PAPER • OPEN ACCESS

A universal rewrite system adapted for formal language theory, classical and quantum computing

To cite this article: Sydney Rowlands and Peter Rowlands 2022 *J. Phys.: Conf. Ser.* **2197** 012024

A universal rewrite system adapted for formal language theory, classical and quantum computing

[1]Sydney Rowlands, [2]Peter Rowlands

[2]Physics Department University of Liverpool
Oliver Lodge Laboratory
Oxford St, Liverpool. L69 7ZE, UK

[1]jrist29@gmail.com
[2]p.rowlands@liverpool.ac.uk

Abstract. The meta-pattern of the universe, first formulated by Rowlands and Diaz [2002], is a universal rewrite system (URS). This universal pattern finds a formulation in formal language theory centred around the fundamental semantic unit of the zero word or the zero string: $0 = X_0X^{\#}_0$. This is realized successively in the computational procedures of Turing machines, Post machines and Finite machines with two pushdown stores.

1. Introduction

A universal alphabet and rewrite system, first formulated by Rowlands and Diaz [1-6], has a strong claim to be the fundamental meta-system in Nature, sought by Bateson among others [7]. Essentially, it predicates an infinitely degenerate totality zero as the state of the universe at all time, which is realized by an infinite succession of zero-totality alphabets, each of which ensures uniqueness by incorporating its predecessor. The succession is not necessarily temporal, but rather supervenient, as time is a product of the process, rather than an assumed component. It is a succession of zero cardinalities, rather than a succession of infinite ones.

The technical details of the process, which are remarkably simple as we would expect, are described in many publications but will be outlined more extensively in this paper. Essentially, there is only one process of transition between successive states of the universe, but it simultaneously requires two aspects, signified respectively by $\rightarrow$ and $\Rightarrow$, and referred to, for convenience, as 'conserve' and 'create'. A state of the universe is described by a zero-totality alphabet, in which each component is always accompanied by a conjugate (signified by $^{\#}$). The alphabet can then be concatenated either with one or more components of itself or 'subalphabet' (conserve) or with its entire self (create). The first aspect yields only automorphisms of the alphabet, whereas the second produces an entirely new alphabet. However, the new alphabet will only be valid if it both contains the previous alphabet and also fulfils the requirements of 'conserve' in that all the new components concatenate with the new alphabet to produce automorphisms of that alphabet. (The automorphisms differ in producing different ordering of the terms, but are identical in totality.)

We can describe the process in practical terms using symbols, though symbols are not themselves necessary to the process. There is no fixed start or end state, though we can define a start and end for our convenience, and the process is effectively a fractal. The alphabet is not fixed but extends continuously, and the production rules are recreated at every stage, although there are generic similarities between the stages. The concatenation process can be conveniently described using replacement rules for the symbols, which are illustrated in the way that the automorphisms occur. At each stage, just one entirely new symbol is created, but this is accompanied by concatenations with all the previously created symbols. The replacement rules are determined by the requirement that the new symbol must also describe a new process – newness cannot be created by the symbol itself. Various things emerge from the symbolic representation and the need for it always to produce something new, including the fact that any symbol, other than the starting symbol R (or identity), does not concatenate with itself to produce R, but rather its conjugate, and that successive new symbols following R, beginning, say with A and B, concatenate to produce AB, which also concatenates with itself to produce the conjugate of R, which we represent as $R^{\#}$. In principle, an *anticommutativity* is introduced into the system to ensure that A and B are new and not just a new representation of R. The anticommutativity also introduces an aspect of closure and discreteness, not previously assumed. It additionally means that the only way to continue the sequence is to introduce new pairs of symbols which are anticommutative to each other but commutative to all the others. The series can then continue to infinity with the uniqueness of each new symbol assured by the fact that it has a unique partner with which it anticommutes. In effect the alphabet continues to infinity by incorporating a generically repeating aspect.

In addition, entropy is built into the structure in a significantly pure form in that each successive stage effectively doubles the alphabet for the previous one by adding a new symbol and all its concatenations. If the number of independent 'microstates' is $W = 2^n$ at level n, then taking a logarithmic function of this reveals that the entropy is simply an index of the relative level reached. In effect, entropy is a description of the working of the system of Nature, not an additional property requiring explanation [4,6,8]. The system is also deterministic in that no symbol repeats; each is distinct, only repeating generically at a higher level.

The system has, since its first conception, been used in many applications, for example, in generating mathematical structures, such as the real numbers, integers, quaternions, Clifford algebra, and even Conway's surreal numbers, in addition to those of mathematical logic. The mathematics that it resembles most is Clifford algebra, suggesting the particular significance of this algebra in the description of many aspects of Nature, but no mathematical structure is excluded. The only basic assumption is totality zero and no further assumptions need to be made to generate them. It is even possible to generate the full sequence of Cayley-Dickson algebras *as mathematic*s, though the evidence suggests that the primary version has no need to introduce antiassociativity along with anticommutativity. In fact, the rewrite system as a purely *natural* process needs no inputs precisely because it is not antiassociative. Antiassociativity forces us to choose between options, whereas anticommutativity merely forces us into the only available one. When time emerges as an intrinsic component of physics and all sciences based upon it, we see that antiassociativity would require time as well as space reversal at a fundamental level, and, in effect, time, because of the algebraic structure which emerges with its definition, has an associativity which cannot be changed.

By contrast with mathematics, it has been shown that physics has a special structure in that its four basic parameters, mass, time, charge and space have the properties (real / imaginary, conserved / nonconserved, dimensional / nondimensional) and mathematical structures (real, complex, quaternion and complex quaternion) required by the first four alphabets starting from R, and that these, when combined into the highest alphabet, lead to structures which concatenate to a complete zero, meaning that all subsequent alphabets will automatically become zero without being specified. This *nilpotent structure* is ubiquitous in physics at all levels and in all natural systems defined by the conservation of energy or Newton's third law of motion, or with a changing energy that can be defined by a known process. In effect, the principles of relativistic quantum mechanics, defined by the nilpotent structure, become the template for investigating all higher order systems [9-21]. A parallel system of genetics, also using four component units (A, T, G, C) uses exactly the same 64-part mathematics as physics (the algebra of a double space or space and conjugate space); while the identical algebra also leads to fundamental particle structures [4,22-25].

The structures from the rewrite system are determined by characteristic mathematical patterns (in particular, duality, anticommutativity and symmetry-breaking, associated with the numbers 2, 3 and 5) which scale upward via a replacement of the original components by higher order ones, and which ultimately include such areas as nuclear physics, atomic structure and the Periodic Table, chemistry, systems (physical, biological, higher order and constructed), physiology, evolutionary and cell biology, and consciousness among others, as has been demonstrated in a long series of publications. Computing aspects include automated reasoning or AI and the complexity problem with an investigation of the p / np question by Marcer and Rowlands suggesting that the structured nature of the rewrite system and the regularity of its application must lead to an answer favouring the p (or polynomial) alternative [4, 26-32]. A category theory application is currently under development by the present authors.

A key area of investigation is in the theory of formal languages in computer science. Applications of this in current technology include programming languages such as C++, Java, xml, html, extensible markup languages, compiling, parsing (text mining). The aim of this paper is to reveal the formal language aspect of the rewrite system, and to demonstrate that the pattern of this system conforms to the rules of its own language generation structure and that this language is recognizable by a Turing machine algorithm. In principle, we show that the rewrite system can simulate language defined by a set of meaningful pattern units ('words'). This language, which we identify as the language of nature, is a type 1 (context sensitive) language.

Current formal language theory suggests that an infinite alphabet requires a finite repeating unit and an infinite countable set of symbols. These are related to the duality and anticommutativity (2 and 3) of the rewrite system. Diaz and Rowlands have already used the rewrite structure to develop computer language in terms of an algebraic interpretation of the infinite square roots of –1 [3], including a special unit repeated (quaternions), and extending to infinity in Clifford algebra, which was itself originally developed from the closed group of quaternions plus Grassmann's infinite tensor outer product. We have already shown that physics, derived from the rewrite system, has such a structure, combining nilpotent fermions as a generically repeated unit, with an infinite Hilbert space derived from the Grassmann algebra. In fact the rewrite system itself is a universal version of this pattern, probably the most general that can be derived. The remainder of the paper shows how to construct a computer-compatible simulation of the rewrite system, up to a limit determined by the user.

2. The URS as a formal language can be expressed using two types of alphabets

The finite alphabet form of the URS indicates that there exist four types of characters

1	–1	***i***	***j***
a	b	c	e
X_0	$X^{\#}_0$	X_{2u+1}	X_{2u+2}

where u = $u(n-1) = \frac{1}{4}(2(n-1) - 3 - (-1)^{n-1}) \in \{0, 1, 2, 3, \ldots\}$ and the order of the URS is 2^n where $n \in \{1, 2, 3, \ldots\}$.

The infinite alphabet form of the URS shows the infinite number of quaternion cycles

1	–1	$\boldsymbol{i_1}$	$\boldsymbol{j_1}$	$\boldsymbol{i_2}$	$\boldsymbol{j_2}$	$\boldsymbol{i_3}$	…
a	b	$\boldsymbol{c_1}$	$\boldsymbol{e_1}$	$\boldsymbol{c_2}$	$\boldsymbol{e_2}$	$\boldsymbol{c_3}$	…
X_0	$X^{\#}_0$	X_1	X_2	X_3	X_4	X_5	…

The URS can be represented as a language consisting of an infinite sequence of empty strings

$L_{URS} = \{\epsilon_1 = ab, \epsilon_2 = abac_1bc_1, \epsilon_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1,$

$\epsilon_4 = abac_1bc_1ae_1be_1\ ac_1e_1bc_1e_1ac_2bc_2\ ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2c_1e_1bc_2c_1e_1$,

$\epsilon_5=$
$abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2c_1e_1bc_2c_1e_1ae_2be_2ae_2c_1be_2c_1ae_2e_1be_2e_1ae_2c_1e_1be_2c_1$
$e_1ae_2c_2be_2c_2ae_2\ c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2e_1ae_2c_2c_1e_1be_2c_2c_1e_1$,

$\epsilon_6=abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2c_1e_1bc_2c_1e_1ae_2be_2ae_2c_1be_2c_1ae_2e_1be_2e_1ae_2c_1$
$e_1be_2c_1e_1ae_2c_2be_2c_2ae_2c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2e_1ae_2c_2c_1e_1be_2c_2c_1e_1ac_3bc_3ac_3c_1bc_3c_1ac_3e_1bc_3e_1ac_3c_1e_1bc_3c_1$
$e_1ac_3c_2bc_3c_2ac_3c_2c_1bc_3c_2c_1ac_3c_2e_1bc_3c_2e_1ac_3c_2c_1e_1bc_3c_2c_1e_1ac_3e_2bc_3e_2ac_3e_2c_1bc_3e_2c_1ac_3e_2e_1bc_3e_2e_1ac_3e_2c_1e$
${}_1bc_3e_2c_1e_1ac_3e_2c_2bc_3e_2c_2\ ac_3e_2c_2c_1bc_3e_2c_2c_1\ ac_3e_2c_2e_1bc_3e_2c_2e_1ac_3e_2c_2c_1e_1bc_3e_2c_2c_1e_1, \ldots\}$

3. URS generation via tensor products

The first six orders of the Universal Rewrite System (URS) written as Clifford Algebra tensor products of anti-commutative quaternion cycles

order 2	± 1
order 4	$\pm 1, \pm \boldsymbol{i}_1$
order 8	$\pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1$
order 16	$\pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$
order 32	$\pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1,$
	$\pm \boldsymbol{j}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$
order 64	$\pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1,$
	$\pm \boldsymbol{j}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$
	$\pm \boldsymbol{i}_3, \pm \boldsymbol{i}_3\boldsymbol{i}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_2, \pm \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1,$
	$\pm \boldsymbol{i}_3\boldsymbol{j}_2, \pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{i}_3\,\boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1,$
	$\pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$

There are two ways to generate the empty strings in the URS. Both procedures are connected by the conserve part of the CREATE and CONSERVE algorithms of the URS. The first procedure involves taking an infinite tensor product of the alphabetic symbols. The second way is to use the replacement rules of the URS grammar. (Note: to build transition tables for the different types of machines running through the Chomsky hierarchy of generative grammars → rewrite all words of URS using the same alphabets = finite alphabet and infinite alphabet.)

The behaviour of the URS symbols can be represented in the following identities:

$$\begin{aligned} aa &= a \\ ab &= ba = b \\ ac &= ca = c \\ bc &= cb \\ ae &= ea = e \\ be &= eb \\ ce &= bec \\ ec &= bce \\ bcbe &= bce \\ bcbc &= bcc \\ bebe &= bee \end{aligned}$$

and the tensor products presented in the left of the table shown below reduce to the words of the URS in the right side of the table shown below:

Tensor Product = Conserve part of URS processing		ϵ_n	ϵ_n (infinite alphabet)
(a × b)	(a, b)	ab	ab
(a × b) ⊗ (a × c)	((a, a), (a, c), (b, a), (b, c))	aaacbabc = abacbc	$abac_1bc_1$
(a × b) ⊗ (a × c) ⊗ (a × e)	((a, a, a), (a, c, a), (b, a, a), (b, c, a), (a, a, e), (a, c, e), (b, a, e), (b, c, e))	aaaacabaabcaaaeaceba ebce = abacbcaebeacebce	$abac_1bc_1ae_1be_1ac_1e_1bc_1$ e_1
(a × b) ⊗ (a × c) ⊗ (a × e) ⊗ (a × c)	((a, a, a, a), (a, c, a, a), (b, a, a, a), (b, c, a, a), (a, a, e, a), (a, c, e, a), (b, a, e, a), (b, c, e, a), (a, a, a, c), (a, c, a, c), (b, a, a, c), (b, c, a, c), (a, a, e, c), (a, c, e, c), (b, a, e, c), (b, c, e, c))	aaaaacaabaaabcaaaaea aceabaeabceaaaacacac baacbcacaaecacecbaec bcec = abacbcaebeacebceacbc accbccacebceaccebcce	$abac_1bc_1ae_1be_1$ $ac_1e_1bc_1e_1ac_2bc_2$ $ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2$ $c_1e_1bc_2$ c_1e_1
(a × b) ⊗ (a × c) ⊗ (a × e) ⊗ (a × c) ⊗ (a × e)	((a, a, a, a, a), (a, c, a, a, a), (b, a, a, a, a), (b, c, a, a, a), (a, a, e, a, a), (a, c, e, a, a), (b, a, e, a, a), (b, c, e, a, a), (a, a, a, c, a), (a, c, a, c, a), (b, a, a, c, a), (b, c, a, c, a), (a, a, e, c, a), (a, c, e, c, a), (b, a, e, c, a), (b, c, e, c, a), (a, a, a, a, e), (a, c, a, a, e), (b, a, a, a, e), (b, c, a, a, e), (a, a, e, a, e), (a, c, e, a, e), (b, a, e, a, e), (b, c, e, a, e),	aaaaaacaaabaaaabcaaa aaeaaaceaabaeaabceaa aaacaacacabaacabcaca aaecaacecabaecabceca aaaaeacaaebaaaebcaae aaeaeaceaebaeaebceae aaaceacacebaacebcace aaeceacecebaecebcece = abacbcaebeacebceacbc accbccacebceaccebcce aebeaecbecaeebeeaece beceaecbecaeccbeccae cebeceaeccebecce	$abac_1bc_1ae_1be_1ac_1e_1bc_1$ e_1ac_2b $c_2ac_2c_1bc_2c_1ac_2e_1bc_2e_1a$ $c_2c_1e_1bc_2c_1e_1ae_2be_2ae_2c$ $_1be_2c_1ae_2e_1be_2e_1ae_2c_1e_1$ $be_2c_1e_1ae_2c_2be_2c_2ae_2$ $c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2$ $e_1ae_2c_2c_1e_1be_2c_2c_1e_1$

