Duality in Logic and Computation

(Invited Paper)

Prakash Panangaden
School of Computer Science
McGill University
Montréal, Québec
Canada
Email: prakash@cs.mcgill.ca

Abstract

I give a brief introduction to Stone duality and then survey a number of duality theories that arise in logic and computer science. I mention some more unfamiliar dualities at the end which may be of importance to emerging fields within computer science.

1. Introduction

The Stone representation theorem [Sto36] is a recognized landmarks of mathematics. The theorem states that every (abstract) Boolean algebra is isomorphic to a (concrete) Boolean algebra of sets. This is not hard to prove and appears as an exercise in many books: usually decorated with two stars! However, this does not convey the full import of the theorem. Stone duality is an inherently categorical statement: it says that one can transfer ideas *back and forth* across two different mathematical universes.

Stone duality embodies completeness theorems, but goes far beyond them. Proofs of completeness typically work by constructing an instance of a model from maximal consistent sets of formulas. Stone duality works in the same way, but gives a correspondence not just for syntactically generated algebras, but for *any* suitable algebra. This includes both smaller algebras, which could be finite and generate finite structures, or larger algebras, which could be uncountable, thus not syntactically generated. Furthermore, homomorphisms of algebras give rise to maps between the corresponding structures in the opposite direction. Thus mathematical arguments can be transferred in both directions.

The Stone representation theorem [Sto36] began a revolution in the way we understand the relationship between different mathematical structures. It was soon followed by an equally striking theorem, due to Gelfand [Gel39], [GK39], with the modest title "On normed rings." These theorems are the earliest *duality theorems*: they say that one mathematical "universe" is the "mirror image" of another. Roughly speaking this means the following. Given a mathematical structure of type $\mathcal{A}$ (boolean algebras, rings, vector spaces, or whatever) one can construct a structure of type $\mathcal{B}$, where $\mathcal{B}$ may superficially look very different from $\mathcal{A}$. Furthermore, from each structure of type $\mathcal{B}$ one can construct an object of type $\mathcal{A}$. Finally, starting from an object, say A of type $\mathcal{A}$, if one constructs the corresponding structure B and type $\mathcal{B}$ *and them comes back to* $\mathcal{A}$ with the reverse construction, *one obtains essentially the same object as one started with.*

It was not until the emergence of category theory [EL45] that the full import of these duality theorems was felt. One of the messages of category theory is to take seriously the *maps* between mathematical objects. It is here that the phrase "mirror image" becomes important. A duality between two categories $\mathcal{C}$ and $\mathcal{D}$ means the following: given objects A, A' of $\mathcal{C}$ *and a map* $f : A \longrightarrow A'$ there is a way to construct objects —a functor— $F(A), F(A')$ of $\mathcal{D}$ and a map $F(f)$ from $F(A')$ to $F(A)$ and all these can be reversed; in other words there is a similar functor going the other way. The key point is that the direction of the arrow has been reversed: just as one expects in a mirror.

Stone-type dualities are now recognized as being ubiquitous in computer science. Plotkin [Plo83] and Smyth [Smy83] emphasized that the duality between state-transformer semantics and predicate-transformer semantics is an instance of a Stone-type duality. Kozen [Koz85] discovered such a duality for probabilistic transition systems. Abramsky [Abr91] identified dualities in domain and concurrency theory. Recently several authors (e.g. [BHK03], [BK05],

1043-6871/13 $26.00 © 2013 IEEE
DOI 10.1109/LICS.2013.64

[KKV04], [Sil10]) have emphasized the duality between logics and transition systems from a coalgebraic perspective. Mislove et al. [MOPW04] found a duality between labeled Markov processes and C^*-algebras based on the closely related classical Gelfand duality.

