http://letters.isomorph.it
an online Journal

Isomorph Letters A: Physics of Information, 1 (2007)

ISSN: 1973-2260

On the mathematical structure of messages and message processing systems

H. Grassmann

Isomorph srl, Visogliano 9R1, I-34013 Duino-Aurisina and
Department of Physics, University of Udine, Via delle Scienze 208, I-33100 Udine
grassman@fisica.uniud.it

Received 1 September 2007, accepted 10 October 2007
Available online 16 October 2007

In this paper we want to explore the prospects for a physics of information. In the first chapter we will summarise the present situation. Chapter 2 demonstrates the present limitations of information processing. Chapter 3 suggests, that these practical limitations might be traced back to limitations in the underlying theoretical concepts. Chapter 4 suggests a more physics oriented way of defining concepts like “program” or “complexity”. Chapter 5 explores, whether the concepts discussed in chapter 4 can be applied to the processing of physical messages, for instance video images. In a first step in chapter 5.1 we note a difference between messages in information theory and physical messages – physical messages can always be finite. In chapter 5.2 we note, that messages can be considered vectors and that already this very simple statement is leading to relevant practical consequences, leading to a new type of OTP cipher. Chapter 5.3 discusses some general mathematical properties of vector spaces and creates a connection between the mathematical concepts and the physical observables (measurement). In chapter 5.4 we try to gain first hints, of how the mathematical structure of the visible world might be used in order to calculate the description of an image to mathematical precision.

1) Information Theory and Physics : the present situation

Messages (like books, or electric signals) and message processing systems (like computers or brains) are part of the physical world. Therefore physics should be able to make some kind of statements on messages and message processing systems.

Some first attempt towards a physics of information was done by Szilard [1] in 1929. He concluded, that storing one bit of information would necessarily dissipate the energy 1/2k·T, corresponding to the thermal energy of one degree of freedom (with T absolute temperature, k Boltzmann constant). Landauer later claimed, that it is rather the erasure of information, which dissipates energy [2]. Bennett later concluded diversely, that energy must and must only be dissipated, when a system inputs information from the outside world ("reversible computing") [3].
A possible interpretation of these disagreements is, that Szilard, Landauer and Bennett have not clarified their basic concepts - like "information" or "outside world" - in an universal and fundamental way, but have rather explained them in the context of some particular scenario. While their findings were conclusive in this respective scenario, they did not apply in general.

Nowadays information processing is discussed in the framework of information theory, as founded by Shannon [4] and more recently in algorithmic information theory [5,6,7]. Shannon's original studies concerned mainly the transmission of messages, in particular the effect of noise in the signal line on the message. His theory is therefore a probabilistic theory. In this context, the amount of information of a message is measured by the probability of the "letters" of the message to occur. For example the amount of information of a digital message is equal to the length of the message, if "1" and "0" occur with probability 0.5.
Modern algorithmic Information theory still follows this line of thought: similar to the definition of the "amount of information of a message" as the "length of the message" (as explained above), the "complexity of a program" is defined as the minimum length of this program.
[6].

But although modern information theory is a very successful theory with many important applications in IT technology, there is no connection to physics: Shannon has claimed the information content of a message to be equal to the thermodynamic entropy of the message [4], and if this claim were correct, an important connection between physics and information theory would exist. But Shannon's claim is not correct, since the entropy of a message strongly depends on its temperature, while its amount of information does not. For instance, the amount of information of a newspaper does not change, when we put it into a refrigerator, while its entropy does change.
Even more evident is the non-physicality of the concept of "program complexity": the complexity of a program is not computable. This means, that one cannot even determine it in a Gedankenexperiment, which shows the large conceptual gap between information theory and physics.
Last but not least, the most relevant feature of a message in the real world is the meaning of the message. While instead Shannon explicitly stated, that the goal of his work was not to study the "meaning" or "semantics" of messages at all.
In order to create a connection between information theory and physics, it is instructive to ask, how modern physics functions, how it proceeds:
It proceeds by connecting physical "objects" (space, time, microstates etc) to certain mathematical structures. Next, from these mathematical structures, the relevant laws of physics are deduced.
For instance, it is found that space and time can be described by four-vectors, spanning a vector space with certain properties. For instance, the physical laws are assumed to be invariant under space translations and rotations, and this is sufficient to deduce conservation laws for energy, linear- or angular momentum. For which reason a large part of modern

physics is rooted in group theory: >> The entire theory of group representations is built on homomorphisms of abstract groups (often symmetry groups of physics) to groups of linear operators or matrices on vector spaces (for our purposes, spaces of physical states) …the interest in group theory, therefore, centres on the realization of group transformations as linear transformations on vector spaces of classical and quantum physics<< [8].
One possible approach (there may be others, too) for including the field of "information" into physics might therefore be, to explore the mathematical structure of physical messages and of physical message processing systems.

2) A practical limitation of the state of the art theory

In general, human beings can describe what they see in an image, for instance an image taken by a video camera. Due to the limited resolution of the video camera, or due to other technical problems like electronics noise, there are images, where one given person will not be able to clearly identify the content of the image, or where different persons will come to different conclusions. But this kind of technical problem is not what we want to discuss here (while it rather resembles the concerns of Shannon's work).
In general, computers cannot describe what they see in an image, even if the image is of good technical quality so that it can be unambiguously described by a human being: >>The field of computer vision can be characterized as immature and diverse … there is no standard formulation of "the computer vision problem". Also, and to an even larger extent, there is no standard formulation of how computer vision problems should be solved. Instead, there exists an abundance of methods for solving various well-defined computer vision tasks, where the methods often are very task specific and seldom can be generalized over a wide range of applications<< [9].

Let us for the ease of discussion assume, that the video image be black and white only, for instance it might be the result of a difference filter or edge detection routine. The image then can be written as a chain of "1" and "0", describing whether a pixel is black or white, or as a digital number, a_i. This number is the input message for the information processing system, T, which will transform the input message into an output message, b_i, consisting in the description of the image.
The state of the art technique tries to approximate this transformation T over a limited range of values a_i [10]. But there is neither a fundamental and universal method for determining this range, nor can one quantify the quality of the approximation.