	(a, a, a, c, e), (a, c, a, c, e), (b, a, a, c, e), (b, c, a, c, e), (a, a, e, c, e), (a, c, e, c, e), (b, a, e, c, e), (b, c, e, c, e))		
(a × b) ⊗ (a × c) ⊗ (a × e) ⊗ (a × c) ⊗ (a × e) ⊗ (a × c)	((a, a, a, a, a, a), (a, c, a, a, a, a), (b, a, a, a, a, a), (b, c, a, a, a, a), (a, a, e, a, a, a), (a, c, e, a, a, a), (b, a, e, a, a, a), (b, c, e, a, a, a), (a, a, a, c, a, a), (a, c, a, c, a, a), (b, a, a, c, a, a), (b, c, a, c, a, a), (a, a, e, c, a, a), (a, c, e, c, a, a), (b, a, e, c, a, a), (b, c, e, c, a, a), (a, a, a, a, e, a), (a, c, a, a, e, a), (b, a, a, a, e, a), (b, c, a, a, e, a), (a, a, e, a, e, a), (a, c, e, a, e, a), (b, a, e, a, e, a), (b, c, e, a, e, a), (a, a, a, c, e, a), (a, c, a, c, e, a), (b, a, a, c, e, a), (b, c, a, c, e, a), (a, a, e, c, e, a), (a, c, e, c, e, a), (b, a, e, c, e, a), (b, c, e, c, e, a), (a, a, a, a, a, c), (a, c, a, a, a, c), (b, a, a, a, a, c), (b, c, a, a, a, c), (a, a, e, a, a, c), (a, c, e, a, a, c), (b, a, e, a, a, c), (b, c, e, a, a, c),	aaaaaaacaaaabaaaaabc aaaaaaeaaaaceaaabaea aabceaaaaaacaaacacaa baacaabcacaaaaecaaac ecaabaecaabcecaaaaaa eaacaaeabaaaeabcaaea aaeaeaaceaeabaeaeabc eaeaaaaceaacaceabaac eabcaceaaaeceaacecea baeceabceceaaaaaacac aaacbaaaacbcaaacaaea acaceaacbaeaacbceaac aaacacacacacbaacacbc acacaaecacacecacbaec acbcecacaaaaecacaaec baaaecbcaaecaaeaecac eaecbaeaecbceaecaaac ecacacecbaacecbcacec aaececacecececbaececbc ecec = abacbcaebeacebceacbc accbccacebceaccebcce aebeaecbecaeebeeaece beceaecbecaeccbeccae cebeceaeccebecceacbc accbccacebceaccebcce accbccacccbcccaccebc ceacccebccceacebceac ecbcecaceebceeaceceb ceceacecbcecaceccbce ccacecebceceaceccebc ecce	$abac_1bc_1ae_1be_1ac_1e_1bc_1$ e_1 ac_2bc_2 $ac_2c_1bc_2c_1ac_2e_1bc_2e_1$ $ac_2c_1e_1bc_2c_1e_1ae_2be_2$ $ae_2c_1be_2c_1ae_2e_1be_2e_1$ $ae_2c_1e_1be_2c_1e_1ae_2c_2be_2c$ $_2ae_2$ $c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2$ e_1 $ae_2c_2c_1e_1be_2c_2c_1e_1ac_3bc$ $_3$ $ac_3c_1bc_3c_1ac_3e_1bc_3e_1$ $ac_3c_1e_1bc_3c_1e_1ac_3c_2bc_3c$ $_2$ $ac_3c_2c_1bc_3c_2c_1ac_3c_2e_1bc$ $_3c_2e_1$ $ac_3c_2c_1e_1bc_3c_2c_1e_1ac_3e_2$ bc_3e_2 $ac_3e_2c_1bc_3e_2c_1ac_3e_2e_1bc$ $_3e_2e_1$ $ac_3e_2c_1e_1bc_3e_2c_1e_1$ $ac_3e_2c_2bc_3e_2c_2$ $ac_3e_2c_2c_1bc_3e_2c_2c_1$ $ac_3e_2c_2e_1bc_3e_2c_2e_1$ $ac_3e_2c_2c_1e_1bc_3e_2c_2c_1e_1$

	(a, a, a, c, a, c), (a, c, a, c, a, c), (b, a, a, c, a, c), (b, c, a, c, a, c), (a, a, e, c, a, c), (a, c, e, c, a, c), (b, a, e, c, a, c), (b, c, e, c, a, c), (a, a, a, a, e, c), (a, c, a, a, e, c), (b, a, a, a, e, c), (b, c, a, a, e, c), (a, a, e, a, e, c), (a, c, e, a, e, c), (b, a, e, a, e, c), (b, c, e, a, e, c), (a, a, a, c, e, c), (a, c, a, c, e, c), (b, a, a, c, e, c), (b, c, a, c, e, c), (a, a, e, c, e, c), (a, c, e, c, e, c), (b, a, e, c, e, c), (b, c, e, c, e, c))		

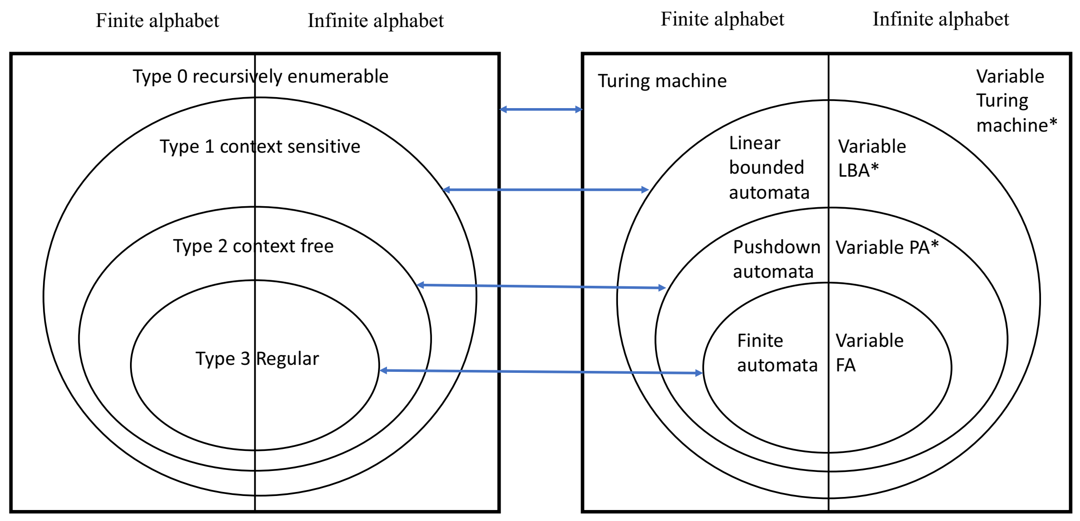
The URS represented as one long sequence of empty strings might be justification for the one fermion theory of the universe

$$\epsilon = \epsilon_1\epsilon_2\epsilon_3\epsilon_4\epsilon_5\epsilon_6\ldots.$$

Every new symbol introduced by create has to be tested by conserve to maintain the pattern and prevent anomalies. Subsequently, the conserve process is responsible for the size of each ϵ_n, $n \in \{1, 2, 3, \ldots\}$.

4. URS generation via grammar replacement rules

Venn Diagram of Grammar Types:



* conjectured as plausible – yet to be completely determined

In the Chomsky hierarchy, a phrase structure grammar G is a quadruple, G = (V_N, V_T, P, S), V_N is the set of variable symbols, V_T is the set of terminal symbols, P is the set of production rules, S is the starting symbol. The grammar of the URS, G_{URS}, is a triple, G_{URS} = ($V_{N\text{-}URS}$, P_{URS}, ϵ), $V_{T\text{-}URS}$ does not exist in G_{URS}. The existence $V_{T\text{-}URS}$ would terminate the application of the production rules P_{URS}. Since the URS never ends, the production rules P_{URS} never stop being applied therefore $V_{T\text{-}URS}$ does not exist. However, for practical coding purposes, the set of symbols. {X_0, $X^{\#}{}_0$, X_{2u+1}, X_{2u+2}} would serve as the set of terminal symbols, $V_{T\text{-}URS}$, to create an arbitrary termination point for computing hardware.

The production rules, P_{URS}, are as follows:

(0) create first new symbol (assume a nonzero symbol): $\epsilon \longrightarrow$ a
(1) conserve first zero totality: a $\longrightarrow$ ab
(2.1) conserve zero totality: ab $\longrightarrow$ aab
(2.2) conserve zero totality: ab $\longrightarrow$ bab
(3.1) create a new symbol: abab $\longrightarrow$ abacbc
(3.2) create a new symbol: baab$\longrightarrow$ baacbc

Rules (2) and (3) split into positive and negative versions of each other. Both options lead to the same result but either branch must be rule replaced in a certain sequence consistently and independently of each other to give the same universal rewrite system. (2.1) conserve positive must be replaced by (2.2) negative substitution and vice versa. To apply the rules to generate the universal rewrite system:

(0) create $\epsilon \longrightarrow$ a
(1) conserve a $\longrightarrow$ ab
(2.1) conserve positive ab$\longrightarrow$ aab
OR
(2.1) conserve negative ab$\longrightarrow$ bab

(2.2) negative substitution into conserve positive a(**ab)** → a(**bab)**
OR
(2.2) positive substitution into conserve negative b(**ab)** → b(**aab)**
(3) create from negative substitution into conserve positive abab → abacbc
(3) create from positive substitution into conserve negative baab → baacbc

In (2.2), we are using a substitution rule (highlighted in bold). This *anticipatory process* is necessary to create the spaces for the symbols coming, in the same way as we create the space, by anticipation, for the expanded algebra in the basic URS. In this way it explains how the create process actually 'calculates' the size of the new algebra which emerges. The URS generated by P_{URS}, the URS grammar replacement rules, is the same URS independently of whether the URS language is expressed using the finite or the infinite alphabet. The infinite alphabet will be used for the URS the rest of this paper.

L_{URS} is generated from an infinite number of tensor products: $\epsilon_1 = (a \times b)$, $\epsilon_2 = (a \times b) \otimes (a \times c_1)$, …

P_{URS} 1.0 in the style of [1]	**P_{URS} 2.0**
create a new symbol: (X, X*)(X, X*) ⇒ (X, X*, Y, Y*) conserve zero totality: X(X, X*) → (X, X*) X*(X, X*) → (X*, X) = (X, X*) [The zero-totality alphabets are not ordered n-tuples; the conserve rule of the rewrite system produces automorphic alphabets; generally, (X, X*) = (X*, X)] ((+) bias reflecting the handedness of nature) 0 ⇒ R RR ⇒ (R, R*) R(R, R*) → (R, R*) R*(R, R*) → (R*, R) (R, R*) (R, R*) ⇒ (R, R*, A, A*) R(R, R*, A, A*) → (R, R*, A, A*) R*(R, R*, A, A*) → (R, R*, A, A*) A(R, R*, A, A*) → (R, R*, A, A*) A*(R, R*, A, A*) → (R, R*, A, A*) (R, R*, A, A*)(R, R*, A, A*) ⇒ (R, R*, A, A*, B, B*, AB, AB*) and so on	**P_{URS} using the finite alphabet** (0) create first new symbol (assume a symbol ≠ ϵ): ϵ → a (1) conserve first zero totality: a → ab (2) conserve zero totality: ab→ aab (2) conserve zero totality: ab → bab (3) create a new symbol: abab → abacbc (3) create a new symbol: baab→ baacbc
	Application of P_{URS} using the finite alphabet (0) create ϵ → a (1) conserve a → ab (2.1) conserve positive ab→ aab OR (2.1) conserve negative ab→ bab (2.2) negative substitution into conserve positive a(**ab**)→ a(**bab**) OR (2.2) positive substitution into conserve negative b(**ab**) → b(**aab**) (3) create from negative substitution into conserve positive abab→ abacbc (3) create from positive substitution into conserve negative baab → baacbc
	P_{URS} using the infinite alphabet (using conserve positive and negative substitution into conserve positive rules) (0) ϵ → a (1) → (ab) (2.1) → (a(ab)) (2.2) → (a(bab)) (3) → (ab)(ac_1bc_1)

	(2.1) → (a(ab))(ac$_1$bc$_1$) (2.2) → (a(bab))ac$_1$bc$_1$ (3) → (ab)(ac$_1$bc$_1$)(ae$_1$be$_1$) (2.1) → (a(ab))(ac$_1$bc$_1$)(ae$_1$be$_1$) (2.2) → (a(bab))(ac$_1$bc$_1$)(ae$_1$be$_1$) (3) → (ab)(ac$_1$bc$_1$)(ae$_1$be$_1$)(ac$_1$e$_1$bc$_1$e$_1$) and so on

In the table, the left-hand column lists the replacement rules of the original Rowlands-Diaz universal rewrite system, as discussed in the Introduction (version 1.0). Each successive alphabet is represented by a string of letters, R, A, B, C, etc., with their conjugates R*, A*, B*, C* ..., which also generate composite structures, such as AB, AC, ABC, ABC*… The symbols are always taken in pairs with their conjugates, ensuring totality zero at all levels. The first complete alphabet is (R,R*), the successive alphabets are then (R,R*, A,A*); (R,R*, A,A*, B,B*, AB,AB*); (R,R*, A,A*, B,B*, AB,AB*, C,C*, AC,AC*, BC,BC*, ABC,ABC*); and so on. Each new alphabet requires the creation of a new symbol and its conjugate: (R,R*); (A,A*); (B,B*); (C,C*) … The creation of a new alphabet is symbolized by ⇒, so (R,R*)(R,R*) ⇒ (R,R*, A,A*) means the first alphabet leads to the creation of the second …

The composition of the new alphabet is determined by the simultaneous application of an anticipatory conservation property, symbolized by →, in which the concatenation of the new alphabet with each of its terms leads to an automorphism with the same terms rearranged in a new order. As the order of terms makes no difference to the alphabet as a totality, the automorphisms are versions of the same alphabet. So (R,R*, A,A*) successively concatenated with R, R*, A and A* becomes (R,R*, A,A*); (R*,R, A*,A); (A,A*, R*,R) and (A*,A, R,R*). Rather than 'multiplication', we should see the concatenation as a system of *replacement*, using a set of replacement rules. While R takes on the form of an identity element and R* switches terms with their conjugates, A, B, C, D, etc concatenate with themselves to produce R*, not R, and the concatenation of AB with itself is 'anticommutative', again producing R* rather than R. These results are only possible because each symbol is always paired with its conjugate and the order between these is not significant in absolute terms. The create-conserve combination is tightly controlled, each term having unique properties forced upon them. The successive pairs (A, B), (C, D), etc. are anticommutative within the pairing, but commutative everywhere else. This is reflected in the machine descriptions where the pairings are referred to as (R(2u+1), R(2u+2)).

