Sustainable Numerals: The Identity of Symbol and Value

Armands Strazds

ABSTRACT

The purpose of this paper is to propose a new numeral system, which possesses quantitative equality (identity) between symbol and value, and can be used to represent large numbers. As all the traditional numeral systems (with exception of tally marks) are limited to the qualitative equality, in which a symbol is *assigned* to a value by convention instead of being *derived* from it by some unchanging principle (e.g. an arithmetic or geometric method), they are potentially vulnerable to becoming obsolete and undecipherable under certain unfavorable conditions. We argue, therefore, that an alternative approach with higher degree of protection for storing critical information should be considered. Our proposition for such an approach is to utilize a proportional geometric binary expansion of the first two prime numbers.

Key words: numeral system, delta expansion, mathematical semiotics.

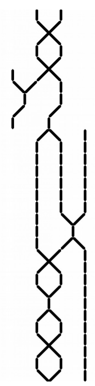
1 Introduction

Symbols used to represent numeric values range from simple unary tally marks to complex daxie-characters of Chinese finance:

壹拾貳億參仟肆佰伍拾陸萬柒仟捌佰玖拾

However, all the traditional numeral systems are limited in one of two ways: they are either like daxie where there is no means to derive the value from the symbol, since the value and the symbol have only a qualitative, but no quantitative equality (identity), or they are like the tally marks unsuitable for representation of large numbers. The most commonly used Hindu-Arabic system belongs to the first category. We argue that due to this limitation traditional numeral systems are in the long term unsafe for information retention, as they tend to become obsolete and replaced by new cultural conventions.

Therefore we propose a solution where the number's value can be derived from the numeral itself by purely geometric methods, and at the same time such a numeral can be used for representation of sufficiently large numbers. For example the daxie-number given above can be encoded in a form of 1234567890, or in a geometric form:



While the former case has no identity between the symbol and value, the latter has. In the paragraphs below we will describe the establishing of the identity in detail:

§2 gives the basic definitions used with the conversion, §3 describes the geometric primitives, §4 explains the mathematical properties of the geometric digits, §5 gives an overview about the geometric number components composition principles, §6 describes the rules of logic for constructing the geometric numbers, §7 Presents the formal grammar for the geometric components, and §8 discusses the research results, and suggests the possible fields of application.

2 Definitions

We define a set *z* with even number elements

(a) z := {0, 2, 4, 6, …}

and a set *s* with odd number elements

(b) s := {-1, 1, 3, 5, 7, …}.

Further, we define a delta number as a distance (absolute value) between exponents of the adjacent powers of two, and a set Δ with natural number elements

(c) Δ := {1, 2, 3, 4, …}.

3 Geometric Primitives

Our goal was to develop geometric[1] marks, which like tally marks can carry quantitative information, and unlike tally marks compress this information efficiently. We start with description of the geometric primitives, the simplest (irreducible) geometric objects responsible for the appearance of the marks. First, we define a two-dimensional space composed of magnitudes A and B (Fig. 3.1). The extension of magnitude A is two units, and of magnitude B three units[2].

FIG. 3.1 `A-B--`

1 Arithmetic, Geometry and other sciences of that kind which only treat of things that are very simple and very general, without taking great trouble to ascertain whether they are actually existent or not, contain some measure of certainty and an element of the indubitable. For whether I am awake or asleep, two and three together always form five. (Descartes, 1641)

2 According to Leibniz's classification of numbers, the prime numbers are of type 1, even numbers of type 2, numbers divisible by 3 of type 3, etc. (Leibniz ca. 1680) For our basic magnitudes we have chosen the first and only even prime number 2, and on the first odd prime number 3.

All magnitudes are extended in horizontal dimension, which we call level. The vertical dimension is a set of levels. The distance between the adjacent levels is one unit[3]. We number the levels top down p = 0..∞. Each level, according to its number, represents a distinct power of 2, i.e. 2^0, 2^1, 2^2, etc.

The two-units-magnitude (A) we call the short syllable, and the three-units-magnitude (B) the long syllable[4]. Within a level syllables expose two kinds of behavior: runs and cycles. Runs are repetitions of the same type of syllable, e.g. short syllable run of length 2 is `A-A-`, and long syllable run of lenght 4 is `B--B--B--B--`. Cycles are repetitions of the complete runs of both syllables, e.g. `(A-A-B--B--)(A-A-B--B--)` etc. The run length of a level is 2^p, which gives on level 0 the run length 1, on level 1 the run length 2, on level 2 the run length 4, etc. On every next level the run length doubles, thus making the runs of short and long syllables of one level proportional to the runs of short and long syllables of any other level. Fig. 3.2 shows the first twenty horizontal units of the three top levels (p = 0..2). The cycle length of a level in units is 5 * 2^p.