2. Duality categorically

If $\mathcal{C}$ and $\mathcal{D}$ are categories a *duality* is a pair of contravariant functors $F : \mathcal{C} \longrightarrow \mathcal{D}$ and $G : \mathcal{D} \longrightarrow \mathcal{C}$

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend right, "F"'] & \mathcal{D}^{op} \arrow[l, bend right, "G"'] \end{tikzcd}$$

such that there are natural transformations $\eta : I_{\mathcal{C}} \longrightarrow G \circ F$ and $\varepsilon : F \circ G \longrightarrow I_{\mathcal{D}}$ satisfying some equations; most importantly that in this case the natural transformations are actually isomorphisms.

This is probably gobbledegook to someone unfamiliar with categorical language. The way to think about it is as follows. For any object $X \in \mathcal{C}$ one can construct a new object $F(X)$ in $\mathcal{D}$ and for any object Y in $\mathcal{D}$ one can construct an object $G(Y)$ in $\mathcal{C}$. If one starts with X and constructs $GF(X)$ the result is isomorphic to X. Similarly, if one starts with Y, $FG(Y)$ is isomorphic to Y. But that is not all. If there is a map $f : X \longrightarrow X'$ in $\mathcal{C}$ one obtains a map $F(X) :\longrightarrow F(X') \longrightarrow F(X)$ (note the reversal in the direction) and similarly for maps in $\mathcal{D}$ under the action of G. Given X, Y as above there is a bijection from the collection of maps from FX to Y in $\mathcal{D}$ (written $\mathcal{D}(FX, Y)$) to the maps $\mathcal{C}(X, GY)$ and this bijection is *natural*. The precise meaning of "natural" need not worry us here: it suffice to think that it means that the bijections do not depend on special features of X or Y but are defined in a uniform way.

3. Stone duality in logic

Boolean algebra appeared originally as an algebraic form of propositional logic thus the original Stone representation theorem was manifestly connected with logic.

Definition 1. A Boolean algebra is a set A equipped with two constants, $0, 1$, a unary operation $(\cdot)'$ and two binary operations $\vee, \wedge$ which obey the following axioms, p, q, r are arbitrary members of A:

$$\begin{array}{rr} 0' = 1 & 1' = 0 \\ p \wedge 0 = 0 & p \vee 1 = 1 \\ p \wedge 1 = p & p \vee 0 = p \\ p \wedge p' = 0 & p \vee p' = 1 \\ p \wedge p = p & p \vee p = p \end{array}$$

$$\begin{array}{rcl} p'' & = & p \\ (p \wedge q)' & = & p' \vee q' \\ (p \vee q)' & = & p' \wedge q' \\ p \wedge q & = & q \wedge p \\ p \vee q & = & q \vee p \\ p \wedge (q \wedge r) & = & (p \wedge q) \wedge r \\ p \vee (q \vee r) & = & (p \vee q) \vee r \\ p \wedge (q \vee r) & = & (p \wedge q) \vee (p \wedge r) \\ p \vee (q \wedge r) & = & (p \vee q) \wedge (p \vee r) \end{array}$$

The operation $\vee$ is called *join*, $\wedge$ is called *meet* and $(\cdot)'$ is called *complement*.

A natural interpretation of these axioms is to take A to be the power set of some set X (written $\mathcal{P}(X)$), to take $\vee$ to be union, $\wedge$ to be intersection, $(\cdot)'$ to be complement, 1 to be X and 0 to be $\emptyset$. Not all Boolean algebras arise in this way. Those that do are special, they are called *complete atomic* Boolean algebras (CABA); I will not define them formally here. The adjective "complete" signifies that arbitrary meets and joins exist not just finite ones. Atomic means that there are "smallest nonzero elements" called atoms and every element can be expressed as a join of atoms.

There is a natural notion of Boolean algebra homomorphism: maps that preserve all the structure. Bijective homomorphisms whose inverse is also a homomorphism are called *isomorphisms*. Boolean algebras and Boolean algebra homomorphisms form a category called $\mathbf{BoolAlg}$.

Examples of Boolean algebras that are not atomic are typically obtained from topological situations. A basic theorem—a warmup to the Stone representation theorem—states that a Boolean algebra is complete and atomic if and only if it is isomorphic to $\mathcal{P}(X)$ for some X.

