The Complete Arithmetization Program

"God created the integers. All esle is the work of humans." (L. Kronecker)

Raptis, T. E.

[1]Computational Applications Group, Division of Applied Technologies, NCSRD

[2]Physical Chemistry Lab(Assoc.), National University Athens

2021

Arithmetization of Automata

- 1 Analog machines add/subtract/mutliply quantities directly (currents, voltages, frquencies)
- 2 Symbolic/Digital Automata require expansions of inputs in symbolic series (TM tape, 'Registers')
- 3 Let $A : 0, 1^M \to 0, 1^N$ an arbitrary automaton as a symbolwise morphism between two "Hypercubes"
- 4 Let $P : 0, 1^k \to N$ the polynomial representation of all integers
- 5 Redefine as

$$\nu \leftarrow (P \circ A \circ P^{-1})(\mu), \mu, \nu \in N$$

("Arithmetized" Automaton)

The Hamming Hierarchy

- 1 Let $L_k = 2^k - 1$, a "Mersenne integer" and $[0, ..., L_k]$ a "Mersenne interval".
- 2 Let $H(L) \leftarrow \{P^{-1}(0); P^{-1}(1); ..., P^{-1}(L_k)\}$ a $k \times L_k + 1$ Lex-ordered array of same length k strings
- 3 $L_1 \subset L_2 \subset .... \overset{P^{-1}}{\rightarrow} H(1) \subset H(2) \subset ...$ defines a naturally self-similar hierarchy of Hamming Spaces
- 4 Equivalent simultanous decoder $H^k_{ij} = \lfloor Mod(j/2^i, 2) \rfloor$ Extendible to any higher alphabet b via $2 \rightarrow b$
- 5 Arithmetized automata can be studied through their global maps across the hierarchy

The Hamming Hierarchy

Properties

- 1 All arrays $H(L_k)$ hide a special geometry associated with "counter systems" and rooted binary trees
- 2 All strings (columns) equivalent to the enumeration of paths of any rooted binary tree
- 3 Lex-ordering associated with the action of an exponential "Dilation" group across rows (discrete hyperbolic space)
- 4 Action taken over $Aut[Z_2^k]$ provides the global structure of a semi-direct "Wreath" product of Z_2 with D_{2^i}

The Hamming Hierarchy

Inductive Combinatorics

- 1 Geometry of Lex-ordered Hamming spaces inherited by many fundamental sequences and global maps with a recursive, "arithmetic fractal" structure.
- 2 Mersenne intervals are natural clauses for bitwise operands (AND, OR, XOR)
- 3 Bit-XOR $2^k \times 2^k$ matrices form the multiplication tables of any $Aut[Z_2^k]$ of whose the Cayley diagram is the associated $k-$Dim. hypercube
- 4 Many sequences compressible via a universal list iteration scheme

$$S_{n+1} \leftarrow [S_n, M(S_n)]$$

with $M(x)$ an appropriate "Reproducing Map" and $[,]$ as concatenation. Effective "replacement" of the $P \circ A \circ P^{-1}$ operator.

Applications of the Hierarchy

Some standard fractal sequences

Sequence	Reproducing Map $M(x)$	S_0
Digit Sum (h_w)	$x_n + 1$	[0]
Runlength Dim.	$(\hat{R})(x_n) + 1$	[0]
Symmetric Gray	$(\hat{R})(x_n) + 2^{n-1}$	[0 1]
Stern-Brocot	$(\hat{R})(x_n) + x_n$	[1 2]
Radon-Hurwitz	$\mathbf{c}_0 x_n + \mathbf{c}_1 \delta(n)$	[1 2]

where $\hat{R}$ is a left-right reflection of an array or a "mirroring" operation, the auxilliary vectors $\mathbf{c}_0 = [ones(n-1), 0]$, $\mathbf{c}_1 = [zeros(n-1), 1]$ and $\delta(n) = [2, 4, 1, 1, 2, 4, 1, ...]$. The last one also has a nown resolvant (individual) formula

Applications of the Hierarchy

Concrete Examples of Recursions

All bitwise Boolean operands admit an iterative 2D array representation as

$$B_{n+1} \leftarrow \begin{pmatrix} B_n + 2^{n-1}B_0(1,1) & B_n + 2^{n-1}B_0(1,2) \\ B_n + 2^{n-1}B_0(2,1) & B_n + 2^{n-1}B_0(2,2) \end{pmatrix}$$