FIG. 3.2

```
A-B--A-B--A-B--A-B--
A-A-B--B--A-A-B--B--
A-A-A-A-B--B--B--B--
```

There exist six possible configurations (relation types) of syllables of the adjacent levels. The variable X in Fig. 3.3 stands for either A or B.

FIG. 3.3

(0a)	(0b)	(0c)	(1a)	(1b)	(1c)

`A-X-`
`A-A-`
	`-B--`
	`A-X-`
		`--B-`
		`-B--`
		or
		`-B--`
		`--B-`
			`-A-B`
			`B--B`
				`A-X-`
				`-B--`
					`-B--`
					`-B--`

3 Magnitudes 2 and 3 of the horizontal and 1 of the vertical dimension can be regarded as the expressions of diversity and unity.

4 Based on Pingala, ca. 200 BC, s. Weber, 1863.

4 Elements

Each of the configurations as shown in Fig. 3.3 can be represented as a distinct geometric shape (Fig. 4.1). We call these basic geometric shapes elements, and they correspond to the notion of digits in positional number systems. Unlike with the traditional number systems, digits in our system do not posses a distinct value, instead they share a value, giving in each case three different shapes for one value. For instance, all the three elements in the left column of Fig. 4.1 represent the binary value zero, or the logical value false; and all the three elements in the right column represent the binary value one, or the logical value true.

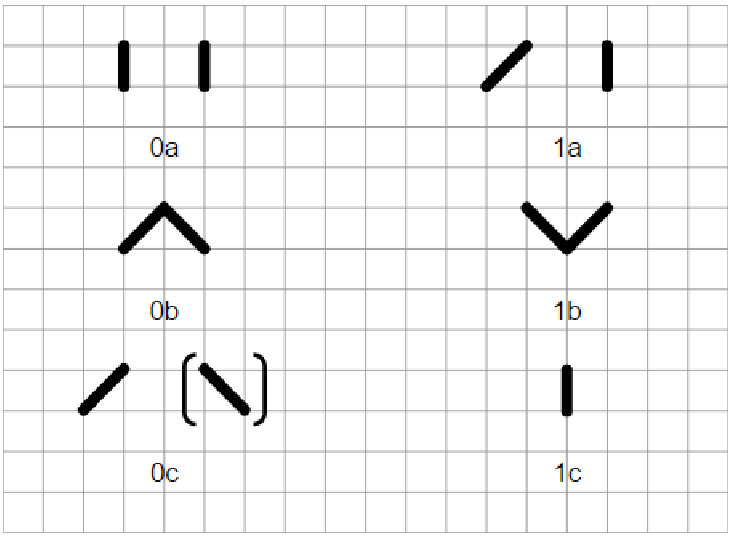


FIG. 4.1

In the next paragraph we will describe how elements are connected to build the components.

5 Components

How is the symbol given in the introduction related to its value? First we need to obtain the binary form of the value, which in this case is

100100110010110000001011010010.

In the second step we convert it to what we call the delta form, which means counting the zero runs within the binary number, increased by 1:

3-3-1-3-2-1-8-2-1-2-3-2

In the third and last step we substitute the individual delta values with the corresponding geometric structures (see §5), the parameters (shape, height, and shift) of which are unambiguously determined by the respective delta value and its position within the delta number. In the example above the first delta value (from the right) is 2, and it is encoded using the geometric V-component, which is 2 units high; the next value 3 is encoded using the O-component, which is 3 units high; the next value 2 is encoded using the U-component, which is 2 units high. Why the V-component is not used here instead? Because the V-component is a so called alpha component, it marks the beginning of the symbol thus encoding the vertical reading direction (here top down). Other components, like ZI, Z and S, encode the horizontal reading direction.

Every natural number can be expressed as the sum of distinct powers of two:

$$\sum_{p=0}^{\infty} a_p * 2^p \quad \text{where} \quad a_p \in \{0,1\}$$

The distinct powers of two where $a_p = 1$ we call components. The set of components with even *p* values we describe as 2^z and the set with odd values as 2^s. The set of even delta numbers we describe as Δ_z and the set of odd delta numbers as Δ_s (definitions see chapter 2). The component with the lowest exponent's value we call the alpha component, and all other components beta components. The delta value of the alpha component 2^{p_0} is $p_0 - (-1)$ or simply $p_0 + 1$. The delta value of the beta component 2^{p_n} is $p_n - p_{n-1}$.

