Life as interplay of information and matter

Gérard Battail
Retired from ENST Paris

Code Biology Conference, Paris,
21–23 May 2014

Outline

- Defining information according to the engineering practice shows that it is basically an abstract entity. Being borne by symbolic sequences inscribed on a physical medium, information bridges the abstract and the concrete.
- Physical perturbations permanently degrade information-bearing sequences but error-correcting codes enable their regeneration, which conserves information if it is performed frequently enough.
- Genomes act on matter by instructing the assembly of living structures, including semantic feedback loops locked by enzymes which catalyse their own assembly.
- The successive establishment of semantic feedback loops induced constraints which endowed genomes with the system of nested error-correcting soft codes which protect them, and originated Barbieri's organic codes.

Information is an abstract entity

Only *discrete* information is useful in biology. The extension to continuous information is possible but involves mathematical difficulties and will be left aside. A discrete information is assumed to be represented by a *sequence* of symbols belonging to some finite set referred to as the *alphabet*.

Defining information according to the engineering practice shows that it is basically non-physical (at variance with the opinion of Schrödinger, Brillouin, and of most contemporary physicists.) Since any sequence can be transformed into an equivalent one by alphabet change and/or encoding, an information cannot be identified to a single sequence. It should indeed be defined as the *equivalence class* among sequences with respect to such transformations. It is thus an *abstract entity*.

An information may then be represented by its 'information message' defined as the shortest binary sequence in this class, whose length in binary digits (bits) quantitatively measures the information. Its bits are mutually independent, and each of them is essential to the information's integrity. An information is thus a *nominable entity* in Barbieri's meaning. No topology exists within informations.

Associating with each bit of an information message a dichotomic choice (answering a question or executing an action) endows the information it represents with a *semantic content* which possibly refers to the *concrete* world. An information then appears as a content for semantics, just like a shell contains a hermit crab. Thus, the length of the information message is a measure of the semantic specificity besides being that of the information quantity.

Information dwells in the physical world

As an abstract entity, an information is represented by a sequence of symbols. Since any sequence must be borne by a physical medium, information actually dwells in the physical world.

Information thus *bridges* the *abstract* and the *concrete*.

Information can be annihilated if its physical support is destroyed, but it can also be shared if it is written on several distinct supports.

Life is interpreted here as resulting from the interplay of information and matter: the physical world acts on information, but information too acts on the physical world. This interplay is highly dissymmetrical: perturbations in the physical world result in random symbol errors affecting information-bearing sequences, while information may instruct the assembly of physical objects by the agency of the semantics it bears.

Physical perturbations irreversibly degrade sequences

The physical world is seen since Ludwig Boltzmann as basically *chaotic*. 'Thermal noise' is just the macrocospic average result of random molecular movements. The second law of thermodynamics states that any physical system incurs an irreversible degradation.

A physical medium which bears some sequence does not escape this degradation, which results in random errors affecting the sequence symbols. During time intervals as short as a human life, genomic mutations can indeed be observed. The average number of erroneous symbols in the sequence is an increasing function of time, so the ability of the medium to store information, measured by its *capacity*, approaches zero as time passes. This fact seems to preclude the conservation of any sequence.

Information may instruct the assembly of physical objects

On the other hand, genetics tells that genomes instruct the assembly of living things within the physical world. Then the information borne by genomes acts on material structures by the agency of its semantic content.

It turns out that heredity lasts for at least 3.5 billion years, and this undeniable fact seems in absolute contradiction with the above statement hat any sequence-bearing medium suffers an inescapable degradation, which however is no less undeniable. We show how information theory succeeds in solving this blatant contradiction.

Errorless communication is possible despite symbol errors

A fundamental theorem of information theory tells that *errorless* communication of a sequence is possible in the presence of *symbol errors*. This paradoxical but highly favourable result is obtained by channel encoding, which consists of replacing the given sequence by a longer but fully equivalent one, belonging to a set of sequences as different from each others as to enable identifying one of them even if a limited number of its symbols are in error. Such a set of sequences is referred to as an *error-correcting code*. Being necessarily longer than the original sequence, a sequence which belongs to the code, or *codeword*, is *redundant*. The more redundant, the more efficient can be a code. Encoding an information message by means of such a code protects the information it represents.

Notice that 'error-correcting' is somewhat misleading for finite-length codes: correction is highly likely, not guaranteed.

Error-correcting codes enable conserving genomes

Physical perturbations result in symbol errors occurring with non-zero probability. Their permanency entails that the average cumulated number of errors increases with time. Assuming it is encoded into an error-correcting code, a genome can be *exactly regenerated* provided the cumulated number of symbol errors does not exceed the correction ability of the code. The genome is thus conserved almost indefinitely if it is regenerated frequently enough. The number of cumulated symbol errors within some time interval is random, but if it is large enough (hence for a code long enough) its variance is comparatively small as a consequence of the law of large numbers. Then the genome conservation is highly probable if the time interval beween successive regenerations properly matches the code performance.