For the rewrite rules adapted for language generation (2.0), we use a different symbolism, with X(0) replacing R, and X(1), X(2), X(3), etc. replacing A, B, C, etc., and # replacing *. The commas are also removed. In this case we read operations from right to left, rather than left to right. To keep continuity in the process, the rewrite rules output from one replacement rule becomes the input for the next, i.e. working out output from rules rather than rules from output, forward rather than reverse engineering. This is to set up the rewrite rules as a practical process, and we recommend the reader, following Turing's original conception of his machine, to work out the computations by hand using pencil and paper. To set up the process, we have to read the operations in 1.0 from right to left. In the first section of the right-hand column, (0) and (1) start the process, leading to (2). (2) and (3) are the conjugate conserve rules X(0)X(0)$^{\#}$ → X(0)**X(0)X$^{\#}$(0)** and X(0)X$^{\#}$(0) → **X$^{\#}$(0)X(0)X$^{\#}$(0)** written in reverse. The key step now is to now to insert the output of (3) in bold into that part of the output of (2) (also in bold) which was the same as the input, that is **X$^{\#}$(0)X(0)X$^{\#}$(0)** in (3) replaces **X(0)X$^{\#}$(0)** in (2), producing X(0)**X$^{\#}$(0)X(0)X$^{\#}$(0)**. This allows the creation of the next alphabet to take place with the right number of terms, with the index shifting by 1 to give X(0)**X$^{\#}$(0)X(1)X$^{\#}$(1)**, in what is identified as step (4). Exactly the same process continues into the next iteration of an alphabet with the procedures following in the same order.

5. The global language of the Universal Rewrite System (URS)

From a global interpretation, the language of the URS, LURS, is the set of words = $\{\epsilon^{2^n}: n \geq 0\}$. The 0 symbol is a Clifford algebra totality, made up of + and – versions of each symbol. In the following table, the symbol '=' represents the number of algebraic terms (+ and –) within each 0 totality. For each order n, the number of symbols per order in the URS = 2^n, and the number of words per order in the URS = 2^{n-1}.

n	n – 1	$0 = 2^{n-1}$	0^{2^n}	ϵ^{2^n}
0			$0^{2^0} = 0 = \pm 1$	$\epsilon^{2^0} = ab$
1	0	$0 = \pm 1$	$0^{2^1} = 00 = \pm 1, \pm \boldsymbol{i}_1$	$\epsilon^{2^1} = abac_1bc_1$
2	1	$00 = \pm 1, \pm \boldsymbol{i}_1$	$0^{2^2} = 0000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1$	$\epsilon^{2^2} =$ $abac_1bc_1ae_1be_1ac_1$ $e_1bc_1e_1$
3	2	$0000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1$	$0^{2^3} = 00000000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$	$\epsilon^{2^3} =$ $abac_1bc_1ae_1be_1$ $ac_1e_1bc_1e_1ac_2bc_2a$ $c_2c_1bc_2c_1ac_2e_1bc_2$ $e_1ac_2c_1e_1bc_2$ c_1e_1
4	3	$00000000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$	$0^{2^4} = 0000000000000000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$	$\epsilon^{2^4} =$ $abac_1bc_1ae_1be_1ac_1$ $e_1bc_1e_1ac_2bc_2ac_2c_1$ $bc_2c_1ac_2e_1bc_2e_1ac_2$ $c_1e_1bc_2c_1e_1ae_2be_2a$ $e_2c_1be_2c_1ae_2e_1be_2$ $e_1ae_2c_1e_1be_2c_1$ e_1a $e_2c_2be_2c_2ae_2$ $c_2c_1be_2c_2c_1ae_2c_2e_1$ $be_2c_2e_1ae_2c_2c_1e_1b$ $e_2c_2c_1e_1$
5	4	$0000000000000000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$	$0^{2^5} = 00000000000000000000000000000000 = \pm 1, \pm \boldsymbol{i}_1, \pm \boldsymbol{j}_1, \pm \boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_2, \pm \boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm\boldsymbol{j}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1, \pm \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_3, \pm \boldsymbol{i}_3\boldsymbol{i}_1, \pm \boldsymbol{i}_3\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_1\boldsymbol{j}_1, \pm \boldsymbol{i}_3\boldsymbol{i}_2, \pm \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{i}_1,$	$\epsilon^{2^5} =$ $abac_1bc_1ae_1be_1ac_1$ $e_1bc_1e_1$ $ac_2bc_2ac_2c_1bc_2c_1a$ $c_2e_1bc_2e_1ac_2c_1e_1b$ $c_2c_1e_1ae_2be_2ae$ $_2c_1be_2c_1ae_2e_1be_2e_1$ ae_2 $c_1e_1be_2c_1e_1ae_2$ c_2b $e_2c_2ae_2$ $c_2c_1be_2c_2c_1ae_2$ $c_2e_1be_2c_2e_1ae_2c_2c_1$ $e_1be_2c_2c_1e_1ac_3$ $bc_3ac_3c_1bc_3c_1ac_3e_1$ bc_3

			$\pm\ \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{j}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{j}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_1\boldsymbol{j}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2$, $\pm\ \boldsymbol{i}_3\ \boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{j}_1$, $\pm\ \boldsymbol{i}_3\boldsymbol{j}_2\boldsymbol{i}_2\boldsymbol{i}_1\boldsymbol{j}_1$	$e_1ac_3c_1e_1bc_3c_1e_1a$ $c_3c_2bc_3c_2ac_3c_2c_1b$ $c_3c_2c_1ac_3c_2e_1bc_3c_2$ $e_1ac_3c_2c_1e_1bc_3c_2c_1e_1ac_3e_2bc_3e_2ac_3$ $e_2c_1bc_3e_2c_1ac_3e_2e_1bc_3e_2e_1ac_3e_2c_1$ $e_1bc_3e_2c_1e_1ac_3e_2c_2bc_3e_2c_2ac_3e_2c_2c_1bc_3e_2c_2c_1ac_3e_2c_2e_1bc_3e_2c_2e_1ac_3e_2c_2c_1e_1bc_3e_2c_2c_1e_1$
…	…	…	…	…

6. Language generation

To produce sentences in a particular language requires knowledge of the rules of sentence formation. Sentient intelligence possesses this ability and computational devices simulate this ability. But the universe itself seems to exist on a simple fundamental meta-pattern (never repeated) that can be formulated as a language with appropriate grammar rules. This universal pattern was discovered by Peter Rowlands and Bernard Diaz, together with the initial form of the grammar rules [1-6]. A formulation of the grammar rules of the universal rewrite system (URS) adapted to formal language theory engineered by Sydney Rowlands, assisted by Peter Rowlands, faithfully reproduces the same output as the original formulation of the grammar rules. However, this adapted and re-engineered version of the grammar rules introduces the universal rewrite system to the subject of formal language analysis and computation theory, which might have applications for computerized simulation of the physical laws operating in the universe, in addition to technological consequences. The generative grammar is given in this section, followed by the linear bounded automata in section 3. It is significant that this procedure only accepts zero words, as required by the universal rewrite system, but the simulated process (unlike the natural one) can be terminated by a halting condition set by the user.

6.1 The Universal Rewrite System (URS) as a Generative Grammar

In conventional language theory, the alphabets that compose the definition of generative grammars are finite sets of symbols. A grammar G is a tuple, G = (V_N, V_T, P, S), where V_N is a finite set of non-terminal symbols, V_T is a finite set of terminal symbols, P is a set of production rules and S is the start symbol. To treat the URS as a computational language, the URS needs to be modelled as a language generated by a very specific type of grammar. To compute the sets of words the grammar this language generates requires a machine with the minimum power capabilities of linear bounded automata (lba) at the least.

To bring out the progressive repetition structures of the URS in a computational manner, the grammar of the URS needs to contain finite alphabets. The way to accomplish this is to let every symbol in the URS be labelled with one of two types of index numbers q_1 or q_2 that both vary as functions of the variable u, $q_1 = g(u(n-1)) = 2(u(n-1)) + 1 = g \circ f = g(f(n))$ and $q_2 = s(u(n-1)) = 2(u(n-1)) + 2 = s \circ f = s(f(n))$. If we leave these indices q_1 and q_2 unevaluated, then we can reduce the infinite number of URS symbols to four basic types of symbols, which as mentioned earlier are really unevaluated functions of u. It will be shown later in this section that the variable u is itself a function of the exponent n of the order 2^n of the URS in the form u(n – 1) = f(n) $= \frac{1}{4}(2(n-1)-3-(-1)^{n-1})$.

In an analogous fashion, if the generating rules of the URS are treated in the context of the definition of a grammar, that grammar would be formulated as a tuple G(URS) = (V_N(URS), V_T(URS), P(URS), S) where V_N(URS) is a finite alphabet of nonterminal symbols, V_T(URS) is a finite alphabet of terminal symbols, P(URS) is the set of production or replacement rules, and S is the start symbol, which is a symbol independent of the symbols in the finite sets of nonterminal and terminal alphabets. For the URS as defined by Nature, V_T(URS) does not exist because the rewrite system doesn't terminate, though we can include it to stop arbitrarily for technical convenience.

Rules (2) and (3) split into positive and negative versions of each other. Both options lead to the same result but either branch must be rule replaced in a certain sequence consistently and independently of each other to give the same universal rewrite system. (2.1) conserve positive must be replaced by (2.2) negative substitution and vice versa. To apply the rules to generate the universal rewrite system. In (2.2), we are using a substitution rule (highlighted in bold). This *anticipatory process* is necessary to create the spaces for the symbols coming, in the same way as we create the space, by anticipation, for the expanded algebra in the basic URS. In this way it explains how the create process actually 'calculates' the size of the new algebra which emerges.

The language generated by the grammar URS, L(URS) is the set of words defined at the end of Section 2, assuming the symbols operate as though they are quaternions following the quaternion multiplication rules where X_0 is the identity, $X_0^{\#}$ minus identity, X_{2u+1} quaternion ***i***, $X_{2u+1}^{\#}$ quaternion –***i***, X_{2u+2} quaternion ***j***, $X_{2u+2}^{\#}$ quaternion –***j***, $X_{2u+1}X_{2u+2}$ quaternion ***ij***, $X_{2u+1}^{\#}\ {}^{X}{}_{2u+2}{}^{\#}$ quaternion –***ij***. Instead of continuing the product to infinity, as we propose that Nature does in principle, we can choose to adapt this product to a real machine by terminating the product at some particular power *n* of the order 2^n in the URS, with the help of a function of *n*: u = f(*n*-1). u(*n*-1) numerically labels each conjugate pair symbol, using 2u+1 (= ***i*** in a quaternion representation) in power *n* and 2u + 2 (= ***j*** in a quaternion representation) in power *n*+1 with the same number u(*n*-1), excluding the conjugate unit pair R(0), $R^{\#}$(0) at power *n* = 1 where u(*n*-1) is undefined because there does not exist a set of anticommutative cycles at the power *n* = 1 (see table below).

$$u(n-1) = \frac{1}{4}(2(n-1)-3-(-1)^{n-1})$$

where u(*n*-1) ranges over {0, 1, 2, 3, …} and *n* ranges over {1, 2, 3, …}.

Additionally, for every power *n* in each order 2^n of the URS there exist (*n* — 1)/2 anticommutative sets that alternate between complete and incomplete sets. These correspond, respectively, to *n* odd values, with 2u+1 (quaternion ***i***) combined with 2u + 2 (quaternion ***j***) (complete), and *n* even values with 2u + 2 (quaternion ***i***) only (incomplete). The process is driven by the value of *n* chosen, which then produces a value of u(*n*-1), according to the formula.

n = even value → u(*n*-1) → X_{2u+1} (quaternion ***i***)

same label at next level *n* + 1 unique label

n = odd value →. u(*n*-1) → X_{2u+2} (quaternion ***j***)

same label at level *n* unique label

In the rewrite system, at the most fundamental level, anticommutativity is the source of all discreteness and, ultimately, uniqueness. For every power *n* in each order 2^n of the URS there exist (*n* – 1)/2 anti-commutative sets that alternate between complete and incomplete sets. These correspond, respectively, to n odd values, with 2u+1 (quaternion $\boldsymbol{i}$) combined with 2u + 2 (quaternion $\boldsymbol{j}$) (complete), and n even values with 2u + 2 (quaternion $\boldsymbol{i}$) only (incomplete). The process is driven by the value of n chosen, which then produces a value of u(n – 1), according to the formula. Here is a table explaining the progression of the URS through increasing values of n:

Power of URS n	Order of URS 2^n	Quaternion $\boldsymbol{i}$ and Quaternion $\boldsymbol{j}$ same label u(n-1)	Quaternion $\boldsymbol{i}$ and Quaternion $\boldsymbol{j}$ same canonical labelling u(n-1) + 1 [*Zero to Infinity* page 11]	Quaternion $\boldsymbol{i}$ unique label $R_{2u(n-1)+1}$	Quaternion $\boldsymbol{j}$ unique label $R_{2u(n-1)+2}$	Number of anticommutative cycles at n
1	2	No quaternions at level n = 1	No quaternions at level n = 1	No quaternions at level n = 1	No quaternions at level n = 1	0
2	4	0	1	1		0.5
3	8	0	1		2	1
4	16	1	2	3		1.5
5	32	1	2		4	2
6	64	2	3	5		2.5
7	128	2	3		6	3
8	256	3	4	7		3.5
9	512	3	4		8	4
10	1024	4	5	9		4.5
11	2048	4	5		10	5

Index numbers on the symbols in the rewrite system represent different levels of the ‘repeated units’ become the numbers in the exponent specifying the number of concatenated copies of the ‘repeated units’ in regular expressions and the next alphabet (zero totality) consumes the previous alphabet (zero totality).

6.2 The role of anticommutativity

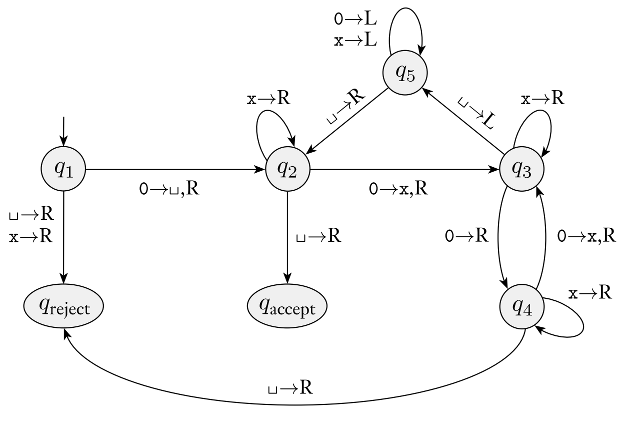
In the rewrite system, at the most fundamental level, anticommutativity is the source of all discreteness and, ultimately, uniqueness. For every power n in each order 2^n of the URS there exist (n – 1)/2 anti-commutative sets that alternate between complete and incomplete sets. These correspond, respectively, to n odd values, with 2u+1 (quaternion $\boldsymbol{i}$) combined with 2u + 2 (quaternion $\boldsymbol{j}$) (complete), and n even values with 2u + 2 (quaternion $\boldsymbol{i}$) only (incomplete). The process is driven by the value of n chosen, which then produces a value of u(n – 1), according to the formula.
This lends itself to a tensor product calculation in a computer algebra system with a tensor product capability (as in the Mathematica code used above). All variations of the code can be reduced in a program to a structure of this form, whether or not they are intrinsically algebraic. Without a computer algebraic system, the coding would be much more difficult.