There are two* types of alpha components (p_n, n = 0):

1) ZI, an alpha component with odd delta parity and even exponent parity

$$ZI := p_0 \in \Delta_s$$

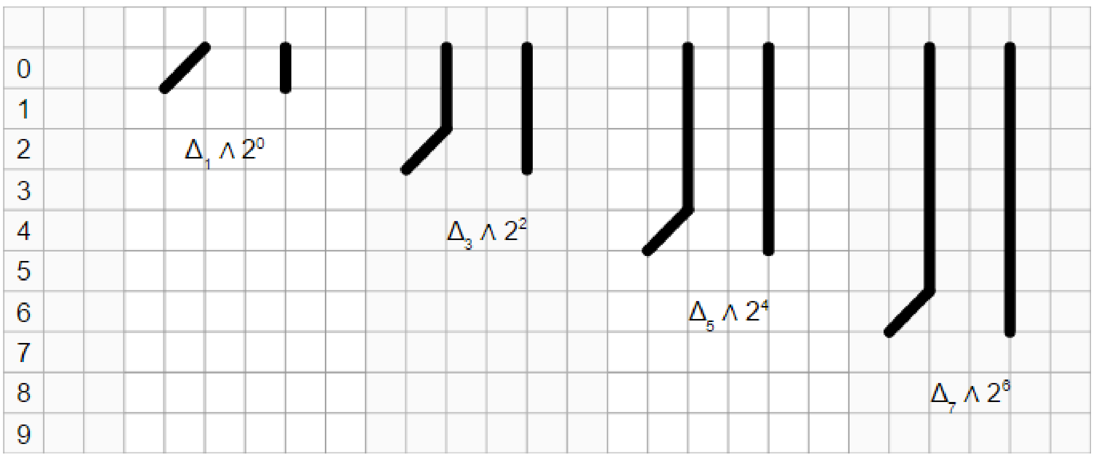


FIG. 5.1

Fig. 5.1 Shows the first four ZI components, representing values from the set
2^s = {1, 4, 16, 64, ...}.

2) V, an alpha component with even delta parity and odd exponent parity

$V := p_0 \in \Delta_z$

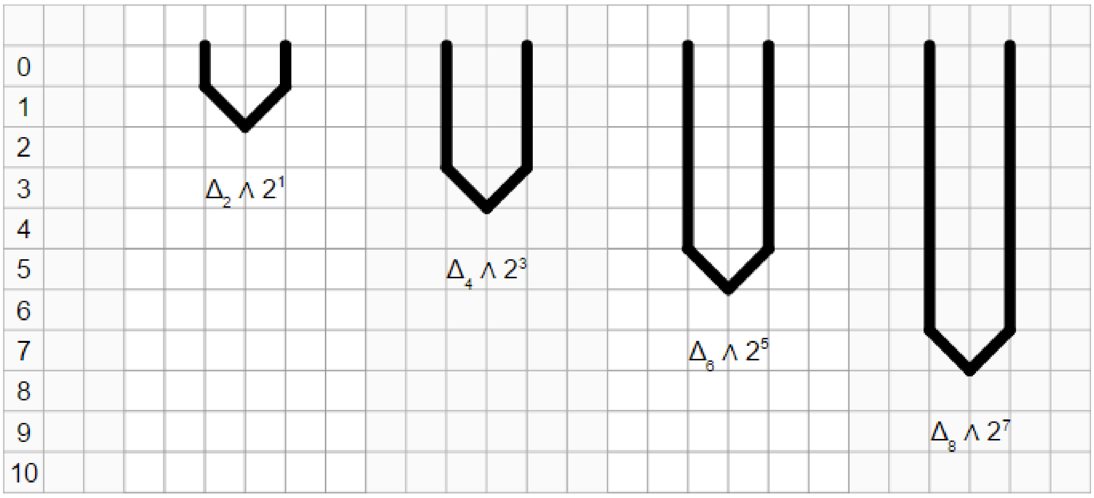


FIG. 5.2

Fig. 5.2 Shows the first four V components, representing values from the set $2^z = \{2, 8, 32, 128, ...\}$.

