

vii. for each natural number n, if $\mathcal{X}_1 = (X_1, 0_1, T_1)$, $\mathcal{X}_2 = (X_2, 0_2, T_2)$, ..., $\mathcal{X}_n = (X_n, 0_n, T_n)$ are objects of $\mathcal{D}$, then the projection function $\pi_j : X_1 \times X_2 \times \ldots \times X_n \longrightarrow X_j$ mapping each tuple to its j^{th} component is a $\mathcal{D}$-differential from $\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_n$ to $\mathcal{X}_j$ $(j = 1, 2, \ldots, n)$.

viii. for each natural number n, if $\mathcal{X} = (X, 0, T)$, $\mathcal{Y}_1 = (Y_1, 0_1, T_1)$, $\mathcal{Y}_2 = (Y_2, 0_2, T_2)$, ..., $\mathcal{Y}_n = (Y_n, 0_n, T_n)$ are objects of $\mathcal{D}$, and $f_i : \mathcal{X} \longrightarrow \mathcal{Y}_i$ are $\mathcal{D}$-differentials $(i = 1, 2, \ldots, n)$, then the parametric function

$$(f_1, f_2, \ldots, f_n) : \mathcal{X} \longrightarrow \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_n$$

given by

$$(f_1, f_2, \ldots, f_n)(x) = (f_1(x), f_2(x), \ldots, f_n(x))$$

is also a $\mathcal{D}$-differential.

Observation 3.2.41. *Let $\mathcal{D}$ be a differential calculus on reflexive digraphs which has Kronecker products. Then*

1. *the category $\mathcal{D}$ has finite products, and*

2. *the forgetful functor from $\mathcal{D}$ to the category of reflexive digraphs and digraph homomorphisms preserves finite products.*

□

Example 3.2.42. *The calculus of complete finite Boolean digraphs (Example 3.2.3) has Kronecker products.*

□

In the following, let $\mathcal{X} = (X, 0_X, T_X)$ and $\mathcal{Y}_i = (Y_i, 0_i, T_i)$ $(i = 1, 2, \ldots, n)$ be objects of a differential calculus with Kronecker products, and let A, B_1, B_2, ..., B_n be full subdigraphs of X, Y_1, Y_2, ..., Y_n, respectively.

In the following, assume that $\mathcal{D}$ has Kronecker products, and let $\mathcal{X}$ and $\mathcal{Y}_i$ $(i = 1, 2, \ldots n)$ be as before.

Observation 3.2.43. $D(\mathcal{X}, \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_n) = D(X, \mathcal{Y}_1) \times D(X, \mathcal{Y}_2) \times \ldots \times D(X, \mathcal{Y}_n)$

□

Lemma 3.2.44. *Let $\mathcal{F}_1$, $\mathcal{F}_2$, ..., $\mathcal{F}_n$ be filters on $D(\mathcal{X}, \mathcal{Y}_1)$, $D(\mathcal{X}, \mathcal{Y}_2)$, ..., $D(\mathcal{X}, \mathcal{Y}_n)$, respectively. Let L_i be a member of $D(\mathcal{X}, \mathcal{Y}_i)$ $(i = 1, 2, \ldots, n)$.*

Then $\{ F_1 \times F_2 \times \ldots \times F_n \mid F_i \in \mathcal{F}_i \ (i = 1, 2, \ldots, n) \}$ is a base for a filter $\mathcal{F}$ on $D(\mathcal{X}, \mathcal{Y}_1) \times D(\mathcal{X}, \mathcal{Y}_2) \times \ldots \times D(\mathcal{X}, \mathcal{Y}_n)$. Moreover,

1. *If $\mathcal{F}_i$ strongly converges to L_i $(i = 1, 2, \ldots, n)$, then $\mathcal{F}$ strongly converges to $(L_1, L_2, \ldots, L_n)$ in $D(\mathcal{X}, \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_n)$.*

2. *If $\mathcal{F}_i$ weakly converges to L_i $(i = 1, 2, \ldots, n)$, then $\mathcal{F}$ weakly converges to $(L_1, L_2, \ldots, L_n)$ in $D(\mathcal{X}, \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_n)$.*

Proof: Define $\mathcal{F}$ as in the statement of the theorem.

1. Suppose that $\mathcal{F}_i$ strongly converges to L_i in $\mathcal{D}(\mathcal{X}, \mathcal{Y}_i)$ $(i = 1, 2, \ldots, n)$. Let (u, v) be an edge of X.

Let $L : \mathcal{X} \longrightarrow \mathcal{Y} = (L_1, L(2), \ldots L(n))$. Then L is a member of $\mathcal{D}(X, Y_1 \times Y_2 \times \times Y_n$.

For $i = 1, 2, \ldots, n$, there exist F_i in $\mathcal{F}_i$ and vertices $w_1, w_2 \ldots w_n$ of $Y_1, Y_2, \ldots Y_n$ respectively such that, for $1 = 1, 2, \ldots n$,

(a) the edge $(L_i(u), w_i)$ is present in Y_i, and

(b) every member of F_i maps v to w_i.

Let $F = F_1 \times F_2 \times \ldots F_n$. Then F belongs to $\mathcal{F}$. Furthermore,

(a) the edge $(L_1(u), L_2(u), \ldots L_n(u)), (w_1, w_2, \ldots w_n))$ is present in $Y_1 \times Y_2 \times \ldots \times Y_n$, and

(b) every member of F maps v to $(w_1, w_2, \ldots w_n)$.

In short, $\mathcal{F}$ strongly converges to $(L_1, L_2, \ldots L_n)$.

2. Suppose that $\mathcal{F}_i$ weakly converges to L_i in $\mathcal{D}(\mathcal{X}, \mathcal{Y}_i)$ $(i = 1, 2, \ldots, n)$. Let u be a vertex of X.

   For $i = 1, 2, \ldots n$, there exists $F_i \in \mathcal{F}_i$ such that for each M in F_i and each vertex v such that the edge (u, v) is present in X, the edge

   $$(L_i(u), M(v))$$

   is present in Y_i.

   Let $F = F_1 \times F_2 \times \ldots \times F_n$. Then F belongs to $\mathcal{F}$. Furthermore, for each $(M_1, M_2, \ldots, M_n)$ in F and each vertex v such that the edge (u, v) is present in X, the edge

   $$((L_1(u), L_2(u), \ldots L_n(u)), (M_1(v), M_2(v), \ldots M_n(v)))$$

is present in $Y_1 \times Y_2 \times \ldots \times Y_n$

In short, $\mathcal{F}$ weakly converges to $(L_1, L_2, \ldots L_n)$. □

Theorem 3.2.45. *Let $\mathcal{X} = (X, 0_X, T_X)$, $\mathcal{Y}_1 = (Y_1, 0_1, T_1)$, $\mathcal{Y}_2 = (Y_2, 0_2, T_2)$, ..., $\mathcal{Y}_n = (Y_n, 0_n, T_n)$ be objects of a differential calculus $\mathcal{D}$ which has Kronecker products. Let $L_i : \mathcal{X} \longrightarrow \mathcal{Y}_i$ be $\mathcal{D}$-differentials $(i = 1, 2, \ldots, n)$.*

Let A, B_1, B_2, ..., B_n be full subdigraphs of X, Y_1, Y_2, ..., Y_n, respectively. Let $f_i : A \longrightarrow B_i$ be functions $(i = 1, 2, \ldots, n)$. Let a be a vertex of A.

1. *If L_1, L_2, ..., L_n are discrete differentials of f_1, f_2, ...,, f_n respectively at a, then*

$$(L_1, L_2, \ldots, L_n) : X \longrightarrow Y_1 \times Y_2 \times \ldots \times Y_n$$

is a discrete differential of

$$(f_1, f_2, \ldots, f_n) : A \longrightarrow B_1 \times B_2 \times \ldots \times B_n$$

at a.

2. *If L_1, L_2, ..., L_n are graph differentials of f_1, f_2, ...,, f_n respectively at a, then $(L_1, L_2, \ldots, L_n)$ is a graph differential of $(f_1, f_2, \ldots, f_n)$ at a.*

3. *If L_1, L_2, ..., L_n are postdiscrete differentials of f_1, f_2, ...,, f_n respectively at a, then $(L_1, L_2, \ldots, L_n)$ is a postdiscrete differential of $(f_1, f_2, \ldots, f_n)$ at a.*

4. *If L_1, L_2, ..., L_n are pretopological differentials of f_1, f_2, ...,, f_n respectively at a, then $(L_1, L_2, \ldots, L_n)$ is a pretopological differential of $(f_1, f_2, \ldots, f_n)$ at a.*

Proof: Let $\mathcal{X}$, $\mathcal{Y}_1$, $\mathcal{Y}_2$, ..., $\mathcal{Y}_n$, L_1, L_2, ..., L_n, A, B_1, B_2, ..., B_n, f_1, f_2, ...,, f_n, and a be as in the statement of the theorem.

1. Suppose that L_1, L_2, ..., L_n are discrete differentials of f_1, f_2, ...,, f_n, respectively, at a. Let x be a vertex of A such that there is an edge from a to x.

   Then

   $$L_i({\tau_a}^{-1}(x)) = L_i(x-a) = f_i(x) - f_i(a) = {\tau_{f_i(a)}}^{-1}(f_i(x)) \quad (i = 1, 2, \ldots, n)$$

   and hence

   $$\begin{aligned}
   &(L_1, L_2, \ldots, L_n)(x-a) \\
   &= (L_1(x-a), L_2(x-a), \ldots, L_n(x-a)) \\
   &= (L_1({\tau_a}^{-1}(x)), L_2({\tau_a}^{-1}(x)), \ldots, L_n({\tau_a}^{-1}(x))) \\
   &= (\tau_{f_1}(a)^{-1}(f_1(x)), \tau_{f_2}(a)^{-1}(f_2(x)), \ldots, \tau_{f_n}(a)^{-1}(f_n(x))) \\
   &= (\tau_{f_1}(a)^{-1} \times \tau_{f_2}(a) \times \ldots \times \tau_{f_n}(a)^{-1})(f_1(x), f_2(x), \ldots, f_n(x)) \\
   &= {\tau_{(f_1(a), f_2(a), \ldots, f_n(a))}}^{-1}(f_1(x), f_2(x), \ldots, f_n(x)) \\
   &= (f_1(x), f_2(x), \ldots, f_n(x)) - (f_1(a), f_2(a), \ldots, f_n(a)) \\
   &= (f_1, f_2, \ldots, f_n)(x) - (f_1, f_2, \ldots, f_n)(a)
   \end{aligned}$$

   In short, $(L_1, L_2, \ldots, L_n)$ is a discrete differential of $(f_1, f_2, \ldots, f_n)$ at a.

2. Suppose that L_1, L_2, ...,, L_n are graph differentials of f_1, f_2, ...,, f_n, respectively, at a. Let x be a vertex of A such that there is an edge from a to x.

   Then there exist differentials M_1, M_2, ..., M_n such that, for $(i = 1, 2, \ldots, n)$,

   (a) the edge L_i, M_i is present in $\mathcal{D}(\mathcal{X}, \mathcal{Y}_i)$, and

   (b) $M_i({\tau_a}^{-1}(x)) = M_i(x-a) = f_i(x) - f_i(a) = {\tau_{f_i(a)}}^{-1}(f_i(x))$

   Equivalently,

(a) the edge $((L_1, L_2, \ldots, L_n), M_1, M_2, \ldots, M_n)$ is present in $\mathcal{D}(, \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_n)$, and

(b)

$$\begin{aligned}
&(M_1, M_2, \ldots, M_n)(x - a) \\
&= (M_1(x - a), M_2(x - a), \ldots, M_n(x - a)) \\
&= (M_1({\tau_a}^{-1}(x)), M_2({\tau_a}^{-1}(x)), \ldots, M_n({\tau_a}^{-1}(x))) \\
&= ({\tau_{f_1(a)}}^{-1}(f_1(x)), {\tau_{f_2(a)}}^{-1}(f_2(x)), \ldots, {\tau_{f_n(a)}}^{-1}(f_n(x))) \\
&= ({\tau_{f_1(a)}}^{-1} \times {\tau_{f_2(a)}}^{-1} \times \ldots \times {\tau_{f_n(a)}}^{-1})(f_1(x), f_2(x), \ldots, f_n(x)) \\
&= {\tau_{(f_1(a), f_2(a), \ldots, f_n(a))}}^{-1}(f_1(x), f_2(x), \ldots, f_n(x)) \\
&= (f_1(x), f_2(x), \ldots, f_n(x)) - (f_1(a), f_2(a), \ldots, f_n(a)) \\
&= (f_1, f_2, \ldots, f_n)(x) - (f_1, f_2, \ldots, f_n)(a)
\end{aligned}$$

In short, $(L_1, L_2, \ldots, L_n)$ is a graph differential of $(f_1, f_2, \ldots, f_n)$ at a.