Every Boolean algebra can be given a natural partial order. We define $p \leq q$ if $p \vee q = q$ or, equivalently, if $p \wedge q = p$. With this order the join operation is the least upper bound and the meet operation is the greatest lower bound. Boolean algebra homomorphisms are monotone with respect to this order.

A very special Boolean algebra is the algebra with just two elements, namely 0 and 1, which have to be there by definition, and nothing else; the definitions of the operations are evident. This Boolean algebra is called $\mathbf{2}$. The Stone representation theorem can be stated as follows.

Theorem 2. Let A be a Boolean algebra. Let X be the set of homomorphisms from A to $\mathbf{2}$. Then the map $f : A \longrightarrow \mathcal{P}(X)$ given by $f(p) = \{h \in X \mid h(p) = 1\}$ is an injective homomorphism.

This theorem does not exhibit the duality. In order to understand the "mirror image" of Boolean algebras it is necessary to introduce topological ideas. I assume familiarity with concepts like open and closed sets, bases and subbases, compactness, Hausdorff spaces, continuity and homeomorphisms. A set is said to be *clopen* if it is both closed and open. The presence of a clopen set makes a space disconnected. A space is said to be *zero-dimensional* if it has a base of clopen sets. A Stone space is a compact, Hausdorff, zero-dimensional space. Stone spaces also form a category. Notice that the clopen sets of a topological space form a Boolean algebra. The topological spaces one typically uses in analysis are always Hausdorff, often compact but never zero-dimensional. Hausdorff zero-dimensional spaces are totally disconnected (the only connected sets are singletons), the converse is not true in general but it is true if the space is locally compact.

Now given a Boolean algebra one can define a Stone space. Start with an algebra A. Consider all the homomorphisms from X to $\mathbf{2}$. For each a of A we define $U_a := \{f \in X \mid f(a) = 1\}$. This is the basis for a topology which makes X into a Stone space with the U_a all being clopen. The clopens viewed as sets form a Boolean algebra. This gives our constructions in the two directions. If we have a Boolean algebra homomorphism $h : A \longrightarrow A'$ we get a "natural" map from X' (the Stone Space constructed from A') to X by the obvious device of composing functions in X' with h:

$$A \xrightarrow{h} A' \xrightarrow{f} \mathbf{2}.$$

This clearly reverses the direction of the map.

Going the other way, let $Cl(X)$ stand for the Boolean algebra of clopens of X. A continuous function $f : X \longrightarrow X'$ yields a Boolean algebra homomorphism $f^{-1} : Cl(X') \longrightarrow Cl(X)$; again the direction is flipped. We thus have two contravariant functors $Cl : \mathbf{BoolAlg} \longrightarrow \mathbf{Stone}$ and $\mathcal{U} : \mathbf{Stone} \longrightarrow \mathbf{BoolAlg}$. Given a Boolean algebra A we have an isomorphic Boolean algebra $Cl(\mathcal{U}(A))$. Given a Stone space X we have the homeomorphic Stone space $\mathcal{U}(Cl(X))$.

Theorem 3. The functors Cl and $\mathcal{U}$ define a dual equivalence of categories.

There is another way to think about the Stone space. Instead of constructing it from homomorphisms we can use concepts closer to logic than to algebra. A *filter* in a Boolean algebra is simply a set that is upward closed and meet-closed. If we think logically, with the order relation of the Boolean algebra signifying deducibility ($a \leq b$ means a proves b) then a filter is deductively closed and closed under conjunctions. A filter is trivial it contains 0: this is like a theory proving $\perp$, it means that the theory is inconsistent and everything is derivable. A maximal filter is one that cannot be expanded without making it trivial: it corresponds to a maximal consistent set. In a Boolean algebra, a maximal filter contains every p or its complement p' but not both. One calls them *ultrafilters*. Now given a homomorphism h from A to $\mathbf{2}$; we can define an ultrafilter $[h] = \{p \in A \mid h(p) = 1\}$. It is easy to see that this is an ultrafilter. It is possible to prove that every ultrafilter can be defined from a homomorphism to $\mathbf{2}$. Thus the points of the Stone space are precisely the ultrafilters: we can write $\mathcal{U}(A)$ for the space of ultrafilters of A.