* A special (undetermined) type of alpha component is zero, which can have any delta value 1..∞

$H := p_0 \in \Delta_{1..\infty}$

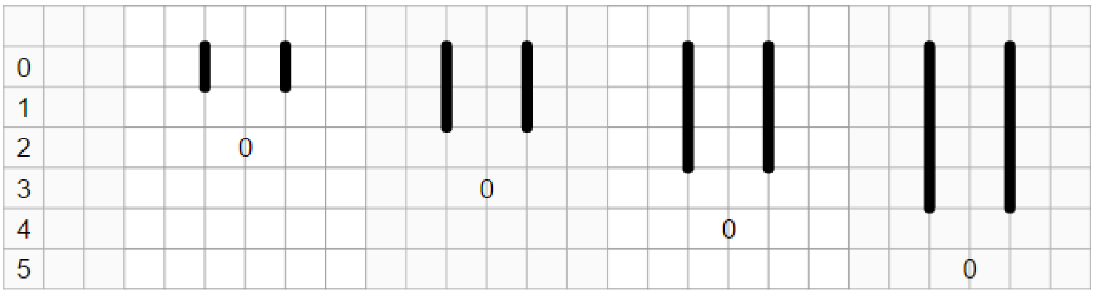


FIG. 5.3

Fig. 5.3 Shows some variants of the H component, representing a value 0 or $2^{(-\infty)}$. A stand-alone unterminated H component is presumed to have an endless delta value; therefore with any extension it still retains its value 0.

The four basic types of beta components (p_n, n > 0) are:

3) U, a beta component with even delta parity and even exponent parity

$U := p_n \in (\Delta_z \wedge 2^z)$, z > 0

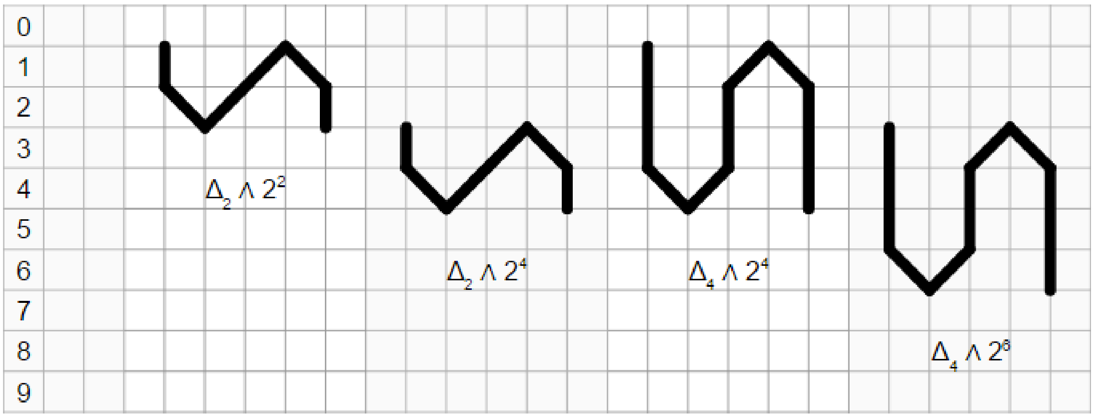


FIG. 5.4

Fig. 5.4 shows some variants of the U component, representing values from the conjunction of the sets Δ_z and 2^z.