One way of obtaining the correct T, without approximation, would be to create a complete list of all possible images with their corresponding description. (One would consider a description to be correct, if it could not be distinguished, whether it comes from a computer or from a human being.)
If for instance the video image has 10^6 pixels, this list would contain $2^{1.000.000}$ elements. While the use of such a list is surely possible in a Gedankenexperiment, its technical realisation is impossible.
Still it remains a fact, that the brain does have this T at its disposal, and that T can therefore be represented with limited hardware means. One possible explanation could be, that the

visual messages processed by the brain do have a certain mathematical structure, and that the brain makes use of this mathematical structure in order to express the lookup table T by a relatively small number of elements.

3) A simple computing machine

The practical limitations described in the previous chapter might reflect limitations in the underlying theoretical concepts, like “Turing machine” or “program”.
Figure 1 shows a hardware device, or “computer”, which is performing a calculus, in the example of figure 1 it multiplies the input number, *a*, by a factor of two, $b=2\cdot a$. The input to the device is organized as a digital number of n bits, transmitted on n signal lines.
The input signal travels through the device of figure 1 at the speed of light, there is no algorithm and nothing which resembles a program, or which has properties attributable to a program, for instance a “program complexity”.

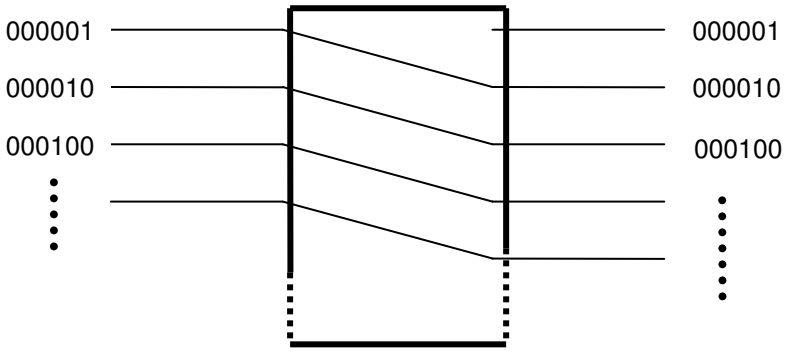

Figure 1: device performing the operation $b=2\cdot a$ on digital numbers.

In figure 2 we show a device, which computes $b=a^2$. In difference to the device of figure 1, the device of figure 2 operates on canonical numbers. By “canonical numbers” we mean the representation of the numbers (0, 1, 2, 3, 4, …) as (0, 1, 10, 100, 1000, ….).
(Note for the later discussion, that canonical numbers can be superposed by means of an logical-or, without changing them. For instance, the superposition of the canonical numbers 0001 and 0100 would be 0101, which still can be unambiguously identified as superposition of 0001 and 0100. While instead the superposition of the digital numbers 0001 (“1”) and 0100 (“4”) 0101 (“5”) could not be identified as a superposition of the two numbers “1” and “4”, since it has now a new meaning of its own, namely “5”. The reason for this is, that canonical numbers are forming the base of a vector space.)

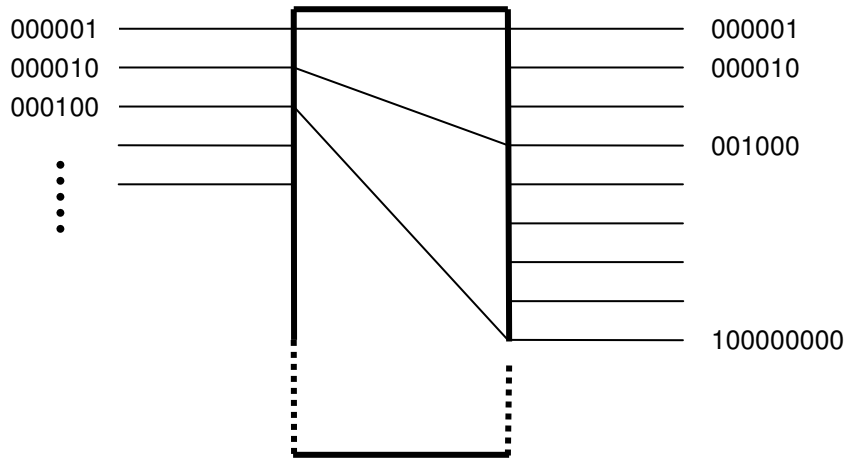

Figure 2: device performing the operation $b=a^2$ on canonical numbers.

Again, it is difficult to say, what the "program" is, and again the calculation is performed in one step - on the light cone of the message. In addition, the device of figure 2 operates on any number of input numbers at the same time. This is a notable difference to a general Turing machine, which can process only one number at a time.
The use of canonical numbers is nothing unusual, but it rather is state of the art already: each RAM (random access memory) has a multiplexer, which transforms digital numbers in canonical numbers, and the memory of the RAM uses these canonical numbers as its input.
In the following we will try to adopt a description of "computing machine", which is based on something which can be measured or counted, like for instance the number of lookup table elements.

4) The elementary entities of algorithms

Let us study for example the addition of two digital numbers, *a* and *b*, of n bits each, $a+b = c$. We can perform this operation by means of a lookup table with $2^n \cdot 2^n = 2^{2n}$ elements. Such a lookup table allows to perform the addition of two numbers with only one step of operation.
Instead, we can perform the same operation in a more conventional way, as indicated in figure 3 for the case n=2. Again the operation can be executed without organizing it in a sequence of time steps or operational steps, in the sense that the input signal can travel through the device on its light cone or without the use of a clock.
Each of the fundamental adders (adding two bits) is representing a lookup table. The addition of the two numbers, *a* and *b*, is now performed by a total of $2 \cdot n$ lookup tables with 4 elements each, for a total of $8 \cdot n$ lookup table elements.
We have therefore substituted a lookup table with 2^{2n} elements by a system of lookup tables with $8 \cdot n$ elements.