Genomic error-correcting codes are needed

There is no other way to solve the contradiction between the conservation of genomes and the unavoidable degradation of sequences than assuming the existence of genomic error-correcting codes. We now examine their needed properties:

- they should be (very) redundant;
- at variance with engineering codes, they are not necessarily defined by mathematical equalities, but may result as well from physical-chemical or linguistic *constraints*. In this case, we refer to them as *soft codes*;
- the conservation of very old parts of genomes demands that they are made of several *nested component codes* which appeared successively during the ages.

Genomes are redundant

With a 4-symbol alphabet, a sequence of 133 nucleotides suffices to count the atoms of the visible universe since the number of distinct sequences having this length, 4^{133}, approximately equals 10^{80}, their estimated number. This moderate length of 133 nucleotides is to be compared with that of genomes, 1,000 or so for the simplest viruses, 10^6 at least for bacteria and much more for animals and plants, e.g., 3.2×10^9 for humans. There is thus room for an *immense* redundancy.

Stuffing cannot be responsible for the excess beyond the strictly necessary number of symbols because any symbol in a codeword contributes in the conservation of this codeword, hence in its own conservation. There are thus no 'junk' symbols. Else, it would be impossible to associate a genome with a species. The living world would be populated with chimeras, not with species members.

Genomes are protected by soft error-correcting codes

The DNA molecule is affected with constraints of several kind, especially steric. For instance its wrapping around histone octamers (in eukaryotic cells) induces constraints on the successive nucleotides. Moreover similar constraints affect polypeptidic chains and induce constraints on the genes which instruct their assembly (by the agency of semantic feedback loops, to be introduced later). All these constraints generate soft component codes.

Besides containing the genes which specify proteins, genomes instruct the assembly of larger-scale living structures, which demands some syntax, thus implying the existence of linguistic constraints, hence of other component codes in addition to those already mentioned.

Genomic error-correcting codes are made of nested component codes

In the absence of coding, the oldest parts of the genome would be the most degraded. It is the exact contrary which is true: the best conserved parts of the genomes, e.g., the *HOX* genes, are also the oldest ones. This fact is easily explained by assuming that the genomic error-correcting code has progressively been established during the ages, by successive encodings resulting in *nested codes*. Some information message has once been encoded. Another information message has been later appended to the result of the first encoding, and the message thus obtained has been encoded again. The initial information is thus protected twice, by its first encoding and because the result of this encoding has been itself later encoded. This process can be repeated arbitrarily many times. The older an information message, the more numerous component codes protect it and the better it is conserved. A very redundant and heterogeneous code results from numerous component codes.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