4) N, a beta component with even delta parity and odd exponent parity

$N := p_n \in (\Delta_z \wedge 2^s)$, z > 0, s > 1

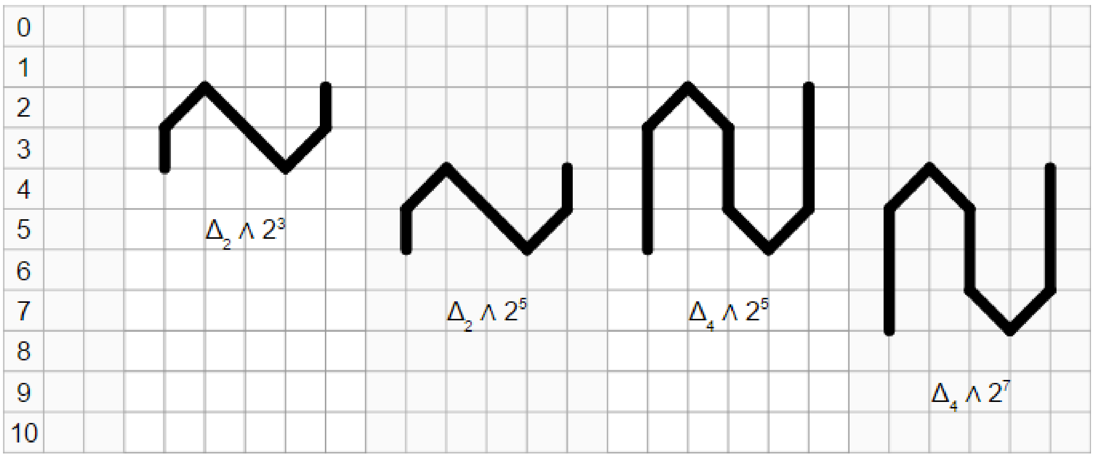


FIG. 5.5

5) O, a beta component with odd delta parity and odd exponent parity

$O := p_n \in \Delta_s$, s > 1

6) I, a beta component with delta value of 1 and odd exponent parity

$I := p_n \in \Delta_1$

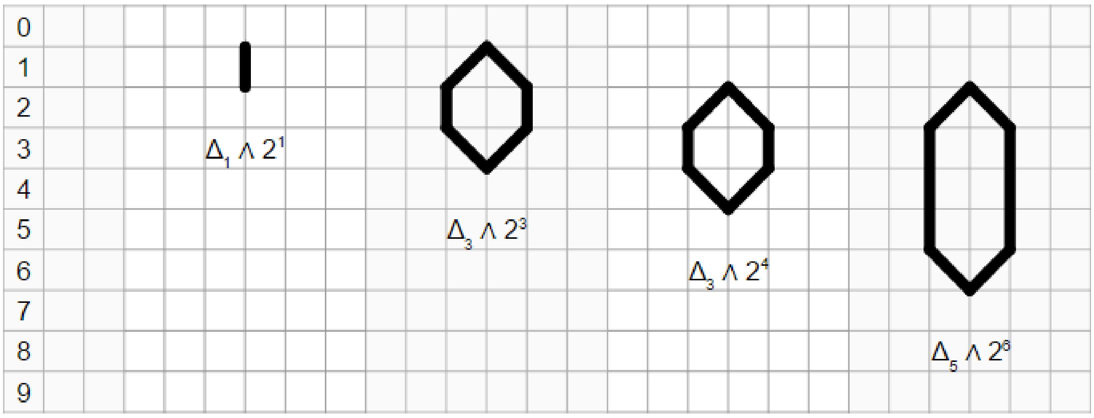


FIG. 5.6

and two types of partial beta components, which are used with split components (see Fig. 5.7):

7) Z, a partial beta component with delta value of 1 and even exponent parity

$Z := p_n \in (\Delta_1 \wedge 2^z)$

8) S, a partial beta component with delta value of 1 and odd exponent parity

$S := p_n \in (\Delta_1 \wedge 2^s)$, s > 1

Fig. 5.7 shows the Z and S partial components in conjunction with the respective complementary components.

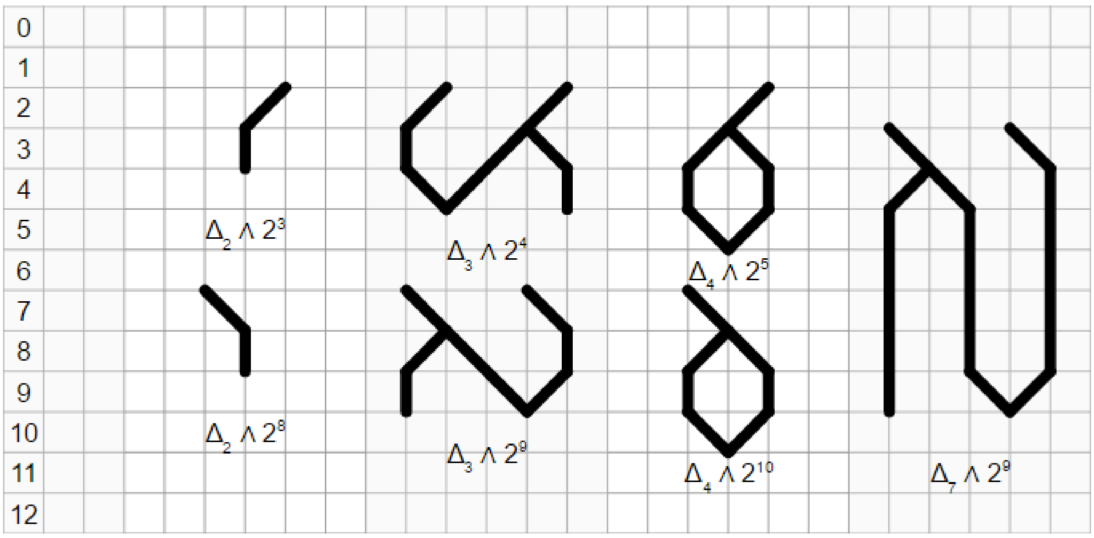


FIG. 5.7

We recognize split components by the presence of Z or S elements (Fig. 4.1 case 0c). Split components substitute the unsplit components in cases when triggers (Fig. 5.8) appear. The delta value (total height, i.e. the height of the two partial components) of the split component is equal to the delta value of unsplit component, which it substitutes.