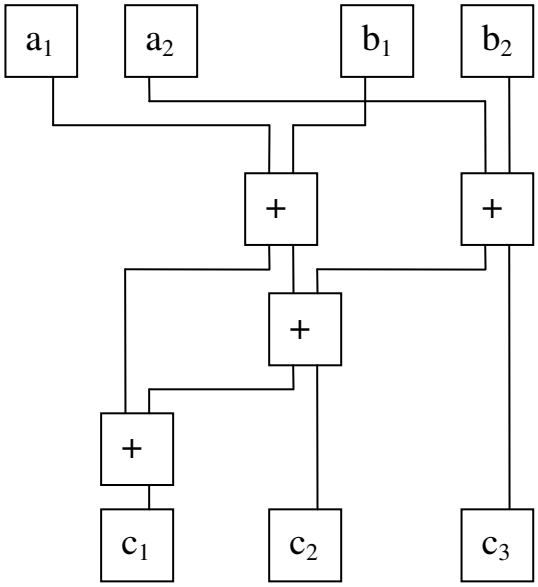

Figure 3: Addition of two digital numbers of 2 bits each by means of fundamental adders.

We have achieved this by breaking down the calculus to its elementary "constituents" (the fundamental adders), which are embedded into a (mathematical) structure, corresponding to the calculus under consideration.
We have paid a price for this achievement: The device in figure 3 can process only one input at a time.
A similar argument can be made for the operation "multiplication", and for many other numerical mathematical operations, which can be performed by a computer.

We can further reduce the number of lookup table elements in the device of figure 3, if we perform the addition in steps, storing the intermediate results. Now one and the same fundamental adder must be used repeatedly, and the operation must be organized in time steps. This would now be a "program": a program is the repeated use of a lookup table.
The complexity of the program could be measured by the number of repeated uses of a lookup table. In correspondence, the complexity of a function would be the number of lookup table elements, which are needed in order to perform the function in one step of operation (as in figure 3).
This would be a physics definition, since lookup table elements are well defined objects, which can be counted in a reproducible way.
The reason why we can now give an absolute number for a program complexity is, that we are referring to elementary constituents. Once chosen, they provide an absolute scale. As in physics, where one refers either to the quarks and electrons or instead to the atoms as fundamental, depending on the problem to be solved, one may also here choose different entities as fundamental constituents, for instance "addition of two bits" and "multiplication of two bits", or instead "addition of four bits" and "multiplication of four bits" or similar. However, once we have chosen what we intend as "fundamental", then we can express the complexity of a program without having to refer to a certain programming language.

5) The mathematical structure of physical messages

For the time being we discuss digital messages.

5.1) physical messages can always be finite

A first mathematical feature of physical messages is their being of finite length. Messages of infinite length are simply not possible in our universe, since the energy of the universe, the number of atoms, as well as all other physics quantities are finite – therefore there are no resources which permit the creation of infinitely long messages. There is also no need for infinitely long messages: Since the energy of the Universe is finite, and since the number of microstates of the universe is linked to its energy by the Boltzmann formula, S=k*lnW (k Boltzmann constant, S entropy, W number of microstates), also the number of possible microstates of the Universe is finite. Therefore each and any possible physical state of the Universe can be described by a message of finite length.
The finite length of physics messages is presumably a first important difference between physics and mathematics – in mathematics and also in classical information theory, messages of infinite length are possible, and play an important role in the so called halting problem. We suppose, and this needs to be verified, that for messages of finite length, the halting problem does not exist.

5.2) messages as vectors – a practical example

"0" and "1" are the elements of the remainder class modulo 2. The remainder class modulo 2 is a field of dimension 2, referred to as F_2. And therefore digital messages of n bit length can be considered elements of the arithmetic vector space F_2^n, which we may refer to as "message space", as is discussed in [11].
Already this very simple and basic statement is leading to a practical application, a new kind of OTP cipher (one time pad). OTP ciphers are the only mathematically secure ciphers, as has been shown by Shannon [12]. (While instead all public key ciphers are only relatively secure, and can in principle be broken, for instance by means of a quantum computer.)
But unfortunately, traditional OTP ciphers suffer from the problem, that one needs to submit a key on a secret line, and this key needs to be as long as the plain text, and this is not very practical, because if such a secret line does exist, then one can as well send the plain text on it, without encrypting it [12]. There have been many attempts by the experts of the field to overcome this limitation, and it was shown, that it is in principle possible to construct OTP methods, without the need for a secret line [12]. While these attempts were successful from a theoretical point of view, they were totally unpractical. Demonstrations of these methods and a more detailed discussion can be found in [13].
If we take into account the vector nature of messages, the problem can now easily be solved: note, that the addition modulo 2 has an addition table, which is isomorphic to the truth table of the exclusive disjunction, also referred to as "exclusive-or", or "XOR", as it is used in the OTP ciphers. Therefore, encrypting a (digital) plain text with a key (of equal length) by means of an exclusive-or can be seen as a vector addition in the message space.
It remains true, that for each message of n bits one needs a separate key of n bit length for encryption, and it is also true that in order to be able to transmit for instance m different messages, one needs to first transmit a set of m keys.

However, it is also true, that all of the 2^{1024} strings of 1024 bit length can be formed from a total of only 1024 base vectors. It is therefore sufficient to handle once a set of 1024 base vectors to the prospective receiver of the encrypted messages (1024 strings of 1024 bits each fit on a floppy disc). All keys, which are possible, can be created from these base vectors.
In different words: taking into account the vector nature of messages, we can substitute the submission of 2^{1024} strings with the submission of 1024 strings.
In order to encrypt a message, one can now form a key from a combination of these base vectors (by means of an exclusive-or), and send a listing of the base vectors used (not the vectors themselves) together with the encrypted message. This is described in more detail in [13].
This method is secure, as long as a crypto-analyst does not obtain a certain number of plain texts and their corresponding cipher texts. The method becomes absolutely secure, if one in addition encrypts with the same method also the list of base vectors.
If for instance we form the key by choosing different combinations of 512 vectors, then we can form 10^{300} different keys, which means in practice, that we can submit any amount of information (possible in our Universe) without ever using the same key twice.
Not only is this method mathematically secure (since each of the encrypted key lists has the same probability to occur), so that the cipher cannot even be broken by a quantum computer, but the method is also extremely fast - even with a cheap and popular Playstation 3 (Sony), 1024 bits can be encrypted in one single step of operation, allowing also the encryption of high speed date lines.
This shows, that a problem, which could not be solved by the conventional methods, becomes trivial just by considering that messages are vectors. Or said in more general, this example shows, that it is justified to try understand better the mathematical properties of messages.