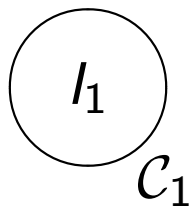

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

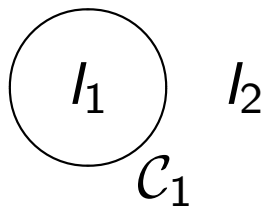

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

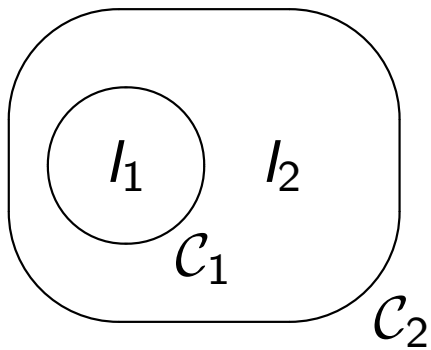

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

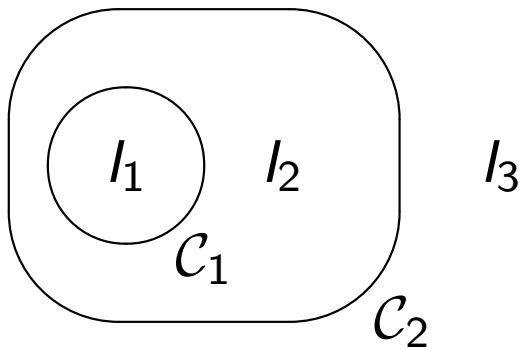

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

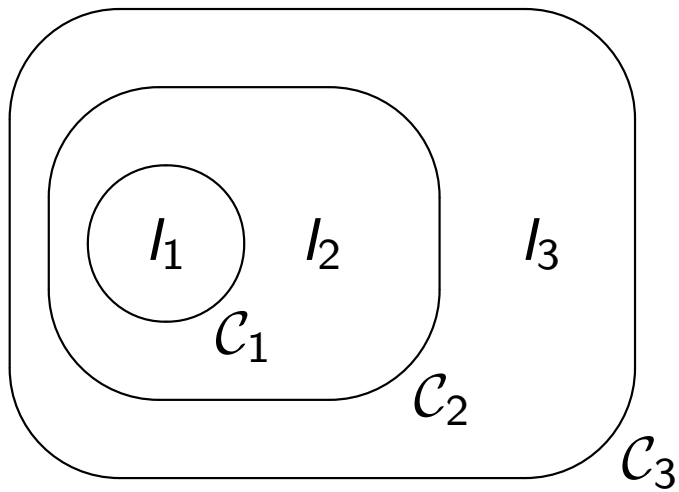

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

Nested component codes: Fortress metaphor

The fortress metaphor provides an intuitive illustration of nested component codes: a code is represented as a wall which protects what is inside it against outside attackers. Several concentric walls have been successively built to enclose information, so the content of the oldest, most central, wall is much better protected than the more recent and peripheral information. A multiplicity of walls is much safer than each of them separately.

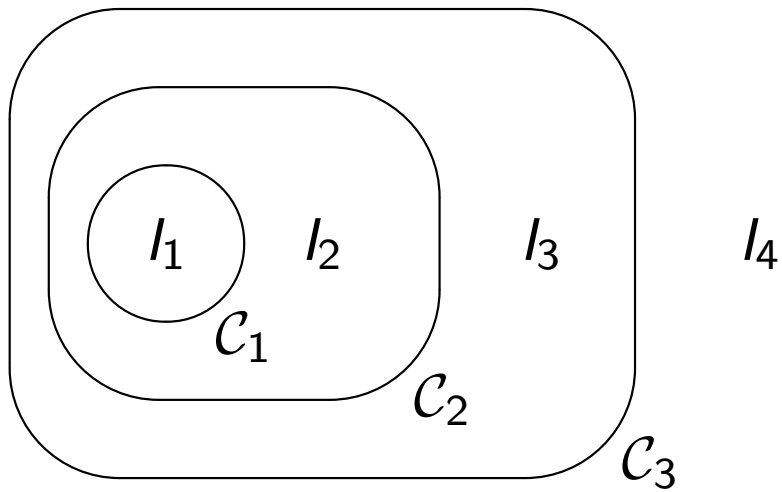

Figure: A code $\mathcal{C}_j$, $j = 1, 2, 3$, is represented as a closed wall which protects what is inside it. I_1, I_2, I_3 and I_4 are successive information messages. I_1 is protected by 3 codes, I_2 by 2 codes, I_3 by a single code and I_4 is left uncoded.

These hypotheses explain basic features of the living world

The assumed properties of genomic error-correcting codes explain many basic features of the living world left unexplained by mainstream biology. To list a few of them:

- Nature prodeeds with successive generations (which actually imply **re**generations).
- Living beings belong to discrete species which, moreover, can be ordered according to a hierarchical taxonomy as a consequence of the nested-code structure.
- Evolution trends towards increasing complexity because longer codes are more efficient in terms of error correction, hence have been favoured by the Darwinian selection.

How information acts on matter

Up to now, we have seen the effects of physical perturbations on information-bearing sequences and how information can be conserved despite them. We now turn on how information acts on matter.

The genes bear the information which instructs the synthesis of proteins. Their transcription and their translation into polypeptidic chains, becoming proteins when properly folded, are controlled by enzymes. As proteins, these enzymes are needed for their own synthesis.

This process can be represented as a 'semantic feedback loop', as shown in the next slide.