Fig. 5.8 Shows the three (or five, if the symmetrical variants are count) types of component combinations, called triggers, which cause a component splitting with utilization of the S or Z partial components.

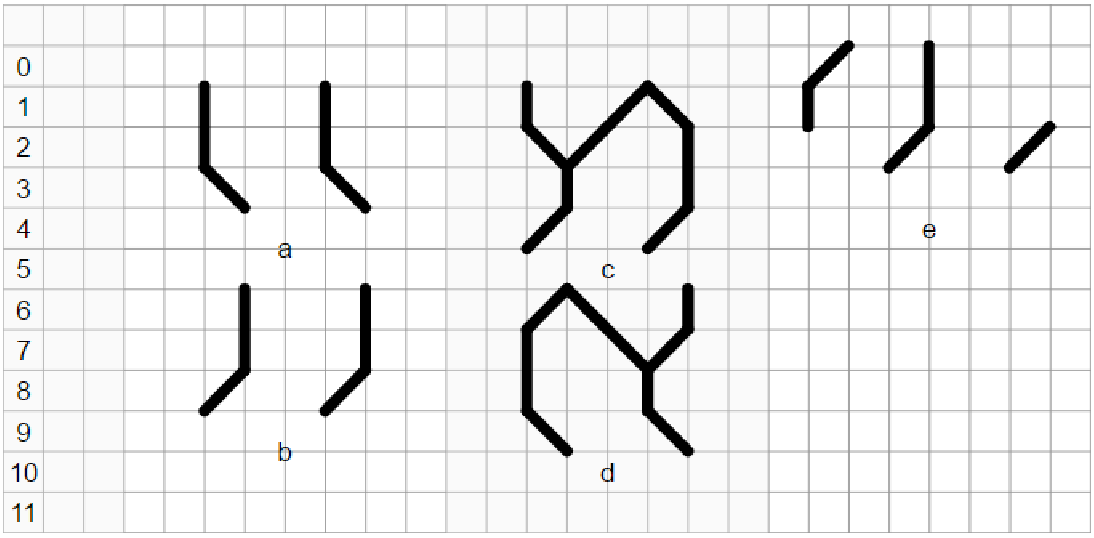


FIG. 5.8

Cases *a* and *b* show a combination of double I-components ($2^2 + 2^1$ resp. $2^7 + 2^6$), if a component of a delta value grater than 1 follows, a split component must be used.

Cases *c* and *d* show a combination of a U resp. N components of delta value 2 with an I component. This combination also produces a split of the following component if its delta value is larger than 1.

Case *e* is a special variant of case *b*: as the exponent 0 never produces an I component, the combination of I and ZI_1 causes here a split in the following component if its delta value is larger than 1.

Unlike the alpha components, stand-alone beta components cannot represent numbers, since they possess merely a relative value. In order to obtain an absolute value, an alpha component must be supplemented.

6 Component Logic

Components are combined according to the rules of logic. The rules are:

1) the initial truth carrier of the H element (Fig. 6.1 cases a and c, cf. Fig. 4.1 case 0a) with exponent 0 is the right strand, and
the initial truth carrier of the ZI element (Fig. 6.1 cases b and d, cf. Fig. 4.1 case 1a) with exponent 0 is the left strand;