3. Suppose that L_1, L_2, ..., L_n are postdiscrete differentials of f_1, f_2, ...,, f_n, respectively, at a. Let x be a vertex of A such that there is an edge from a to x.

   There are filters $\mathcal{H}_1$, $\mathcal{H}_2$, ... $\mathcal{H}_n$ strongly converging to L_1, L_2, ..., L_n respectively such that, for $i = 1, 2, \ldots n$,

   (a) L_i belongs to every member of $\mathcal{H}_i$, and

   (b) for every $H \in \mathcal{H}_i$, there is at least one M in H such that

$$M(x - a) = f_i(x) - f_i(a)$$

   Let $\mathcal{H}$ be the filter generated by all $H_1 \times H_2 \times \ldots \times H_n$ such that H_i belongs to $\mathcal{H}_i$ $(i = 1, 2, \ldots n)$. Then $(L_1, L_2, \ldots L_n)$ belongs to every member of $\mathcal{H}$. Furthermore, by Lemma 3.2.44, $\mathcal{H}$ strongly converges to $L_1, L_2, \ldots, L_n$.

Finally, let H be an arbitrary member of $\mathcal{H}$. There exist H_1, H_2, ... H_n in $\mathcal{H}_1$, $\mathcal{H}_2$, ... $\mathcal{H}_n$ respectively such that

$$H_1 \times H_2 \times \ldots \times H_n \subseteq H$$

There exist M_1, M_2, ... M_n in H_1, H_2, ... H_n respectively such that, for $i = 1, 2, \ldots n$,

$$M_i(x - a) = f_i(x) - f_i(a)$$

Then $(M_1, M_2, \ldots M_n)$ is a member of H such that

$$(M_1, M_2, \ldots M_n)(x - a) = (f_1, f_2, \ldots, f_n)(x) - (f_1, f_2, \ldots, f_n)(a)$$

In short, $(M_1, M_2, \ldots M_n)$ is a postdiscrete differential of $(f_1, f_2, \ldots, f_n)$ at a.

4. For the case of pretopological differentials, modify the proof for postdiscrete differentials by replacing strong convergence by weak convergence. □

3.3 Differential calculus on hybrid structures

3.3.1 Differentiating a function from $3\mathbb{R}$ to $\mathbf{K}_3^-$

$\mathbf{K}_3^-$ is the complete directed graph on 3 vertices, but with one edge from one vertex to another removed. It is universal for all the pretopological convergence spaces in the sense that every pretopological space embeds in some Cartesian power of it (see Bordaud [Bor76]). The space $3\mathbb{R}$ is our designation for the set of real numbers equipped with Euclidean filter structure at each real number r, and in addition at r, all filters containing the filters generated by the open intervals whose right end point is r, and all filters containing the filters generated by the open intervals whose

left end point is r. Take all functions from $3\mathbb{R}$ to $\mathbf{K}_3^-$ that are piecewise constant at 0 for differentials. Then $g : 3\mathbb{R} \longrightarrow \mathbf{K}_3^-$ is a differential of $f : 3\mathbb{R} \longrightarrow \mathbf{K}_3^-$ at r iff f is constant on an open interval whose right end point is r and constant on an open interval whose left end point is r.

Chapter 4

Dynamical Systems as Instances of the Differential Scheme

Now that convergence spaces and differential calculi based upon them have been covered in some detail, we can formally present example instances of the differential scheme.

4.1 Discrete Dynamical Systems

4.1.1 Classical Computation

Recall the standard reduction of a Turing Machine (TM) to an equivalent semi-Thue system [Dav82], which yields a set of productions describing transitions between the instantaneous descriptions of the TM. Assuming such a reduction has been made for a given TM, we show how to obtain a (1d,1r)-CA, i.e., a one-dimensional, radius 1 cellular automaton, equivalent to the original TM.

The differential scheme representing the CA has the following components:

1. Comp = $\mathbb{Z}$ (the set of CA cells indexed by integers corresponds to a two-way infinite tape)

2. Time = $\mathbb{N}$ (discrete time steps starting from 0)

3. $\mathcal{L}$ = original TM tape alphabet

   $\cup$ $\{[q,a]\,|\,q$ a TM internal state, a a TM alphabet symbol$\}$

   (see below for the explanation of this local state space, shared by all elements of the computation space)

4. the differentials are just the CA update rules corresponding the to TM program; an update rule is just a finite function, and all finite functions can be specified via differentials in the appropriate differential calculus.

Suppose the semi-Thue system corresponding to the given TM includes the right-moving transition $\mu qab\nu \implies \mu a'q'b\nu$ between successive instantaneous descriptions of the TM. The global state of the CA corresponds to an instantaneous description of the TM, except that precisely one of the new symbols $[q,a]$ is used to indicate the cell corresponding to the TM square under scan as well as that squares's value and the current TM internal state. Thus, the equivalent step of the CA's computation is $\cdots\mu[q,a]b\nu\cdots \implies \cdots\mu a'[q',b]\nu\cdots$. The remaining TM transition types are emulated similarly. Note that only the CA cell "under scan" and perhaps its left or right neighbor can be affected in a single step. It is thus clear that only 1-dimensional cellular automata of radius 1 need be considered for full computational generality.

A *computation* (as defined in §1.2) of this instance of the differential scheme is a map:

$$u : \text{Time} \longrightarrow \text{GlSt}$$

i.e.,

$$u : \mathbb{N} \longrightarrow \Pi_{x \in \mathbb{Z}} \mathcal{L}$$

which, in this case, clearly traces the progression of global configurations of our (1d,1r)-CA, which, in turn, clearly tracks the progression of instantaneous descriptions of the original TM.

4.2 Continuous Dynamical Systems

We consider continuous dynamical systems described by systems of ordinary differential equations (ODEs). Such systems progress from the simplest ones consisting of a single ODE, to finite systems of ODEs, to systems of infinitely many ODEs. The following examples show how to interpret such continuous dynamical systems as instances of the differential scheme.

4.2.1 A Simple ODE

Consider the single autonomous ODE:

$$\frac{dx}{dt} = f(x) \tag{4.1}$$

As an ODE over the real domain, a solution to (4.1) is a function

$$u : \text{Time} \longrightarrow \mathbb{R}$$

such that

$$\left.\frac{du}{dt}\right|_{t_0} = f(u(t_0)), \quad \text{for all } t_0 \in \mathbb{R}$$

In differential form, we have:

$$\begin{aligned} D_{t_0}u &= \lambda\Delta.[f(u(t_0))](\Delta) \\ &= [f(u(t_0))] \end{aligned}$$

As an instance of the differential scheme, we have:

1. Comp $= \{x\}$ (a singleton set)
2. $\mathcal{L} = \{\mathbb{R}\}$
3. Time $= \mathbb{R}$
4. differentials are the linear operators on $\mathbb{R}$, of classical analysis; we have established that these form an instance of our generalized definition of a differential calculus in §3.1.1

Thus, a solution is just a *computation*:

$$u : \text{Time} \longrightarrow \text{GlSt} = \Pi_{x\in\{x\}}\mathcal{L}$$

which is, in this case:

$$u : \mathbb{R} \longrightarrow \mathbb{R}$$

4.2.2 A Finite System of ODEs

Consider the finite system of autonomous ODEs:

$$\frac{dx_i}{dt} = f_i(x_1, \ldots, x_n), \quad i = 1, \ldots, n \tag{4.2}$$

A solution to (4.2) is a function

$$u : \text{Time} \longrightarrow \mathbb{R}^n$$

(via $t \mapsto [u_1(t), \ldots, u_n(t)]^T$; i.e., we view elements of $\mathbb{R}^n$ as column vectors) such that

$$\left.\frac{du_i}{dt}\right|_{t_0} = f_i(u_1(t_0), \ldots, u_n(t_0)), \quad \text{for all } t_0 \in \mathbb{R}$$

Expressed in differential form:

$$\begin{aligned} D_{t_0} u_i &= \lambda\Delta.[f_i([u_1(t), \ldots, u_n(t)]^T)](\Delta) \\ D_{t_0} u &= \lambda\Delta.[f([u_1(t), \ldots, u_n(t)]^T)](\Delta) \\ &= [f([u_1(t), \ldots, u_n(t)]^T)] \end{aligned}$$

As an instance of the differential scheme, we have:

1. Comp $= \{x_1, \ldots, x_n\}$, with the convergence structure (to be defined) of K_n, the complete graph on n vertices
2. $\mathcal{L} = \{\mathbb{R}\}$ ($\mathbb{R}$ is the local state space for each each x_i; i.e., $x_i : \text{Time} \longrightarrow \mathbb{R}$)
3. Time $= \mathbb{R}$
4. the differentials are linear operators $\mathbb{R} \longrightarrow \mathbb{R}^n$

Chapter 5

Quantum Computation

> *The miracle of the appropriateness of the language of mathematics for the formulation of the laws of physics is a wonderful gift which we neither understand nor deserve.*
>
> - EUGENE WIGNER, Richard Courant Lecture, New York University (1959)

Informally, quantum computation is computation based on quantum mechanics. Since all actual computation is physical, and all physical interactions are ultimately quantum mechanical, this informal definition must be sharpened to distinguish quantum computation from "classical" computation. Quantum computations proceed by "quantum interactions," interactions of individual quantum-level objects, microscopic entities such as individual electrons and photons. In quantum theory, the "classical interactions" we observe in the macroscopic world are accounted for as limiting cases of interactions of vast numbers of quantum-level objects, treated statistically.[1] Classical interactions do not share some rather bizarre-seeming properties enjoyed by quantum interactions. The essential idea behind quantum computing is to exploit those properties peculiar to quantum interactions for computational pur-

[1] This is a version of Bohr's famous "Correspondence Principle." [Boh13, Boh76]

poses.[2]

In a quantum computer, the current state of a computation would be the state of some quantum system, an ensemble of quantum-level entities, which state would be updated by quantum-level interactions. In contrast, the current state of a computation in a "classical" computer can be updated only through classical interactions. The standard abstract model of computation via Turing machines [Dav82] assumes only classical interactions, as do programs for modern general purpose digital computers.[3]

Broad interest in quantum computing is widely credited to an influential paper of Feynmann [Fey82][4], who reasoned that a quantum computer would be more efficient in simulating quantum systems than a classical computer. His rationale, essentially, was that one of those bizarre quantum features — *superposition*[5]— provides a kind of parallelism to quantum computers that is denied their classical counterparts. Moreover, the same principle makes a quantum computer capable of simulating larger and larger quantum systems without the exponential blow-up in resource requirements that a classical computer appears to demand.

We caution that quantum computing as discussed herein is mostly theoretical. Constructing actual quantum computers has proven highly problematic, largely due to the difficulty of finding quantum systems that are both suitable for maintaining the evolving state of a quantum computation and that can be effectively isolated from

[2]Useful sources for quantum computing include the texts [Hir04], [KSV02], [NC00], [Gru99] and [Pit99], and their references, particularly the bibliography of [NC00]. Many relevant articles are downloadable from the arXiv.org e-Print archive for quantum physics at `http://www.arxiv.org/archive/quant-ph`.

[3]Of course, modern computer hardware is based on semiconductor technology, and semiconductivity is a quantum-level phenomenon. However, here the state of a computation is viewed macroscopically, and updated accordingly; one cannot write programs that take direct advantage of the quantum-level interactions occuring inside the transistors of current digital computers.

[4]The last few paragraphs of the Introduction to Manin's [Man80] are cited by Kitaev, Shen and M.N. Vyalyi [KSV02] as bearing first mention of the prospects for quantum computing; see [Man99] for a translation of this excerpt. Benioff [Ben80, Ben82] also considered quantum mechanical computation; this work grew out of a line of research on the physics of computation originating with Landauer [Lan61] and continuing with the work of Bennett [Ben73, Ben89] and others.

[5]See below for definition.

the environment. In the literature, undesirable interaction with the environment has been styled the "decoherence problem."