Basic scheme of a semantic feedback loop

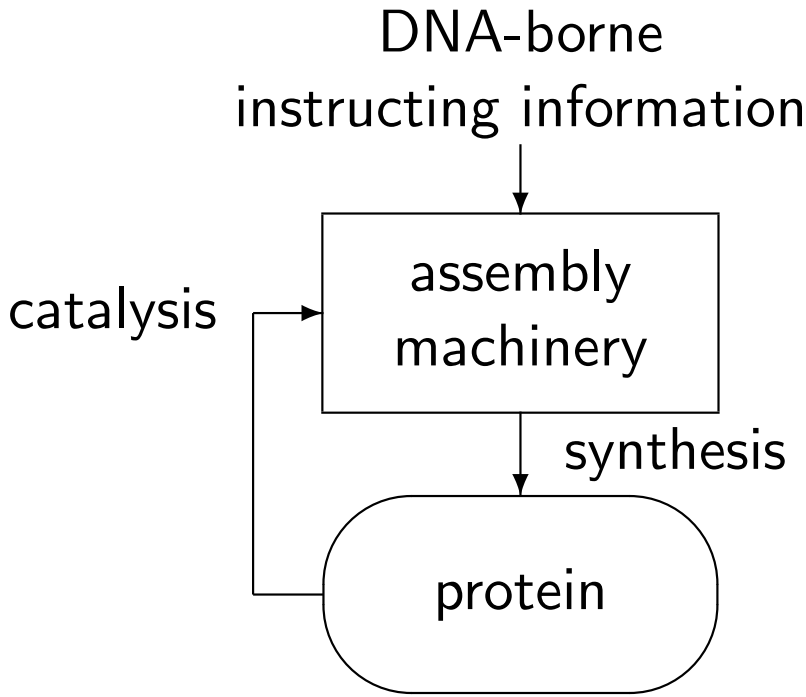

Figure: The genomic information instructs the assembly of a protein needed for its very assembly. As an enzyme, it controls its *own* synthesis, thus locking a loop. The arrows in the figure represent *irreversible* actions, so the whole loop is one-way.

Locking of a semantic feedback loop

The simplest example of semantic feedback loop is the molecular machinery depicted in the previous slide which performs the assembly of a protein according to semantic rules, as instructed by a symbolic information of genomic origin. This protein moreover controls the machinery operation since the instructed assembly is performed only if it is present.
The synthesized protein is then an *enzyme* needed for executing the instructions, which should thus *match* the enzyme specificity. The enzyme just acts as a *key* which enables its *own* assembly. A semantic feedback loop acts as a *trap* since, once assembled, it keeps its own structure and conserves the semantic rules it implements. We may think of this property of semantic feedback loops as performing what has been referred to by Crick as a 'frozen event' and by Barbieri as 'codepoesis'. Semantics then depends on *implementation*.

Interwoven semantic feedback loops

Living structures actually involve several combined semantic feedback loops. These loops are *interwoven* in the sense that the assembly of a single protein demands that several functions are performed. Several elementary loops according to the initial basic scheme are thus combined, so that *all* the proteins which act as enzymes for these functions are needed for the synthesis of *any* of them. Instead of a single key as in the initial basic scheme, as many keys as elementary loops now lock the system, which is the more enduring, the more numerous the keys.

As a first example the following figure schematically represents the genetic communication in a prokaryote, which performs the functions of replication-regeneration of the genome and those of transcription and of translation of the genes. The more complicated example of eukaryotic cells will be examined later.