It is significant, however, that the ‘tensor product 'formalism produced by the Mathematica program is not *restricted* to the anticommutative Clifford algebra. At the fundamental level [2-8], this occurs because it ensures the uniqueness of all states in the system, but, at a higher level, there may be cases in which the uniqueness is already guaranteed, and the anticommutative aspect is not required. This occurs, for example, when we have a system with multiple $n = 6$ fermionic states produced by the Dirac equation, in which the wavefunction is a nilpotent or square root of zero, of the form ($\pm\, \boldsymbol{i}kE \pm \boldsymbol{i}\mathbf{p} + \boldsymbol{j}m$) [6]; each of these is already guaranteed as unique by the nilpotency (which is equivalent to Pauli exclusion), and so the production of new symbols by the rewrite structure as new nilpotent fermions are added to the system does not need symbols that are anticommutative. This, in effect, creates the Hilbert space in which the characteristics of the orthogonal components are created by the nilpotent wavefunctions and not those of the coefficients needed to extend the rewriting to higher levels. So, we find that systems made up of many quantum states are described by tensor products with 2^n terms, and that, at this level, the entire quantum universe has a perfect description as a rewrite system defined by the structure revealed in the Mathematica program. Physically, this means that every state is connected to every other state, and that the entire quantum universe is in some way entangled. Analogues also occur with systems described in classical terms.

7. The machines that recognize the URS

The Turing Machine for the language $L_{URS} = \{0^{2^n} : n \geq 0\}$ is the universal operating system for the URS that is already extant in the computer science literature.

Transition Function δ for $L_{URS} = \{0^{2^n} : n \geq 0\}$			
State q	0	x	⊔
q_1	(q_2, ⊔, R)	(q_{reject}, x, R)	(q_{reject}, ⊔, R)
q_2	(q_3, x, R)	(q_2, x, R)	(q_{accept}, ⊔, R)
q_3	(q_4, 0, R)	(q_3, x, R)	(q_5, ⊔, R)
q_4	(q_3, x, R)	(q_4, x, R)	(q_{reject}, ⊔, R)
q_5	(q_5, 0, L)	(q_5, x, L)	(q_2, ⊔, R)



(Diagram from Sipser [33])

7.1 Recognizing languages

To understand sentences in a particular language requires knowledge of the rules of sentence formation. Sentient intelligence possesses this ability and computational devices simulate this ability. In the Rowlands-Diaz Rewrite System there do not exist choices – the URS recognized by a Turing Machine over an infinite alphabet moves in only one direction. The repeating part of the pattern of the Rowlands-Diaz Rewrite System is caused by anticommutativity which involves the symmetries of the number 3 or 3-dimensionality. This is the reason for the existence of the repetition in the infinite sequence of square roots of –1.

The transition tables for the machines in this work were worked out using two related methods: the suffix function used to determine the values of the transition function borrowed from the finite automata algorithm for the string matching problem and the pseudocode used to run code to compute the transition function [35]. Both methods can be used to cross check the accuracy of the values computed in the transition tables. Using the suffix function method, all the zero totalities of the rewrite system are empty strings with unique patterns. As such these patterns are subject to sorting functions according to the positions of the characters that compose the empty strings. Therefore, these empty strings can be sorted according to the methods of sorting arrays. One such sorting function that is particularly relevant is the suffix function.

7.2 Some notation and terminology

To construct a transition function table for a machine recognizing the 0 words, we find that a particular string-matching algorithm is an exact fit to our requirements. Part of this process involves the suffix function, that is, we find the longest *suffix* that matches the *prefix* of the pattern. A string w is a prefix of a string x, denoted $w \sqsubset x$, if $x = wy$ for some string $y \in \Sigma^*$, where Σ^* is the set of all finite-length strings over the alphabet Σ. A string w is a suffix of a string x, denoted $w \sqsupset x$, if $x = yw$ for some string $y \in \Sigma^*$.[35]

Given a string x and a pattern P[1.. m] from the same alphabet Σ, the suffix function $\sigma(x)$ maps the characters from Σ^*, which is the set of all the finite length words formed from the given alphabet Σ, to the indexed positions {0, 1, …., m} of the given pattern P[1.. m] such that $\sigma(x)$ is the length of the longest prefix of P[1… m] that is also a suffix of x:

$$\sigma(x) = \max\{k: P_k \sqsupset x\}$$

7.3 Computing the transition function

The following algorithm [35], computes the transition function δ from a given pattern *P*[1..*m*].

```
COMPUTE-TRANSITION-FUNCTION($P$, $\Sigma$)
    3.  $m = P.length$
    4.  for $q = 0$ to $m$
    5.      for each character $a \in \Sigma$
    6.          $k = \min(m + 1, q + 2)$
    7.          repeat
    8.              $k = k - 1$
    9.          until $P_k \sqsupset P_q a$
    10.         $\delta(q, a) = k$
    11. return $\delta$
```

We interpret this algorithm as follows

- The values for q are [0, …, m] = m + 1.
- The values for k are [0, …, q + 1] = q + 2.
- The length of P_k is $|P_k| = k$ and the length of $P_q a$ is $|P_q a| = q + 1$.
- The condition $P_k \sqsupset P_q a$ implies that $|P_k| \leq |P_q a|$ which implies $k \leq q + 1$.
- To start the algorithm, for each state q, choose k = min(m + 1, q + 2). Then repeatedly decrease k by 1 and assign k = k – 1 until the condition $k \leq q + 1$ is true.
  Then assign $\delta(q, a) = k$.

State q	Allowable values of k for each state q, $k \leq q + 1$

0	$0, 1 \leq 1$
1	$0, 1, 2 \leq 2$
2	$0, 1, 2, 3 \leq 3$
3	$0, 1, 2, 3, 4 \leq 4$
…	…
m	$0, 1, \ldots, m - 2, m - 1, m, m + 1 \leq m + 1$

The only difference between the URS defined using the finite alphabet versus the infinite alphabet is that the alphabet defined for the URS using the finite alphabet has only four types of characters. Since the URS is an infinitely sized language, the finite alphabet form reveals the fundamental anticommutative cyclical pattern that infinitely repeats as the URS develops. The URS defined using the infinite alphabet reveals the unending development of the URS using an infinite number of symbols. The URS defined using the infinite alphabet shows the infinite number of anticommutative cycles that compose the URS increases as the URS develops.

7.4 Machines for the URS defined iteratively using the infinite alphabet
The 0 for n = 6 is made of the following order 64 word:
$\epsilon_6 =$
$abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2c_1e_1bc_2c_1e_1ae_2be_2ae_2c_1be_2c_1ae_2e_1be_2e_1ae_2c_1e_1be_2c_1$
$e_1ae_2c_2be_2c_2ae_2c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2e_1ae_2c_2c_1e_1be_2c_2c_1e_1ac_3bc_3ac_3c_1bc_3c_1ac_3e_1bc_3e_1ac_3c_1e_1bc_3c_1e_1ac_3c_2$
$bc_3c_2ac_3c_2c_1bc_3c_2c_1ac_3c_2e_1bc_3c_2e_1ac_3c_2c_1e_1bc_3c_2c_1e_1ac_3e_2bc_3e_2ac_3e_2c_1bc_3e_2c_1ac_3e_2e_1bc_3e_2e_1ac_3e_2c_1e_1bc_3e_2c$
$_1e_1ac_3e_2c_2bc_3e_2c_2ac_3e_2c_2c_1bc_3e_2c_2c_1ac_3e_2c_2e_1bc_3e_2c_2e_1ac_3e_2c_2c_1e_1bc_3e_2c_2c_1e_1$

We now apply the suffix function to give the complete instruction table for the transition function for the machine that will recognize this word.

Pattern $P = \epsilon_6$
$\delta(0, a) = 1$, since $P_0a = \epsilon a$ and $\sigma(P_0a) = \sigma(\epsilon a) = 1$
$\delta(0, b) = 0$, since $P_0b = \epsilon b$ and $\sigma(P_0b) = \sigma(\epsilon b) = 0$
$\delta(0, c_1) = 0$, since $P_0c_1 = \epsilon c_1$ and $\sigma(P_0c_1) = \sigma(\epsilon c_1) = 0$
$\delta(0, e_1) = 0$, since $P_0e_1 = \epsilon e_1$ and $\sigma(P_0e_1) = \sigma(\epsilon e_1) = 0$
$\delta(0, c_2) = 0$, since $P_0c_2 = \epsilon c_2$ and $\sigma(P_0c_2) = \sigma(\epsilon c_2) = 0$
$\delta(0, e_2) = 0$, since $P_0e_2 = \epsilon e_2$ and $\sigma(P_0e_2) = \sigma(\epsilon e_2) = 0$
$\delta(0, c_3) = 0$, since $P_0c_3 = \epsilon c_3$ and $\sigma(P_0c_3) = \sigma(\epsilon c_3) = 0$
$\delta(1, a) = 1$, since $P_1a = aa$ and $\sigma(P_1a) = \sigma(aa) = 1$
$\delta(1, b) = 2$, since $P_1b = ab$ and $\sigma(P_1b) = \sigma(ab) = 2$
$\delta(1, c_1) = 0$, since $P_1c_1 = ac_1$ and $\sigma(P_1c_1) = \sigma(ac_1) = 0$
$\delta(1, e_1) = 0$, since $P_1e_1 = ae_1$ and $\sigma(P_1e_1) = \sigma(ae_1) = 0$
$\delta(1, c_2) = 0$, since $P_1c_2 = ac_2$ and $\sigma(P_1c_2) = \sigma(ac_2) = 0$
$\delta(1, e_2) = 0$, since $P_1e_2 = ae_2$ and $\sigma(P_1e_2) = \sigma(ae_2) = 0$
$\delta(1, c_3) = 0$, since $P_1c_3 = ac_3$ and $\sigma(P_1c_3) = \sigma(ac_3) = 0$
$\delta(2, a) = 3$, since $P_2a = aba$ and $\sigma(P_2a) = \sigma(aba) = 3$
$\delta(2, b) = 0$, since $P_2b = abb$ and $\sigma(P_2b) = \sigma(abb) = 0$
$\delta(2, c_1) = 0$, since $P_2c_1 = abc_1$ and $\sigma(P_2c_1) = \sigma(abc_1) = 0$
$\delta(2, e_1) = 0$, since $P_2e_1 = abe_1$ and $\sigma(P_2e_1) = \sigma(abe_1) = 0$
$\delta(2, c_2) = 0$, since $P_2c_2 = abc_2$ and $\sigma(P_2c_2) = \sigma(abc_2) = 0$
$\delta(2, e_2) = 0$, since $P_2e_2 = abe_2$ and $\sigma(P_2e_2) = \sigma(abe_2) = 0$

$\delta(2, c_3) = 0$, since $P_2c_3 = abc_3$ and $\sigma(P_2c_3) = \sigma(abc_3) = 0$

$\delta(3, a) = 1$, since $P_3a = abaa$ and $\sigma(P_3a) = \sigma(abaa) = 1$

$\delta(3, b) = 2$, since $P_3b = abab$ and $\sigma(P_3b) = \sigma(abab) = 2$

$\delta(3, c_1) = 4$, since $P_3c_1 = abac_1$ and $\sigma(P_3c_1) = \sigma(abac_1) = 4$

$\delta(3, e_1) = 0$, since $P_3e_1 = abae_1$ and $\sigma(P_3e_1) = \sigma(abae_1) = 0$

$\delta(3, c_2) = 0$, since $P_3c_2 = abac_2$ and $\sigma(P_3c_2) = \sigma(abac_2) = 0$

$\delta(3, e_2) = 0$, since $P_3e_2 = abae_2$ and $\sigma(P_3e_2) = \sigma(abae_2) = 0$

$\delta(3, c_3) = 0$, since $P_3c_3 = abac_3$ and $\sigma(P_3c_3) = \sigma(abac_3) = 0$

$\delta(4, a) = 1$, since $P_4a = abac_1a$ and $\sigma(P_4a) = \sigma(abac_1a) = 1$

$\delta(4, b) = 0$, since $P_4b = abac_1b$ and $\sigma(P_4b) = \sigma(abac_1b) = 5$

$\delta(4, c_1) = 0$, since $P_4c_1 = abac_1c_1$ and $\sigma(P_4c_1) = \sigma(abac_1c_1) = 0$

$\delta(4, e_1) = 0$, since $P_4e_1 = abac_1e_1$ and $\sigma(P_4e_1) = \sigma(abac_1e_1) = 0$

$\delta(4, c_2) = 0$, since $P_4c_2 = abac_1c_2$ and $\sigma(P_4c_2) = \sigma(abac_1c_2) = 0$

$\delta(4, e_2) = 0$, since $P_4e_2 = abac_1e_2$ and $\sigma(P_4e_2) = \sigma(abac_1e_2) = 0$

$\delta(4, c_3) = 0$, since $P_4c_3 = abac_1c_3$ and $\sigma(P_4c_3) = \sigma(abac_1c_3) = 0$

$\delta(5, a) = 1$, since $P_5a = abac_1ba$ and $\sigma(P_5a) = \sigma(abac_1ba) = 1$

$\delta(5, b) = 0$, since $P_5b = abac_1bb$ and $\sigma(P_5b) = \sigma(abac_1bb) = 0$

$\delta(5, c_1) = 6$, since $P_5c_1 = abac_1bc_1$ and $\sigma(P_5c_1) = \sigma(abac_1bc_1) = 6$

$\delta(5, e_1) = 0$, since $P_5e_1 = abac_1be_1$ and $\sigma(P_5e_1) = \sigma(abac_1be_1) = 0$

$\delta(5, c_2) = 0$, since $P_5c_2 = abac_1bc_2$ and $\sigma(P_5c_2) = \sigma(abac_1bc_2) = 0$

$\delta(5, e_2) = 0$, since $P_5e_2 = abac_1be_2$ and $\sigma(P_5e_2) = \sigma(abac_1be_2) = 0$

$\delta(5, c_3) = 0$, since $P_5c_3 = abac_1bc_3$ and $\sigma(P_5c_3) = \sigma(abac_1bc_3) = 0$

$\delta(6, a) = 7$, since $P_6a = abac_1bc_1a$ and $\sigma(P_6a) = \sigma(abac_1bc_1a) = 7$

$\delta(6, b) = 0$, since $P_6b = abac_1bc_1b$ and $\sigma(P_6b) = \sigma(abac_1bc_1b) = 0$

$\delta(6, c_1) = 0$, since $P_6c_1 = abac_1bc_1c_1$ and $\sigma(P_6c_1) = \sigma(abac_1bc_1c_1) = 0$

$\delta(6, e_1) = 0$, since $P_6e_1 = abac_1bc_1e_1$ and $\sigma(P_6e_1) = \sigma(abac_1bc_1e_1) = 0$

$\delta(6, c_2) = 0$, since $P_6c_2 = abac_1bc_1c_2$ and $\sigma(P_6c_2) = \sigma(abac_1bc_1c_2) = 0$

$\delta(6, e_2) = 0$, since $P_6e_2 = abac_1bc_1e_2$ and $\sigma(P_6e_2) = \sigma(abac_1bc_1e_2) = 0$

$\delta(6, c_3) = 0$, since $P_6c_3 = abac_1bc_1c_3$ and $\sigma(P_6c_3) = \sigma(abac_1bc_1c_3) = 0$

$\delta(7, a) = 1$, since $P_7a = abac_1bc_1aa$ and $\sigma(P_7a) = \sigma(abac_1bc_1aa) = 1$

$\delta(7, b) = 2$, since $P_7b = abac_1bc_1ab$ and $\sigma(P_7b) = \sigma(abac_1bc_1ab) = 2$

$\delta(7, c_1) = 0$, since $P_7c_1 = abac_1bc_1ac_1$ and $\sigma(P_7c_1) = \sigma(abac_1bc_1ac_1) = 0$

$\delta(7, e_1) = 8$, since $P_7e_1 = abac_1bc_1ae_1$ and $\sigma(P_7e_1) = \sigma(abac_1bc_1ae_1) = 8$