Perhaps the quantum computational result of the greatest potential impact, to date, was Shor's discovery of an efficient ($O(n^3)$) quantum algorithm for factoring numbers and finding discrete logarithms [Sho94]. Shor's algorithms could have enormous practical impact on cryptography.[6] For example, the popular RSA public-key cryptosystem depends upon the infeasibility of factoring large numbers for its security — efficient large number factorization would render RSA encryption nugatory.[7] Other well-known cryptosystems, e.g., the ElGamal public key cryptosystem, rely on the infeasibility of solving the discrete logarithm problem.[8]

Quantum mechanics, and so quantum computing, is a highly formalized theory, founded on the theory of Hilbert spaces and the operators that live there. In typical physical situations, the relevant Hilbert spaces are infinite-dimensional, whose treatment requires much of the armamentarium of functional analysis.[9] For the quantum computing algorithms treated in the literature, however, only finite-dimensional Hilbert spaces need be considered, which simplifies matters considerably.

[6]The well-known work of Bennett and Brassard [BB84] on "quantum key distribution" is a fascinating application of quantum-level phenomena to cryptography, but we do not consider it a milestone in quantum computation per se because its effectiveness actually *requires* the decoherence normally assumed absent in theoretical discussions of quantum algorithms. Because such quantum cryptosystems do not rely on the stability and isolation of an evolving quantum system, which is of the essence for quantum computation in general, significant progress has been made in constructing working quantum cryptoystems, up to claims of imminent commercialization.

[7]It should be pointed out that, to date, actual quantum computers constructed to factor numbers can handle only very small numbers, while breaking the RSA cryptosystem would require factoring numbers on the order of hundreds of decimal digits.

[8]See [TW02] for details of the ElGamal and RSA cryptosystems.

[9]See the last chapter of [Kre89] for applications of functional analysis to quantum mechanics in the infinite-dimensional case.

Mathematical Background, Notations and Conventions

We assume basic finite-dimensional vector space theory is familiar,[10] though some review is given below mainly to fix the Dirac "bra-ket" notation used in the literature of quantum physics and quantum computing.

- Let the natural numbers, integers, rationals, reals and complex numbers be denoted by $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$, respectively.
- Let $\mathbb{B} = \{0, 1\}$ be the set of Boolean values.
- For $\alpha \in \mathbb{C}$, let α^* denote the conjugate of α; i.e., if $\alpha = x+yi$ ($x, y \in \mathbb{R}$, $i = \sqrt{-1}$), then $\alpha^* = x - yi$.
- Let V be a vector space over $\mathbb{C}$. If V has finite dimension $n > 0$, then V is isomorphic to $\mathbb{C}^n$, for which we write $V \cong \mathbb{C}^n$.
- A vector in V is denoted by a **ket**, $|v\rangle$, where v is some convenient label.[11]

  For n-dimensional V, a ket $|v\rangle$ is represented by a column vector whose n entries are the coordinates of $|v\rangle$ with respect to some fixed basis of V.
- Linear combinations of vectors like $\alpha_1|v_1\rangle + \alpha_2|v_2\rangle$ may be written $|\alpha_1 v_1 + \alpha_2 v_2\rangle$.
- An **inner product** on V is a function $V \times V \longrightarrow \mathbb{C}$ that maps each pair of vectors $(|v\rangle, |w\rangle)$ to a complex number, written $\langle v|w\rangle$, such that:

  1. (linearity) For all $|v\rangle, |w_1\rangle, |w_2\rangle \in V$ and $\alpha \in \mathbb{C}$, we have

     $$\langle v|w_1 + \alpha w_2\rangle = \langle v|w_1\rangle + \alpha\langle v|w_2\rangle$$

  2. (conjugacy) For all $|v\rangle, |w\rangle \in V$, we have $\langle w|v\rangle = \langle v|w\rangle^*$

[10] See [Lan04] or [Axl97] for the necessary background.

[11] We use $\mathbf{0}$ to denote the zero vector, however, to avoid possible confusion with $|0\rangle$, often found in the literature as denoting a (non-zero) basis element.

3. (positivity) For all $|v\rangle \in V$, $|v\rangle \neq \mathbf{0}$, we have $\langle v|v\rangle > 0$

Note that 1 and 2 together imply that $\langle v_1 + \alpha v_2, w\rangle = \langle v_1, w\rangle + \alpha^*\langle v_2, w\rangle$,[12] condition 2 implies that $\langle v, v\rangle \in \mathbb{R}$, so the inequality $\langle v|v\rangle > 0$ in condition 3 makes sense, and 1 and 3 imply that $\langle v, v\rangle = 0$ if and only if $v = 0$.

- If V is equipped with an inner product it is called a **complex inner product space**. A finite-dimensional complex inner product space is called a **Hilbert space**.[13]

  Let H denote a Hilbert space (of unspecified finite dimension), and H_n denote n-dimensional Hilbert space. Thus, $H_n \cong \mathbb{C}^n$.

- We say vectors $|v\rangle, |w\rangle \in H$ are **orthogonal** if and only if $\langle v|w\rangle = 0$.

- The **length** of a vector $|v\rangle \in H$, denoted by $\||v\rangle\|$, is defined to be the positive square root of $\langle v|v\rangle$.[14] We call $|v\rangle$ a **unit vector** if and only if $\||v\rangle\| = 1$.

- An **orthonormal** basis for H_n is a basis $\{|u_1\rangle, \ldots, |u_n\rangle\}$ such that $\langle u_i|u_j\rangle = 1$ if $i = j$, and $\langle u_i|u_j\rangle = 0$ if $i \neq j$. Every Hilbert space has an orthonormal basis.

- For $|v\rangle \in H$, the linear functional $H \longrightarrow \mathbb{C}$ via $|w\rangle \mapsto \langle v|w\rangle$ is denoted by a **bra**, $\langle v|$. Thus, $\langle v|w\rangle$ (a **bra-ket**) is shorthand for the function application $\langle v||w\rangle$.

  With respect to some fixed orthonormal basis of $H \cong \mathbb{C}^n$, say $\{|u_1\rangle, \ldots, |u_n\rangle\}$, a bra is represented by a row vector whose n entries are the conjugates of the corresponding entries of the column vector representing $|v\rangle$ in the same basis. For example, if $|v\rangle = |\sum_{i=1}^n \alpha_i u_i\rangle$ and $|w\rangle = |\sum_{i=1}^n \beta_i u_i\rangle$, then $\langle v|w\rangle = \sum_{i=1}^n \alpha_i^* \beta_i$, the usual inner product in $\mathbb{C}^n$.

[12] This property is called *conjugate-linearity*, or *anti-linearity*.

[13] An infinite-dimensional complex inner product space must also be analytically *complete* with respect to the metric induced by its inner product in order to be designated a Hilbert space. A good, concise source for the basic theory of Hilbert spaces, both finite- and infinite-dimensional, is [Ger85].

[14] The metric induced by the inner product is just $d(|v\rangle, |w\rangle) = \||v\rangle - |w\rangle\|$.

- The **adjoint** of a linear operator $T : H \longrightarrow H$ is the unique linear operator T^* such that $\langle v|Tw\rangle = \langle T^*v|w\rangle$. T is said to be **self-adjoint** if $T^* = T$. It is **unitary** if it is invertible and $T^{-1} = T^*$.

  Note that, if T is unitary, then $\langle Tv|Tw\rangle = \langle T^*Tv|w\rangle = \langle T^{-1}Tv|w\rangle = \langle v|w\rangle$; i.e., T preserves inner products (and therefore lengths). The composition of unitary operators is unitary, but the composition of self-adjoint operators is not self-adjoint unless the operators commute.[15]

5.1 Quantum Mechanics and Quantum Systems

A discussion of the physical conceptual background to quantum mechanics would take us too far afield.[16] We cover just enough quantum mechanics to establish models of quantum computation. However, the following example should help motivate the formal theory of quantum mechanics, as well as provide a glimpse of the "quantum weirdness" the theory must encapsulate.

Suppose a stream of photons with random polarization orientations is shot at a vertically polarized filter. Half the photons are observed to pass through the filter, though this cannot mean that half the photons were vertically oriented to begin with. It is as though half the photons "chose" to align themselves vertically when confronted with the filter. The photons passing through the filter are now vertically oriented, and if a second vertical filter is placed behind the first one, all of the photons that passed through the first vertical filter are observed to pass through the second. If a horizontal filter is placed behind a vertical filter, none of the photons passing through the vertical filter would pass through the horizontal filter. With respect to the set of orientations $R = \{\text{vertical}, \text{horizontal}\}$, once a photon "chooses"

[15] For unitary U_1 and U_2, we have $(U_1U_2)^* = U_2^*U_1^* = U_2^{-1}U_1^{-1} = (U_1U_2)^{-1}$, but for self-adjoint S and T, $(ST)^* = T^*S^* = TS$.

[16] See [Boh89], [Mes99] and [Per95] for this; [AK03] is a charming non-technical source.

an orientation, its "choice" seems permanent.

However, if a filter oriented diagonally at 45° from the vertical were placed behind a vertical filter, half the (now vertically polarized) photons passing through the vertical filter would pass through the diagonal filter as well. Somehow, it seems that when another choice is offered from the distinct orthogonal orientation set $D = \{45°, 135°\}$, the previous choice of orientation isn't so permanent. Such experiments challenge the common notion that objects have "intrinsic state," and underscore the non-intuitiveness of quantum physics.

5.1.1 Hilbert Space Formalization

An n-level quantum system is an ensemble of quantum-level objects for which n distinct states can be observed. That is, as the result of an *observation* (or *measurement*) of the system, n different outcomes are possible.

In the standard Hilbert space formalism of quantum mechanics, n-level quantum systems are identified with n-dimensional Hilbert space, H_n. The states of an n-level quantum system are represented by unit vectors of H_n, collectively called the **state space**. An **observation** of the system yields a state. Observations are always made with respect to a given orthonormal basis, and the set of states it is possible to observe are the elements of the basis with respect to which the observation is made. That is, the choice of orthonormal basis for H_n determines which states may be observed. Symbolically, if the orthonormal basis $B = \{|x_1\rangle, \ldots, |x_n\rangle\}$ is chosen, then an observation with respect to B yields one of the basis states $|x_i\rangle$.

In general, the state of an n-level quantum system that is not being observed may be *any* unit vector of H_n; i.e., the state may be a linear combination, or

superposition, of basis states:

$$\alpha_1|x_1\rangle + \cdots + \alpha_n|x_n\rangle$$

The complex coefficients α_i of the basis states are called **amplitudes**. Because states are unit vectors, we have $\sum_1^n |\alpha_i|^2 = 1$.[17] The square of the absolute value of an amplitude is interpreted as the probability that, upon observation, the corresponding basis state will be the state observed; symbolically, $|\alpha_i|^2 = \Pr(|x_i\rangle)$. The mapping $\Psi(x_i) = \alpha_i$ is called the **wave function** with respect to basis $\{|x_1\rangle, \ldots, |x_n\rangle\}$.[18]

In the example preceding this section, we may take either set of orientations R or D as a basis for H_2. A photon observed with a vertical filter may only be seen to have a vertical orientation in the R basis (if it passes through the filter) or an implied horizontal orientation (if it doesn't). Similarly, the photon may only be seen to have a 45° or 135° orientation when measured with respect to the D basis (say using a 45° filter).

We won't delve further into the subject of quantum measurement, except to mention that observations of a quantum system involve classical interactions of some kind and "force" the state of the system to a basis element. However, the state of an n-level quantum system may evolve when it is not being observed. Operations which change the state of the system without observation (and therefore, in general, map superpositions to superpositions) are represented by unitary operators on H_n. Thus, if $|v\rangle \in H_n$ is the current state of the system and $U : H_n \longrightarrow H_n$ is a unitary operator, the next state of the system may be $U|v\rangle$. A system may evolve through the action of any number of unitary operators.

Observations with respect to a basis $B = \{|x_1\rangle, \ldots, |x_n\rangle\}$ are represented by self-

[17] Clearly, there are uncountably many superposition states, but only n observable states, for $n > 1$.

[18] It is the finite-dimensional analog to the wave function ψ appearing in Schrödinger's equation: $i\hbar\frac{\partial\psi}{\partial t} = H\psi$. Here, H represents the Hamiltonian operator appropriate to the physical situation and $\hbar = h/2\pi$, where h is Planck's constant: $6.62608 \times 10^{-34} Js$.

adjoint operators on the space for which B is a basis.

5.1.2 Compound Quantum Systems

Let H_n with basis $\{|x_1\rangle, \ldots, |x_n\rangle\}$ represent some n-level quantum system and H_m with basis $\{|y_1\rangle, \ldots, |y_m\rangle\}$ represent some m-level quantum system. The state space for the compound system comprising H_n and H_m is given by the **tensor product** of H_n and H_m, denoted $H_n \otimes H_m$.