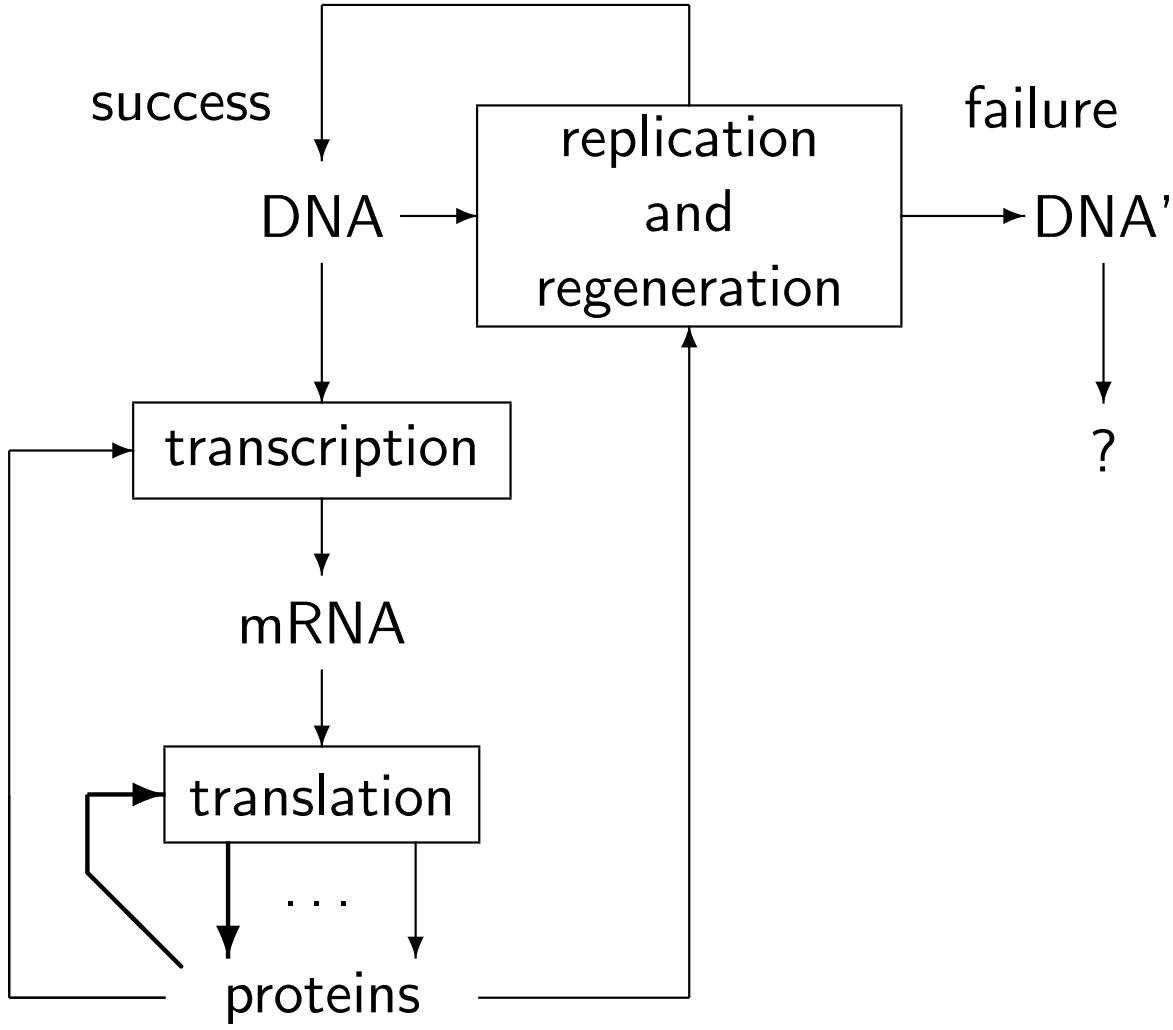

Figure: Modelling genetic communication in a prokaryote as a system of semantic feedback loops. The arrows originating in 'proteins' denote enzymatic actions. The semantic feedback loop which pertains to the genetic mapping is drawn in heavy lines (bottom left). DNA' denotes a DNA string which differs from the original one.

Reproductive feedback loop

The upper part of the figure in the previous slide concerns DNA replication and regeneration. It pertains to each individual in a homogeneous population of cells which descend from a single ancestor and possess the same DNA.

We assumed that the function of regeneration, similarly to others, needs the agency of enzymes. Since the very existence of codes enabling genome regeneration is not recognized by mainstream biology, how regeneration is performed remains unknown and, in particular, the needed enzymes have not been identified. The feedback structure of this 'reproductive loop' entails that *constraints on proteins induce constraints on the genome which instructs their own assembly*, resulting in a seeming teleology. Notice that the reproductive feedback loop works regardless of the genome size, which is thus unlimited.

A seeming teleology

Constraints on proteins induce constraints on the genome which instructs their assembly because any element of the reproductive loop is located both upstream and downstream from any other one, including itself (remember that the loop is one-way). Any element of the loop acts on the one immediately upstream by the agency of all other downstream ones. This looks like teleology but causality is not violated. It is thus possible to associate with any semantic feedback loop a genomic soft code defined by the specific constraints affecting its enzymes. These constraints are superimposed to the others, and together they result in the genomic error-correcting system of nested codes.

Origin of a new semantic feedback loop

A new semantic feedback loop results from the insertion in the genome of a gene which instructs the assembly of a protein acting as an enzyme for controlling this very assembly. Such a new gene may result from the erroneous regeneration of a gene already belonging to the genome, or from genetic material being appended to the initial genome by horizontal genetic transfer. In any way, the insertion of a gene which specifies a protein controlling its own assembly is as infrequent as to make the arising of a new semantic feedback loop an exceptional event.

Origin of Barbieri's organic codes

The onset of a new semantic feedback loop may also be interpreted as creating a new organic code in Barbieri's meaning since it establishes a dependence between two sequences of entirely foreign kinds, one of them being the genome itself. New constraints are created in the genome due to this dependence. We may thus identify the onset of an organic code to the introduction of a new component in the genomic error-correcting system of nested codes, which results itself from the onset of a new semantic feedback loop.

Success or failure of regeneration

Successful regeneration results in a genome strictly identical to the original one. It is why the replication and regeneration process has been drawn as a loop in the left part of the previous figure, which is labelled ‘success’. Then the genomic information is conserved.

The genome denoted by DNA’, obtained in the highly infrequent case of unsuccessful regeneration (indicated by ‘failure’ in the figure), markedly differs from the original one as a consequence of the error correction ability of the genomic code. It can replicate itself only if it instructs the assembly of the machineries which are needed to this end despite this difference. Else the whole process is aborted. That the process continues or not is expressed by the interrogation mark. If it continues, the genome bears a different information and may originate a new species.

New information, lengthened genomes

The error-correction ability of the genomic code maintains the species integrity. When regeneration fails, new information arises. It originates a new species if the phenotype it specifies withstands the Darwinian selection. In contrast, the specificity of the enzymes which control the operations of replication, regeneration, transcription, and translation results in the whole process being merely possible. The failure of any of them would abort this process.