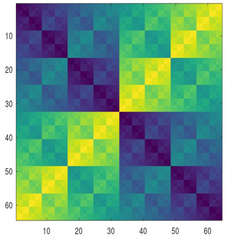
where B_0 the associated "2-gate" representation of AND,OR and XOR 2×2 matrices.
Using the epxonential periodicities one can also extract the relevant "Hamming Weights" (Sums of Digits) $h_w(B_n)$ of these matrices as

$$hB_{n+1} \leftarrow \begin{pmatrix} hB_n + B_0(1,1) & hB_n + B_0(1,2) \\ hB_n + B_0(2,1) & hB_n + B_0(2,2) \end{pmatrix}$$

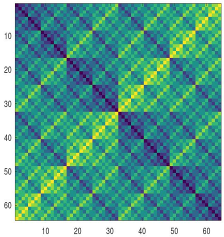
Parity matrices can also be extracted directly from $Mod(h_w(B_n), 2)$. The $h_w(BitAND)$ matrices form fundamental template matrices for all DNF/CNF-SAT problems (work in progress).

Applications of the Hierarchy

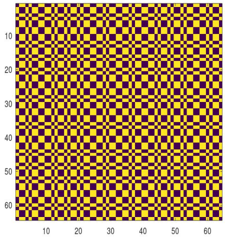
64 by 64 Bit-XOR Fractal Matrices



Basic Bit-XOR



Hamming Distances



XOR Parities

An Algebra of Patterns

- 1 Let $s_x = [\sigma_0, \sigma_1, ..., \sigma_k]$ a column of $H(L_k)$ and $x \leftarrow |s_x| = P(s_x)$ it's integer representative. Two natural involutions in every $H(L)$ instance ($\hat{L}$ involutory iff $\hat{L}^2 = Id$)
- 2 Row-wise involution: Not complement $\bar{x} : 1 - s_x(i) \rightarrow 2^k - x - 1$ Fixed-point-free!
- 3 Column-wise involution: reflection/mirroring (eg 100110 -¿ 011001) Fixed points: subset of all palindromes
- 4 Double (*) involution $x* \rightarrow \hat{R}(\bar{x}) = 2^k - R(x) - 1$ (R and NOT commute!) Invariant: Runlength "Block" Dimension
- 5 Concatenation as "addition": $[s_x, s_y] \rightarrow x + 2^k y$. Hamming weights as a ("hyperbolically") deformed homomorphism: $h_w(x + 2^k y) = h_w(x) + h_w(y)$
- 6 Possible C* norm def: $|x, y*| = |x, x*||y, y*|$. Choose $||.|| \sim exp(f(x, y)) : f(x, y*) = f(x, x*) + f(y, y*)$

The Free (Non-Commuting) Algebra of all Structured Programs

- 1 Algebraic equivalent for each and every instance of the Hamming Hierarchy over any alphabet $b \geq 2$ as an expansion of a multinomial of non-commuting operators with an additional Lex-ordering axiom as
$$H(L_k, b) \equiv (\hat{L}_0 + ... + \hat{L}_{b-1})^k$$
Using the properties fo the hierarchy as introduced previously, certain simple propositions may be easier to prove/compute with recourse to the abstract words formed by "symbols" L_i
- 2 Fundamental Th. of structured programming: equivalent to a special device termed a "Universal Multiplexer" (UMUX) turning each program into brancheless recursive functions via a kind of data "tagging" equivalent to complexification.
- 3 Given a set of N Boolean characteristic functions or indicators $\chi_i{}_{i=1}^{N}$ satisfying $\chi_i \circ \chi_j = 0$ and an associated set of maps $f_i(x)$. Any program equivalent to a multinomial expansion with k equivalent to discrete computational time.

Arithmetization, Chu Spaces and Quantum Logic (Future work)

- 1 Chu Spaces as most general models of topology. Abramsky showed the equivalence between concurrent automata, complementarity and ucertainty.
- 2 A Boolean "Membership" function $M(data, classes)$ is a "Multilabel Classifier" with classes "labelled".
- 3 Natural expansion of the Hamming Weight sequence as a function over the integers into a set of separate Boolean classifiers

$$h_w \leftarrow \sum_{i=0}^{k} \mathbf{1}_\chi \left(s_n \in \binom{k}{i} \right)$$

Arithmetization, Chu Spaces and Quantum Logic (Future work)

- 1 Mappings between classifiers ("evolution") possible with one of two adjointness conditions

$$M_1(f(data), labels) = M_2(data, f^{-1}(labels))$$

$$M_1(f(data), labels) \leq M_2(data, f^{-1}(labels))$$

- 2 The last is a form of "implication" relevant also in ortholattices that came out of the original 1939 paper of Birkhoff and Von Neumann.
- 3 All in all, the whole QM business might be no more than a play between some "card-thieving" automata!