The tensor product $H_n \otimes H_m$ is the Hilbert space generated by the basis elements $|x_i\rangle \otimes |y_j\rangle$, where $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. As is convenient, we may abbreviate $|x_i\rangle \otimes |y_j\rangle$ by writing $|x_i\rangle|y_j\rangle$ or $|x_i y_j\rangle$. The function $\otimes : H_n \times H_m \longrightarrow H_n \otimes H_m$ is **bilinear** (linear in both arguments), so, for arbitrary $|v\rangle = |\sum_{i=1}^{n} \alpha_i x_i\rangle \in H_n$ and $|w\rangle = |\sum_{j=1}^{m} \beta_i y_j\rangle \in H_m$, we have:

$$|v\rangle \otimes |w\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j |x_i y_j\rangle$$

The states of a compound system are its unit vectors, as always, so $|v\rangle \otimes |w\rangle$ above is a state of $H_n \otimes H_m$ if and only if $\sum_{i=1}^{n} \sum_{j=1}^{m} |\alpha_i \beta_j|^2 = 1$. Note that

$$\dim(H_n \otimes H_m) = \dim(H_n) \cdot \dim(H_m).$$

In fact, it is easily shown that $H_n \otimes H_m \cong H_{n \cdot m}$. This stands in contrast to the case of direct sums, for which

$$\dim(H_n \oplus H_m) = \dim(H_n) + \dim(H_m).$$

Let $|x\rangle = \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} |x_i y_j\rangle$ be a state of $H_n \otimes H_m$. We say $|x\rangle$ is **decomposable** if it can be expressed as $|v\rangle \otimes |w\rangle$ for some states $|v\rangle \in H_n$ and $|w\rangle \in H_m$. A non-decomposable state is called an **entangled** state. Decomposability means there are

complex $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$ such that $\sum_{i=1}^{n} |\alpha_i|^2 = \sum_{j=1}^{m} |\beta_j|^2 = 1$ and

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \gamma_{ij}|x_i y_j\rangle = (\sum_{i=1}^{n} \alpha_i|x_i\rangle)(\sum_{j=1}^{m} \beta_j|y_j\rangle)$$

5.1.3 Examples of Quantum Systems

Example 5.1.1. ***A 2-level Quantum System***

We are concerned with states in H_2, represented with respect to an orthonormal basis $\{|0\rangle, |1\rangle\}$. Let W be the following unitary evolution operator:

$$W(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$W(|1\rangle) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

Applied to initial state $|0\rangle$, the evolution operator yields the state $W(|0\rangle)$:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle,$$

a superposition for which $\Pr(0) = \Pr(1) = \frac{1}{2}$. If we update $W(|0\rangle)$ with another application of W, we get $W^2(|0\rangle)$:

$$\begin{aligned} &\frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) + \frac{1}{\sqrt{2}}(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle) \\ &= (\frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle) + (\frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle) \\ &= |0\rangle, \end{aligned}$$

with $\Pr(0) = 1$ and $\Pr(1) = 0$. Similarly, $W^2(|1\rangle) = |1\rangle$. Thus, W^2 is the identity operator on H_2, so W acts as a "square root" of the identity on H_2. In matrix

notation, we have

$$W = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

This matrix is called a Walsh-Hadamard *matrix.*

Note the constructive interference *of the amplitudes of* $|0\rangle$ *and the* destructive interference *of the coefficients of* $|1\rangle$ *as* W *operates on* $W(|0\rangle)$*. Quantum systems permit destructive interference because the amplitudes are complex-valued. In classical probabilistic systems, the probability coefficients analogous to quantum probability amplitudes must be non-negative real numbers, making destructive interference impossible.*

Example 5.1.2. ***A Compound Quantum System***

Consider the compound quantum system $H_2 \otimes H_2$*, with orthonormal basis*

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

and the following update operator:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

We see that $M(|00\rangle) = |00\rangle$*,* $M(|01\rangle) = |01\rangle$*,* $M(|10\rangle) = |11\rangle$ *and* $M(|11\rangle) = |10\rangle$*.*

5.2 Standard Models of Quantum Computation

The machinery of classical computation operates on standard units of information called bits, where the value of a bit is either 0 or 1. In quantum computation, the

qubit [Deu85] is the standard unit of information. A qubit is simply a state in a 2-level quantum system, that is, in H_2. It is conventional to use $\{|0\rangle, |1\rangle\}$ as the orthonormal basis of H_2. Generally, a state is a superposition of $|0\rangle$ and $|1\rangle$. Systems of n qubits are modeled by $H_2 \otimes \cdots \otimes H_2$ (n H_2 factors) or, equivalently, by H_{2^n}, and are called **n-bit quantum registers**.

5.2.1 Quantum Turing Machines

Deutsch [Deu85] was the first to define a quantum analog of the Turing machine (abbreviation: TM) model of classical compuation. We assume the standard Turing machine model is familiar.[19]. Let Q be the set of states, A the tape alphabet and δ the transition function of a Turing machine, so δ is a partial map:

$$\delta : Q \times A \rightharpoonup Q \times A \times \{-1, 0, 1\} \tag{5.1}$$

with the usual interpretation: if the machine is in state $q \in Q$ scanning a tape cell containing alphabet symbol $a \in A$, then $\delta(q, a) = (q', a', d)$ provides the next state q', the new symbol a' is printed in that cell, and the new position of the tape head relative to its previous position (i.e., if p is the position of the tape head, then $p + d$ is its next position). As usual, the computation halts if $\delta(q, a)$ is undefined.

A **probabilistic Turing machine (PTM)** is a TM for with an altered transition function

$$\delta : Q \times A \times Q \times A \times \{-1, 0, 1\} \longrightarrow [0, 1] \tag{5.2}$$

where $[0, 1]$ is the unit interval in $\mathbb{R}$. We interpret the new transition function as follows: $\delta(q, a, q', a', d)$ is the probability that, when the machine is in state q scanning a cell with symbol a, it will switch to state q', print a' in that cell, and

[19] See [Dav82] for a model using Post-style quadruples, and [Pap94] for a model employing the more common Turing-style quintuples

adjust the head position according to d. For the probability interpretation to make sense, it is required that $\sum_{(q',a',d)\in Q\times A\times\{-1,0,1\}} \delta(q,a,q',a',d) = 1$.

A **Quantum Turing Machine (QTM)** is similar to a probabilistic TM, except, instead of a transition function like (5.2), we have a transition function

$$\delta : Q \times A \times Q \times A \times \{-1,0,1\} \longrightarrow \mathbb{C} \tag{5.3}$$

where $\sum_{(q_2,a_2,d)} |\delta(q_1,a_1,q_2,a_2,d)|^2 = 1$. We assume that $\delta(\cdot,\cdot,\cdot) = x + yi$ with $x, y \in \mathbb{Q}$ to avoid "real" problems. As with a PTM, at any step of a computation of a QTM Q, a finite number of tape configurations ("basis configurations") are possible; over all possible Q computations, the number of basis configurations is clearly countably infinite. In general, a configuration of Q at a particular time step is a superposition

$$\alpha_1|c_1\rangle + \cdots + \alpha_n|c_n\rangle$$

where the $|c_i\rangle$ are the possible basis configurations at that step and $\sum_{i=1}^{n} |\alpha_i|^2 = 1$.[20] Let M_δ be the evolution operator on (infinite-dimensional) configuration space determined by δ. Then M_δ must be *unitary*.

It has been shown [Deu85] that QTMs and ordinary TMs compute the same set of functions. The promise of quantum computation has always been that some functions not *efficiently* (i.e., polynomial time) computable by any TM may be efficiently computed by a QTM.

[20]Note that there would be uncountably many general configurations if the α_i were allowed to have arbitrary elements of $\mathbb{R}$ for their real and imaginary parts — this is the "real" problem mentioned earlier. By insisting these parts be rational, we avoid this problem. Wherever amplitudes with irrational real or imaginary parts are shown, assume a rational approximation is used instead.

5.2.2 Quantum Circuits

As one might guess from the above, QTMs are rather unwieldy devices for the specification of quantum computations. Instead, **quantum circuits**, the quantum analog to classical Boolean circuits, are used. Quantum circuits are composed of **quantum gates**, which serve a purpose analogous to ordinary Boolean logic gates. However, whereas ordinary gates are Boolean functions $f : \mathbb{B}^n \longrightarrow \mathbb{B}$ carrying fixed-length sequences of bits to bits, a quantum gate is a unitary operator $F : \bigotimes_{i=1}^{n} H_2 \longrightarrow \bigotimes_{i=1}^{n} H_2$ mapping quantum registers (fixed-length sequences of qubits) to quantum registers. As unitary maps, all quantum gates are reversible, which certainly isn't the case classically.[21]

Example 5.2.1. ***The Quantum NOT Gate***

This is a unary operator $M_\neg : H_2 \longrightarrow H_2$, for which $M_\neg(|0\rangle) = |1\rangle$ and $M_\neg(|1\rangle) = |0\rangle$. Clearly,

$$M_\neg = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

inverts the basis vectors. For general superpositions $\alpha_1|0\rangle + \alpha_2|1\rangle$, we have $M_\neg(\alpha_1|0\rangle + \alpha_2|1\rangle) = \alpha_2|0\rangle + \alpha_1|1\rangle$.

Example 5.2.2. ***The Quantum $\sqrt{NOT}$ Gate***

Let

$$M_{\sqrt{\neg}} = \begin{pmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{pmatrix}$$

Note that $(M_{\sqrt{\neg}})^2 = M_\neg$, so $M_{\sqrt{\neg}}$ acts as a "square root of NOT" on H_2.

Example 5.2.3. ***The "Controlled NOT" Quantum System***

This is a binary quantum gate, that is, a unitary $M_{cnot} : H_2 \otimes H_2 \longrightarrow H_2 \otimes H_2$ (equivalently, a unitary $M_{cnot} : H_4 \longrightarrow H_4$), which, when presented with a pair of

[21] For example, consider the classical two-input AND gate: three different combinations of inputs yield the 0 output.

qubits, acts as the quantum NOT gate on the second qubit if the first qubit is $|1\rangle$ *and as the identity operator on the second qubit if the first qubit is* $|0\rangle$*. That is,* $M_{cnot}(|00\rangle) = |00\rangle$*,* $M_{cnot}(|01\rangle) = |01\rangle$*,* $M_{cnot}(|10\rangle) = |11\rangle$ *and* $M_{cnot}(|11\rangle) = |10\rangle$*. But this is precisely the action of the unary operator*

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

of Example 5.1.2.

We have not specified the physical implementation of any quantum gates, nor how to combine such gates into quantum circuits for the implementation of quantum algorithms. For mathematical purposes, we simply assume this can be done. In practice, a quantum computation implemented with a quantum circuit proceeds in "Prepare-Evolve-Observe" lock-step:

1. Preparing a sequence of qubits (quantum register) with some initial value.
2. Evolving the system via various unitary operations, i.e., sending the qubits through various quantum gates.
3. Observing the qubits, thereby projecting any superposition the final state may represent into one of the possible basis elements.[22]

Of course, the basis element "chosen" when a superposition is measured depends probabilistically on the amplitudes of that superposition at the moment of observation. Hence, the output observed most frequently from a given quantum circuit, when the computation is repeated many times with the same initial qubit values, should reflect the basis element whose final amplitude has the highest absolute value.

[22] Also referred to as "collapsing the wave function."

Chapter 6

Quantization of Cellular Automata

Besides the well-known models of quantum computation reviewed in Chapter 5, another quantum computational model, the **quantum cellular automata** (QCA), has been defined by Watrous [Wat95]. Because each instance of the differential scheme is a kind of generalized cellular automata, we investigate and generalize this model.

6.1 The Watrous QCA Construction

According to [Wat95], a **one-dimensional quantum cellular automaton** M is a quadruple (Q, δ, k, A), where Q is a finite set of *states* that includes a distinguished state called the *quiescent state* (denoted by ϵ), δ is a *local transition function*, k is an integer denoting the *acceptance cell*, and $A \subseteq Q$ is a set of *accepting states*. M is taken to have a two-way infinite sequence of *cells* indexed by the integers, $\mathbb{Z}$. (Think of a 2-way infinite quantum bit register.) The *neighborhood* of a given cell is that cell together with the cells immediately to its left and right; i.e., neighborhoods are assumed to have radius 1.

In an instance M of Watrous's one-dimensional, radius-1 quantum cellular au-

tomaton ((1d,1r)-QCA) model, a *configuration* is a map

$$a : \mathbb{Z} \longrightarrow Q$$

where, $\forall n \in \mathbb{Z}$, $a(n)$ is the state of the cell indexed by n. Importantly, it is assumed that, for any configuration a, there are only finitely many integers n for which $a(n) \neq \epsilon$, the quiescent state. In the sequel, we refer to this restriction as the assumption of *finite support*. Assuming finite support, the set $\mathcal{C} = \mathcal{C}(M)$ of all configurations of (1d,1r)-QCA M is countable.

