On Convolution

David B. Benson
School of Electrical Engineering and Computer Science
Washington State University, Pullman, WA 99164-2752

Abstract

Convolution describes a variety of physical system dynamics, both quantum and classical. Convolution, in a discrete form, describes the nondeterministic dynamics of computational systems as well. Petri nets are used as the motivating model for the discrete convolution. The similarities are formal, but the description of manuals for physical experiments by Foulis and Randall also receives expression in terms of Petri nets by this correspondence. The observations of Petri nets are contrasted with the observations of quantum systems. This brief, only descriptive, paper just mentions that there are relationships to linear logic.

Summary

The all-paths view of quantum dynamics, as expressed for example in [1], has a formal correspondence with an all-possibilities formalism for Petri nets and similar distributed computational settings. Within computer science several researchers have noticed a strong connection between the resource constrained notions of computation, such as Petri nets, [5, 6, 7, 8], and J.-Y. Girard's linear logic, [2, 4, 3]. There is a formal connection between linear logic and the Foulis-Randall notion of manuals for physical quantum experiments described in [1]. This is will be reported elsewhere.

While there are several potential beginnings to the description of these formal correspondences, perhaps the notion of convolution abstracts most readily to the situation of interest to the computer scientist. It is the discrete convolution over a finite number of paths, as used in [1], which we can most readily abstract.

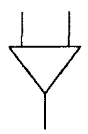
The indeterminism of quantum systems is formalized through probability theories. Its special flavor arises from the probability amplitudes and the rules for combining these. The nondeterminism of computational systems arises through both abstraction and ignorance, that is, incomplete state information. The rules for combining in these distributed computational settings have their own peculiarities, with a nonclassical logic related to the nonclassical logic of manuals.

Despite the formal similarities which are emphasized here, there is no close connection between quantum systems and computational systems. For instance, the nondeterminism in computational settings is, in effect, a hidden-variables theory. See [9]. Even so, the goal here is to find a common setting in which to compare these two conceptual areas.

Splitting and togetherness

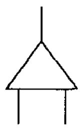
Abstract convolution requires a way of splitting data apart and a way of putting data together. The former is usually called a coalgebra and the latter an algebra. Precisely, our coalgebras are counitary coassociative cocommutative coalgebras of a comonad and our algebras are unitary associative commutative algebras of a monad. That being said, our examples are actually quite easy to understand.

For us, an algebra is simply a way to put data together. We have data coming in on two lines and going out on one line, as in the following picture.



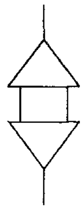
For Petri nets, the algebra is simply the collection of all the tokens from transitions into a place. In dataflow, one thinks of streams of datons being merged into a single stream of datons. You may also think of an adder for complex numbers, or even such an adder followed by taking the squared modulus.

A coalgebra, on the other hand, splits the data into two paths. In dataflow, the incoming stream of datons is split, nondeterministically, into two streams. For Petri nets, a coalgebra takes tokens from a place to send them, nondeterministically, to the connected transitions. There is a certain sophistication to seeing how this must work for Petri nets, [5]. Here is the picture for a coalgebra.



A unitary property for coalgebra-algebra pairs.

Fixing a sending coalgebra and a receiving algebra, one might combine them as in the following picture.



This system is supposed to require *no* time between input and output. One could experiment with such a simple system by sending in data and looking at the output. The experiments can only be thought experiments since no time is allowed to elapse between input and output.

For Petri nets, the number of tokens sent into this system is equal to the number of tokens coming out of this system. Each token either goes to the left or else to the right. In either case, the tokens are all collected by the algebra into one pile of tokens.

In dataflow, the output stream of datons is any permutation of the incoming stream of datons. Since there is no guarantee of recovering the original daton stream, we may say that there is some form of interference occurring in this setting. Perhaps it would be better to call this a form of confusion. In any case, it is not the sort of example we want to consider further.

For particle physics, this simple picture is to denote the mathematics of splitting and then recombining a beam. The example of spin manuals in [1] is most helpful. As is well known, there is interference in a realistic experiment, but as no time elapses in ours, there is no time for the amplitude phase to change and so there is no interference here either.