How does one topologize the Stone space? For each point a of the algebra, define $U_a = \{u \in \mathcal{U}(A) \mid a \in u\}$. These are exactly the same U_a defined above. Thus the maximal consistent sets are the basic open sets of the topology.

This makes the connection between logic and Stone duality easy to explain. Essentially Stone duality subsumes completeness. In a completeness theorem one wishes to show that every consistent set has a model. In Stone duality the Stone space is the space of models and the algebra is the space of formulas, quotiented by logical equivalence of course. The maximal consistent sets are precisely the ultrafilters and the duality theorem essentially says that we can build models out of them. The added bonus of the duality theorem is the statement about the maps between models. Also the correspondence is not just between the syntactically generated algebras but all the algebras.

A big advance in terms of logic was the work of Jonsson and Tarski [JT51]. They considered what happens when one adds modalities to the logic. In modern terms one can describe their results as follows.

Definition 4. A *modal algebra* is a Boolean algebra A with an additional unary operation, written $\Box : A \longrightarrow A$ such that $\Box 1 = 1$ and $\Box(p \wedge q) = \Box p \wedge \Box q$. There is a derived operator $\Diamond$ which is the de Morgan dual of $\Box$. We have $\Diamond 0 = 0$ and $\Diamond(p \vee q) = \Diamond p \vee \Diamond q$.

One can also enrich Stone spaces with additional structures. The semantics of modal logics is given by *frames*: essentially a set M, a boolean algebra V of subsets of M and a binary relation $\longrightarrow$ on M. From the modal operator $\Box$ in the modal algebra one can define a binary relation $\longrightarrow$ on the Stone space by

$$x \longrightarrow y \Leftrightarrow \forall a \in A(\Box a \in x \longrightarrow a \in y).$$

Given a frame defined on a Stone space one can construct a modal algebra as follows. The passage from Stone spaces to Boolean algebras is as before. From the relation $\longrightarrow$ one can define $\Box$ by

$$\Box a\{x \mid \forall y(x \longrightarrow y) \Rightarrow y \in a\}.$$

From a modal algebra one can construct a frame and from a frame one can construct a modal algebra. Modulo some details about pinning down exactly the right class of frames, one gets a duality theorem very much like the Stone duality theorem.

The basic message is that one can get Stone type duality theorems for modal logics by enriching the algebra appropriately. I do not want to suggest that these are trivial extensions; a lot of hard mathematics may be required to prove new duality theorems.

4. Stone duality in the semantics of programming languages

Plotkin, Smyth, Kozen, algebra and coalgebra

In programming language semantics Stone-type dualities were discovered by Plotkin [Plo79], generalized by Smyth [Smy83] and extended to probabilistic programs by Kozen [Koz81], [Koz85]. In this case the duality is between *weakest precondition semantics* due to Dijkstra [Dij76] and state-transformer semantics [Plo81].

State-transformer semantics is the basic way one understands programs. There is a notion of state describing the values of variables and other aspects of the program state and the execution of commands changes the state. In predicate-transformer semantics one specifies properties of the state (predicates) which may be thought of as subsets of the collection of possible states. Given a statement S and a predicate P (property or subset of states) the *weakest precondition* $wp(S; P)$ is another predicate such that if a state σ satisfies $wp(S; P)$ then *after* execution of S the new state will satisfy P. Furthermore, if *any* predicate Q satisfies the same condition then $Q \Rightarrow wp(S; P)$. Notice that weakest-precondition semantics is a *backward* flow of information through the program.