The local transition function δ is a map

$$\delta : Q^4 \longrightarrow \mathbb{C}$$

describing the evolution of M. If three consecutive cells of M are in states q_1, q_2 and q_3, respectively, at time step t, then

$$\delta(q_1, q_2, q_3, q)$$

gives the (complex) probability amplitude with which a cell containing state q_2 (with left and right neighbor cells containing states q_1 and q_3, respectively) will update to state q at time $t + 1$. All cells update simultaneously according to δ. Thus, in the usual quantum computational fashion, any given configuration transforms into a superposition of possible successor configurations, each with its own amplitude.

The significance of the quiescent state in the Watrous model is that every local transition function must observe the following restriction:

$$\delta(\epsilon, \epsilon, \epsilon, q) = \begin{cases} 1 & \text{if } q = \epsilon \\ 0 & \text{otherwise} \end{cases}$$

That is, a cell in the quiescent state whose left and right neighbors are also in the quiescent state, will update to the quiescent state with probability 1.

A configuration of $M = (Q, \delta, k, A)$ where the cell indexed by k contains a state in A is called an *accepting configuration*. Given M and some initial configuration a, the question is: with what probability will we observe M in an accepting configuration after some number of (unobserved) evolution steps according to δ?

The Hilbert space containing the quantum states of M is

$$\ell^2(\mathcal{C}) = \{\, x : \mathcal{C} \longrightarrow \mathbb{C} \mid \sum_{c \in \mathcal{C}} x(c)^* x(c) < \infty \,\}$$

which means that each superposition of M is identified with some $x \in \ell^2(\mathcal{C})$, where $x(c) \in \mathbb{C}$, and $|x(c)|^2$ is the probability of observing configuration c from superposition x. As discussed in Chapter 5, it must be the case that $\sum_{c \in \mathcal{C}} |x(c)|^2 = 1$.

The local transition function δ must ensure that any superposition, produced at any step starting from any valid initial configuration (i.e., from a configuration with finite support), must be such that the sum of the squares of that superposition's amplitudes is 1. This will be so if and only if the global time-evolution operator of M uniquely determined by δ is *unitary*.

It is difficult to determine whether or not a given local transition function yields a unitary global time-evolution. In fact, much of [Wat95] is devoted to specifying a restricted class of (1d,1r)-QCA called **partitioned QCA** and then proving that the question of unitarity for this smaller class can be relatively easily decided.

Watrous's partitioned QCA is based on the **partitioned cellular automata** defined in Morita and Harao [MH89]. These automata yield a state transition by dividing each cell of a (1d,1r)-CCA into left, middle and right subcomponents and updates the 3-component state of each cell by functionally updating as a single unit the triple of values consisting of the values of the right subcomponent of the left-

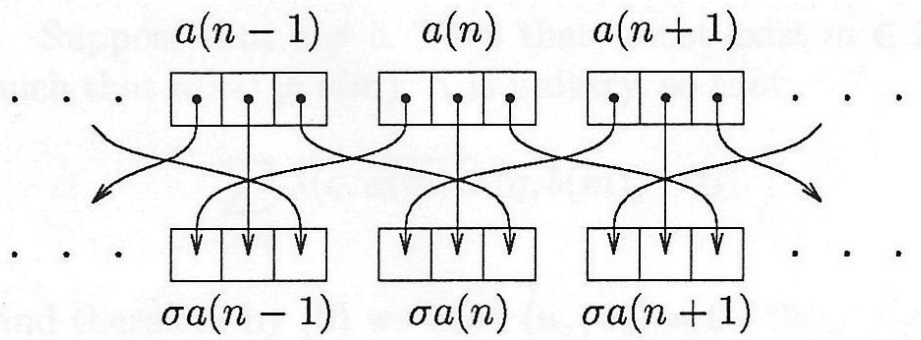

Figure 6.1: Partitioned 1-dimensional QCA [Wat95]

neighbor cell, the middle subcomponent of the middle cell and the left-subcomponent of the right-neighbor cell. The resulting dependency of a new global state, or configuration, of a partitioned (1d,1r)-QCA on an immediately preceding configuration was diagrammatically depicted by Watrous in Figure 6.1.

If we begin with a configuration of the cellular automaton that has *finite support*, i.e. where the state of each cell is quiescent for all but finitely many cells, then every configuration in the sequence of configurations generated by the update rule has finite support. This is sufficient for computational purposes and simplifies the approach to the corresponding quantum state space and the state update operations on it, but substantially restricts the expressive power of the specifications that quantum cellular automata can satisfy and is not completely in the spirit of cellular automata — completely random initial states are not supported, for example.

6.2 The Utility of Finite Support

The set $\mathcal{C}$ of configurations with finite support is countably infinite. We can identify the configuration space with the classical states with finite support of a 2-way infinite classical bit register where the bits of the register are indexed by the integers and by convention, the bit value of 0 is taken to be quiescent.

Suppose, for now, that we had a unitary operator U on the Hilbert space containing the quantum states of a 3-bit quantum register that respects the quiescent state; i.e., U maps the quantum state $|000\rangle$ to $|000\rangle$. Suppose we applied U simultaneously and independently to each contiguous block of 3 cells indexed by the integers $3n$, $3n+1$ and $3n+2$. The resulting operation $\hat{U}$ is guaranteed unitary since U respects the quiescent state. That claim follows since the action of $\hat{U}$ on the quantum state representing any finitely supported configuration is the same as the action of some finite tensor power of U on that quantum state, the power depending on the state. But every finite tensor power of U is unitary. Hence $\hat{U}$ is norm preserving. To see that $\hat{U}$ is surjective, consider any finitely supported configuration c and an interval of the register's bits indexed from $3m$ to $3n-1$, $m < n$, sufficiently long for the configuration to assign the quiescent state to all bits outside the interval. To each element c of the configuration space we associate the vector $|c\rangle$ of the quantum state space, where $|c\rangle(c') = 1$ if $c = c'$ and $|c\rangle(c') = 0$ otherwise. The collection of vectors $\{\, |c\rangle \,|\, c \in \mathcal{C} \,\}$ is a complete orthonormal sequence. It is thus sufficient to show that each $|c\rangle$ has a preimage with respect to $\hat{U}$. This is clear since U preserves quiescent 3-cell blocks and the intervals of c containing the support of c have a preimage with respect to a sufficiently high, but finite, tensor powers of U.

The state update rule for a partitioned (1d,1r)-QCA is the composition of two unitary operators: the first permutes the qubits of the 2-way infinite quantum register in the manner described above in the diagram, and the second is an operator of the form we denoted by $\hat{U}$.

We will show how to construct quantum cellular automata based on Watrous's formalism, but without that formalism's quiescent states, by using shift-invariant Lebesgue measure on Cantor space. Although QCA's with quiescent states are strictly sufficient for computational purposes, removing quiescent states as a requirement allows global QCA states with infinite support that allows the state space of

the QCA to be identified with the class of interpretations of logic-based formalism in a formal methods approach to proving the correctness of QCAs with respect to formal specifications.

6.3 Eliminating the Quiescent State

If there is no quiescent state available with respect to suitable state-update rules, then the configuration space becomes uncountably infinite. Let $\mathbb{Z}$ be the set of integers. The configuration space is given by

$$\mathcal{C} = \{\, c : \mathbb{Z} \longrightarrow \{\, 0, 1 \,\} \,\}$$

which we can identify with the direct product

$$\prod_{k \in \mathbb{Z}} \{\, 0, 1 \,\}$$

Intuitively, the configuration space is the set of all configurations of a 2-way infinite bit register. It is important to realize that the free complex vector space on $\mathcal{C}$ is not separable since $\mathcal{C}$ is uncountable [Ger85], and is therefore unsuitable as a Hilbert space containing the quantum states of our infinite register. Regarding the configuration space itself, there is no reason to give any sort of priority to any component of this direct product. Obviously there is no uniform probability distribution on the configuration space, but if we ask what the probability is of drawing "at random" a configuration with a specified finite set of j components having bit value 0 and k components having bit value 1 with, say, a probability distribution on $\{\, 0, 1 \,\}$ with $\Pr(\{\, 0 \,\}) = p$ and $\Pr(\{\, 1 \,\}) = 1 - p$, then the probability is $p^j(1-p)^k$. The separable Hilbert space $L^2(\mathcal{C})$ will produce the correct statistics for the results of observations of bit values of the qubits. To set up $L^2(\mathcal{C})$ requires

measurable subsets of the configuration space and measurable functions mapping the configurations to complex numbers.

We begin with the Cantor topology on $\mathcal{C}$: each standard basic open set of this topology on the configuration space is determined by a configuration and a finite set of components, i.e., a finite set of integers. Let F be a finite set of integers and let c be a configuration. The standard basic open set determined by F and c is the set of configurations

$$\{\, c' \,|\, c(i) = c'(i), \forall i \in F \,\}$$

i.e., the set of configurations that agree with c on F. The open sets of this topology are arbitrary unions (including the empty union) of the standard basic open sets. The collection of standard basic opens sets gives no priority to any component over any other. No subjective weighting of the components is involved. The collection of Borel sets generated by the Cantor topology consists of the smallest collection of sets of configurations that contains every open set and is closed under complements relative to $\mathcal{C}$ and countable union. Let $B_{c,F}$ be the standard basic open set determined by c and F. There is a standard approach to constructing Lebesgue measure on our set of configurations that we are following.[CR, Fan71] We assign measure 2^{-n} to each standard open set $B_{c,F}$ where n is the size of F. Once this assignment is made, the measure of each Borel set is uniquely determined. The measure assigned to the Borel sets is said to be incomplete: arbitrary subsets of sets with measure 0, which need not be Borel, do not have an assigned measure. This poses difficulties for integrating multi-variable functions defined on direct products of the configuration space. To complete the measure we include all subsets of Borel sets with measure 0 and assign them measure 0 as well. It turns out that, by the Carathéodory extension theorem, the measure of each set in the collection of sets, the Lebesgue measurable sets, that are unions of Borel sets and the newly added measure 0 sets and that is closed under relative complement and countable union must be the same as the

measure of the Borel set if it is to extend the measure on the Borel sets. The measure of a Lebesgue measurable set S is denoted by $\mu(S)$. The Lebesgue measurable functions mapping configurations to complex numbers are those functions for which every preimage of a Borel set of complex numbers (relative to the standard Euclidean topology on the set of complex numbers) is a measurable set of configurations. The considerations concerning the construction of the Lebesgue measure are important for the unitary operations appropriate for the state transitions of quantum cellular automata that we will consider below. Once the Lebesgue measure and Lebesgue measurable functions are in place, the Lebesgue integral of any Lebesgue measurable function over any measurable set of configurations E, denoted as usual by

$$\int_E f \, d\mu$$

can then be defined. We do not need the details [Fan71] of the construction of the Lebesgue integral, but we will need several crucial properties of the integral as we proceed that we will explicitly mention. Intuitively, two functions that agree on all inputs except on a set of measure 0 are measure-theoretically indistinguishable. The next step is to set up an equivalence relation on the measurable functions: f and g are equivalent if, and only if,

$$\int_{\mathcal{C}} |f - g| \, d\mu = 0$$

This equivalence relation is a congruence on the set of Lebesgue integrable functions with respect to addition, multiplication and composition. This means, operationally, that when we add, multiply or compose functions, we are manipulating their equivalence classes without worry. This is important to an isomorphism we will consider below. Also, we now have enough to define the Hilbert space of interest. $L^2(\mathcal{C})$ is the Hilbert space consisting of the equivalence classes defined by the equivalence of

measure-theoretically indistinguishable measurable functions that satisfy

$$\int_{\mathcal{C}} |f|^2 \, d\mu < \infty$$

The inner product $\langle f|g\rangle$ on $L^2(\mathcal{C})$ is defined in the well-known way:

$$\langle f|g\rangle = \int_{\mathcal{C}} f^* g \, d\mu$$

$L^2(\mathcal{C})$ is isomorphic to $L^2([0,1])$. We call the functions whose equivalence classes are in $L^2(\mathcal{C})$ *square summable*. A configuration c is *dyadic* iff $c(i) = 0$ for all but finitely many i, or $c(i) = 1$ for all but finitely many i. The Borel set of dyadic configurations is countable, and so has measure 0. Therefore, the values of a square summable function on the dyadic configurations can be ignored, as well as the dyadic configurations themselves. That is, call two Borel sets of configurations equivalent iff their intersections with the non-dyadic configurations are equal. Re-index the components of the configurations in any way in 1-to-1 correspondence with the nonnegative integers, and map each configuration c to

$$\sum_{k \in \omega} c(k) 2^{-k}$$

which we can denote by $\eta(c)$. Dyadic configurations map to dyadic rationals; i.e., integer multiples of 2^{-k} for some k. Again, the dyadic rationals can be ignored. Given a Borel set of configurations E, and a Lebesgue measurable function f on $\mathcal{C}$,

$$\int_E f \, d\mu = \int_{\eta(E)} \hat{f} \, d\mu'$$

where μ' is the standard Lebesgue measure on the unit interval, and $\hat{f}(\eta(c)) = f(c)$. In particular, $L^2(\mathcal{C})$ is separable and the familiar complete orthonormal sequences in $L^2([0,1])$ transfer. The probability that a configuration is a member of a particular

measurable set E is just the measure of E.