Specifically, for small particles, the coalgebra is simply an expression of the distributive law for multiplication over addition. The algebra is simply complex addition. We are expressing the Feynman rules in terms of certain coalgebra-algebra pairs.

In all of these examples there are indeed coalgebras and algebras to give precision to these ideas. The question in all three of of these examples is whether or not this little system acts like a very short piece of conductor. Indeed, a piece of conductor of zero length.

In the case of Petri nets and particle physics, the answer is yes. In the case of dataflow, the answer is no. When the answer is yes, we say that this system is equal to the identity and that the pair consisting of our algebra and coalgebra is *unital.* This is the situation that we consider further.

Convolution exposed

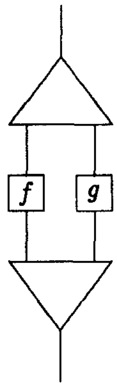
Given two functions, f and g, the convolution of f and g simply requires splitting the data so that some of it is transformed by f and some is transformed by g. The resulting, transformed data is then combined by the algebra.

For Petri nets, the transforming functions are called *transitions.* The tokens are sent from a place to the transitions by the coalgebra and the tokens resulting from the transition are sent by the collecting algebra into the place. Nothing essential changes if there are many places in the Petri net. With many places there are equally many coalgebras choosing where to direct each token. This list of choice coalgebras is again a coalgebra. Dually, there is a collecting algebra for each place and the resulting list of collecting algebras is again an algebra.

Part of the Petri net specification includes the possibility of tokens staying in a place, not being consumed by transitions. To treat this as convolution, we include a special *image* transition which just copies its input to its output. In this exposition, think of the transition function g as the image transition.

For quantum probability, the transforming functions are called *amplitude transition functions.* These specify the change in amplitude from the source to the target. Of course, the algebra at the target combines the amplitudes according to the Feynman rules. With only two possible paths in our little example there are only two amplitude transition functions, f and g.

A picture of the convolution of f and g given by the following diagram, recalling that the coalgebra-algebra pair depends upon the particular situation we are modeling.



This picture does not make obvious the interference between the two paths. In Petri nets the interference occurs in the coalgebra and is often called *conflict*. For example, if there is one place with one token and each of two transitions requires one token from that place, both cannot fire. The choice of which transition receives the token is, in this model, up to the coalgebra which directs each token at a place to either the left or the right.

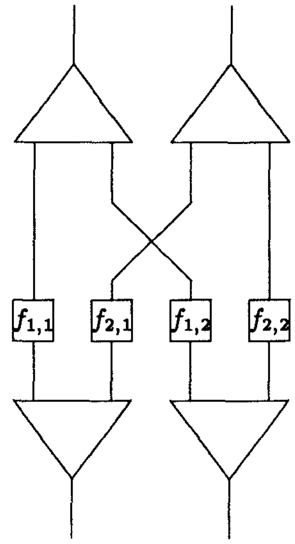
In this attempt to model particle physics, the coalgebra is trivial while all the interference occurs in the crossterms of the collecting algebra. This collecting algebra consists of adding complex numbers. To better understand the classical level, we have a more complicated model, which is still convolution.

Interference

To begin, suppose we have a system with two input channels and two output channels. There is a direct feedthrough transfer function from the first input channels to the first output channel and a direct feedthrough transfer function from the second input channel to the second output channel. There are also the crosstalk terms from the first to the second channel and from the second to the first channel.

As stated, this description will be familiar to many. But we might also classify the transitions of a Petri net as either feedthrough or crosstalk. The first input and the first output channel can be identified with a single place while at the same time we identify the second input and the second output channel with a single place. It is not necessary to do this, but this makes our picture an instance of a two place, four transition net.

Here is the picture of the system with the crosstalk terms $f_{1,2}$ and $f_{2,1}$.



Since our algebra is unitary and our coalgebra is counitary, there is a standard method of indicating the lack of crosstalk. Denoting the composition of the counit followed by the unit as 0, one may write $f_{1,2} = 0$ to indicate that there is no crosstalk from channel one to channel two. In Petri nets, we usually say that there is no transition from place one to place two – and we do not ordinarily draw such nontransitions in a Petri net diagram.