$\delta(7, c_2) = 0$, since $P_7c_2 = abac_1bc_1ac_2$ and $\sigma(P_7c_2) = \sigma(abac_1bc_1ac_2) = 0$

$\delta(7, e_2) = 0$, since $P_7e_2 = abac_1bc_1ae_2$ and $\sigma(P_7e_2) = \sigma(abac_1bc_1ae_2) = 0$

$\delta(7, c_3) = 0$, since $P_7c_3 = abac_1bc_1ac_3$ and $\sigma(P_7c_3) = \sigma(abac_1bc_1ac_3) = 0$

$\delta(8, a) = 1$, since $P_8a = abac_1bc_1ae_1a$ and $\sigma(P_8a) = \sigma(abac_1bc_1ae_1a) = 1$

$\delta(8, b) = 9$, since $P_8b = abac_1bc_1ae_1b$ and $\sigma(P_8b) = \sigma(abac_1bc_1ae_1b) = 9$

$\delta(8, c_1) = 0$, since $P_8c_1 = abac_1bc_1ae_1c_1$ and $\sigma(P_8c_1) = \sigma(abac_1bc_1ae_1c_1) = 0$

$\delta(8, e_1) = 0$, since $P_8e_1 = abac_1bc_1ae_1e_1$ and $\sigma(P_8e_1) = \sigma(abac_1bc_1ae_1e_1) = 0$

$\delta(8, c_2) = 0$, since $P_8c_2 = abac_1bc_1ae_1c_2$ and $\sigma(P_8c_2) = \sigma(abac_1bc_1ae_1c_2) = 0$

$\delta(8, e_2) = 0$, since $P_8e_2 = abac_1bc_1ae_1e_2$ and $\sigma(P_8e_2) = \sigma(abac_1bc_1ae_1e_2) = 0$

$\delta(8, c_3) = 0$, since $P_8c_3 = abac_1bc_1ae_1c_3$ and $\sigma(P_8c_3) = \sigma(abac_1bc_1ae_1c_3) = 0$

$\delta(9, a) = 1$, since $P_9a = abac_1bc_1ae_1ba$ and $\sigma(P_9a) = \sigma(abac_1bc_1ae_1ba) = 1$

$\delta(9, b) = 0$, since $P_9b = abac_1bc_1ae_1bb$ and $\sigma(P_9b) = \sigma(abac_1bc_1ae_1bb) = 0$

$\delta(9, c_1) = 0$, since $P_9c_1 = abac_1bc_1ae_1bc_1$ and $\sigma(P_9c_1) = \sigma(abac_1bc_1ae_1bc_1) = 0$

$\delta(9, e_1) = 10$, since $P_9e_1 = abac_1bc_1ae_1be_1$ and $\sigma(P_9e_1) = \sigma(abac_1bc_1ae_1be_1) = 10$

$\delta(9, c_2) = 0$, since $P_9c_2 = abac_1bc_1ae_1bc_2$ and $\sigma(P_9c_2) = \sigma(abac_1bc_1ae_1bc_2) = 0$
$\delta(9, e_2) = 0$, since $P_9e_2 = abac_1bc_1ae_1be_2$ and $\sigma(P_9e_2) = \sigma(abac_1bc_1ae_1be_2) = 0$
$\delta(9, c_3) = 0$, since $P_9c_3 = abac_1bc_1ae_1bc_3$ and $\sigma(P_9c_3) = \sigma(abac_1bc_1ae_1bc_3) = 0$
$\delta(10, a) = 11$, since $P_{10}a = abac_1bc_1ae_1be_1a$ and $\sigma(P_{10}a) = \sigma(abac_1bc_1ae_1be_1a) = 11$
$\delta(10, b) = 0$, since $P_{10}b = abac_1bc_1ae_1be_1b$ and $\sigma(P_{10}b) = \sigma(abac_1bc_1ae_1be_1b) = 0$
$\delta(10, c_1) = 0$, since $P_{10}c_1 = abac_1bc_1ae_1be_1c_1$ and $\sigma(P_{10}c_1) = \sigma(abac_1bc_1ae_1be_1c_1) = 0$
$\delta(10, e_1) = 0$, since $P_{10}e_1 = abac_1bc_1ae_1be_1e_1$ and $\sigma(P_{10}e_1) = \sigma(abac_1bc_1ae_1be_1e_1) = 0$
$\delta(10, c_2) = 0$, since $P_{10}c_2 = abac_1bc_1ae_1be_1c_2$ and $\sigma(P_{10}c_2) = \sigma(abac_1bc_1ae_1be_1c_2) = 0$
$\delta(10, e_2) = 0$, since $P_{10}e_2 = abac_1bc_1ae_1be_1e_2$ and $\sigma(P_{10}e_2) = \sigma(abac_1bc_1ae_1be_1e_2) = 0$
$\delta(10, c_3) = 0$, since $P_{10}c_3 = abac_1bc_1ae_1be_1c_3$ and $\sigma(P_{10}c_3) = \sigma(abac_1bc_1ae_1be_1c_3) = 0$
$\delta(11, a) = 1$, since $P_{11}a = abac_1bc_1ae_1be_1aa$ and $\sigma(P_{11}a) = \sigma(abac_1bc_1ae_1be_1aa) = 1$
$\delta(11, b) = 2$, since $P_{11}b = abac_1bc_1ae_1be_1ab$ and $\sigma(P_{11}b) = \sigma(abac_1bc_1ae_1be_1ab) = 2$
$\delta(11, c_1) = 12$, since $P_{11}c_1 = abac_1bc_1ae_1be_1ac_1$ and $\sigma(P_{11}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1) = 12$
$\delta(11, e_1) = 0$, since $P_{11}e_1 = abac_1bc_1ae_1be_1ae_1$ and $\sigma(P_{11}e_1) = \sigma(abac_1bc_1ae_1be_1ae_1) = 0$
$\delta(11, c_2) = 0$, since $P_{11}c_2 = abac_1bc_1ae_1be_1ac_2$ and $\sigma(P_{11}c_2) = \sigma(abac_1bc_1ae_1be_1ac_2) = 0$
$\delta(11, e_2) = 0$, since $P_{11}e_2 = abac_1bc_1ae_1be_1ae_2$ and $\sigma(P_{11}e_2) = \sigma(abac_1bc_1ae_1be_1ae_2) = 0$
$\delta(11, c_3) = 0$, since $P_{11}c_3 = abac_1bc_1ae_1be_1ac_3$ and $\sigma(P_{11}c_3) = \sigma(abac_1bc_1ae_1be_1ac_3) = 0$
$\delta(12, a) = 1$, since $P_{12}a = abac_1bc_1ae_1be_1ac_1a$ and $\sigma(P_{12}a) = \sigma(abac_1bc_1ae_1be_1ac_1a) = 1$
$\delta(12, b) = 0$, since $P_{12}b = abac_1bc_1ae_1be_1ac_1b$ and $\sigma(P_{12}b) = \sigma(abac_1bc_1ae_1be_1ac_1b) = 0$
$\delta(12, c_1) = 0$, since $P_{12}c_1 = abac_1bc_1ae_1be_1ac_1c_1$ and $\sigma(P_{12}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1c_1) = 0$
$\delta(12, e_1) = 13$, since $P_{12}e_1 = abac_1bc_1ae_1be_1ac_1e_1$ and $\sigma(P_{12}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1) = 13$
$\delta(12, c_2) = 0$, since $P_{12}c_2 = abac_1bc_1ae_1be_1ac_1c_2$ and $\sigma(P_{12}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1c_2) = 0$
$\delta(12, e_2) = 0$, since $P_{12}e_2 = abac_1bc_1ae_1be_1ac_1e_2$ and $\sigma(P_{12}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_2) = 0$
$\delta(12, c_3) = 0$, since $P_{12}c_3 = abac_1bc_1ae_1be_1ac_1c_3$ and $\sigma(P_{12}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1c_3) = 0$
$\delta(13, a) = 1$, since $P_{13}a = abac_1bc_1ae_1be_1ac_1e_1a$ and $\sigma(P_{13}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1a) = 1$
$\delta(13, b) = 14$, since $P_{13}b = abac_1bc_1ae_1be_1ac_1e_1b$ and $\sigma(P_{13}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1b) = 14$
$\delta(13, c_1) = 0$, since $P_{13}c_1 = abac_1bc_1ae_1be_1ac_1e_1c_1$ and $\sigma(P_{13}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1c_1) = 0$
$\delta(13, e_1) = 0$, since $P_{13}e_1 = abac_1bc_1ae_1be_1ac_1e_1e_1$ and $\sigma(P_{13}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1e_1) = 0$
$\delta(13, c_2) = 0$, since $P_{13}c_2 = abac_1bc_1ae_1be_1ac_1e_1c_2$ and $\sigma(P_{13}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1c_2) = 0$
$\delta(13, e_2) = 0$, since $P_{13}e_2 = abac_1bc_1ae_1be_1ac_1e_1e_2$ and $\sigma(P_{13}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1e_2) = 0$
$\delta(13, c_3) = 0$, since $P_{13}c_3 = abac_1bc_1ae_1be_1ac_1e_1c_3$ and $\sigma(P_{13}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1c_3) = 0$
$\delta(14, a) = 1$, since $P_{14}a = abac_1bc_1ae_1be_1ac_1e_1ba$ and $\sigma(P_{14}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1ba) = 1$
$\delta(14, b) = 0$, since $P_{14}b = abac_1bc_1ae_1be_1ac_1e_1bb$ and $\sigma(P_{14}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bb) = 0$
$\delta(14, c_1) = 15$, since $P_{14}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1$ and $\sigma(P_{14}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1) = 15$
$\delta(14, e_1) = 0$, since $P_{14}e_1 = abac_1bc_1ae_1be_1ac_1e_1be_1$ and $\sigma(P_{14}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1be_1) = 0$
$\delta(14, c_2) = 0$, since $P_{14}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_2$ and $\sigma(P_{14}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_2) = 0$
$\delta(14, e_2) = 0$, since $P_{14}e_2 = abac_1bc_1ae_1be_1ac_1e_1be_2$ and $\sigma(P_{14}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1be_2) = 0$
$\delta(14, c_3) = 0$, since $P_{14}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_3$ and $\sigma(P_{14}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_3) = 0$
$\delta(15, a) = 1$, since $P_{15}a = abac_1bc_1ae_1be_1ac_1e_1bc_1a$ and $\sigma(P_{15}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1a) = 1$
$\delta(15, b) = 0$, since $P_{15}b = abac_1bc_1ae_1be_1ac_1e_1bc_1b$ and $\sigma(P_{15}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1b) = 0$
$\delta(15, c_1) = 0$, since $P_{15}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1c_1$ and $\sigma(P_{15}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1c_1) = 0$
$\delta(15, e_1) = 16$, since $P_{15}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1$ and $\sigma(P_{15}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1) = 16$
$\delta(15, c_2) = 0$, since $P_{15}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1c_2$ and $\sigma(P_{15}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1c_2) = 0$
$\delta(15, e_2) = 0$, since $P_{15}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_2$ and $\sigma(P_{15}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_2) = 0$
$\delta(15, c_3) = 0$, since $P_{15}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1c_3$ and $\sigma(P_{15}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1c_3) = 0$
$\delta(16, a) = 17$, since $P_{16}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1a$ and $\sigma(P_{16}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1a) = 17$
$\delta(16, b) = 0$, since $P_{16}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1b$ and $\sigma(P_{16}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1b) = 0$

δ(16, c_1) = 0, since $P_{16}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_1$ and $\sigma(P_{16}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_1) = 0$
δ(16, e_1) = 0, since $P_{16}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1e_1$ and $\sigma(P_{16}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1e_1) = 0$
δ(16, c_2) = 0, since $P_{16}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_2$ and $\sigma(P_{16}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_2) = 0$
δ(16, e_2) = 0, since $P_{16}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1e_2$ and $\sigma(P_{16}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1e_2) = 0$
δ(16, c_3) = 0, since $P_{16}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_3$ and $\sigma(P_{16}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1c_3) = 0$
δ(17, a) = 1, since $P_{17}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1aa$ and $\sigma(P_{17}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1aa) = 1$
δ(17, b) = 2, since $P_{17}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ab$ and $\sigma(P_{17}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ab) = 2$
δ(17, c_1) = 0, since $P_{17}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_1$ and $\sigma(P_{17}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_1) =$
0
δ(17, e_1) = 0, since $P_{17}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ae_1$ and $\sigma(P_{17}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ae_1) =$
0
δ(17, c_2) = 18, since $P_{17}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2$ and $\sigma(P_{17}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2)$
= 18
δ(17, e_2) = 0, since $P_{17}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ae_2$ and $\sigma(P_{17}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ae_2) =$
0
δ(17, c_3) = 0, since $P_{17}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_3$ and $\sigma(P_{17}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_3) =$
0
δ(18, a) = 1, since $P_{18}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2a$ and $\sigma(P_{18}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2a) =$
1
δ(18, b) = 19, since $P_{18}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2b$ and $\sigma(P_{18}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2b)$
= 19
δ(18, c_1) = 0, since $P_{18}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_1$ and $\sigma(P_{18}c_1) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_1) = 0$
δ(18, e_1) = 0, since $P_{18}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2e_1$ and $\sigma(P_{18}e_1) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2e_1) = 0$
δ(18, c_2) = 0, since $P_{18}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_2$ and $\sigma(P_{18}c_2) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_2) = 0$
δ(18, e_2) = 0, since $P_{18}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2e_2$ and $\sigma(P_{18}e_2) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2e_2) = 0$
δ(18, c_3) = 0, since $P_{18}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_3$ and $\sigma(P_{18}c_3) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2c_3) = 0$
δ(19, a) = 1, since $P_{19}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2ba$ and $\sigma(P_{19}a) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2ba) = 1$
δ(19, b) = 0, since $P_{19}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bb$ and $\sigma(P_{19}b) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bb) = 0$
δ(19, c_1) = 0, since $P_{19}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_1$ and $\sigma(P_{19}c_1) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_1) = 0$
δ(19, e_1) = 0, since $P_{19}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2be_1$ and $\sigma(P_{19}e_1) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2be_1) = 0$
δ(19, c_2) = 20, since $P_{19}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2$ and $\sigma(P_{19}c_2) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2) = 20$
δ(19, e_2) = 0, since $P_{19}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2be_2$ and $\sigma(P_{19}e_2) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2be_2) = 0$
δ(19, c_3) = 0, since $P_{19}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_3$ and $\sigma(P_{19}c_3) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_3) = 0$
δ(20, a) = 21, since $P_{20}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2a$ and $\sigma(P_{20}a) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2a) = 21$
δ(20, b) = 0, since $P_{20}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2b$ and $\sigma(P_{20}b) =$
$\sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2b) = 0$