5.3) some properties of message vectors

We next need to connect the mathematical structure to the real world by means of some postulates. We also need to define some measuring procedure – at least we need to define, when two messages are identical:
We assume an information processing system, T, which processes an input message, *a*, and transforms it into an output message, *b*, $b=T(a)$. We postulate: *a* and *b* are identical messages with respect to T, if $b=T(a)$ and $a=T^{-1}(b)$.
(This postulate is inspired by experience: the best, if not only way to verify whether an English text has been translated correctly into Chinese, is, to have it re-translated by an independent translator, and compare whether original and re-translation are the same.)
And we define what “information” is by postulating, that T conserves information, if it is injective, otherwise T erases information.
It seems furthermore reasonable to state, that identical messages must contain the same information.
We will assume throughout this chapter T to be injective.

For the particular case of T being linear, the following statements are then true: since the input messages form a vector space, U, also the output messages are forming a vector space, V, and U is isomorphic to V. T itself can then be expressed as a matrix, built from vectors, which are again isomorphic to U and V.

With reference to above postulates for identical messages it follows, that input- und output messages as well as the information processing system itself are composed of identical messages and contain the same information. Which means for instance, that in order to function, such an information processing system must first contain already all the information which it is supposed to process in the future.
For the case of a linear T this situation - namely input and output messages being identical - can rather intuitively be appreciated, the linearity of T helps to "imagine" the situation in a more intuitive way (this is at least true for the author). Note however, that the identity of input and output messages is not a consequence of linearity, but results from T being injective. One would therefore expect, that also non-linear information processing system do nothing else, and cannot do anything else but "re-labelling" messages.

If T is not linear: Since n is finite, one can map the 2^n different digital input vectors onto a 2^n dimensional base, this base can be formed by canonical numbers. Also the output messages can be considered a 2^n dimensional base. Now the execution of T can without approximation be substituted by a mapping D between these two bases. And a mapping between two bases can always be extended into a linear function.
In practical terms, "extending D into a linear function" means: we can either have D process several different (canonical) input vectors subsequently, or concurrently – it does not make a difference. Provided only, that the input is canonical, D can process any number of input numbers at the same time, or : $D(a+b) = D(a) + D(b)$.
If we create a lookup table in a RAM of a conventional computer instead, then it is true, that the RAM can accept only one input number at a time. But this is not a fundamental property of memories or lookup tables, but the reason is simply the multiplexer, which needs to transform the digital numbers into canonical ones. It is the multiplexer, which can process only one number at a time. From a physical and presumably also from a technical point of view it is possible to built memories, which allow to read out several elements concurrently.
We obtain a model of the ideal information processing system: it would be a "linear computer", operating on 2^n data lines and it would have a memory of size 2^n elements. This computer could process any number of operations in one single step. It would in this sense emulate a quantum computer (but without the problem of entanglement). A future study should clarify, whether this means, that the fundamental physics properties of information processing are identical to the ones of quantum physics.

5.4) the mathematical properties of image processing systems

We assume an image, *a*, as input for the information processing system, T, in the form of a black ("1") and white ("0") pixel matrix with a total of n pixels. For instance, this could be the output of a filter, like an edge extraction- or difference filter applied on a video picture. Here and in the following the expression "image" will always refer to black and white array resulting from the filter, while the original video picture before processing will be called "video picture". As an example the image of a rectangle is shown in figure 4.

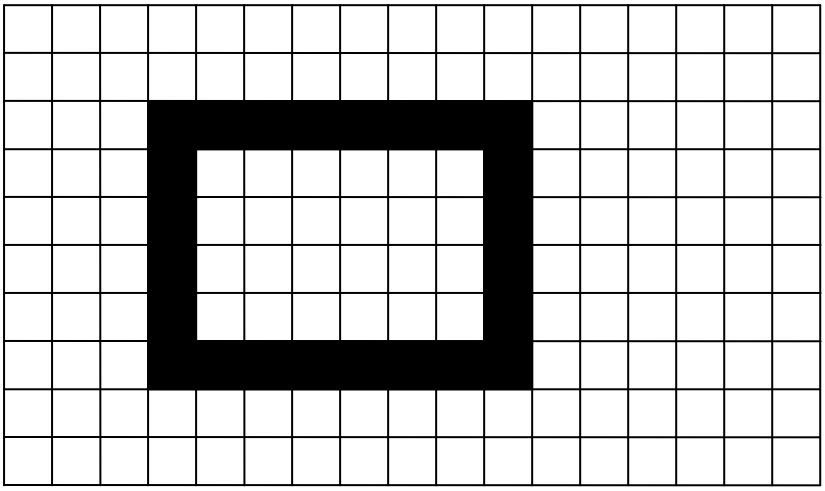

Figure 4: a rectangle after having been processed by a “logical-or” difference filter.

Since there are 2^n different images possible, T can be described by a lookup table of 2^n elements, which provides for each input vector the corresponding output – a description of the image. A hardware model is shown in figure 5: the input image a_i -, which is a string of n bits - is used as an address to read out the corresponding description of this image, b_i.
If the image has one million pixels, the lookup table would need to have $2^{1.000.000}$ elements. Along the lines of the previous discussion we want to reduce the number of lookup table elements without approximating T.