If there is no crosstalk at all, $f_{1,2} = 0$ and $f_{2,1} = 0$, we have two completely independent systems, side by side. This is the situation called the tensor product of quasimanuals in [1] and the biproduct of Petri net behaviors in [5]. There is no possibility of either of the systems influencing the other.

If only one of the crosstalk terms is 0, then there is the possibility of one side influencing the other side, but not the reverse. This is similar, but not identical to, the forward operational product of manuals in [1]. If both crosstalk terms are nonzero, there is the possibility of both sides influencing the other side. This is called the tensor product of Petri nets in [5] and the cartesian product of quasimanuals in [1].

Combining systems

We have just seen some ways in which systems can be combined. All of the above are special cases of the convolution of systems. This works as follows. Fixing the algebra and the coalgebra applicable to the combination of the systems being studied, let f and g be the two system functions. That is, f and g are functions representing the entire behavior of the systems to be combined. The convolution exposed above is then the convolution of systems.

The convolution of systems includes as special cases the series and the parallel connection of systems, as well as the special cases of crosstalk or the lack thereof described in the previous section.

There are other ways to combine systems. In Petri nets, there is a reasonable notion of the intersection and the union of transitions. Two transistions have an intersection if they are connected to the same places and for each possible distribution of input tokens, either both fire and have the same resulting distribution, both do not fire, or one fires and the other does not. The intersection transition fires just in case both argument transitions fire. In the same situation there is a union transition, which fires just in case one of the argument transitions fires.

For example, we always have $f \cup 0 = f$ and $f \cap 0 = 0$.

This is enough to see how to take the union and intersections of sufficiently similar Petri nets. Suppose two nets have the same feedthrough transitions but one has crosstalk only in one direction while the other has crosstalk only in the other direction. The intersection of these two Petri nets is then a Petri net without crosstalk, what we have called a biproduct. The result is the biproduct of the two nets obtainable from either argument Petri net by eliminating the crosstalk transitions.

Similarly, the union of these selfsame argument Petri nets is what we have called the tensor product of Petri nets. In the union Petri net crosstalk occurs in both directions.

These operations have formal similarities to the means of combining manuals in [1].

Linear logic

The connection between linear logic and Petri nets is explicit in [6, 7, 8]. This model of Petri nets is essentially a linear-time model and so not easily suited to the nondeterministic considerations here. A linear logic requires additives, akin to our $\cup$ and $\cap$, and multiplicatives, akin to our tensor product and biproduct. Thus one obtains at least a sizable fragment of linear logic. We pointed out the connection between these constructions on Petri nets and the constructions on manuals, strictly speaking, on quasimanuals. Linear logic is such an important conception that the notion of manual surely fits into the linear logic framework, although there are many details remaining to be checked.

Regarding manuals and quasimanuals, here is a quotation from [1]. "[Foulis and Randall] stress that they are not attempting to develop or advocate and particular theory, rather they are formulating a precise 'language' in which such theories can be expressed, compared, evaluated, and related to laboratory experiments. In principle, any serious activity which is based upon the scientific method can be expressed in the language of operational statistics."

So there is no particular surprise that there are quasimanuals which describe the operation of Petri nets. Moreover, from [4] we are guaranteed that we can start with the notion of quasimanual to end up with a model of linear logic.

What is surprising and pleasing, at least to this author, is that all these differently motivated conceptions are essentially the same underlying idea.

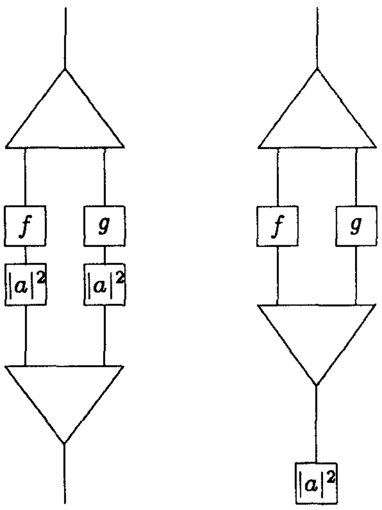
Observation

The notion of process and the observations on processes has received considerable attention recently. We cite only [9] which stems from some of the earliest work on abstract computational processes. What is clear is that observing a computational process may, in fact, irreversably disturb its state. Several have noticed that this seemed similar to quantum mechanics. This is so, but the similarities do not extend far into the mathematics beyond the abstract convolution we study here. We turn to what is similar and dissimilar about these two situations, beyond the obvious fact that the setting of Petri nets is classical whilst the quantum setting is not.