6.3.1 Decompositions in terms of tensor products

Consider the following claim:

$$L^2(\mathcal{C}) = L^2(\mathcal{C} \upharpoonright F \times \mathcal{C} \upharpoonright (\mathbb{Z} - F)) = L^2(\mathcal{C} \upharpoonright F) \otimes L^2(\mathcal{C} \upharpoonright (\mathbb{Z} - F))$$

If, instead of L^2, we had Free, then this claim would be a category-theoretic theorem about tensor construction in the category of complex vector spaces. Does the claim hold for $L^2(\mathcal{C})$? For any two disjoint sets sets S_1 and S_2 of integers,

$$\mathcal{C} \upharpoonright (S_1 \cup S_2) = \mathcal{C} \upharpoonright S_1 \times \mathcal{C} \upharpoonright S_2$$

More generally, does

$$L^2(\mathcal{C} \upharpoonright (S_1 \cup S_2)) = L^2(\mathcal{C} \upharpoonright S_1 \times \mathcal{C} \upharpoonright S_2) = L^2(\mathcal{C} \upharpoonright S_1) \otimes L^2(\mathcal{C} \upharpoonright S_2)$$

hold, as it would if L^2 were replaced with Free? The answer is *yes* and should be explicitly seen. It first needs to be realized that Lebesgue measure, Lebesgue measurable functions and Lebesgue integrals are defined straightforwardly on finite sets equipped with uniform distributions, but there is a slight mathematical surprise if one hasn't previously considered $L^2(\mathcal{C})$ over finite sets. Note that every subset of a finite set is Borel if every singleton is, as is the case with the discrete topology. It follows that we don't need to restrict consideration to configurations over infinite sets.

Example 6.3.1. ***Using the Lebesgue integral defined on a uniformly distributed finite set***

Consider the configuration space $\{0, 1\}$ *representing the classical values 0 and 1 of*

a single bit. Let each of the two elements of the configuration space be assigned measure $\frac{1}{2}$*. The members of* $L^2(\{0,1\})$ *are singleton equivalence classes, so we need only consider functions from* $\{0,1\}$ *to the complex numbers* $\mathbb{C}$*. Consider the function* x *where* $x(0) = \sqrt{2}$ *and* $x(1) = 0$*. The norm of* x *is* $(\sqrt{2})^2$ *times the measure of* $\{0\}$ *plus 0 times the measure of* $\{1\}$*. The norm is therefore 1. Thus,* $|0\rangle = x$ *and similar considerations apply to* $|1\rangle$*. Generally then, the norm of* $a|0\rangle + b|1\rangle$ *is* $|a|^2 + |b|^2$*.*

6.3.2 Factoring L^2 spaces

A Hamel basis of a vector space is a linearly independent set of vectors that spans, with *finite* linear combinations, the entire space. It follows using Zorn's Lemma that every vector space has a Hamel basis.[Ger85] Choose Hamel bases for $L^2(\mathcal{C} \upharpoonright S_i)$, $i = 1, 2$. Since these spaces are separable, it follows that there are complete orthonormal sequences of vectors in each. Choose such a sequence Q_i, $i = 1, 2$, in each. It also follows by the construction of the tensor product space that the set of all pairwise tensor products of u_1 and u_2 such that u_i is in the chosen Hamel basis of $L^2(\mathcal{C} \upharpoonright S_i)$, $i = 1, 2$, is a Hamel basis for $L^2(\mathcal{C} \upharpoonright S_1) \otimes L^2(\mathcal{C} \upharpoonright S_2)$. [Ger85]. No special properties of the L^2 construction are involved. It follows from the definition of the inner product on the tensor product space that the countable collection R of tensor products of vectors $|p\rangle$ in Q_1 and $|q\rangle$ in Q_2 are pairwise orthogonal. For completeness it suffices to note that every element of the chosen Hamel basis of the tensor product space is given by a convergent (generally infinite) linear combination of vectors in R, and this follows from the identity

$$(\sum_i a_i|p_i\rangle) \otimes (\sum_j b_j|q_j\rangle) = \sum_{i,j} a_i b_j (|p_i\rangle \otimes |q_j\rangle)$$

for all square summable sequences $\{ a_i \}_i$, $\{ b_j \}_j$. The identity follows from the completeness of Q_1 and Q_2 , and the definition of inner product and its continuity on the tensor product space. (The identity of course holds for finite linear combinations; there is only something to prove in the case of countably infinite linear combinations.)

A problem with naïve infinitary tensor products

It is tempting to define and establish

$$L^2(\mathcal{C}) \cong (L^2(\{ 0,1 \}))^{\otimes *}$$

where the right-hand side is a countably infinite tensor product. The problem is that sequences such as

$$\{ |0\rangle^{\otimes n} \,|\, n \in \omega \}$$

don't converge. $|0\rangle^{\otimes n}$ is the function that maps the zero state of the classical n bit register to the square root of 2^n and all other states of the register to 0. The sequence of functions does not converge.

Cellular unitary operators

We consider unitary operators on $L^2(\mathcal{C})$. Partition the integers into intervals

$$I(n) = \{ nk, \ldots, nk + k - 1 \}$$

of length k. Now choose a fixed n and consider all standard Cantor topology basis sets $B_{c,I(n)}, \forall c \in \mathcal{C}$. To each $B_{c,I(n)}$ there corresponds a subspace of $L^2(\mathcal{C})$ given by $\chi_{B_{c,I(n)}} L^2(\mathcal{C})$, the subspace of pointwise products of the elements of $L^2(\mathcal{C})$ multiplied by the characteristic function of $B_{c,I(n)}$. These subspaces are all isomorphic to each

other. Moreover there are exactly 2^k of them for each fixed n and $L^2(\mathcal{C})$ splits into their orthogonal direct sum. Consider a unitary operator U on the finite dimensional space $L^2(\{0,\ldots,2^k-1\})$. Factor $L^2(\mathcal{C})$ as

$$L^2(\mathcal{C} \upharpoonright \{0,\ldots,2^k-1\}) \otimes L^2(\mathcal{C} \upharpoonright (\mathbb{Z} - \{0,\ldots,2^k-1\}))$$

Let operator U' be the Kronecker product of U with the identity operator on the right-hand factor. Now replace n by n' and repeat the construction obtaining, say, operator V'. The direct sum decomposition corresponding to n' is isomorphic to the direct sum decomposition corresponding to n and, up to isomorphism between these two direct sum decompositions, U' and V' are the same. We seek unitary operators on $L^2(\mathcal{C})$ that are same up to isomorphism on all such orthogonal direct sum decompositions, for all n. Specifically, for radius 1 QCA with $k = 3$.

Let U_1 be a unitary operation on $L^2(0,\ldots,7)$. Consider U_1' as above. We have the isomorphism

$$L^2(\mathcal{C}) \cong L^2(\mathcal{C} \upharpoonright (\mathbb{Z} - \{0,\ldots,7\}))$$

and so there is an operation U_2' on $L^2(\mathcal{C} \upharpoonright (\mathbb{Z} - \{0,\ldots,7\}))$ isomorphic to U_1'. Let U_2' be the Kronecker product of U_1 and U_1'. Choose an integer n and repeat the construction for $L^2(\mathcal{C} \upharpoonright (\mathbb{Z} - \{3n,\ldots,3n+6\}))$ to obtain U_3', etc. A cellular unitary operator is a unitary operator $W = \lim_{n\to\infty} U_n'$. If we let $k = 1$ and begin with the Hadamard operator on $L^2(\{0,1\})$, then the limit is not defined, for example, but the limit is of course defined and unitary if we begin with the identity operator or any permutation of the computational basis of $L^2(\{0,1\})$. Similar considerations apply when $k = 3$.

6.4 Specification Logic

In classical computation, there are formal verification methods based on *specification logic.*[Sho00] Formulas in specification logic take the form

$$\{\Phi\}C\{\Psi\}$$

where $\{\Phi\}$ and $\{\Psi\}$ are formulas expressed in some underlying logic (usually first-order predicate calculus) and C is a command, i.e., a program or fragment of a program. Formal semantics are given by the notion that the state map defined by C maps a computation state about which $\{\,\Phi\,\}$ is true to a state about which $\{\,\Psi\,\}$ is true; i.e.,

$$[\![C]\!]\{\,\Phi\,\} \subseteq \{\,\Psi\,\}$$

where $[\![C]\!]$ is the usual semantic map on computation states defined by the command C. The analogous specification formulas for quantum computation can be taken to be

$$\{\mathrm{Tr}(\rho P^A_\Phi) \geq p\}U\{\mathrm{Tr}(\rho P^A_\Psi) \geq q\}$$

where A is an observable, $P^A_\Delta = \chi_\Delta(A)$, χ_Δ is the characteristic function of effective Borel set Δ, and U is a unitary evolution operator.

6.5 Effective Borel Sets

Let $P(\alpha, x_1, \ldots, x_n)$ be a computable $(1, n)$-ary relation, where the x_i are nonnegative integers and α is a function mapping nonnegative integers to nonnegative integers. *Computable* means that, with an oracle for computing α, we need only use the oracle to compute some finite sequence $\alpha(0), \ldots, \alpha(k)$ in order to effectively

decide whether or not $P(\alpha, x_1, \ldots, x_n)$ is true. Consider the formula

$$\forall x_1 \exists x_2 \cdots \mathrm{Q} x_n P(\alpha, x_1, \ldots, x_n)$$

where Q is either $\forall$ or $\exists$, as appropriate, in the alternating quantifier prefix in the above expression. In the conventional notation widely used in mathematical logic [Sho00], the sets of all α satisfying formulas such as the above are the (1,0)-ary arithmetic relations when interpreted on the natural numbers or any other effectively presented countably infinite set such as the integers. These sets are all Borel and the set of their characteristic functions are the Borel sets of configurations we have been discussing. In a set-theoretic sense these constitute a countable subcollection of the uncountably many distinct Borel sets of configurations we have been implicitly considering, but they include all Borel sets of configurations that can be explicitly defined via the apparatus of formal number theory.

Chapter 7

Conclusion and Future Prospects

> *Semantics is a strange kind of applied mathematics; it seeks profound definitions rather than difficult theorems.*
>
> - JOHN C. REYNOLDS (1980)

In the foregoing, we have established the generality of the differential scheme and the flexibility of the differential calculi at its heart, in particular. A "profound definition" was certainly being sought, and has hopefully been achieved, in the formulation of the differential scheme.

7.1 What We Have Done

We have completed the formalization of the differential scheme as envisioned in [Bla00], presenting novel uses for convergence spaces along the way and detailing the properties of convergence spaces that lend themselves to their new application to the differential scheme.

From the formal definition, we have provided specific and meaningful examples of differential calculi, including the continuous differential calculus of classical analysis

as well as differential calculi on discrete structures (reflexive digraphs).

In keeping with the motivation behind the differential scheme, we have provided specific examples of dynamical systems as instances of the differential scheme, including classical (Turing) computation, via classical cellular automata.

Lastly, we have shown how to extend Watrous's quantum cellular automata model, overcoming its restriction to configurations with finite support, via probability measures on appropriate L^2 spaces.

7.2 Future Work

Further development of our line of investigation could proceed along two general paths. One path would be to enlarge the stock of compelling examples that fit into the differential scheme. Along this path, if a particularly compelling example turns out to not fit into the scheme, the opportunity to *conservatively* extend the current definition beckons.[1] Another path is to seek theorems about our generalized dynamical systems that would apply to the broad set of continuous and/or discrete systems already accounted for in the theory (by rough analogy, like the category theoretic solution of recursive domain equations in programming language semantics).

7.2.1 Further Examples

Significant prospects include quantum computation and higher-order computation.

[1] Our existing machinery is tuned to mesh neatly with, e.g., the standard differentials of analysis; extensions of this machinery should not expand the set of differentiable functions in these cases.