$\delta(20, c_1) = 0$, since $P_{20}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_1$ and $\sigma(P_{20}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_1) = 0$
$\delta(20, e_1) = 0$, since $P_{20}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2e_1$ and $\sigma(P_{20}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2e_1) = 0$
$\delta(20, c_2) = 0$, since $P_{20}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_2$ and $\sigma(P_{20}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_2) = 0$
$\delta(20, e_2) = 0$, since $P_{20}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2e_2$ and $\sigma(P_{20}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2e_2) = 0$
$\delta(20, c_3) = 0$, since $P_{20}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_3$ and $\sigma(P_{20}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2c_3) = 0$
$\delta(21, a) = 1$, since $P_{21}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2aa$ and $\sigma(P_{21}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2aa) = 1$
$\delta(21, b) = 2$, since $P_{21}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ab$ and $\sigma(P_{21}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ab) = 2$
$\delta(21, c_1) = 0$, since $P_{21}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_1$ and $\sigma(P_{21}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_1) = 0$
$\delta(21, e_1) = 0$, since $P_{21}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ae_1$ and $\sigma(P_{21}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ae_1) = 0$
$\delta(21, c_2) = 22$, since $P_{21}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2$ and $\sigma(P_{21}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2) = 22$
$\delta(21, e_2) = 0$, since $P_{21}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ae_2$ and $\sigma(P_{21}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ae_2) = 0$
$\delta(21, c_3) = 0$, since $P_{21}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_3$ and $\sigma(P_{21}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_3) = 0$
$\delta(22, a) = 1$, since $P_{22}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2a$ and $\sigma(P_{22}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2a) = 1$
$\delta(22, b) = 0$, since $P_{22}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2b$ and $\sigma(P_{22}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2b) = 0$
$\delta(22, c_1) = 23$, since $P_{22}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1$ and $\sigma(P_{22}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1) = 23$
$\delta(22, e_1) = 0$, since $P_{22}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2e_1$ and $\sigma(P_{22}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2e_1) = 0$
$\delta(22, c_2) = 0$, since $P_{22}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_2$ and $\sigma(P_{22}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_2) = 0$
$\delta(22, e_2) = 0$, since $P_{22}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2e_2$ and $\sigma(P_{22}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2e_2) = 0$
$\delta(22, c_3) = 0$, since $P_{22}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_3$ and $\sigma(P_{22}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_3) = 0$
$\delta(23, a) = 1$, since $P_{23}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1a$ and $\sigma(P_{23}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1a) = 1$
$\delta(23, b) = 24$, since $P_{23}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1b$ and $\sigma(P_{23}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1b) = 24$
$\delta(23, c_1) = 0$, since $P_{23}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_1$ and $\sigma(P_{23}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_1) = 0$
$\delta(23, e_1) = 0$, since $P_{23}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1e_1$ and $\sigma(P_{23}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1e_1) = 0$

$\delta(23, c_2) = 0$, since $P_{23}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_2$ and $\sigma(P_{23}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_2) = 0$
$\delta(23, e_2) = 0$, since $P_{23}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1e_2$ and $\sigma(P_{23}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1e_2) = 0$
$\delta(23, c_3) = 0$, since $P_{23}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_3$ and $\sigma(P_{23}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1c_3) = 0$
$\delta(24, a) = 1$, since $P_{24}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1ba$ and $\sigma(P_{24}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1ba) = 1$
$\delta(24, b) = 0$, since $P_{24}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bb$ and $\sigma(P_{24}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bb) = 0$
$\delta(24, c_1) = 0$, since $P_{24}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_1$ and $\sigma(P_{24}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_1) = 0$
$\delta(24, e_1) = 0$, since $P_{24}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1be_1$ and $\sigma(P_{24}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1be_1) = 0$
$\delta(24, c_2) = 25$, since $P_{24}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2$ and $\sigma(P_{24}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2) = 25$
$\delta(24, e_2) = 0$, since $P_{24}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1be_2$ and $\sigma(P_{24}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1be_2) = 0$
$\delta(24, c_3) = 0$, since $P_{24}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_3$ and $\sigma(P_{24}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_3) = 0$
$\delta(25, a) = 1$, since $P_{25}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2a$ and $\sigma(P_{25}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2a) = 1$
$\delta(25, b) = 0$, since $P_{25}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2b$ and $\sigma(P_{25}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2b) = 0$
$\delta(25, c_1) = 25$, since $P_{25}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1$ and $\sigma(P_{25}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1) = 25$
$\delta(25, e_1) = 0$, since $P_{25}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2e_1$ and $\sigma(P_{25}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2e_1) = 0$
$\delta(25, c_2) = 0$, since $P_{25}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_2$ and $\sigma(P_{25}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_2) = 0$
$\delta(25, e_2) = 0$, since $P_{25}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2e_2$ and $\sigma(P_{25}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2e_2) = 0$
$\delta(25, c_3) = 0$, since $P_{25}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_3$ and $\sigma(P_{25}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_3) = 0$
$\delta(26, a) = 27$, since $P_{26}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1a$ and $\sigma(P_{26}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1a) = 27$
$\delta(26, b) = 0$, since $P_{26}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1b$ and $\sigma(P_{26}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1b) = 0$
$\delta(26, c_1) = 0$, since $P_{26}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_1$ and $\sigma(P_{26}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_1) = 0$
$\delta(26, e_1) = 0$, since $P_{26}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1e_1$ and $\sigma(P_{26}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1e_1) = 0$
$\delta(26, c_2) = 0$, since $P_{26}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_2$ and $\sigma(P_{26}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_2) = 0$
$\delta(26, e_2) = 0$, since $P_{26}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1e_2$ and $\sigma(P_{26}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1e_2) = 0$
$\delta(26, c_3) = 0$, since $P_{26}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_3$ and $\sigma(P_{26}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1c_3) = 0$

$\delta(27, a) = 1$, since $P_{27}a = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1aa$ and $\sigma(P_{27}a) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1aa) = 1$
$\delta(27, b) = 2$, since $P_{27}b = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ab$ and $\sigma(P_{27}b) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ab) = 2$
$\delta(27, c_1) = 0$, since $P_{27}c_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_1$ and $\sigma(P_{27}c_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_1) = 0$
$\delta(27, e_1) = 0$, since $P_{27}e_1 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ae_1$ and $\sigma(P_{27}e_1) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ae_1) = 0$
$\delta(27, c_2) = 28$, since $P_{27}c_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2$ and $\sigma(P_{27}c_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2) = 28$
$\delta(27, e_2) = 0$, since $P_{27}e_2 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ae_2$ and $\sigma(P_{27}e_2) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ae_2) = 0$
$\delta(27, c_3) = 0$, since $P_{27}c_3 = abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_3$ and $\sigma(P_{27}c_3) = \sigma(abac_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_3) = 0$

and so on. The full instructions are in the table below:

Transition Function $\delta(q, a)$ for ϵ_6								
State q	Character a							Pattern P
	a	b	c_1	e_1	c_2	e_2	c_3	
$\rightarrow$ 0	1, a	0	0	0	0	0	0	a
1	1	2, b	0	0	0	0	0	b
◎2	3, a	0	0	0	0	0	0	a
3	1	2	4, c_1	0	0	0	0	c_1
4	1	5, b	0	0	0	0	0	b
5	1	0	6, c_1	0	0	0	0	c_1
◎6	7, a	0	0	0	0	0	0	a
7	1	2	0	8, e_1	0	0	0	e_1
8	1	9, b	0	0	0	0	0	b
9	1	0	0	10, e_1	0	0	0	e_1
10	11, a	0	0	0	0	0	0	a
11	1	2	12, c_1	0	0	0	0	c_1
12	1	0	0	13, e_1	0	0	0	e_1
13	1	14, b	0	0	0	0	0	b
14	1	0	15, c_1	0	0	0	0	c_1
15	1	0	0	16, e_1	0	0	0	e_1
◎16	17, a	0	0	0	0	0	0	a
17	1	2	0	0	18, c_2	0	0	c_2
18	1	19, b	0	0	0	0	0	b
19	1	0	0	0	20, c_2	0	0	c_2
20	21, a	0	0	0	0	0	0	a
21	1	2	0	0	22, c_2	0	0	c_2
22	1	0	23, c_1	0	0	0	0	c_1
23	1	24, b	0	0	0	0	0	b
24	1	0	0	0	25, c_2	0	0	c_2

25	1	0	25, c_1	0	0	0	0	c_1
26	27, a	0	0	0	0	0	0	a
27	1	2	0	0	28, c_2	0	0	c_2
28	1	0	0	29, e_1	0	0	0	e_1
29	1	30, b	0	0	0	0	0	b
30	1	0	0	0	31, c_2	0	0	c_2
31	1	0	0	32, e_1	0	0	0	e_1
32	33, a	0	0	0	0	0	0	a
33	1	2	0	0	34, c_2	0	0	c_2
34	1	0	35, c_1	0	0	0	0	c_1
35	1	0	0	36, e_1	0	0	0	e_1
36	1	37, b	0	0	0	0	0	b
37	1	0	0	0	38, c_2	0	0	c_2
38	1	0	39, c_1	0	0	0	0	c_1
39	1	0	0	40, e_1	0	0	0	e_1
◎40	41, a	0	0	0	0	0	0	a
41	1	2	0	0	0	42, e_2	0	e_2
42	1	43, b	0	0	0	0	0	b
43	1	0	0	0	0	44, e_2	0	e_2
44	45, a	0	0	0	0	0	0	a
45	1	2	0	0	0	46, e_2	0	e_2
46	1	0	47, c_1	0	0	0	0	c_1
47	1	48, b	0	0	0	0	0	b
48	1	0	0	0	0	49, e_2	0	e_2
49	1	0	50, c_1	0	0	0	0	c_1
50	51, a	0	0	0	0	0	0	a
51	1	2	0	0	0	52, e_2	0	e_2
52	1	0	0	53, e_1	0	0	0	e_1
53	1	54, b	0	0	0	0	0	b
54	1	0	0	0	0	55, e_2	0	e_2
55	1	0	0	56, e_1	0	0	0	e_1
56	57, a	0	0	0	0	0	0	a
57	1	2	0	0	0	58, e_2	0	e_2
58	1	0	59, c_1	0	0	0	0	c_1
59	1	0	0	60, e_1	0	0	0	e_1
60	1	61, b	0	0	0	0	0	b
61	1	0	0	0	0	62, e_2	0	e_2
62	1	0	63, c_1	0	0	0	0	c_1
63	1	0	0	64, e_1	0	0	0	e_1
64	65, a	0	0	0	0	0	0	a
65	1	2	0	0	0	66, e_2	0	e_2
66	1	0	0	0	67, c_2	0	0	c_2
67	1	68, b	0	0	0	0	0	b

68	1	0	0	0	0	69, e_2	0	e_2
69	1	0	0	0	70, c_2	0	0	c_2
70	71, a	0	0	0	0	0	0	a
71	1	2	0	0	0	72, e_2	0	e_2
72	1	0	0	0	73, c_2	0	0	c_2
73	1	0	74, c_1	0	0	0	0	c_1
74	1	75, b	0	0	0	0	0	b
75	1	0	0	0	0	76, e_2	0	e_2
76	1	0	0	0	77, c_2	0	0	c_2
77	1	0	78, c_1	0	0	0	0	c_1
78	79, a	0	0	0	0	0	0	a
79	1	2	0	0	0	80, e_2	0	e_2
80	1	0	0	0	81, c_2	0	0	c_2
81	1	0	0	82, e_1	0	0	0	e_1
82	1	83, b	0	0	0	0	0	b
83	1	0	0	0	0	84, e_2	0	e_2
84	1	0	0	0	85, c_2	0	0	c_2
85	1	0	0	86, e_1	0	0	0	e_1
86	87, a	0	0	0	0	0	0	a
87	1	2	0	0	0	88, e_2	0	e_2
88	1	0	0	0	89, c_2	0	0	c_2
89	1	0	90, c_1	0	0	0	0	c_1
90	1	0	0	91, e_1	0	0	0	e_1
91	1	92, b	0	0	0	0	0	b
92	1	0	0	0	0	93, e_2	0	e_2
93	1	0	0	0	94, c_2	0	0	c_2
94	1	0	95, c_1	0	0	0	0	c_1
95	1	0	0	96, e_1	0	0	0	e_1
◎96	97, a	0	0	0	0	0	0	a
97	1	2	0	0	0	0	98, c_3	c_3
98	1	99, b	0	0	0	0	0	b
99	1	0	0	0	0	0	100, c_3	c_3
100	101, a	0	0	0	0	0	0	a
101	1	2	0	0	0	0	102, c_3	c_3
102	1		103, c_1	0	0	0	0	c_1
103	1	104, b	0	0	0	0	0	b
104	1	0	0	0	0	0	105, c_3	c_3
105	1	0	106, c_1	0	0	0	0	c_1
106	107, a	0	0	0	0	0	0	a
107	1	2	0	0	0	0	108, c_3	c_3
108	1	0	0	109, e_1	0	0	0	e_1
109	1	110, b	0	0	0	0	0	b
110	1	0	0	0	0	0	111, c_3	c_3

111	1	0	0	112, e_1	0	0	0	e_1
112	113, a	0	0	0	0	0	0	a
113	1	2	0	0	0	0	114, c_3	c_3
114	1	0	115, c_1	0	0	0	0	c_1
115	1	0	0	116, e_1	0	0	0	e_1
116	1	117, b	0	0	0	0	0	b
117	1	0	0	0	0	0	118, c_3	c_3
118	1	0	119, c_1	0	0	0	0	c_1
119	1	0	0	120, e_1	0	0	0	e_1
120	121, a	0	0	0	0	0	0	a
121	1	2	0	0	0	0	122, c_3	c_3
122	1	0	0	0	123, c_2	0	0	c_2
123	1	124, b	0	0	0	0	0	b
124	1	0	0	0	0	0	125, c_3	c_3
125	1	0	0	0	126, c_2	0	0	c_2
126	127, a	0	0	0	0	0	0	a
127	1	2	0	0	0	0	128, c_3	c_3
128	1	0	0	0	129, c_2	0	0	c_2
129	1	0	130, c_1	0	0	0	0	c_1
130	1	131, b	0	0	0	0	0	b
131	1	0	0	0	0	0	132, c_3	c_3
132	1	0	0	0	133, c_2	0	0	c_2
133	1	0	134, c_1	0	0	0	0	c_1
134	135, a	0	0	0	0	0	0	a
135	1	2	0	0	0	0	136, c_3	c_3
136	1	0	0	0	137, c_2	0	0	c_2
137	1	0	0	138, e_1	0	0	0	e_1
138	1	139, b	0	0	0	0	0	b
139	1	0	0	0	0	0	140, c_3	c_3
140	1	0	0	0	141, c_2	0	0	c_2
141	1	0	0	142, e_1	0	0	0	e_1
142	143, a	0	0	0	0	0	0	a
143	1	2	0	0	0	0	144, c_3	c_3
144	1	0	0	0	145, c_2	0	0	c_2
145	1	0	146, c_1	0	0	0	0	c_1
146	1	0	0	147, e_1	0	0	0	e_1
147	1	148, b	0	0	0	0	0	b
148	1	0	0	0	0	0	149, c_3	c_3
149	1	0	0	0	150, c_2	0	0	c_2
150	1	0	151, c_1	0	0	0	0	c_1
151	1	0	0	152, e_1	0	0	0	e_1
152	153, a	0	0	0	0	0	0	a
153	1	2	0	0	0	0	154, c_3	c_3