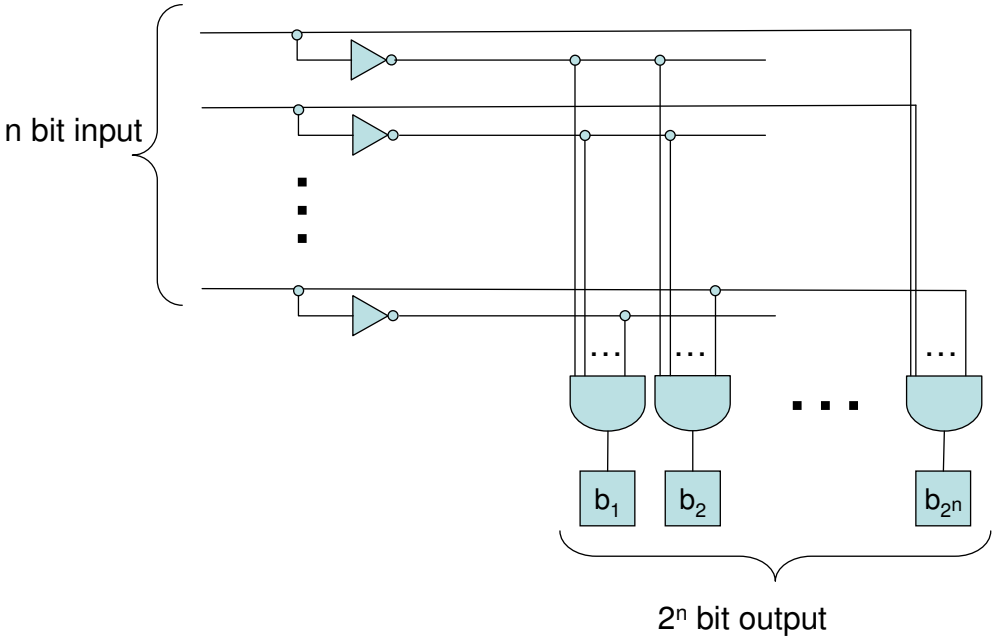

Figure 5: a RAM provides a complete list of all images (with n pixels), and a list of their descriptions.

5.4.1) the mathematical properties of images

The set (0,1) with the two binary operation “and”, Λ, and “or”, V, and a unary operation “not”, ¬, forms a Boolean Algebra. Correspondingly and under the same operations, the images (with n pixels) form a 2^n-element Boolean Algebra.

Also the physics objects, which we want to identify in the image, as well as their descriptions, form a Boolean Algebra (because the power set of any set forms a Boolean Algebra). The main problem in the construction of a pattern reconstruction system is now, that these two algebras are not a priori isomorphic for two reasons:

a) if there are m objects, they form a 2^m-element algebra, not a 2^n-element algebra, like the images do.

b) a logical-or between the images of two rectangles would look like in figure 6. But an image of a photograph showing the two physical rectangles would look different, like in figure 7.

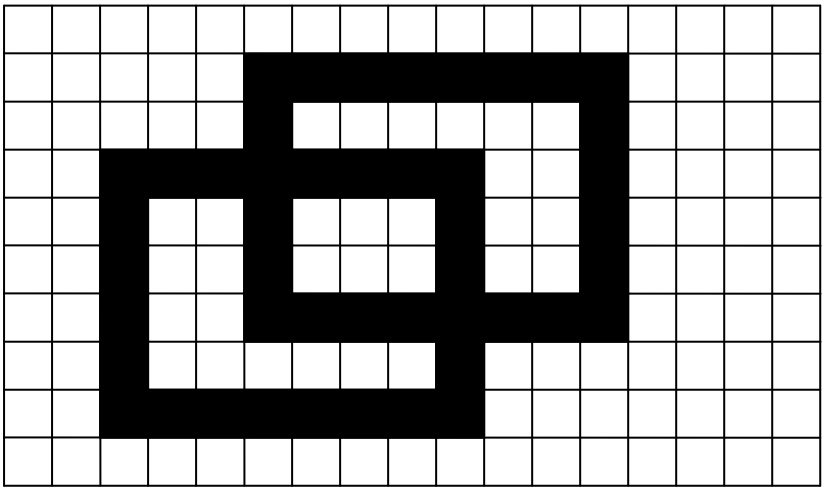

Figure 6: superposition of two images, each of them showing one rectangle, respectively.

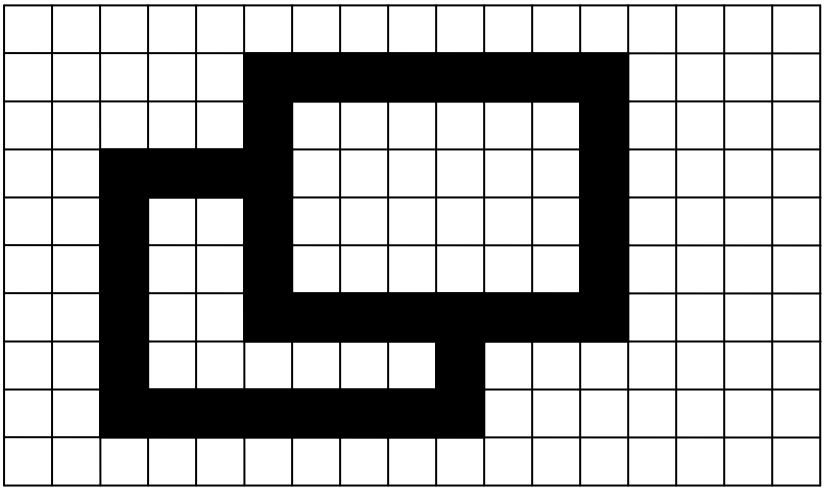

Figure 7: image of a photograph of two physical rectangles.

Problem (a) can be solved with the tools of Boolean Algebra, which allows to define atoms: an atom in a Boolean Algebra is a nonzero element *a* such that there is no element *b* with $0 < b < a$ [14]. If we define the images of the m objects as atoms, the images become a 2^m-element Boolean Algebra, which is isomorphic to the Boolean Algebra of the physical objects.

Problem (b) is not really a problem of information processing, but a problem of representation: physics objects exist in a three-dimensional space, while the images are two dimensional. If the physics world were two-dimensional, overlap of objects would not occur.
In a first step we simply limit our discussion to objects, which do not have overlapping contours – that is, we arrange m objects in the image in such a way, so that their contours do not overlap nor touch, and make only photographs of one or several of these m objects.
(Note, that in this context we count two objects, which are at different positions but otherwise identical as two different objects, because their position is part of their description.)
It is then possible to associate to each (atomic) image a_i its correct description b_i, $b_i = D(a_i)$. And it is by construction true that the image of a photograph of two objects will be the same as the logical-or of two images, each one showing one of these same objects: $D(a_i \vee a_j) = b_i \cup b_j$. D is then an homomorphism, and since D is bijective, it is also an isomorphism.
In the next chapter we will propose the structure of a corresponding device and discuss some of its properties.