Quantum Computation

We have shown that classical computation fits neatly into the differential scheme in §4.1.1. The case of quantum computation is currently unresolved. It may be that the differential scheme, as defined herein, already subsumes quantum computation. The main issue, as is typical with quantum computation, is establishing the unitarity of the differentials falling out of any natural differential calculi. It is not clear that this can be done under the current differential scheme.

Higher Order Computation

Many applications of filter spaces, including the convergence space-equivalent filter spaces of Hyland [Hyl79], were made in the context of research on higher-order computation. This is a natural area in which to apply our generalized dynamical systems approach.

7.2.2 Further Theorems

Besides classical Turing machines (TMs), the differential scheme accommodates weaker classes of abstract automata, for example, deterministic finite automata (DFAs). The evolution of the global state of different classes of automata are determined by differentials of different types. There should be differential equations whose solutions provide the appropriate state evolution. What might these look like? A TM would require a partial differential equation, as both time and the cell space change. A DFA would only require an ordinary differential equation. What else may be said remains to be investigated.

Bibliography

[AHS90] J. Adámek, H. Herrlich, and G. E. Strecker. *Abstract and Concrete Categories*. Wiley Interscience, 1990.

[AK03] J. Al-Khalili. *Quantum: A Guide for the Perplexed*. Weidenfeld & Nicolson, 2003.

[Ale37] P. S. Alexandrov. Discrete Räume. *Math. Sb. (New Series)*, 2 (44)(3):501–519, 1937.

[All84] J. Allen. Towards a general theory of action and time. *Art. Intell.*, 23:123–154, 1984.

[AM75] M. A. Arbib and E. Manes. *Arrows, Structures, and Functors: The categorical imperative*. Academic Press, 1975.

[Are46] R. F. Arens. A topology for spaces of transformations. *Annals of Mathematics*, 47:480–495, 1946.

[AS68] W. I. Averbukh and O. G. Smolyanov. The various definitions of the derivative in linear topological spaces. *Russian Math. Surveys*, 23(4):67–113, 1968.

[Axl97] S. Axler. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer, 2nd edition, 1997.

[Bar93] M. Barnsley. *Fractals Everywhere*. Academic Press, 2nd edition, 1993.

[BB84] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, pages 175–179, Bangalore, India, December 1984. IEEE Press.

[Ben73] C.H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[Ben80] P.A. Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22(5):531–591, 1980.

[Ben82] P.A. Benioff. Quantum mechanical hamiltonian models of discrete processes that erase their own histories: application to turing machines. *International Journal of Theoretical Physics*, 21:177–202, 1982.

[Ben89] C.H. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal of Computing*, 18:766–776, 1989.

[BHL91] H. L. Bentley, H. Herrlich, and R. Lowen. Improving constructions in topology. In H. Herrlich and H.-E. Porst, editors, *Category Theory at Work*, pages 3–20. Heldermann Verlag, 1991.

[Bin66] E. Binz. Ein Differenzierbarkeitsbegriff limitieren Vektorraäume. *Comment. Math. Helv.*, 41:137–156, 1966.

[Bin75] E. Binz. *Continuous Convergence on C(X).* Number 469 in Lecture Notes in Mathematics. Springer-Verlag, 1975.

[BJ] H. A. Blair and D. W. Jakel. Reflexive digraphs as pretopological spaces and as other convergence spaces. in preparation.

[BJIR07] H.A. Blair, D.W. Jakel, R.J. Irwin, and A. Rivera. Elementary differential calculus on discrete and hybrid structures. In Sergei N. Artëmov and

Anil Nerode, editors, *LFCS*, volume 4514 of *Lecture Notes in Computer Science*, pages 41–53. Springer, 2007.

[BK66] E. Binz and E. Keller. Functionenräume in der Kategorie der Limesräume. *Ann. Acad. Sci. Fenn. (Ser. A. I.)*, pages 1–21, 1966.

[Bla00] H. A. Blair. The differential scheme for models of computation. *Electr. Notes Theor. Comput. Sci.*, 40, 2000.

[Boh13] N. Bohr. On the constitution of atoms and molecules. *Philosophical Magazine*, 26:1–25, 476–502, 857–75, 1913.

[Boh51] D. Bohm. *Quantum Theory*. Prentice-Hall, 1951.

[Boh76] N. Bohr. The correspondence principle. In J.R. Nielsen, editor, *Niels Bohr Collected Works*, volume 3. North-Holland, 1976.

[Boh89] D. Bohm. *Quantum Theory*. Dover, 1989. republication of [Boh51].

[Bor76] G. Bordaud. Some cartesian closed categories of convergence spaces. In *Convergence Spaces: Proc. Conf. Mannheim 1975*, number 540 in Lecture Notes in Mathematics, pages 93–108. Springer-Verlag, 1976.

[Bou49] N. Bourbaki. *Topologie Générale*. Actualités Sci. Ind., **858** (1940), **916** (1942), **1029** (1947), **1045** (1948), **1084** (1949).

[Car37] H. Cartan. Théorie des filtres. *C. R. Acad. Paris*, 205:595–598, 1937.

[Čec66] E. Čech. *Topological Spaces*. Interscience, revised edition, 1966.

[Cho47] C. Choquet. Convergences. *Ann. Univ. Grenoble*, 23:55–112, 1947.

[CR] D. Cenzer and J. Remmel. Effectively closed sets. to appear in ASL Lecture Notes in Logic; draft available at http://www.math.ufl.edu/~cenzer/research_html/list.html.

[Dav58] M. Davis. *Computability and Unsolvability.* McGraw-Hill, 1958.

[Dav82] M. Davis. *Computability and Unsolvability.* Dover, 1982. Republication of [Dav58].

[Dav90] E. Davis. *Representations of Commonsense Knowledge.* Morgan Kaufman, 1990.

[Deu85] D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London A*, 400:97–117, 1985.

[Fan71] G. Fano. *Mathematical Methods for Quantum Mechanics.* McGraw-Hill, 1971.

[FB66] A. Frölicher and W. Bucher. *Calculus in Vector Spaces without Norm.* Number 30 in Lecture Notes in Mathematics. Springer-Verlag, 1966.

[Fey82] R.P. Feynmann. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[Fox45] R. H. Fox. On topologies for function spaces. *Bull. Amer. Math. Soc.*, 51:429–432, 1945.

[Gal03] A. P. Galton. A generalized topological view of motion in discrete space. *Theoretical Computer Science*, 305:111–134, 2003.

[GD71] A. Grothendieck and J. A. Dieudonné. *Eléments de Géométrie Algébrique I.* Springer-Verlag, 1971.

[Ger85] R. Geroch. *Mathematical Physics.* University of Chicago, 1985.

[GHK+80] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices.* Springer-Verlag, 1980.

[Gru99] J. Gruska. *Quantum Computing*. McGraw-Hill, 1999.

[Hec03] R. Heckmann. A non-topological view of dcpo's as convergence spaces. *Theoretical Computer Science*, 305:159–186, 2003.

[Hel79] P. Hell. An introduction to the category of graphs. In F. Harary, editor, *Topics in Graph Theory*, pages 120–136. New York Academy of Sciences, 1979. (Annals of the New York Academy of Sciences **328** (June 20,1979)).

[Her68] H. Herrlich. *Topologische Reflexionen und Coreflexionen*. Number 78 in Lecture Notes in Mathematics. Springer-Verlag, 1968.

[Hir04] M. Hirvensalo. *Quantum Computing*. Natural Computing. Spring-Verlag, 2nd edition, 2004.

[HLCS91] H. Herrlich, E. Lowen-Colebunders, and F. Schwarz. Improving **Top**: **PrTop** and **PsTop**. In H. Herrlich and H.-E. Porst, editors, *Category Theory at Work*, pages 21–34. Heldermann Verlag, 1991.

[HS74] M. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Number 60 in Pure and Applied Mathematics. Academic Press, 1974.

[HW67] F. Harary and G. W. Wilcox. Boolean operations on graphs. *Math. Scand.*, pages 41–51, 1967.

[HW91] J. Hubbard and B. West. *Differential Equations, A Dynamical Systems Approach Part I*. Number 5 in Texts in Applied Mathematics. Springer-Verlag, 1991.

[HW95] J. Hubbard and B. West. *Differential Equations, A Dynamical Systems Approach: Higher-dimensional Systems*. Number 18 in Texts in Applied Mathematics. Springer-Verlag, 1995.

[Hyl79] J. M. E. Hyland. Filter spaces and continuous functionals. *Annals of Mathematical Logic*, 16:101–143, 1979.

[IK00] W. Imrich and S. Klavžar. *Product Graphs*. John Wiley & Sons, 2000.

[Joh82] P. T. Johnstone. *Stone Spaces*. Cambridge University Press, 1982.

[Kat65] M. Katětov. On continuity structures and spaces of mappings. *Comm. Math. Univ. Carol.*, 6(2):257–279, 1965.

[Kel55] J. L. Kelley. *General Topology*. Van Nostrand Reinhold, 1955.

[Kel74] E. Keller. *Differential Calculus in Locally Convex Spaces*. Number 417 in Lecture Notes in Mathematics. Springer-Verlag, 1974.

[Ken64] D. C. Kent. Convergence functions and their related topologies. *Fundamenta Mathematicae*, 54:125–133, 1964.

[KKM90] E. Khalimsky, R. Kopperman, and P. R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology Appl.*, 36:1–17, 1990.

[Kov89] V. A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision Graphics Image Processing*, 46:141–161, 1989.

[KR89] T. Y. Kong and A. Rosenfeld. Digital topology: introduction and survey. *Computer Vision Graphics Image Processing*, 48:357–393, 1989.

[Kre89] E. Kreyszig. *Introductory Functional Analysis with Applications*. Wiley, 1989.

[Kri83] A. Kriegl. Eine kartesische abgeschlossene Kategorie glatter Abbildungen zwischen beleibigen lokalkonvexen Vektoräumen. *Monatshefte für Mathematik*, 95:287–309, 1983.

[KSV02] A.Yu. Kitaev, A.H. Shen, and M.N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

[Lan61] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5:183–191, 1961.

[Lan04] S. Lang. *Linear Algebra*. Springer, 3rd edition, 2004.

[Mac71] S. MacLane. *Categories for the Working Mathematician*. Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1971.

[Man80] Yu. Manin. *Computable and Uncomputable*. Sovetskoye Radio, 1980. (In Russian).

[Man99] Yu. Manin. Classical computing, quantum computing, and shor's factoring algorithm, 1999. arXive e-print quant-ph/9903008.

[Mar63] G. Marinescu. *Espaces Vectoriels Pseudo Topologique et le Théorie de Distributions*. Deutsche Verlag d. Wiss., 1963.

[Mes99] A. Messiah. *Quantum Mechanics*. Dover, 1999.

[MH89] K. Morita and M. Harao. Computation universality of one-dimensional reversible (injective) cellular automata. *Trans. of the IEICE*, E 72(6):758–762, 1989.

[Mic38] A. D. Michal. Differential calculus in linear topological spaces. *Proc. Nat. Acad. Sci.*, 24(8):340–342, 1938.

[NC00] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Per95] A. Peres. *Quantum Mechanics: Concepts and Methods*. Kluwer, 1995.

[Pit99] A.O. Pittenger. *An Introduction to Quantum Computing Algorithms*. Progress in Computer Science and Applied Logic. Birkhäuser, 1999.

[RCC92] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In B. Nebel, C. Rich, and W. Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings 3rd International Conference (KR '92)*, pages 165–176. Morgan Kaufman, 1992.

[Rib83] P. Ribenboim. Algebraic structures on graphs. *Algebra Universalis*, 16(1):105–123, 1983.

[RS03] C. M. Reidys and P. F. Stadler. Combinatorial landscapes. Technical report, Santa Fe Institute, 2003.

[Sch01] L. Schröder. Categories: a free tour. In J. Kozlowski and A. Melton, editors, *Categorical Perspectives*, pages 1–27. Birkhäuser, 2001.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete log and factoring. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 20–22. IEEE Press, December 1994.

[Sho00] J. Shoenfield. *Mathematical Logic*. ASL, 2000. originally published by Addison-Wesley, 1967.

[Shr88] J. Shrimpton. Cartesian closed categories of directed graphs. Master's thesis, University College of North Wales, 1988. U. C. N. W. Maths Preprint **88.5**.

[Smy95] M. B. Smythe. Semi-metrics, closure spaces and digital topology. *Theoretical Computer Science*, 151:257–276, 1995.