154	1	0	0	0	0	155, e_2	0	e_2
155	1	156, b	0	0	0	0	0	b
156	1	0	0	0	0	0	157, c_3	c_3
157	1	0	0	0	0	158, e_2	0	e_2
158	159, a	0	0	0	0	0	0	a
159	1	2	0	0	0	0	160, c_3	c_3
160	1	0	0	0	0	161, e_2	0	e_2
161	1	0	162, c_1	0	0	0	0	c_1
162	1	163, b	0	0	0	0	0	b
163	1	0	0	0	0	0	164, c_3	c_3
164	1	0	0	0	0	165, e_2	0	e_2
165	1	0	166, c_1	0	0	0	0	c_1
166	167, a	0	0	0	0	0	0	a
167	1	2	0	0	0	0	168, c_3	c_3
168	1	0	0	0	0	169, e_2	0	e_2
169	1	0	0	170, e_1	0	0	0	e_1
170	1	171, b	0	0	0	0	0	b
171	1	0	0	0	0	0	172, c_3	c_3
172	1	0	0	0	0	173, e_2	0	e_2
173	1	0	0	174, e_1	0	0	0	e_1
174	175, a	0	0	0	0	0	0	a
175	1	2	0	0	0	0	176, c_3	c_3
176	1	0	0	0	0	177, e_2	0	e_2
177	1	0	178, c_1	0	0	0	0	c_1
178	1	0	0	179, e_1	0	0	0	e_1
179	1	180, b	0	0	0	0	0	b
180	1	0	0	0	0	0	181, c_3	c_3
181	1	0	0	0	0	182, e_2	0	e_2
182	1	0	183, c_1	0	0	0	0	c_1
183	1	0	0	184, e_1	0	0	0	e_1
184	185, a	0	0	0	0	0	0	a
185	1	2	0	0	0	0	186, c_3	c_3
186	1	0	0	0	0	187, e_2	0	e_2
187	1	0	0	0	188, c_2	0	0	c_2
188	1	189, b	0	0	0	0	0	b
189	1	0	0	0	0	0	190, c_3	c_3
190	1	0	0	0	0	191, e_2	0	e_2
191	1	0	0	0	192, c_2	0	0	c_2
192	193, a	0	0	0	0	0	0	a
193	1	2	0	0	0	0	194, c_3	c_3
194	1	0	0	0	0	195, e_2	0	e_2
195	1	0	0	0	196, c_2	0	0	c_2
196	1	0	197, c_1	0	0	0	0	c_1

197	1	198, b	0	0	0	0	0	b
198	1	0	0	0	0	0	199, c_3	c_3
199	1	0	0	0	0	200, e_2	0	e_2
200	1	0	0	0	201, c_2	0	0	c_2
201	1	0	202, c_1	0	0	0	0	c_1
202	203, a	0	0	0	0	0	0	a
203	1	2	0	0	0	0	204, c_3	c_3
204	1	0	0	0	0	205, e_2	0	e_2
205	1	0	0	0	206, c_2	0	0	c_2
206	1	0	0	207, e_1	0	0	0	e_1
207	1	208, b	0	0	0	0	0	b
208	1	0	0	0	0	0	209, c_3	c_3
209	1	0	0	0	0	210, e_2	0	e_2
210	1	0	0	0	211, c_2	0	0	c_2
211	1	0	0	212, e_1	0	0	0	e_1
212	213, a	0	0	0	0	0	0	a
213	1	2	0	0	0	0	214, c_3	c_3
214	1	0	0	0	0	215, e_2	0	e_2
215	1	0	0	0	216, c_2	0	0	c_2
216	1	0	217, c_1	0	0	0	0	c_1
217	1	0	0	218, e_1	0	0	0	e_1
218	1	219, b	0	0	0	0	0	b
219	1	0	0	0	0	0	220, c_3	c_3
220	1	0	0	0	0	221, e_2	0	e_2
221	1	0	0	0	222, c_2	0	0	c_2
222	1	0	223, c_1	0	0	0	0	c_1
223	1	0	0	224, e_1	0	0	0	e_1
⊚224	-	-	-	-	-	-	-	

The number of states in the tables can be found from the formula $2^{n-1}(n + 1)$, where n is the exponent of the order of the rewrite system. The successive orders 1, 2, 3, 4, 5, 6, … will have 2, 6, 16, 40, 96, 224 … states.

8. Boolean circuit for $L_{URS} = \{0^{2^n} : n \geq 0\}$

We can design a digital logic for a processor, using the universal gate set (AND, OR and NOT), that can produce an output equivalent to that from the transition function given the same input. Further developments could simplify this function to one using NAND gates alone.

Turing Machine $M = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$
$Q = \{q_1, q_2, q_3, q_4, q_5, q_{accept}, q_{reject}\}$
$\Sigma = \{0\}$
$\Gamma = \{0, x, \sqcup\}$

Transition Function δ for $L_{URS} = \{0^{2^n} : n \geq 0\}$			
State q	0	x	⊔
q_1	(q_2, ⊔, R)	(q_{reject}, x, R)	(q_{reject}, ⊔, R)
q_2	(q_3, x, R)	(q_2, x, R)	(q_{accept}, ⊔, R)
q_3	(q_4, 0, R)	(q_3, x, R)	(q_5, ⊔, R)
q_4	(q_3, x, R)	(q_4, x, R)	(q_{reject}, ⊔, R)
q_5	(q_5, 0, L)	(q_5, x, L)	(q_2, ⊔, R)

The following blank tape configuration table can be used by anyone who wants to test the standard transition function above against the Boolean circuit constructed below for any word of any length consisting of 0s. (It will only accept if the length is a power of 2.)

	Tape Configurations of δ for $L_{URS} = \{0^{2^n} : n \geq 0\}$						
	1	2	…	n	n + 1	…	t(n)
Input							
1							
2							
3							
…							
n							
n + 1							
…							
t(n)							

where $1 \leq i, j \leq$ t(n) and n = length of 0*

8.1 The Boolean circuit

And gate ∧
Or gate ∨
Not gate ¬

input: 0^n where $n \geq 0$

0^{2^0} ≡ binary[2, 2, ⊔q_{accept}] = (binary[1, 1, $q_1$0] ∧ binary[1, 2, ⊔]) ∧ (binary[2, 1, ⊔] ∧ binary[2, 2, q_2⊔])

⊔ ≡ binary[1, 1, ⊔q_{reject}] = (binary[1, 1, q_1⊔] ∧ binary[1, 2, ⊔])

x ≡ binary[1, 1, xq_{reject}] = (binary[1, 1, q_1x] ∧ binary[1, 2, ⊔])

0^{2^1} ≡ binary[5, 5, ⊔q_{accept}] = (binary[1,1, $q_1$0] ∧ binary[1, 2, 0] ∧binary[1, 3, ⊔]) ∧ (binary[2, 1, ⊔] ∧binary[2, 2, $q_2$0] ∧binary[2, 3, ⊔] ∧binary[2, 4, ⊔]) ∧(binary[3, 1, ⊔] ∧ binary[3, 2, x] ∧binary[3, 3, q_3⊔] ∧binary[3, 4, ⊔] ∧binary[3, 5, ⊔]) ∧(binary[4, 1, ⊔] ∧ binary[4, 2, x] ∧binary[4, 3, ⊔] ∧binary[4, 4, q_5⊔] ∧binary[4, 5, ⊔] ∧binary[4, 6, ⊔]) ∧ (binary[5, 1, ⊔] ∧binary[5, 2, x] ∧binary[5, 3, ⊔] ∧binary[5, 4, ⊔] ∧binary[5, 5, q_2⊔]))

0^3 ≡ binary[4, 4, ⊔q_{reject}] = (binary[1, 1, $q_1$0] ∧binary[1, 2, 0] ∧binary[1, 3, 0]) ∧ (binary[2, 1, ⊔] ∧binary[2, 2, $q_2$0] ∧binary[2, 3, 0] ∧binary[2, 4, ⊔]) ∧
(binary[3, 1, ⊔] ∧binary[3, 2, x] ∧binary[3, 3, $q_3$0] ∧binary[3, 4, ⊔] ∧binary[3, 5, ⊔]) ∧
(binary[4, 1, ⊔] ∧binary[4, 2, x] ∧binary[4, 3, 0] ∧binary[4, 4, q_4⊔])

0^{2^2} ≡ binary[7, 7, ⊔q_{accept}] = (binary[1, 1, $q_1$0] ∧binary[1, 2, 0] ∧binary[1, 3, 0] ∧ binary[1, 4, 0] ∧binary[2, 1, ⊔] ∧binary[2, 2, $q_2$0] ∧binary[2, 3, 0] ∧binary[2, 4, 0] ∧ binary[3, 1, ⊔] ∧binary[3, 2, x] ∧binary[3, 3, $q_3$0] ∧binary[3, 4, 0] ∧binary[4, 1, ⊔] ∧ binary[4, 2, x] ∧binary[4, 3, 0] ∧binary[4, 4, $q_4$0] ∧binary[5, 1, ⊔] ∧binary[5, 2, x] ∧ binary[5, 3, 0] ∧binary[5, 4, x] ∧binary[5, 5, q_3⊔] ∧binary[6, 1, ⊔] ∧binary[6, 2, x] ∧ binary[6, 3, 0] ∧binary[6, 4, x] ∧binary[6, 5, ⊔] ∧binary[6, 6, q_5⊔] ∧binary[7, 1, ⊔] ∧ binary[7, 2, x] ∧binary[7, 3, 0] ∧binary[7, 4, x] ∧binary[7, 5, ⊔] ∧binary[7, 6, ⊔] ∧
binary[7, 7, q_2⊔]

0^5 ≡ binary[6, 6, ⊔q_{reject}] = (binary[1, 1, $q_1$0] ∧binary[1, 2, 0] ∧binary[1, 3, 0] ∧ binary[1, 4, 0] ∧binary[1, 5, 0] ∧binary[2, 1, ⊔] ∧binary[2, 2, $q_2$0] ∧binary[2, 3, 0] ∧ binary[2, 4, 0] ∧binary[2, 5, 0] ∧binary[3, 1, ⊔] ∧binary[3, 2, x] ∧binary[3, 3, $q_3$0] ∧ binary[3, 4, 0] ∧binary[3, 5, 0] ∧binary[4, 1, ⊔] ∧binary[4, 2, x] ∧binary[4, 3, 0] ∧ binary[4, 4, $q_4$0] ∧binary[4, 5, 0] ∧binary[5, 1, ⊔] ∧binary[5, 2, x] ∧binary[5, 3, 0] ∧ binary[5, 4, x] ∧binary[5, 5, $q_3$0] ∧binary[6, 1, ⊔] ∧binary[6, 2, x] ∧binary[6, 3, 0] ∧ binary[6, 4, x] ∧binary[6, 5, 0] ∧binary[6, 6, q_4⊔])

Accept = {0^{2^0} ≡ binary[2, 2, ⊔q_{accept}], 0^{2^1} ≡ binary[5, 5, ⊔q_{accept}], 0^{2^2} ≡ binary[7, 7, ⊔q_{accept}], …}
Reject = {x ≡ binary[1, 1, xq_{reject}], ⊔ ≡ binary[1, 1, ⊔q_{reject}], 0^3 ≡ binary[4, 4, ⊔q_{reject}], 0^5 ≡ binary[6, 6, ⊔q_{reject}], …}

A Boolean and a quantum circuit have been constructed for ϵ_6 circuit and will be included, with full explanation, in a later paper. (The quantum circuit follows via a continuous transition from the 'classical' one.)

9. URS defined using the infinite alphabet with variables

Variable Finite Automata

The URS is effectively a series of closed finite zero-totality alphabets in a process which is repeated to infinity. Variable Finite Automaton (VFA), recognize finite languages (finite set of words) with words taken from an infinite alphabet Σ, as discussed by Grumberg *et al* [9]. We start with a *nondeterministic* Finite Automaton. This is a quintuple A = $< \Gamma, Q, Q_0, \delta, F >$, over a finite alphabet Γ, with Q as a finite set of states, with the set of initial states $Q_0 \subseteq$ Q. The transition function δ : $Q \times \Gamma \rightarrow 2^Q$ where 2^Q is the power set of Q, and F ⊆ Q is a set of accepting states. We say that *a exits q* if δ(q, a) ≠ ∅. A *run* of A on a word w = $\sigma_1\sigma_2...\sigma_n$ in Γ* (the set of all words formed from the letters in Γ) is defined as a sequence of states $q = q_0q_1...q_n$ such that the state $q_0 \in Q_0$ and the state $q_i \in \delta(q_{i-1}, \sigma_i)$ for every 1≤ i ≤ n. If the state $q_n \in$ F then the run q is accepted. Where a run exists, w is said to be *read along A*. L(A), the language of the nondeterministic finite automaton A, becomes the set of all words w in which there exists an accepting run of A on the input word w.

Grumberg *et al* [9] use this quintuple as a second component in their definition of a VFA as a pair $\mathcal{A} = < \Sigma, A >$where Σ is an infinite alphabet and A is a nondeterministic finite automaton, which is referred to as the pattern automaton of A. They state that: 'The (finite) alphabet of A is $\Gamma_A = \Sigma_A \cup X \cup \{y\}$, where $\Sigma_A \subset \Sigma$ is a finite set of constant letters, X is a finite set of bounded variables and y is a free variable. We refer to the number of bounded variables in A as the width of A. The variables in X ∪{y} range over $\Sigma \backslash \Sigma_A$.' (Here, $\Sigma \backslash \Sigma_A$ is the set difference of Σ and Σ_A, that is, all the elements in Σ except those of Σ_A.) Suppose we take a word $\sigma = \sigma_1\sigma_2...\sigma_j \in \Sigma_A$* read along A (meaning that a run exists in A for σ), and then take another word, w = $w_1w_2...w_j \in \Sigma$*. The word w from Σ* is a *legal instance* of σ from Σ_A* if

- $\sigma_i = w_i$ for every $\sigma_i \in \Sigma_A$,
- For x_i , $x_j \in X$, it holds that $w_i = w_j$ if and only if $x_i = x_j$, and w_i , $w_j \notin \Sigma_A$, and
- For $w_i = y$ and $w_j \neq y$, it holds that $w_i \neq w_j$.

We interpret what it means for a word w ∈ Σ to be a *legal instance* of a word v ∈ Σ, given that

$$\Sigma_A = \{\sigma_j : 1 \leq j \leq n\}$$

where Σ_A is a finite alphabet of constant letters that sets the word pattern of the VFA. The particular word pattern will then be transcribed into the infinite alphabet

$$\Sigma = \{w_j : j \geq 1\}$$

Every letter in the word σ ∈ Σ_A is now assigned to every letter in the word w ∈ Σ by matching positions j in both words. Σ is an infinite alphabet where $w_j = \sigma_j$ such that $1 \leq j \leq n$. This is the alphabet that the variables (bound and free) will range over. For the bound variables

$$X = \{x_j : 1 \leq j \leq n\}$$

That is, X is a finite set of bounded variables where $x_j = w_j$ such that $1 \leq j \leq n$. All elements of X have a fixed assignment to an element of Σ that cannot change. For the single free variable

$$\{y\} = \{y : y = w_j, j > n\}$$

That is, y is the free variable in the set $\{y\}, where\ y = w_j$, $w_j \neq x_j$ and $j > n$. In contrast to the bound variables x_j, the y variable is free. Its assignment runs for all the w_j in Σ for all $j > n$. This constraint ensures that the bound variables x_j and the free variable y will not have conflicting assignments because they will not be assigned to the same w_j in Σ.

According to Grumberg et. al [9] (with changed notation), a 'legal instance' of a word σ in $\Sigma_A *$ (which is another word w in Σ^*) 'leaves all occurrences of $\sigma_j \in \Sigma_A$ unchanged, associates all occurrences of $x_j \in$ X with the same unique letter, not in Σ_A, and associates every occurrence of y freely with letters from $\Sigma \setminus \Sigma_A$, different from those associated with X variables'. A word $\sigma \in \Sigma_A *$ becomes a 'witnessing pattern' for a word w $\in \Sigma^*$ iff w is a legal instance of σ, and this may be for zero, one, or greater than one witnessing patterns σ in $\Sigma_A *$. For any word w in Σ^*, a run of A on w is a run of A on a witnessing pattern for w'. The set of words from Σ^* for which there is a witnessing pattern in its subset $\Sigma_A *$ becomes the language of A, L(A).