Although the system of semantic feedbacks as depicted in the figure is locked, nothing prevents the genome lengthening, e.g., according to 'horizontal genetic transfer' mechanisms. Genome lengthening then results at the same time in increasing its redundancy *and* the information quantity it bears. The former improves the error-correction ability of the genome, and the latter specifies phenotypic features perhaps advantageous with respect to the Darwinian selection.

Consequences of genome lengthening

It should indeed be reminded that lengthening any symbolic sequence can have two effects: increasing the length of its information message hence the information quantity it bears, and increasing its redundancy. One may think of such a sequence of length n as actually made of the k symbols of its information message, $k < n$, the remaining $n - k$ redundancy symbols being computed in terms of the information message symbols in order to make the whole sequence resilient to errors. Then k measures the information quantity borne by the sequence, hence its *semantic specificity*. The *resilience* of the sequence to casual errors, which is needed for its conservation, increases as $n - k$ increases if an adequate encoding is employed. This remains true even if the information message is not explicitly present in the sequence.

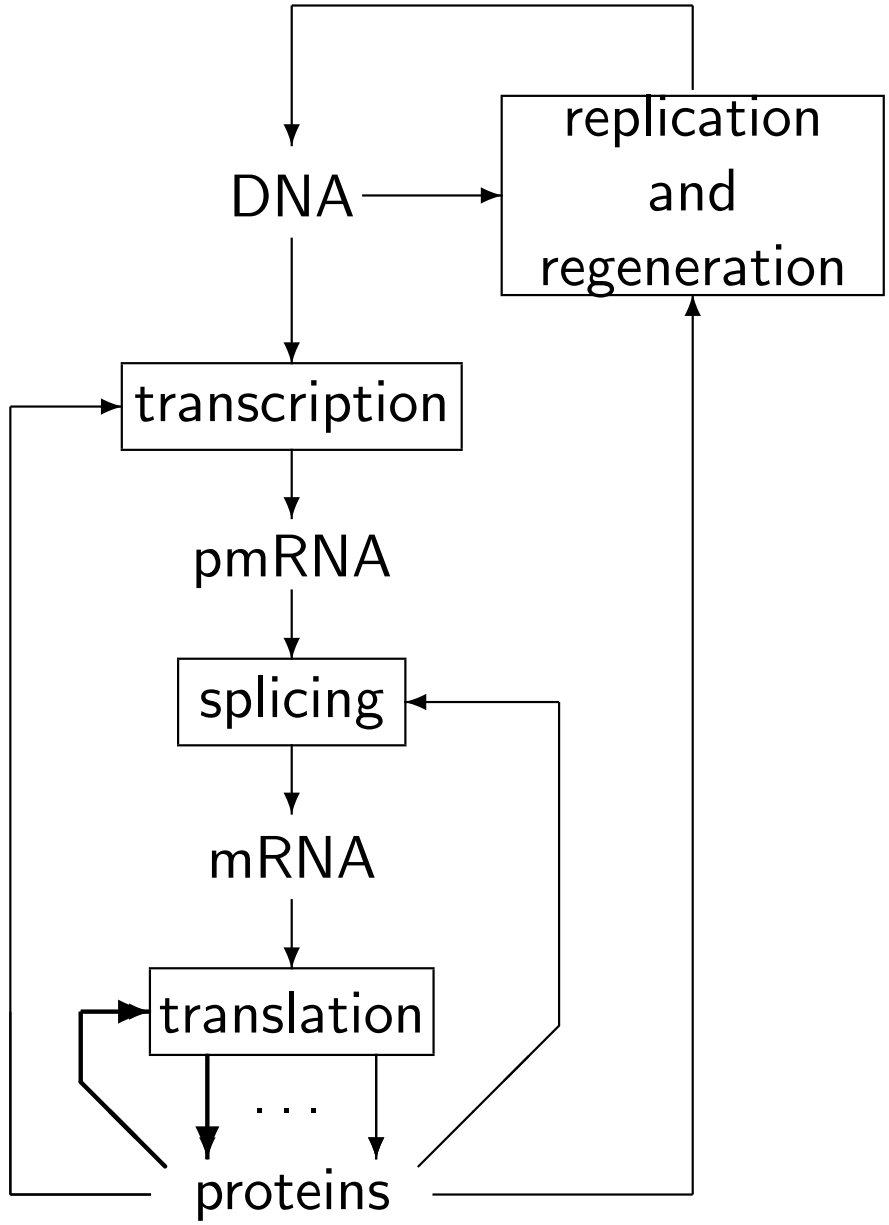

Figure: Further including the function of splicing in the system of semantic feedback loops of the previous figure; pmRNA denotes pre-messenger RNA.

Further inserting the splicing function

The previous figure shows how the function of splicing, typical of the eukaryotes, has been inserted in the system of semantic feedback loops as another such loop mutually locked to those already present. The next figure is a simplified scheme of the lower part of the previous one, showing the structure which is implemented in each individual of the considered population. It has been assumed that each of the mentioned functions is controlled by a single enzyme. That several enzymes are actually needed for each function still increases the mutual locking of the semantic feedback loops.