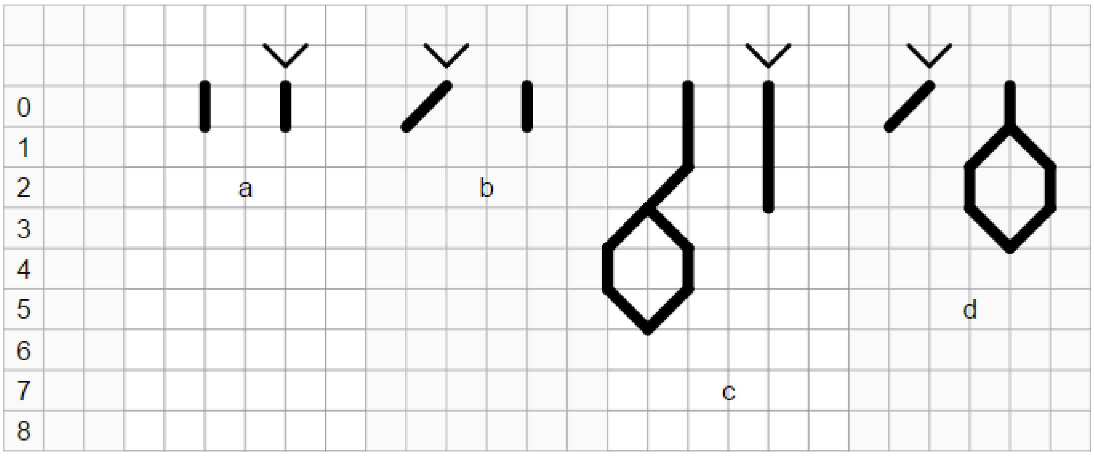


FIG. 6.1

2) the V element (Fig. 4.1 case 1b) is the truth emitter, and
the A element (ibid. case 0b) is the truth receiver;

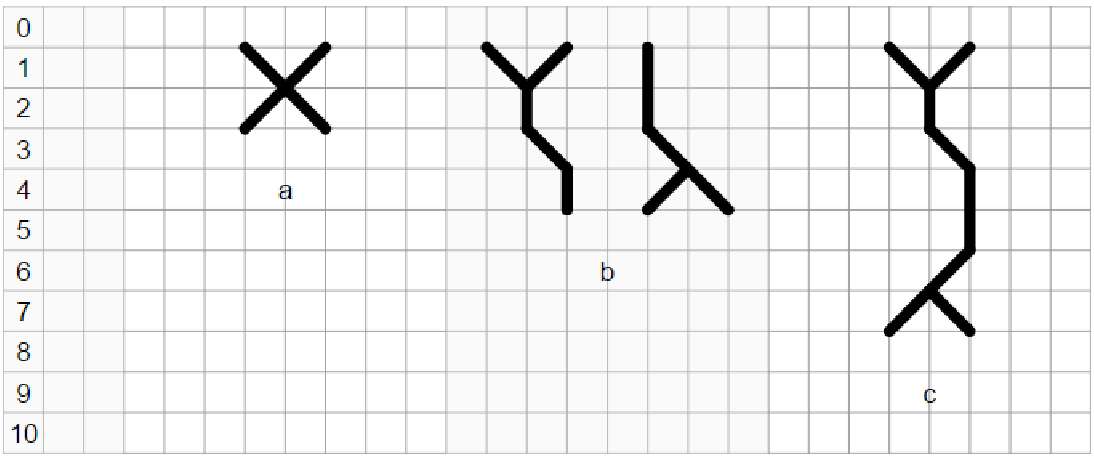


FIG. 6.2

3) the ZI, Z resp. S elements (ibid. cases 1a, 0c) are logical negations, and
the I element (ibid. case 1c) is logical affirmation.

On rule 1: The initial truth carrier can be deduced by observation of how beta components are connected to alpha components. For instance, Fig. 6.1 case c shows that the truth receiver A (see the rule 2) of the O component is connected to the left strand, and in ibid. case d the same component is connected to the right strand. This leads us to the conclusion, that the H and ZI elements must have the opposite initial truth setting: in case of H element the initial truth carrier must be the right strand, and in case of ZI the left strand (see also the rule 3, esp. regarding the ZI element's role as a logical negation).

On rule 2: The beta components N, U, and O contain both the A and the V elements (see Fig. 5.4, 5.5, and 5.6 cf. Fig. 4.1 cases 0b and 1b). Therefore when connecting to the previous alpha or beta component it is important to observe the rule that element A always connects to the true strand. N and U components expose different behavior regarding the logic than the O component: the former two act as definite logical negations, and the latter as a definite logical affirmation. In all three cases the logical operation is called definite since the component itself contains information about the logical value of its strands. By contrast, the S, Z, and I components perform indefinite logical operations, since they contain neither the truth receivers, nor emitters.

The truth emitter endows the strand below with the logical value true, and the strands to the left and right with the logical value false. If a component possesses a truth receiver (element A), it must always be connected to the strand carrying the logical truth. Example see Fig. 6.2 case a: the direct connection of the truth emitter V on level 1, and truth receiver A on level 2.

On rule 3: Fig. 6.2 case b shows how S element swaps the truth values of the two strands: first, the truth emitter V (see the rule 2) endows the left strand on level 1 with the logical value true, ergo the right strand with the value false; then on level 2 the I component (indefinite logical affirmation) forwards the unchanged truth values to the S element on the level 3; the S component, being the indefinite logical negation, swaps the truth values endowing the left strand with the logical value false and the right strand with value true; and finally on the level 4 the truth receiver, element A, connects to the strand currently carrying the logical value true, i.e. to the right strand.