The connection can be written

$$\frac{(S, \sigma) \longrightarrow \sigma'}{\sigma \models wp(S; P) \Leftarrow \sigma' \models P.}$$

Plotkin [Plo79] originally presented this as an isomorphism between domains which could be lifted to an isomorphism between the semantics of a particular programming language. Later, in his Pisa notes [Plo83] he emphasized the categorical aspect and exhibited as a Stone-type duality. Smyth [Smy83] gave a more general topological presentation.

Kozen carried out the same program for probabilistic programs. He made the following striking analogies:

Classical logic	Generalization
Truth values $\{0, 1\}$	Interval $[0, 1]$
Propositional function	Measurable function
State	Measure
The satisfaction relation $\models$	Integration $\int$

Just as the satisfaction relation, $\models$, links states and formulas to give truth values so the integral links measures (generalized states) with measurable functions (generalized formulas) to give real numbers (generalized truth values). He essentially proved a Stone-type duality between a category of probabilistic state transformers and probabilistic analogues of predicate transformers. These are essentially transformers of expectation values of random variables.

The use of expectation-value transformers in programming language semantics was independently invented by the Oxford group [MM04], [MMS96].

The most general form of these Stone-type dualities is in terms of algebras and co-algebras. This is not the place for a review of this topic; fortunately there are excellent reviews available [Rut00], [Ven06] with textbooks forthcoming. In this view algebras for for transition systems appear as algebras and behaviour is described by a coalgebra. The duality between logics and transition systems is then an example of algebra-coalgebra duality [BK05], [BHK03].

5. Stone duality in control theory and automata theory

Before computer scientists looked at duality Kalman [Kal59], [KFA69] had observed the role of duality in linear systems theory. This duality rests on the self duality of finite-dimensional vector spaces. This is the first duality that everyone learns as an undergraduate though it is not always emphasized as a duality.

Given a finite-dimensional vector space V, over some field k the vector space of linear maps from V

to k is called the *dual space*, written V^*: it is easy to show that V^* has the same dimension as V and is hence isomorphic to V but this isomorphism depends on a choice of basis for V. The double-dual V^{**} is isomorphic to V: this time there is a canonical basis independent isomorphism. Let $v \in V$ and $\sigma \in V^*$. Then $\iota : V \longrightarrow V^{**}$ defined by $\iota v(\sigma) = \sigma(v)$ is an isomorphism that does not require one to specify a basis. To write down the inverse explicitly one needs a basis but one can show that it does not depend on the choice of basis.

The dual construction extends to linear maps. Let $\lambda : U \longrightarrow V$ be a linear map between finite-dimensional vector spaces. Then one has a map $\lambda^* : V^* \longrightarrow U^*$. Let $\sigma \in V^*$ and $u \in U$ then $\lambda^*(\sigma)$ is an element of U^*: it should produce an element of k when applied to a vector in U. We define $\lambda^*(\sigma)(u) = \sigma(\lambda(u))$. Note the reversal of direction. Thus $(\cdot)^*$ is a contravariant functor from the category of finite-dimensional vector spaces to itself. It can be shown that it defines a duality: it is, in fact, the prototypical example of a self duality.

In control theory (now usually called systems theory) one (usually) looks at systems defined on linear spaces and governed by linear differential equations. The new features of systems theory over the usual study of differential equations are *partial observability* and *control*. It is assumed that one has some information about the current state but one cannot, in general, determine the state unambiguously. Furthermore, in response to the partial information about the state, one can apply chosen controls that steer the system towards a desired state. Kalman [Kal59] emphasized the importance of duality here and introduced two very important concepts: reachability and observability. Reachability of a state s from another state s_0 means that one can, by an appropriate choice of controls, steer the system from s_0 to s. Observability means that by an appropriate choice of controls one can steer the system into a known state.

A beautiful categorical presentation of this duality was given by Arbib and Manes [AM75] who also introduced these ideas to automata theory. Recently Bonchi et al. [BBRS12] have shown how these ideas explain a hitherto mysterious minimization algorithm due to Brzozowski [Brz62]. Independently, Hundt et al. [HPPP06], [DHP$^+$13] noted that a kind of duality could be used to minimize automata including probabilistic automata; they also emphasized applications to machine learning. The proper categorical treatment of this work emerged later [BKP12] where it was realized that a number of different minimization constructions could be treated under the same rubric.