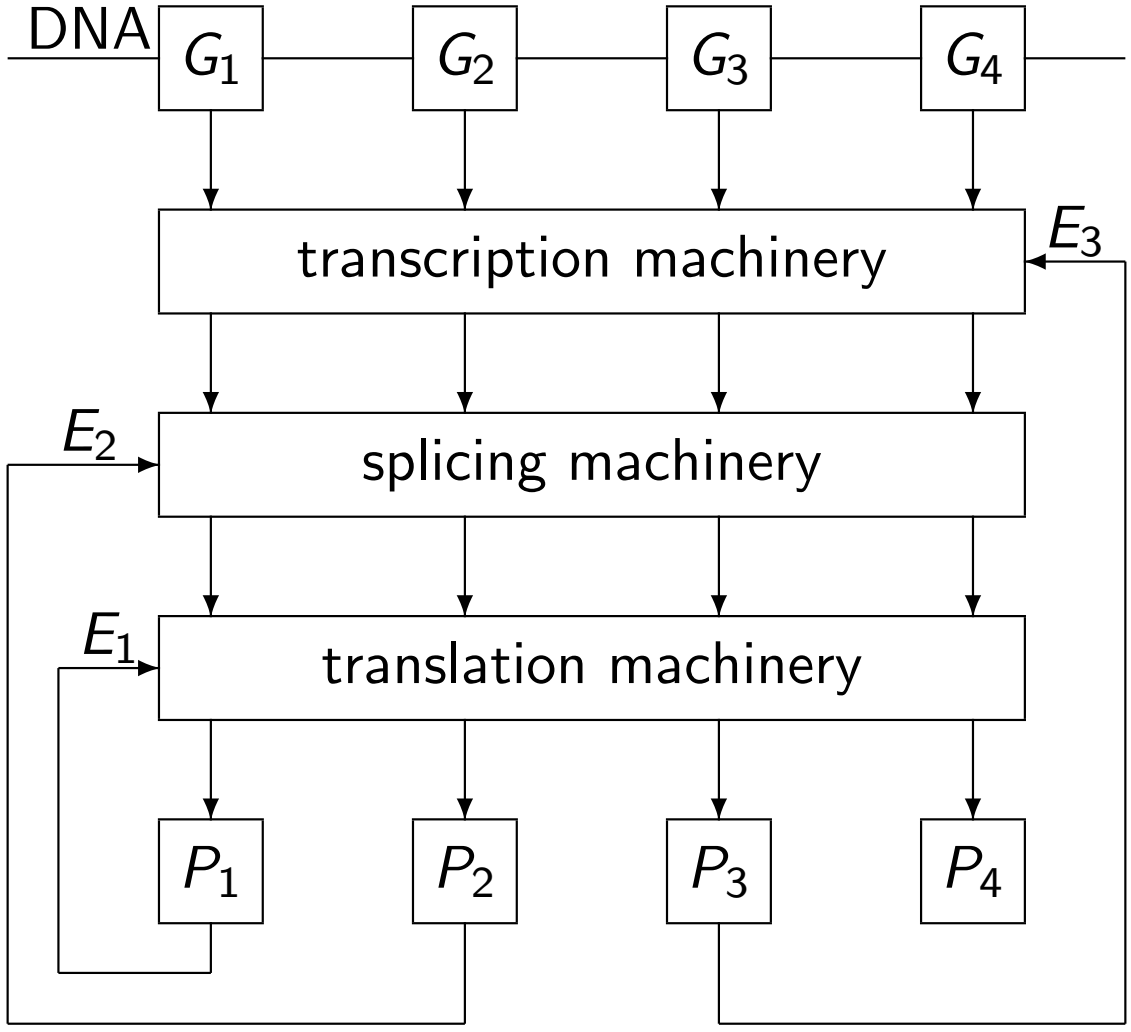

Figure: Simplified lower part of the system of semantic feedback loops in eukaryotic cells. $G_1, \ldots, G_4$ denote genes which instruct the assembly of proteins $P_1, \ldots, P_4$. $E_1, \ldots, E_3$ denote the enzymatic actions of the proteins P_1, P_2 and P_3 which enable the functions of translation, splicing and transcription, respectively. The loops are clearly interwoven.

Proteins as enzymes or building blocks

We may distinguish among proteins: (1) the enzymes which are necessary catalysts for the functions implied in the operation of the semantic feedback loops; and (2) other proteins, i.e., enzymes having other functions or mere building blocks. In the simplified previous figure, emphasis was laid on the enzymes P_1, P_2 and P_3 as belonging to the first category. P_4 was the single representative of the second one. This category actually contains more elements than the first one, namely all the proteins which are used for building the remainder of the phenotype. Not being critically needed, proteins of the second category may incur regeneration errors while their assembly still results in viable phenotypes (later filtered by the Darwinian selection).