The infinite alphabet

$$\Sigma = \{w_1, w_2, w_3, \ldots\}$$

The infinite URS alphabet with variables

$$\Gamma_A = \Sigma_A \cup X \cup \{y\} \text{ and } \Sigma_A \cap X \cap \{y\} = \emptyset$$

$$\Sigma_A = \{w_1 = a, w_2 = b\}$$

$$X = \{w_3 = x_1 = c_1, w_4 = x_2 = e_1\}$$

$$\{y_1, y_2\} = \Sigma - (\Sigma_A \cup X) = \{w_{3+2u} = y_1 = c_{u+1}, w_{4+2u} = y_2 = e_{u+1}\}$$

1	-1	***i_1***	***j_1***	***i_2***	***j_2***	***i_3***	***j_3***	…
a	b	***c_1***	***e_1***	***c_2***	***e_2***	***c_3***	***e_3***	…
		x_1	x_2	y_1	y_2	y_1	y_2	…
w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	…

$L_{URS} = \{\epsilon_1 = ab$, $\epsilon_2 = abax_1bx_1$, $\epsilon_3 = abax_1bx_1ax_2bx_2ax_1x_2bx_1x_2$,
$\epsilon_4 = abax_1bx_1ax_2bx_2ax_1x_2bx_1x_2ay_1by_1ay_1x_1by_1x_1ay_1x_2by_1x_2ay_1x_1x_2by_1x_1x_2$,
$\epsilon_5 = abax_1bx_1\ ax_2bx_2\ ax_1x_2bx_1x_2ay_1by_1ay_1x_1by_1x_1ay_1x_2by_1x_2ay_1x_1x_2by_1x_1x_2ay_2by_2\ ay_2x_1by_2x_1ay_2x_2by_2x_2$
$ay_2x_1x_2by_2x_1x_2ay_2y_1by_2y_1ay_2y_1x_1by_2y_1x_1ay_2y_1x_2by_2y_1x_2ay_2y_1x_1x_2by_2y_1x_1x_2$,
$\epsilon_6=$
$abax_1bc_1ae_1be_1ac_1e_1bc_1e_1ac_2bc_2ac_2c_1bc_2c_1ac_2e_1bc_2e_1ac_2c_1e_1bc_2c_1e_1ae_2be_2ae_2c_1be_2c_1ae_2e_1be_2e_1ae_2c_1e_1be_2c_1$
$e_1ae_2c_2be_2c_2ae_2c_2c_1be_2c_2c_1ae_2c_2e_1be_2c_2e_1ae_2c_2c_1e_1be_2c_2c_1e_1ac_3bc_3ac_3c_1bc_3c_1ac_3e_1bc_3e_1ac_3c_1e_1bc_3c_1e_1ac_3c_2$
$bc_3c_2ac_3c_2c_1bc_3c_2c_1ac_3c_2e_1bc_3c_2e_1ac_3c_2c_1e_1bc_3c_2c_1e_1ac_3e_2bc_3e_2ac_3e_2c_1bc_3e_2c_1ac_3e_2e_1bc_3e_2e_1ac_3e_2c_1e_1bc_3e_2$
$c_1e_1ac_3e_2c_2bc_3e_2c_2\ ac_3e_2c_2c_1bc_3e_2c_2c_1\ ac_3e_2c_2e_1bc_3e_2c_2e_1ac_3e_2c_2c_1e_1bc_3e_2c_2c_1e_1\}$

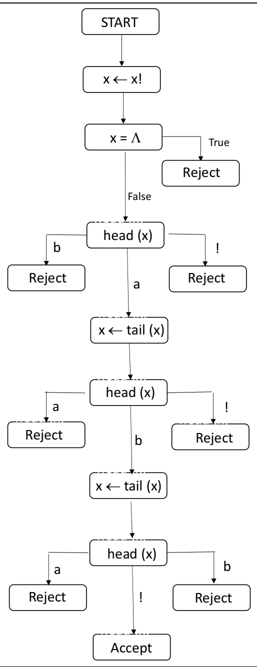
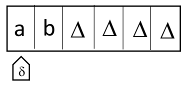
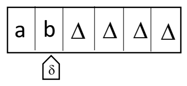
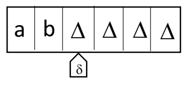
Machines for the URS defined using the infinite alphabet with variables are the same as the machines for the URS defined using the infinite alphabet without variables.

10. Other machines that recognize the URS: Post and finite machine with two pushdown stores

There exist two other procedures which are equivalent in power, i.e. recognizing the same set of words or language, to the Turing machine. These are the Post machine and the Finite machine with two pushdown stores [36]. In the tables below we show the flow diagrams for the simplest URS word: ϵ_1 = ab.

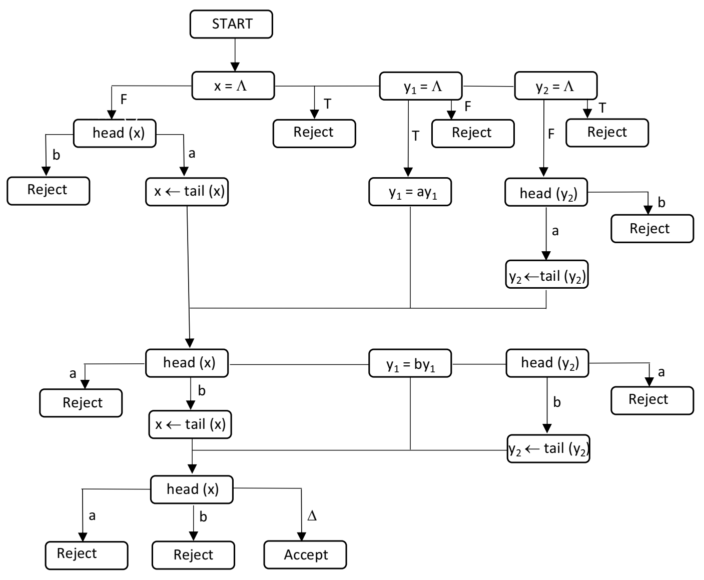
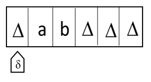
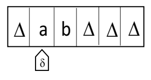
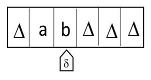
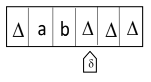
A Post Machine M over Σ ∪ {!} is a flow-diagram with one variable x, which may have as a value any word over Σ ∪ {!}, where ! is a special auxiliary symbol.

Post Machine Model
(Σ = {a, b}) ∪ (V = {a, b}) ∪ {!}
x = ab ∈ Σ*
	Corresponding Turing Machine Model
	(Σ = {a, b}) ∪ (V = {a, b}) ∪ {!}
	x = ab ∈ Σ*

 START
x ← x!
x = Λ
True
Reject
False
head (x)
b
!
Reject
a
Reject
x ← tail (x)
head (x)
a
!
Reject
b
Reject
x ← tail (x)
head (x)
a
b
Reject
!
Reject
Accept
	x = ab!ΔΔΔΔ
	
	x = b!aΔΔΔΔ
	
	x =!abΔΔΔΔ
	
	x = ΔΔΔΔ!ab

A Finite Machine with Two Pushdown Stores M over Σ ∪ {Λ} is a flow-diagram with one variable x, which may have as a value any word over Σ ∪ {Λ}.

Finite Machine with Two Pushdown Stores Model
(Σ = {a, b}) ∪ (V = {a, b}) ∪ {Δ}
x = ab ∈ Σ*, y_1 = Λ, y_2 = ab
	Corresponding Turing Machine Model
	(Σ = {a, b}) ∪ (V = {a, b})
	x = ab ∈ Σ*, y_1 = Δ, y_2 = ab ∈Σ*

 START
x = Λ
y₁ = Λ
y₂ = Λ
head (x)
x ← tail (x)
y₁ = ay₁
head (y₂)
y₂ ← tail (y₂)
y₁ = by₁
Reject
Accept
T
F
a
b
Δ
	
	x = ab, y_1= Δ, y_2 = abΔΔΔ
	
	x = ab, y_1 = aΔ, y_2 = bΔΔΔ
	
	x =ab, y_1 = baΔ, y_2 = ΔΔΔ
	
	x = ab, y_1 = ΔbaΔ, y_2 = ΔΔ

11. Conclusion

The hierarchy of machines discussed in this paper and stemming from the Chomsky hierarchy is relevant to the way the URS is applied in physics and biology. We can see the repeating fractal structure constructed from a finite alphabet in the finite fundamental group of the physical parameters, with its alphabet constructed from mass, time, charge and space and their algebraic representations. However, their combination in the nilpotent fermionic structure leads to an infinite alphabet representing a continuous progression, representing the quantum universe, with an infinite alphabet of unique nilpotents, each constructed from units of mass, time, charge and space. Again the possible types of fermionic structure naturally fit into a finite alphabet construction, whereas the potentially infinite number of fermionic states would require an infinite alphabet representing a continuous progression. In biology, a finite alphabet operates for the letter code of the bases A,T, G, C and the 64 codons which they construct, while an infinite alphabet is needed for the infinite possibility of genomes constructed from them.

The Rowlands-Diaz universal alphabet and rewrite system appears to provide a meta-pattern for mathematics and science, extending from quantum mechanics and particle physics up to biology and even consciousness. Here, we have shown that it is compatible with computational and formal language theory, using a Turing machine, a Post machine and a Finite machine with two pushdown stores, which means that we can apply computational theory and practice directly to all these systems. Mathematics gives us a guide to the patterns of Nature, rather than the meta-pattern, because it is structured on exact repetition, e.g. of the integer series 1, 2, 3, 4, 5 …, rather than uniqueness. The meta-pattern is a unique birth-ordering in an infinite process, and any reproduction of it will only ever be finite. In addition, mathematics only reads, bounded by the zero cardinalities, each of which creates a new algebra. Physics requires a unique sequence of events, never repeated, which means that, to simulate it, we must use machines that can write as well as read, as in the cases discussed here, which makes its requirements different from those of much mathematically-based computational theory, which is frequently concerned with pure reading automata. Physics has a special system for obtaining zeros using nilpotents, which are each unique and hence at its most basic level can describe a unique birthordering. We see that the mathematics most appropriate for this development is a form of Clifford algebra. This does not prevent us from developing alternative mathematical ideas, such as the higher Cayley-Dickson algebras, but they cannot be used to describe the system itself. The development of a computational representation based on the most general devices that can be imagined gives a powerful indication that the Rowlands-Diaz universal rewrite system provides the most general, generic and efficient description so far known for Nature's most fundamental processes.

References

1. Rowlands P and Diaz B 2002 A universal alphabet and rewrite system arXiv:cs.OH/0209026
2. Diaz B and Rowlands P 2005 A computational path to the nilpotent Dirac equation *International Journal of Computing Anticipatory Systems* 16 203-18
3. Diaz B and Rowlands P 2006 The infinite square roots of –1 *International Journal of Computing Anticipatory Systems* 19 229-235
4. Rowlands P 2007 *Zero to Infinity: The Foundations of Physics* (World Scientific)
5. Rowlands P 2010 Mathematics and Physics as Emergent Aspects of a Universal Rewrite System *International Journal of Computing Anticipatory Systems* 25 115-131
6. Rowlands P 2014 *The Foundations of Physical Law* (World Scientific)
7. Bateson G 1979 *Mind and Nature: A Necessary Unity* (Bantam Books Toronto)
8. Marcer P and Rowlands P 2015 Information, Bifurcation and Entropy in the Universal Rewrite System *International Journal of Computing Anticipatory Systems* 27 203-215
9. Rowlands P 2001 A foundational approach to physics, arXiv:physics/0106054
10. Rowlands P 2003 The nilpotent Dirac equation and its applications in particle physics arXiv:quant-ph/0301071
11. Rowlands P 2004 Symmetry breaking and the nilpotent Dirac equation *AIP Conference Proceedings* 718 102-115
12. Rowlands P 2005 Removing redundancy in relativistic quantum mechanic arXiv.org:physics/ 0507188
13. Rowlands P 2008 What is vacuum? arXiv:0810.0224
14. Rowlands P 2010 Physical Interpretations of Nilpotent Quantum Mechanics, arXiv: 1004.1523
15. Rowlands P 2013 Space and Antispace in Amoroso R L Kauffman L H and Rowlands P (eds.), *The Physics of Reality Space, Time, Matter, Cosmos* World Scientific 29-37
16. Rowlands P 2013 Symmetry in Physics from the Foundations *Symmetry* 24 41-56
17. Rowlands P 2017 How symmetries become broken, *Symmetry* 28 244-254
18. Rowlands P 2018 Idempotent or nilpotent? *AIP Conference Proceedings* 2046 020081
19. Marcer P and Rowlands P 2017 Nilpotent Quantum Mechanics: Analogs and Applications *Frontiers in Physics* 5 article 28 1-8, 2017
20. Rowlands P 2019 Constructing the Standard Model fermions? *Journal of Physics Conference Series* 1251 012004
21. Rowlands P and Rowlands S 2019 Are octonions necessary to the Standard Model? *Journal of Physics Conference Series* 1251 012044

22. Hill V J and Rowlands P 2008 Nature’s code *AIP Conference Proceedings* 1051 117-126
23. Hill V J and Rowlands P 2010 Nature’s Fundamental Symmetry Breaking *International Journal of Computing Anticipatory Systems* 25 144-159
24. Hill V J and Rowlands P 2010 The Numbers of Nature’s Code *International Journal of Computing Anticipatory Systems* 25 160-175
25. Hill V J and Rowlands P 2015 A mathematical representation of the genetic code in Amoroso R L Kauffman L H and Rowlands P (eds.) *Unified Field Mechanics* (World Scientific) 553-559
26. Marcer P Mitchell E Rowlands P and Schempp W Zenergy: The ‘phaseonium’ of dark energy that fuels the natural structures of the Universe 2005 *International Journal of Computing Anticipatory Systems* 16 189-202
27. Marcer P and Rowlands P 2007 How intelligence evolved? in Quantum Interaction, Papers from the AAAI Spring Symposium, Technical Report SS-07-08, 2007
28. Marcer P and Rowlands P 2010 Further Evidence in Support of the Universal Nilpotent Grammatical Computational Paradigm of Quantum Physics *AIP Conference Proceedings* 1316 90-101
29. Marcer P and Rowlands P 2010 The ‘Logic’ of Self-Organizing Systems *AAAI Technical Reports* 2010-08-020
30. Marcer P and Rowlands P 2010 The Grammatical Universe and the Laws of Thermodynamics and Quantum Entanglement *AIP Conference Proceedings* 1303 161-167
31. Marcer P and Rowlands P 2013 A Computational Unification of Scientific Law: Spelling out a Universal Semantics for Physical Reality in Amoroso R L Kauffman L H and Rowlands P (eds.) *The Physics of Reality Space, Time, Matter, Cosmos* (World Scientific)
32. Marcer P and Rowlands P 2015 Computational tractability – beyond Turing? in Amoroso R L Kauffman L H and Rowlands P (eds.) *Unified Field Mechanics* (World Scientific) 33-38
33. Sipser M 2013 *Introduction to the Theory of Computation* third edition (Cengage Learning)
34. Grumberg O Kupferman O and Sheinvald S Variable Automata over Infinite Alphabets 2010 in Dediu A-H, Fernau H and Martín-Vide C (eds) LATA 2010 LNCS 6031 561-572
35. Cormen T H, Leiserson C E, Rivest R L and Stein C 2009 *Introduction to Algorithms* third edition (MIT Press)
36. Manna Z 1974 *Mathematical Theory of Computation* (McGraw-Hill)

Acknowledgement. We are grateful to Sarai Sheinvald and Orna Kupferman for clarification of various points raised in their paper [34]