6. Gelfand, Pontryagin and Tannaka

In this section I very briefly mention some other dualities, formal definitions and details are not given here. I recommend books by Arveson [Arv76], [Arv10], Dixmier [Dix77], Murphy [Mur90] and Rudin [Rud62].

An exercise in many algebra books is the following. Let $\mathcal{C}(X)$ be the ring of continuous functions from X, a compact Hausdorff space, to the reals $\mathbb{R}$. Suppose that M is a maximal ideal of $\mathcal{C}(X)$, show that there is a unique point x of X such that $M = \{f \in \mathcal{C}(X) \mid f(x) = 0\}$. It is easy to see that a set of the form $\{f \in \mathcal{C}(X) \mid f(x) = 0\}$ are maximal ideals. What this exercise says is that the maximal ideals precisely correspond to the points. One can recover the points if one is given just the ring of continuous functions. In fact, one can recover the topology by taking a base of the open sets to be the set of maximal ideals containing a given function. This is the beginning of the celebrated Gelfand duality theorem [Gel39] which gives a duality between (complex) commutative unital C^*-algebras and compact Hausdorff spaces. Stone [Sto40] proved a similar theorem for real C^*-algebras; it is this version which is explained in Johnstone's book [Joh82]. It is quite striking that the proofs in the real and complex case are very different but they end up saying that the categories of real and complex commutative unital C^*-algebras are equivalent because each one is dual to the category of compact Hausdorff spaces.

Pontryagin duality [Pon66] is a duality that "everyone" knows in some fashion. Consider a finite abelian group G. Consider the group $\mathbb{C}^\times$ of complex numbers with 0 removed under multiplication. A homomorphism from G to $\mathbb{C}^\times$ is called a *character*. Often one restricts to the subgroup of complex numbers of unit magnitude (the circle), written $U(1)$, and says that a character is a homomorphism from G to $U(1)$. Now under pointwise multiplication the characters form an abelian group called the *dual group*, $\hat{G}$. One can similarly construct the characters of $\hat{G}$ and obtain the double dual $\hat{\hat{G}}$. Remarkably, G and $\hat{\hat{G}}$ are isomorphic. This can be done far more generally. If one has a locally compact abelian group the dual group is also locally compact abelian and the double dual is isomorphic to the original group. This is the original Pontryagin duality theorem.

Why did I say that everyone knows this? The group $\mathbf{Z}$, the integers, is dual to the circle $U(1)$. Functions defined on the circle are the same as periodic functions defined on the real line. Characters of $\mathbf{Z}$ are of the form $e^{in\theta}$ and the duality tells us that functions on the circle can be expanded in terms of the characters of **Z**: in short, periodic functions can be expressed as a sum of complex exponentials (or sines and cosines) this is Fourier series!

What happens if the groups are not abelian? Then one cannot work with just the characters. One needs to work with higher dimensional representations of the group. This is the fundamental reconstruction theorem of Tannaka [Tan39], [HR69] which has spawned many recent new extensions in areas like quantum groups which may be relavent for quantum computation [JS91]. There are other extensions of Pontryagin duality of interest, for example Hofmann, Mislove and Stralka have developed a Pontryagin duality for certain kinds of semilattices; this theory may have some relevance to quantum information theory.

7. Conclusion

I have tried to mention a wide spectrum of duality theories and show that they are connected to fundamental questions in logic and computation. Almost twenty years ago Vaughan Pratt [Pra95] gave a talk at LICS on the Stone gamut[1] in which he presented the subject from the viewpoint of Chu spaces. I have not said much about Chu spaces here because Pratt has said it so much better than I could: he gives it a beautiful relational or matricial presentation where one can think of Stone duality as just a kind of glorified transpose.