Before proceeding to the next chapter, we note: we are now applying on the images the logical-or of the Boolean Algebra, while instead we have used previously the exclusive-or to add vectors. This is not different on a fundamental level, and it does not mean, that the laws of information processing are totally different for cryptography and pattern recognition, because “every Boolean algebra $(S, \wedge, \vee)$ gives rise to a ring $(S,+,*)$ be defining $a+b = (a \wedge \neg b) \vee (b \wedge \neg a)$ … and $a*b = (a \wedge b)$.” [15]

5.4.2) Calculating image descriptions

In order to be able to identify m (non-overlapping) objects in an image, the device of figure 5 needs to have 2^m lookup table elements, provided the image shows nothing but only one or several of these m objects. If the image may show other things as well, be it additional objects or noise, additional lookup table elements are necessarily needed in the arrangement shown in figure 5. This is, because the device by construction necessarily verifies the state of each single pixel in the image.
Following the discussion of the previous chapter we can simplify the device, as is shown in figure 8: again, the input consists of the image vector of n bits. But now, each of the logical units is connected to only those signal lines, which correspond to one of the m image objects, a_i, and its output activates the corresponding description, b_i.

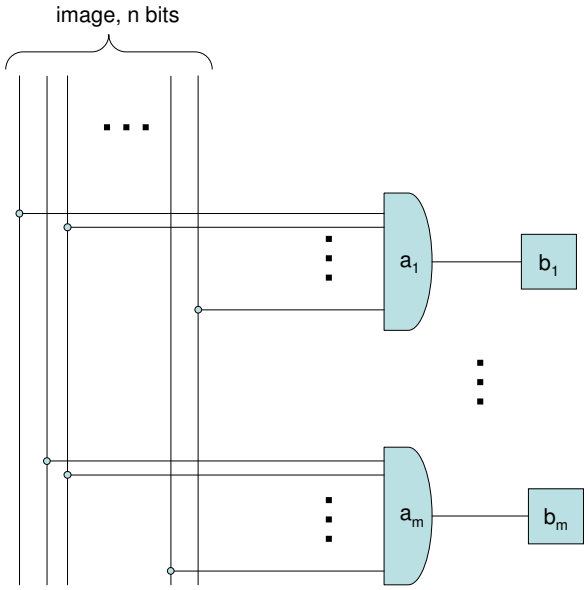

Figure 8: associative RAM for identifying rectangles.

In difference to figure 5, the coincidence units verify only the state of the pixels, which correspond to one of the m images. The device in figure 8 could also be described as an "associative RAM".
The device in figure 8 needs only m lookup table elements in order to describe all images, which contain one or several of the m objects to be identified, compared to a much higher number of lookup table elements needed if we use the "complete lookup table" represented by the device in figure 5. Therefore, we have reduced the number of lookup table element, needed to find rectangles in images by making use of the mathematical properties of Boolean Algebra, and we did not perform any approximation.
The device in figure 8 is essentially having the same function as the device in figure 3 – the device in figure 3 calculates a+b=c, the device in figure 8 calculates the description of images. Both devices pay the same kind of price for reducing the size of the lookup table elements: they are not able anymore to process more than one input vector at a time in a meaningful way.
The associative RAM looks very simple, and with respect to this simplicity it is surprisingly intelligent:

a) the system can easily be extended. Let us assume that the m objects where all rectangles. If we want to identify also triangles, and if there are k different triangles possible, we simply need to add k logical circuits, for a total of m+k circuits and lookup table elements. This is different from traditional pattern reconstruction methods, where the "complexity" of the program or device increases exponentially with the "complexity" of the problem[1] [10].
b) the system can resolve ambiguities: if the smallest rectangle and also the smallest triangle are just one pixel, then an image showing one single pixel "on" will be described as a list, with the elements "rectangle" and "triangle"

[1] We have here used the term "complexity" in the same way as is used in [10].

c) as already Plato has noted, human beings are able to identify an object for instance as being another human being, even if they have never before meet this particular person. Therefore, as Plato put it, there must exist general "ideas" of objects, known to the mind. This concept is known in Boolean Algebra under the name of "ideal": an ideal of the Boolean algebra S is a subset J such that for all *a*, *b* in J we have *a* V *b* in J and for all *x* in S we have (*a* Λ *x*) in J [14]. The associative RAM becomes able to find ideals, if we simply require that not all pixels of a certain object must be "on", but only a certain fraction of them. In particular this also allows to identify partially occluded objects – still within the framework of Boolean Algebra.
d) As mentioned before, Boolean Algebra allows to precisely define "atoms", or fundamental entities of the calculus. In state of the art pattern reconstruction procedures such "atoms" do not exist, or are not defined with mathematical precision[10].
e) Boolean Algebras are already partially ordered, and subsets can be brought to total order by means of linear extensions [15]. This allows to construct images of objects from smaller subsets, by means of set theory. The decisive practical importance of this fact will be demonstrated in some more detail in the next chapter.

We remind briefly the meaning of "ordered set" by quoting again from wikipedia:

>>In mathematics, a total order, linear order, simple order, or (non-strict) ordering on a set *X* is any binary relation on *X* that is anti symmetric, transitive, and total. This means that if we denote one such relation by ≤ then the following statements hold for all *a*, *b* and *c* in *X*:

if $a \leq b$ and $b \leq a$ then $a = b$ (anti symmetry)
if $a \leq b$ and $b \leq c$ then $a \leq c$ (transitivity)
$a \leq b$ or $b \leq a$ (totality or completeness)

A set paired with an associated total order on it is called a totally ordered set, a linearly ordered set, a simply ordered set, or a chain.<<

Here $a \leq b$ precisely when $a = a \wedge b$.