Fig. 6.2 case c is an example of the double negation; the reason why the truth receiver on level 7 can be connected to the same strand as defined true by the emitter on level 1, despite of the previous negation element S on level 3, is the presence of the second negation element Z on level 6, which reverts the truth value again; negation of negation gives a confirmation.

7 The Formal Grammar

Noam Chomsky famously developed a method of constructing formal grammars by means of „rewriting rules“[5]. We apply this method here to demonstrate that a consistent way exists of connecting our finite set of components to produce infinite number of unique syntactically valid combinations, as required for representation of all natural numbers.

5 Chomsky 1959.

First, we use the symbols of components to create an alphabet:

∑ = {ZI, V, U, N, O, I, Z, S}

Then we define the variables; we use dashes to connect the letters so that ZI (one element) can be distinguished from Z-I (two elements); the initial variable ST is the starting point.

V = {ST, U, N, O, I, Z, S, I-ZI, I-I, I-N, I-U}

And finally we design the production rules; terminal symbols are given in lower-case; I^z stands for any I component of the even level (see §2 definition *a*), and I^s stands for any I component of the odd level (see ibid. definition *b*). The index in ZI_1, N_2, U_2, etc. stands for the delta value (see ibid. definition *c)*; in cases index is omitted, any for the specified component valid delta value is implied, i.e. for ZI component delta values 1, 3, 5, etc., for V, U, and N components 2, 4, 6, etc, for O component 3, 5, 7, etc., and for I, Z, and S components 1.

P = {
 ST → U-zi | N-v
  U → I-U | U-u | ε
  N → I-N | N-n | ε
  O → I | O-o
  I → O | ε
  Z → I-z | U-z | ε
  S → I-s | N-s | ε
  $I\text{-}ZI_1 \rightarrow Z\text{-}i\text{-}zi_1$
  $I^z\text{-}I \rightarrow S\text{-}i\text{-}i$
  $I^s\text{-}I \rightarrow Z\text{-}i\text{-}i$
  $I\text{-}N_2 \rightarrow S\text{-}i\text{-}n_2$
  $I\text{-}U_2 \rightarrow Z\text{-}i\text{-}u_2$
}

The following examples show the production sequences of the geometric symbols for the decimal values 100, 1000, and 2051:

100 = I-O-ZI
ST → U-zi
U → I-U = I-U-zi
U → ε = I-zi
I → O = O-zi
O → O-o = O-o-zi
O → I = I-o-zi

$1000 = I\text{-}I\text{-}I\text{-}I\text{-}N\text{-}V_4$
ST → N-v
N → I-N = I-N-v
N → I-N = I-I-N-v
N → I-N = I-I-I-N-v
N → I-N = I-I-I-I-N-v

$2051 = O_9\text{-}Z\text{-}I\text{-}ZI$
ST → U-zi
U → I-U = I-U-zi
U → ε = I-zi
I-ZI → Z-i-zi = Z-i-zi
Z → I-z = I-z-i-zi
I → O = O-z-i-zi

All the component combinations, which cannot be achieved using the production rules are ungrammatical, e.g. N-I-I-V of which the grammatically correct form is I-Z-I-I-V. Further examples of ungrammatical combinations include S-U, Z-N, Z-O, S-O, S-Z, Z-S, S-I-Z, and Z-I-S.

8 Conclusions

The main benefit of a numeral system with quantitative, in addition to qualitative, equality between symbols and values is in its universality and increased safety with respect to information sustainability. A reconstruction of the encoded data can be undertaken without a need for historic knowledge about the meaning of the particular symbols. The shape of the symbol itself contains sufficient key for decoding both the spacial orientation (writing/reading direction) of the recording, and the recorded data itself.

Due to the geometric redundancy of encoded digits, further benefit concerns a possibility of limited error correction: the context-sensitivity of delta components can provide additional protection against information losses through restoration by reapplication of production rules with intact neighboring components.

References

1. René Descartes (1641), Meditationes de prima philosophia.
2. Gottfried Wilhelm Leibniz (ca. 1680), Hauptschriften zur Grundlegung der Philosophie, Volume 1, Part 1: Schriften zur Logik und Methodenlehre. Chapter 4: Die Methoden der universellen Synthesis und Analysis.
3. Weber, A. (1863), Indische Studien 8, Leipzig.
4. Chomsky, Noam (1959), On certain formal properties of grammars, Information and Control, Volume 2, Issue 2: 137–167.