I hope that I have inspired some of the readers at least to learn more about this beautiful and important subject. There are all kinds of new areas where computational thinking is entering. Computer scientists are no longer just working with traditional software and hardware; they are engaged with systems theorists, physicists, engineers and biologists. Duality theories that were considered outside the realm of computer science may well be common place in a few years.

Acknowledgment

I have benefitted from discussions over the years with the following about Stone-type dualities: Samson Abramsky, Michael Barr, Nick Bezhanishvili, Filippo Bonchi, Marcello Bonsangue, Marta Bunge, Brendan Cordy, Vincent Danos, Josée Desharnais, Tobias Fritz, Helle Hvid Hansen, Chris Hundt, Bart Jacobs, Clemens Kupke, Alexander Kurz, François Laviolette, Michael Makkai, Radu Mardare, Vincenzo Marra, Michael Mislove, Drew Moshier, Dusko Pavlovic, Daniela Petrisan, Gordon Plotkin, Vaughan Pratt, Doina Precup, Jan Rutten, Alexandra Silva and James Worrell. I owe a special debt to Dexter Kozen who taught me Stone duality and whose influence on my work has been crucial. I have been supported by research grants from the Natural Sciences and Engineering Research Council (Canada), the Office of Naval Research (USA) and the Engineering and Physical Sciences research Council (UK).

1. He could not say "Stone spectrum" because that phrase was taken.

References

[Abr91] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[AM75] M.A. Arbib and E.G. Manes. Adjoint machines, state behavior machines and duality. *J. Pure Appl. Algebra*, 6:313–343, 1975.

[Arv76] William Arveson. *An Invitation to C^*-Algebra*. Springer-Verlag, 1976.

[Arv10] William Arveson. *A Short Course on Spectral Theory*. Springer-Verlag, 2010.

[BBRS12] Filippo Bonchi, Marcello Bonsangue, Jan Rutten, and Alexandra Silva. Brzozowski's algorithm (co)algebraically. In Robert Constable and Alexandra Silva, editors, *Logics and Program Semantics: Essays Dedicated to Dexter Kozen*, volume 7230 of *Lecture Notes In Computer Science*, pages 12–23. Springer-Verlag, 2012.

[BHK03] Michel Bidoit, Rolf Hennicker, and Alexander Kurz. Observational logic, constructor-based logic, and their duality. *Theor. Comput. Sci.*, 3(298):471–510, 2003.

[BK05] M. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *FoSSaCS*, pages 455–469, 2005.

[BKP12] Nick Bezhanishvili, Clemens Kupke, and Prakash Panangaden. Minimization via duality. In *Logic, Language, Information and Computation - 19th International Workshop, WoLLIC 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings*, volume 7456 of *Lecture Notes in Computer Science*, pages 191–205. Springer, 2012.

[Brz62] J. A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In J. Fox, editor, *Proceedings of the Symposium on Mathematical Theory of Automata*, number 12 in MRI Symposia Series, pages 529–561. Polytechnic Press of the Polytechnic Institute of Brooklyn, April 1962. Book appeared in 1963.

[DHP+13] Monica Dinculescu, Christopher Hundt, Prakash Panangaden, Joelle Pineau, and Doina Precup. The duality of state and observation in probabilistic transition systems. In *Logic, Language, and Computation - 9th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2011, Kutaisi, Georgia, September 26-30, 2011, Revised Selected Papers*, volume 7758 of *Lecture Notes in Computer Science*, pages 206–230. Springer, 2013.

[Dij76] E. W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[Dix77] Jacques Dixmier. *C^*-Algebras*. North-Holland, 1977. Translation of *Les C^* algèbres et leurs représentations*, published 1969.

[EL45] Samuel Eilenberg and Saunders Mac Lane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[Gel39] I. M. Gelfand. On normed rings. *Doklady Akad. Nauk USSR*, 23:430–432, 1939.

[GK39] I. M. Gelfand and A. N. Kolmogorov. On rings of continuous functions on topological spaces. *Doklady Akad. Nauk USSR*, 22:11–15, 1939.