An image showing a straight line, for instance, can be constructed from small line sections, by creating a totally ordered set. We begin with one of the line sections, and refer to it as element *a*. We search for another line section, *a'*, the first pixel of a' should be a neighbor of the last pixel of *a*, and both line sections should have the same direction. We create $b = (a \vee a')$. We next search for a line section b', again with its first pixel neighboring the last pixel of *b*, and having the same orientation as b, and create $c = (b \vee b')$, and so forth, till we do not find line sections anymore, which could be united with the set. Depending on the details of the search, a straight line or a bent curve results, showing the structure of a totally ordered set, under above definitions.
Correspondingly, straight lines can be ordered into, for instance, rectangles.

5.4.3) An experimental hardware device

In order to built a functioning device, two obstacles need to be overcome:

a) we have decreased in the previous chapter the number of lookup table elements to be equal to the number of objects to be identified, which in practice will still be a very large number, too large for a real hardware implementation. A practical number of lookup table elements can be obtained, if we operate only on small sub-images, making use of the possibility to reconstruct larger images as totally ordered sets from smaller elements.
b) For our studies we did not have an associative RAM at disposition. We have therefore emulated an associative RAM in a conventional RAM by means of a software procedure.

The device and procedure following from these requirements is described in the following:

We consider arrays of 5*5 pixels. A computer program creates all line sections, which pass through the centre-pixel. For each line section the description of the line section – in this case the angle of the line section – is known to the computer program. The projection of each line section leads to a corresponding pixel pattern, as shown in figure 9.
This pixel pattern is a message vector, or digital number, a_i, in the example of figure 9 it is "00000000110110011000000000". This number is used as address to an element in the RAM. The description (in this case "angle of 30 degree") of the line section, b_i, is written into the corresponding storage element. This is indicated in figure 10.

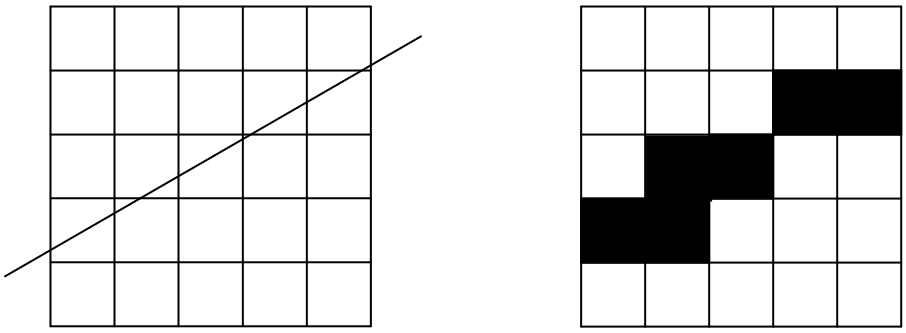

Figure 9: a line section of known properties is projected onto a 5*5 pixel array. In this example, the line section may have an angle of 30 degree.

If we now need to analyze an image, we invert this procedure in order to "translate" the image in a list of descriptions of line sections: we use each pixel of the image as center of a 5*5 pixel array, and translate its pixels into a digital number, a_i. This number is used as address to the lookup table in the RAM, where we will find the description of the line section.

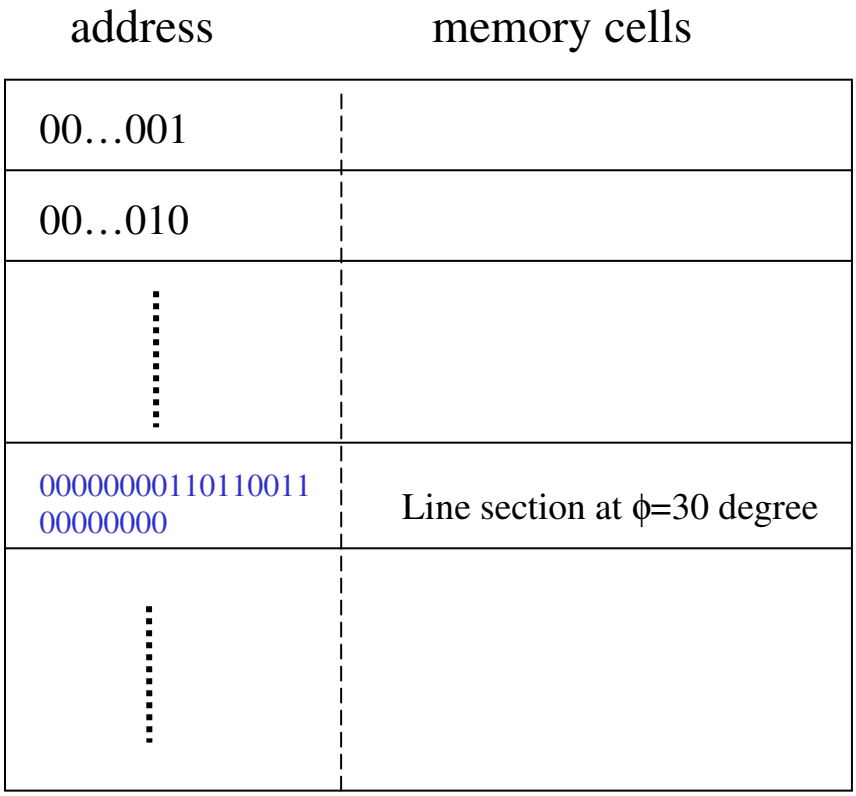

Figure 10: each line segment corresponds to an address in the RAM, and the corresponding storage element contains a description of the line segment. Empty storage elements are possible.

In addition, one can also create sections of circles, ellipses etc, and deposit the corresponding information in the relevant lookup table element.
In this way we will find only 5*5 pixel arrays, which show a line section, and nothing else. If only one of the white pixels from figure 9 would be black due to noise, for example, the 5*5 image could not be identified as a line section.

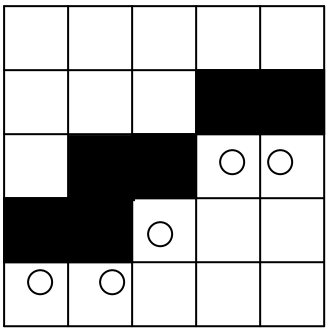

Figure 11: a line section of 30 degree, as in figure 9, where again the black pixels correspond to the line. Some of the “white” pixels, marked with circles are required to be “white”, while all the other pixels may be either “black” or “white”.