[SSWF01] B. M. R. Stadler, P. F. Stadler, G. P. Wagner, and W. Fontana. The topology of the possible: formal spaces underlying patterns of evolutionary change. *Journal of Theoretical Biology*, 213(2):241–274, 2001.

[Sto97] O. Stock, editor. *Spatial and Temporal Reasoning*. Kluwer Academic Publishers, 1997.

[Tof77] T. Toffoli. Computation and construction universality of reversible cellular automata. *J. Comput. Sys. Sci.*, 15:213, 1977.

[TW02] W. Trappe and L.C. Washington. *Introduction to Cryptography with Coding Theory*. Prentice Hall, 2002.

[vN66] J. von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois, 1966. edited and completed by A.W. Burks.

[Wat95] J. Watrous. On one-dimensional quantum cellular automata. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 528–537. IEEE Press, October 1995.

[Wei62] P. M. Weichsel. The Kronecker product of graphs. *Proc. Amer. Math. Soc.*, 13:47–52, 1962.

[Wil70] S. Willard. *General Topology*. Addison-Wesley, 1970.

Curriculum Vitae

Robert J. Irwin

212 Redfield Avenue

Fayetteville, NY 13066

rjirwin@syr.edu

Education

SYRACUSE UNIVERSITY, Syracuse, NY

Ph.D. Candidate in Computer Science

Thesis Title: *The Differential Scheme and Quantum Computation*

Thesis Advisor: Howard A. Blair

M.S. in Computer Science, 1992

ANTIOCH COLLEGE, Yellow Springs, OH

B.S. in Mathematics, 1973

Higher Education Employment

SYRACUSE UNIVERSITY, Syracuse, NY

7/2011 Taught *Cyber Security* teacher training workshop for Project Advance.

1/10 - present ***Teaching Assistant***. Teach *Scripting Languages*, *Mobile App Development* and I*ntroduction to Programming in C* courses.

Summ. '01,'02 ***Adjunct Instructor***. Taught *UNIX and Internet* course.

1/97 - 5/97 ***Adjunct Instructor***. Taught *Introduction to Programming in C* course.

6/96 - 12/96 ***Research Assistant***. Research in Complexity of Higher Order Functions supported by a grant from the National Science Foundation (CCR-9522987; principal investigator: James S. Royer).

8/95 - 5/96 ***Teaching Associate***. Taught undergraduate C and graduate C/C++ courses; co-taught and organized year-long graduate *Logic Seminar*.

6/94 - 8/95 ***Teaching Assistant***. Taught C and C++ courses; developed extensive formal materials for same: lecture notes, lab exercise book, coding standards, etc.

8/92 - 5/93 ***Teaching Assistant***. Assisted instructors of two graduate and three undergraduate CS courses.

1/10 – 5/10 **LEMOYNE COLLEGE**, Syracuse, NY

Adjunct Faculty. Taught *Introduction to Algorithm Analysis and Data Structures*, *Programming in Visual Basic*, and *Object-Oriented Software Design* courses.

8/07 – 7/08 **HAMILTON COLLEGE**, Clinton, NY

Visiting Faculty, Computer Science Department. Taught *Computer Architecture*, *Computer Organization and Assembly Programming*, *Applied Theory* (computability theory), and *Virtual Worlds* (animations and gaming with Alice).

9/02 – 8/07 **SUNY OSWEGO**, Oswego, NY

Assistant Professor, Computer Science Department. Taught *Cryptology*, *Computer Organization*, *Introduction to Computer Programming*, and *Tools for Computing* (computer literacy course), etc.

PACE UNIVERSITY, New York, NY

1/91 – 8/91 ***Adj. Assistant Professor***, Computer Science Department. Taught *Computer Organization I* and graduate *Computer Architecture* courses.

9/83 - 6/85 ***Lecturer***, Computer Science Department. Taught undergraduate (*Assembler Language Programming*, *Computer Programming II*, *Computer Organization I*, *Data Structures & Algorithms I*, *Data Base Design*) and graduate (*Programming Language Implementation*, *Operating Systems*) courses.

Other Employment

1/97 - 7/99 **TEXTWISE, LLC.**, Syracuse, NY

Research Engineer. Development of Natural Language Processing (NLP) systems for venture-capitalized R&D firm; wrote key semantic information extraction software, with generic application potential, for DoD-funded NLP systems CHESS and KNOW-IT (the latter cited in 1998 DoD Tibbetts Award to TextWise); trained/supervised other engineers and served as resident optimization expert; advised company on project management issues and product application strategy; performed make-or-buy analyses; prepared grant proposals and presentations; wrote technical and end user-oriented documentation.

6/85 - 12/90 **APPLIED INTELLIGENCE SYSTEMS, INC.**, New York, NY

Director, Software Engineering. Founding member of start-up AI software and consulting firm; co-developed a proprietary, portable expert system (ES) shell and associated utilities, an intelligent data entry system generator, several custom expert systems (including one of the first life underwriting ES's) and an intra-day futures trading system; chief programmer; supervised/trained engineering staff; prepared/delivered proposals and presentations; documented systems internally and externally; trained clients in use of proprietary software.

7/78 - 7/83 **MERRILL LYNCH**, New York, NY

Project Manager. Supervised development and maintenance of multi-tasking software linking IBM mainframes running incompatible operating systems; supervised systems support of multi-region CICS system and associated message-switching software.

Programmer/Analyst, then ***Senior Prog./Analyst***. Analysis, design and implementation of software systems for stock option transactions; wrote extensive utilities package for DG Eclipse minicomputers, greatly improving on that provided by vendor.

Areas of Interest

Computability & Complexity Theory, Dynamical Systems, Quantum Computation, Cryptology, Virtual Worlds, Artificial Intelligence (Natural Language Processing, Expert Systems), Programming Languages, History of Mathematics and Computer Science

Course and Program Development

LEMOYNE COLLEGE

CSC 272 – *Object-Oriented Software Design*

HAMILTON COLLEGE

CPSCI 320 – *Computer Architecture* (added hands-on digital electronics lab component)

SUNY OSWEGO

CSC 333 – *Privacy, Security and Cryptology* (general education, for non-majors)

CSC/MAT 332 - *Cryptology*

CSC 322 – *Systems Programming* (for new Software Engineering program)

SYRACUSE UNIVERSITY

CIS 700 – *Logic Seminar* (graduate)

CIS 504 - *Programming in C and C++* (graduate)

CIS 400/600 – *Mobile Application Development* (graduate/undergraduate)

CIS 300 – *Scripting Languages* (undergraduate)

CIS 296 - *Programming in C* (undergraduate)

Honors & Awards

SUNY Oswego Faculty Enhancement Program Grant, 2003; for development of Cryptology course

NSF Research Assistantship, June 1996 - December 1996; for research in Complexity of Higher Order Functions; principal investigator: James. S. Royer; grant CCR-9522987

Outstanding Teaching Assistant Award, 1996.

Teaching Associateship, August 1995 - May 1996; higher-level teaching assistantship sponsored by Graduate School/EECS Future Professoriate Project.

"Curriculum Development" Assistantships, Summers of 1994 and 1995; for creating extensive formal course materials for CIS 504 and CIS 296, respectively.

Syracuse University Full Tuition Scholarships, August 1991 - December 1996; to accompany fellowships, assistantships and associateships over this time period.

Syracuse all-University Fellowship, August 1991 - May 1994; University-wide competitive fellowship.

Publications

Robert J. Irwin and Howard A. Blair. Quantum cellular automata without quiescent states. *Proceedings of SPIE Quantum Information and Computation IX Conference*, Orlando, FL, April 28, 2011.

Howard A. Blair, David W. Jakel, Robert J. Irwin, and Angel Rivera. Elementary Differential Calculus on Discrete and Hybrid Structures. In Sergei N. Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, Lecture Notes in Computer Science 4514, 41-53, Springer, 2007.

Robert J. Irwin, James S. Royer, and Bruce M. Kapron. On characterizations of the basic feasible functionals, Part II. (submitted to *Theoretical Computer Science*)

Robert J. Irwin, James S. Royer, and Bruce M. Kapron. On characterizations of the basic feasible functionals, Part I. *Journal of Functional Programming*, 11(1):117-153, January 2001.

Robert J. Irwin, James S. Royer, and Bruce M. Kapron. Separating notions of higher-type polynomial-time. *Proceedings of Second International Workshop on Implicit Computational Complexity*, Santa Barbara, CA, June 29-30, 2000.

W. Paik, E.D. Liddy, E. Allen, E. Brown, A. Farris, R. Irwin, J.H. Liddy, and I. Niles. Applying link analysis to automatically extracted information from texts using KNOW-IT, *Proceedings of the AAAI Symposium on Artificial Intelligence and Link Analysis*, 1998.

Work in Progress

Howard A. Blair, Robert J. Irwin, David W. Jakel, and Angel Rivera. Differential calculus on directed graphs. (*to be submitted*)

Invited Reviews

Review of *Models of Computation: Exploring the Power of Computing* (Prentice Hall, 1998), *SIGACT NEWS*. (to be submitted)

Review of *Derivation and Computation: Taking the Curry-Howard correspondence seriously* (Cambridge, 2000), *SIGACT NEWS*, June 2008

Review of *Coding Theory and Cryptography: The Essentials* (Marcel Dekker, 2000), *SIGACT NEWS*, December 2003

Review of *Set Theory for Computing* (Springer, 2001). *SIGACT NEWS*, September 2003

Review of *The Mind and the Machine: Philosophical Aspects of Artificial Intelligence* (Wiley, 1984). *IEEE Software*, May 1987.

Review of *Effective Design of CODASYL Data Base* (Macmillan, 1985). *IEEE Software*, September 1985.

Presentations

Robert J. Irwin. Invited lectures at Capital Normal University, Beijing, China, Spring 2009:

1. *Essentials of Quantum Cryptography*
2. *Computer Science Education in the United States: graduate and undergraduate*

Robert J. Irwin. *Toward a Model of Classical and Quantum Hybrid Computation*, Dept. of Elec. Engineering and Computer Science Colloquium Lecture, Syracuse University, April 29, 2005.

Robert J. Irwin. *America's Unluckiest Logician*, QUEST 2005, SUNY Oswego, April 20, 2005.

Robert J. Irwin. *Early Computability Theory*, Canadian Mathematical Society Winter Meeting, Montreal, December 12, 2004.

Robert J. Irwin. *Calculus on Discrete and Hybrid Data Structures*, QUEST 2004, SUNY Oswego, April 21, 2004.

H. Blair, R. Irwin, D. Jakel, J. Remmel, A. Rivera. *Toward a Theory of Heterogeneous Computation*, Computer Science Colloquium, University of Massachusetts, Boston, April 30, 2003.

Robert J. Irwin. *Quantum Cryptography*, QUEST 2003, SUNY Oswego, April 23, 2003.

R. Irwin, B. Kapron, J. Royer. *On Characterizations of the Basic Feasible Functionals, Part I.* 14th Annual Mathematical Foundations of Programming Semantics Workshop, May 1998.

Selected Systems Developed

KNOWledge base Information Tool (KNOW-IT), 1998-1999, at TextWise Labs under contract with DARPA; developed the core information extraction component.

Chronological information Extraction SyStem (CHESS), 1997-1998, at TextWise Labs under contract with DARPA; developed the core information extraction com-

ponent (precursor system to KNOW-IT).

Front-End Generator (FEG), 1987-1988, at Applied Intelligence Systems under contract with John Hancock; an early object-oriented RAD system, incorporating a scripting language and expert system shell capabilities; developed the central interpreter and co-developed the object/script compiler and most other components.

Decision Master, 1985-1986, at Applied Intelligence Systems; an expert system shell (not related to FEG) incorporating a variety of rule induction techniques, optimized for high-volume non-interactive applications; co-developed with Joseph S. Rubenfeld.

College/University Service

SUNY OSWEGO

Scholarly and Creative Activities Committee

Provost's Advisory Committee on Academic Quality

Science Planning Committee subcommittee on "Curriculum and New Interdisciplinary Initiatives"

Computer Science Department Ad Hoc ABET Accreditation Committee

CSTEP Mentor (for minority students in technical fields)

Computer Science Department Curriculum Committee

First-Year Advisor

Computer Science Department Laboratory Committee

SYRACUSE UNIVERSITY

Senate Affirmative Action Grievance Committee

EECS Tenure and Promotion Committee

Graduate Student Representative to Dean

Community Service

Meal on Wheels volunteer

Fayetteville-Manlius Astronomical Society (assist public observations)

Fayetteville Elementary Science Club volunteer

Affiliations

Association for Computing Machinery (ACM)

ACM Special Interest Group on Algorithms and Computation Theory (SIGACT)

Mathematical Association of America (MAA)