In the setting of Petri nets, we observe one Petri net by the transitions of another Petri net. So we have a Petri net called the observer and another Petri net being observed. For definiteness we write the observing Petri net to the left of the Petri net being observed. These two Petri nets must have some places in common, else the observer cannot determine anything about the configuration of the observed net. So the convolution of the two Petri nets is an example of a combined Petri net with crosstalk terms. We might attempt to have the observing Petri net influence the observed Petri net as little as possible by only listening in, so to speak. That is, with our given orientation, $f_{1,2} = 0$. Of course, there are nets in which this attempt to influence as little as possible meets with complete and utter failure. In such examples, the influence of the observer is so pervasive that observing even one token brings the activity of the observed Petri net to a halt.

On the other hand, one can always consider designing the Petri net so that observations of its behavior at certain selected places do not modify the behavior of the Petri net. One includes certain sink places which can then be freely attached to the transitions of the observing Petri net. This freedom to redesign is certainly not available in quantum systems and indeed is often not available in classical systems.

In the quantum probability setting, each observation or better, observation-like-physical-setup, results in taking the squared modulus of the amplitude. If we don't observe until the very end this operation is done after taking the sum over all paths. If the various paths contain observations along the way, the squared modulus is taken at the point of observation, and the resulting real numbers summed at the end. Here are pictures of the two convolutions, with $|a|^2$ the taking of the modulus. The picture on the left is the situation of observing which slit the electron came through. The picture on the right is the situation in which the electron strikes a screen after passing through both slits without being observed. This is the situation in which interference is exhibited. In these pictures the algebra involved is simply summing the various quantities received.



Interference reconsidered

As we pointed out earlier, the interference in a Petri net behavior is due to the interaction of the choice coalgebra and the requirements of the various transitions to have enough tokens to fire. This is called conflict and the use of this term distinguishes the situation in Petri nets from the quantum situation.

As the just prior picture makes clear, the interference in quantum systems arises at the end, when the modulus square of the sum has crossterms which mathematically give rise to interference.

This analysis shows some of the differences between this classical setting and the quantum setting. The observations of computational processes may indeed change the state of the observed system, but in a way which is rather unlike the effect in a quantum system.

Acknowledgements

Aurelio Carboni, Peter Freyd, Yves Lafont and Ross Street have each shown me aspects of the nice picture calculus for algebra-coalgebra combinations. Stan Gudder clarified a point about quasimanual products.

References

[1] Stanley P. Gudder, *Quantum Probability*, Academic Press, 1988.

[2] Jean-Yves Girard *et al.*, *Proofs and Types*, Cambridge University Press, 1989.

[3] A. S. Troelstra, *Lectures on Linear Logic*, CSLI Lecture Notes No. 29, University of Chicago Press, 1992.

[4] Michael Barr, *∗-Autonomous categories and linear logic*, Math. Structures Comput. Sci. bf 1(1991), 159-178.

[5] David B. Benson and Raju R. Iyer, *On a nondeterministic step dynamics for place-transition nets*, WSU Tech. Rpt. CS-89-206, 1989.

[6] Narcisco Martí-Oliet and José Meseguer, *From Petri nets to linear logic*, Math. Structures Comput. Sci. bf 1(1991), 69-102.

[7] Carolyn Brown, Doug Gurr, and Valeria de Paiva, *An algebraic theory of Petri nets founded in linear logic*, preprint.

[8] Carolyn Brown, Doug Gurr, and Valeria de Paiva, *A linear specification language for Petri nets*, DAIMI PB-363, Computer Science Department, Aarhus University, Denmark, 1991.

[9] Matthew Hennessy, *Algebraic Theory of Processes*, The MIT Press, 1988.