Therefore we choose some number of white pixels, as indicated by circles in figure 11, in our computer created image. And define, that all ai, which have a “1” at the position, where in figure 11 there is a black pixel, and which have “0” at the position, where in figure 11 is a circle, are to be considered line segments.
Since in our example there are 14 pixels left, which can be either black or white, there are 2^{14} different addresses a_i, which all correspond to the same description “line segment with 30 degree”. This description is therefore copied to all of the corresponding storage elements.
The situation in figure 11 serves only as a typical example. More sophisticated procedures can be applied in addition.

As a result, a black and white image of 1.000.000 pixels can be “translated” into a list of line elements be means of at maximum 1.000.000 calls into the RAM. How this is organised best, will depend on the properties of the computer, for instance one will in general perform calls for only those central pixels, which are black, which makes for a total of typically 10.000 calls per image. This amounts to a processing time of order of 1 millisecond in today’s computing devices. The procedure could also be implemented on dedicated hardware, because in principle the processing of all pixel arrays could occur concurrently, in which case the processing time could be further reduced.

Once the image is translated in line segments or other elements, these can be ordered into totally ordered sets, representing larger size structures.
Following this line of thought, the spin-off company Isomorph [16] has created a first industrial application of a pattern reconstruction system, which turned out to function very well, with results superior to the state of the art technique, it is described in detail in [17].
The technique is sometimes referred to as “linear computing” or “isomorph computing”, and some more examples are shown in [16].

6) Outlook

In this paper we have tried to show, that a study of the mathematical properties of messages and message processing systems can lead to relevant results. In this paper we have not discussed the second pillar on which the physics of information must rest, namely Thermodynamics. One would expect, that a complete physics of information will combine Thermodynamics with the mathematical structure of messages and message processing systems.
Only in order to show, that such a discussion could be interesting and worth the effort, we present a short example: let us assume an injective information processing system. It does not create information, nor does it erase information. According to Bennett, it can operate without dissipating energy. Now let us set all output vectors to the same value, for instance “0”. Now the system is not injective anymore, but one cannot see any reason, why it should necessarily dissipate energy now. One would conclude, that for creating an information processing system energy needs to be dissipated, not for operating it.
This would require a new discussion of the Maxwell Daemon, and Szilard’s work would have to be re-evaluated. Indeed G.Luhn and collaborators are working on this question, based on some surprising and totally new concepts and ideas [18].
As a result one might better understand, how natural information processing systems (brains) form spontaneously.

7) Conclusion

Messages and message processing systems have a mathematical structure, which is imposed on the elementary entities of the calculus. It allows to discuss information processing in terms of algebra and set theory. Interpreting this mathematical structure in terms of physics observables one can arrive at a physics of information.
This is different from the traditional approach, which is based on a probabilistic theory, and has no connection to physics.
The algebraic approach opens the prospective to create artificial information processing systems, which are able to process also messages from the physics world, like video images, in a fundamental and universal way – it allows to calculate the description of an image analytically.
As a consequence, also a new generation of computing systems becomes possible and unavoidable. These computing systems will have direct access to a large amount of intelligently organized storage (lookup table system), and they will operate on much more than 32 or 64 bits at a time (canonical operation). The first example for this development is the CELL processor from IBM.

References

[1] L. Szilard, *über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen*, Zeitschrift für Physik **53**, 840-856.
[2] R.Landauer, *Irreversibility and heat generation in the computing process*, IBM Journal of Research and Development, vol. 5, pp. 183-191, 1961
[3] C.H.Bennett, *Logical reversibility of computation,* IBM Journal of Research and Development, vol. 17, pp 525-532 (1973);
C.H.Bennett, *The thermodynamics of computation – a review*, Int.J.Theor.Phys. 21, pp 905-940 (1982);
C.H.Bennett, *Notes on the history of reversible computing*, IBM Journal of Research and Development, vol. 32, pp 16-23 (1988)
[4] C. E. Shannon: *A mathematical theory of communication.* Bell System Technical Journal, vol. 27, pp. 379–423 and 623–656, July and October, 1948.
[5] Ray Solomonoff, A Formal Theory of Inductive Inference, Part I", Information and Control, Part I: Vol 7, No. 1, pp. 1-22, March 1964

[6] C.S. Wallace, D.L. Dowe, *Minimum Message Length and Kolmogorov Complexity*, Computer Journal, Vol. 42, No. 4, 1999
[7] G.J. Chaitin, Algorithmic information thery, IBM Journal of Research and Development 21 (1977), pp. 350–359, 496
[8] Wu-Ki Tung, *Group Theory in Physics*, World Scientific Publishing Co., 1999
[9] http://en.wikipedia.org/wiki/Computer_vision
[10] Anil K. Jain, Robert P.W. Duin, Jianchang Mao, *Statistical Pattern Recognition: A Review*, IEEE Transactions on pattern analysis and machine intelligence, Vol. 22, No. 1, January 2000
[11] H.Grassmann, *Seemingly by chance* , Proceedings of the 13th Meeting at Bozen, "Entwicklung des Universums und des Menschen", pp 173-185editors I.Hosp, P.Mulser, K.Schredelseker, GCA-Verlag, Herdecke, 2003.
[12] Applied Cryptography, Bruce Schneier, Jon Wiley&Sons, 1996.
[13]http://www.isomorph.it/solutions/information-technology/cryptography/isomorph-cipher/Older-methods
[14] Monk, J. Donald, "The Mathematics of Boolean Algebra", *The Stanford Encyclopedia of Philosophy (Summer 2006 Edition)*, Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/sum2006/entries/boolalg-math/>.
[15] http://en.wikipedia.org/wiki/Boolean_algebra_%28structure%29
[16] www.isomorph.it
[17] Alessandro Prest, *A highly effective inspection system for automated semiconductor manufacturing at Infineon Dresden*, Diploma Thesis, University of Udine, 2007.
[18] G. Luhn: Forming Power - Message – Chance. Approach to a Triadic Information Concept, abstract submitted to the conference "Information Theory and Practice", Duino, 2007. http://www.isomorph.it/science/duino2007