The Ghost of Von Neumann: Automata and Physics

Source: http://sci.tech−archive.net/Archive/sci.physics.research/2007−08/msg00161.html

- *From*: markwh04@xxxxxxxxx

- *Date*: Thu, 30 Aug 2007 05:37:18 +0000 (UTC)

Since going to and returning from Budapest, a long time ago, I've been haunted by the ghost of von Neumann. Though it's probably not widely known, he was a major pioneer in BOTH Computer Science and fundamental Physics, establishing a foundation in the latter based on the primary notions of "state" and "observable". A distinguishing feature of von Neumann's approach was to algebraize everything at the foundational level.

What's not widely appreciated is that the two pursuits were two strands on a single thread. In a similar way, the pursuit in the domain of what's come to be known as formal language and automata theory was to be based on the same foundation, though this apparently never panned out in von Neumann's time.

Under his "ghost", I've (successfully) sought to remedy this since then. A small part of this synthesis is

The Untold Story of Formal Languages
http://federation.g3z.com/CompSci/index.htm#Untold

Originating from 3 article series presented over the USENET in 1993−1996, this details a synthesis of formal language and automata theory essentially within the setting of unversal algebra and category theory. In it, the notion of state plays a central role.

The central theme of von Neumann's enterprise −− if ever there were one −− was this very notion. While this serves as the focal point in the definition of the algebra of observables, of sectors, of transitions, etc.; here it serves a similar critical role where the objects may (in fact) be considered a finitary descriptions of non−linear dynamic systems.

This redacted series is but a small part of the much larger synthesis already on−line at this site. In turn, this is only about 1/4 of what will be present. The oldest emphasis in von Neumann's theory had been BOTH on the combinatorial, as well as stochastic, aspects of the concept of state in automata theory. it is particularly the latter aspect that presents the bridge to non−linear dynamics. The larger synthesis bring this into its scope.

First, we'll start out with the related reference:
Context−Free Expressions
http://federation.g3z.com/CompSci/index.htm#Luecke

Part 2, contained therein, "The von Neumann Correspondence", is of particular interest and expands on parts 1, 3 and 4, below, of the "Untold Story" series.

The Untold Story series:
Part 1. The Algebraic Representation of Formal Languages
An analogy exists between the Fock spaces of classical and quantum physics and the state spaces of automata based on stacks, queues and other lists. Corresponding, respectively, to the create and annihilate operators are the "enqueue" and "dequeue" processes; to the vacuum state, the "empty queue". Based on this correspondence, algebras can be devised that embody the actions of machine and automata models that employ such devices. This section reviews two such algebras, termed the "polycyclic" and "Bra−Ket" algebras, as well as providing the background to the underlying algebraic theories employed throughout the "Untold Story" treatment.

Shades of Finkelstein...

1.1. Background
1.2. Quantales, Dioids and Kleene Algebras
1.3. Algebraic Preliminaries
1.4. Extending Dioids to Quantales
1.5. The Fock Space Representation
1.6. Stack Algebra and Matrix Representations
1.7. An Expression Algebra for Context−Free Languages and Transductions
1.8. The Bra−Ket Algebra
1.9. The Self−Similarity Property

It should be no surprise that quantales and dioids are also seen prominently in the study of non−linear dynamics.

Part 3. The Algebraic Representation of Automata
"We are very far from possessing a theory of automata which deserves that name, that is, a [properly mathematical−logical theory."
−− John von Neumann, _The General and Logical Theory of Automata_

This section features the introduction of an algebra for state diagrams. This is not too unlike the graphic algebra seen, for instance, in one of the writeups on the Baez web site.

Treating a state diagram as a graphical representation of a matrix, one can define the opeators of addition and multiplication over them. The result is an extension of the cycle notation used for representing groups.

The classes of automata seen in classical formal language theory and automata theory may all be seen as instances cut from the cloth of the same "infinite state automaton" mould, where by each class is distinguished by

(a) the FACTORING of its state space into a product of a 'control" space and "device" space
(b) the imposition of SELECTION RULES constraining the allowable transitions over the device space
(c) the imposition of SYMMETRY rules governing the transition in device space, in virtue of which the entire (infinite) set of transitions can be generated from a finite kernel.

The last property (symmetry) is none other than that captured by the results known in the classical theory as the various "pumping lemmas".

Features (b) and (c) play roles analogous to those played by selection rules and symmetry in non−linear dynamics.

In this section, the notion of automata and state diagrams as graphical representations of systems of inequalities over a partially ordered algebra is developed. This concept leads us directly to the issue of operator algebras and their matrix representations.

3.1. State Diagrams, Automata and Matrices
3.2. Infinite State Automata
3.3. Varieties of Infinite State Automata
3.4. Representation of Automata as Systems of Inequalities
3.5. Connection to the Operator and Matrix Algebras

Part 4. Context−Free Expressions and the Bra−Ket Algebra
The class of infinite state automata corresponding to the 1−stack automata can be directly converted into a finite linear system of inequalities over an algebra that incorporates the Bra−Ket algebra. The resulting expressions excent the classical theory of regular expressions (such as those seen in the UNIX facility GREP) up to type 2 in the Chomsky hierarchy and so may be termed "context−free expressions".

A process for converting the corresonding non−linear fixed−point systems into linear form is developed. A key feature of the conversion is the process of "linearization", which may be likened to that seen in mathematical physics of taking the spinor "square root" of a vector. With the use of the Bra−Ket operators, the non−linear systems are decouplined into first−order systems.

4.1. The Bra−Ket Algebra; Tensor Products
4.2. Infinite State Automata and Bra−Ket Notation; Example
4.3. Reduction of 1−Stack Automata

4.4. Embedding into the Algebra C_2 x R(X*)
4.5. Example
4.6. Context−Free Expressions and Fixed−Point Systems: Examples
4.7. Converting EBNF Grammars and Fixed−Point Systems into Bra−Ket Form
4.8. Optimizations

Related Reference:
Stacks as Algebras
http://federation.g3z.com/CompSci/index.htm#StackAlg

Here, the genesis of the formal "Dirac" bra−ket notation is explained in more detail; particularly its relation to the classical N−body Maxwell−Boltzmann Fock space.

Other sections, not outlined here
Part 2. Regular Expressions −> DFA's
(Development of the REX superset of GREP)

Part 5. Turing Expressions and Translation Expressions
(Algebraic representation for ALL computation)

Part 6. The Hardest Context−Free Language

Part 7. Grammars as Fixed−Point Systems

Part 8. Direct Proof of Parikh's Theorem
(Fixed−point theorem for abelian algebras)

Part 9. Properties of Context−Free Systems

.