[HPPP06] Christopher Hundt, Prakash Panangaden, Joelle Pineau, and Doina Precup. Representing systems with hidden state. In *The Twenty-First National Conference on Artificial Intelligence (AAAI)*, 2006.

[HR69] E Hewitt and KA Ross. The tannaka-krein duality theorems. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 71:61–83, 1969.

[Joh82] Peter Johnstone. *Stone Spaces*, volume 3 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1982.

[JS91] André Joyal and Ross Street. An introduction to Tannaka duality and quantum groups. In *Category theory*, pages 413–492. Springer, 1991.

[JT51] Bjarni Jónsson and Alfred Tarski. Boolean algebras with operators I. *American Journal of Mathematics*, 73:891–939, 1951.

[Kal59] R. Kalman. On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3):110–110, 1959.

[KFA69] R. E. Kalman, P. L. Falb, and M. A. Arbib. *Topics in Mathematical Systems Theory*. McGraw Hill, 1969.

[KKV04] Clemens Kupke, Alexander Kurz, and Yde Venema. Stone coalgebras. *Theoretical Computer Science*, 327(1):109–134, 2004.

[Koz81] D. Kozen. Semantics of probabilistic programs. *Journal of Computer and Systems Sciences*, 22:328–350, 1981.

[Koz85] D. Kozen. A probabilistic PDL. *Journal of Computer and Systems Sciences*, 30(2):162–178, 1985.

[MM04] C. Morgan and A. McIver. *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer-Verlag, 2004. to appear.

[MMS96] Carroll Morgan, Annabelle McIver, and Karen Seidel. Probabilistic predicate transformers. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 18(3):325–353, 1996.

[MOPW04] M. Mislove, J. Ouaknine, D. Pavlovic, and J. Worrell. Duality for labelled Markov processes. In I. Walukiewicz, editor, *Foundations of Software Science and Computation Structures, FOSSACS*, volume 2987 of *Lecture Notes In Computer Science*, pages 393–407, 2004.

[Mur90] Gerard J. Murphy. *C^*-algebras and Operator Theory*. Academic Press, 1990.

[Plo79] Gordon D. Plotkin. Dijkstra's predicate transformers and Smyth's power domains. In *Abstract Software Specifications*, volume 86 of *Lecture Notes in Computer Science*, pages 527–553. Springer, 1979.

[Plo81] G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Aarhus University, 1981.

[Plo83] G. D. Plotkin. Lecture notes on domain theory. Available from his home page as The Pisa Notes, 1983.

[Pon66] L. S. Pontryagin. *Topological Groups*. Gordon and Breach, 1966.

[Pra95] Vaughan R Pratt. The stone gamut: A coordinatization of mathematics. In *Logic in Computer Science, 1995. LICS'95. Proceedings., Tenth Annual IEEE Symposium on*, pages 444–454. IEEE, 1995.

[Rud62] Walter Rudin. *Fourier Analysis on Groups*. John Wiley and Sons, 1962.

[Rut00] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249:3–80, October 2000.

[Sil10] Alexandra Silva. *Kleene Coalgebra*. PhD thesis, University of Nijmegen, 2010.

[Smy83] M. Smyth. Powerdomains and predicate transformers. In J. Diaz, editor, *Proceedings of the International Colloquium On Automata Languages And Programming*, pages 662–676. Springer-Verlag, 1983. Lecture Notes In Computer Science 154.

[Sto36] Marshall H. Stone. The theory of representations for boolean algebras. *Trans. Amer. Math. Soc.*, 40:37–111, 1936.

[Sto40] Marshall H. Stone. A general theory of spectra I. *Proc. Nat. Acad. Sci. USA*, 26:280–283, 1940.

[Tan39] T. Tannaka. Über den dualitätssatz der nichtkommutativen topologischen Gruppen. *Tohoku Math. J.*, 45:1–12, 1939.

[Ven06] Yde Venema. Algebras and coalgebras. *Handbook of modal logic*, 3:331–426, 2006.