Extended semantic feedback loops

Up to now, we have seen how semantic feedback loops control the assembly of proteins, as instructed by the genes. But a living thing is not an unorganized cluster of proteins. The molecular machineries of the phenotype which implement the functions of transcription, translation and replication-regeneration, as well as all others, must be assembled as instructed by the genome. Not all details of this process are known, at variance with the fairly well understood scheme which represents the assembly of proteins. Instructing the assembly of phenotypic machineries demands that the whole genome be endowed with some syntax, the constraints of which define some more layers in the system of genomic nested soft codes. The next slide schematically represents a living thing as a system of extended semantic feedback loops.

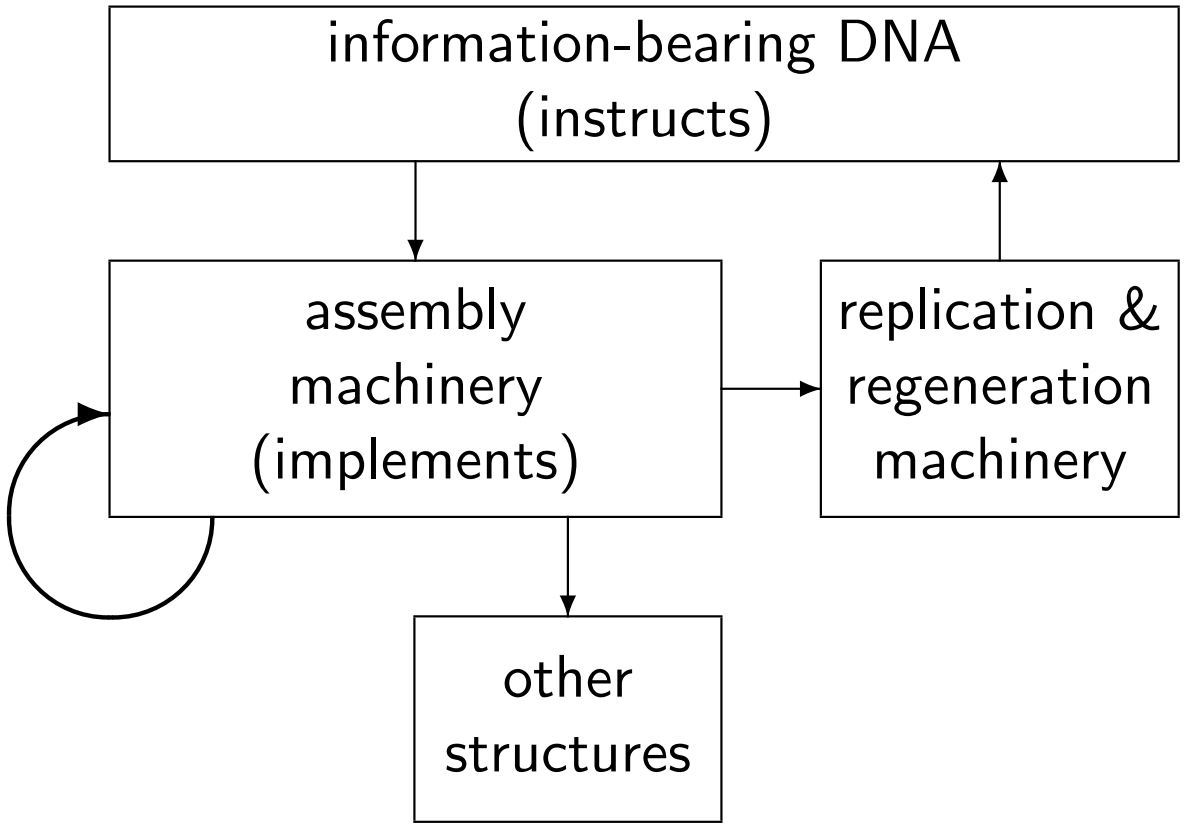

Figure: Schematic representation of a living thing as made of extended semantic feedback loops. The assembly machinery involves multiple feedbacks, schematically represented by the curved arrow at left.

Schematic representation of a living thing

According to schemes similar to those valid for the assembly of a protein, we may attempt to represent that of a whole living thing as a system of interwoven extended semantic feedback loops. We tried to do so in the next slide.

The figure also shows the connections of a living thing with its environment. Important parts of it are devoted to matter and energy exchanges with the environment (metabolism), and the acquisition of information from it (sensing). The horizontal genetic transfer which sporadically occurs tends to increase the genome length. Of course, the living thing as a whole is submitted to the Darwinian selection.

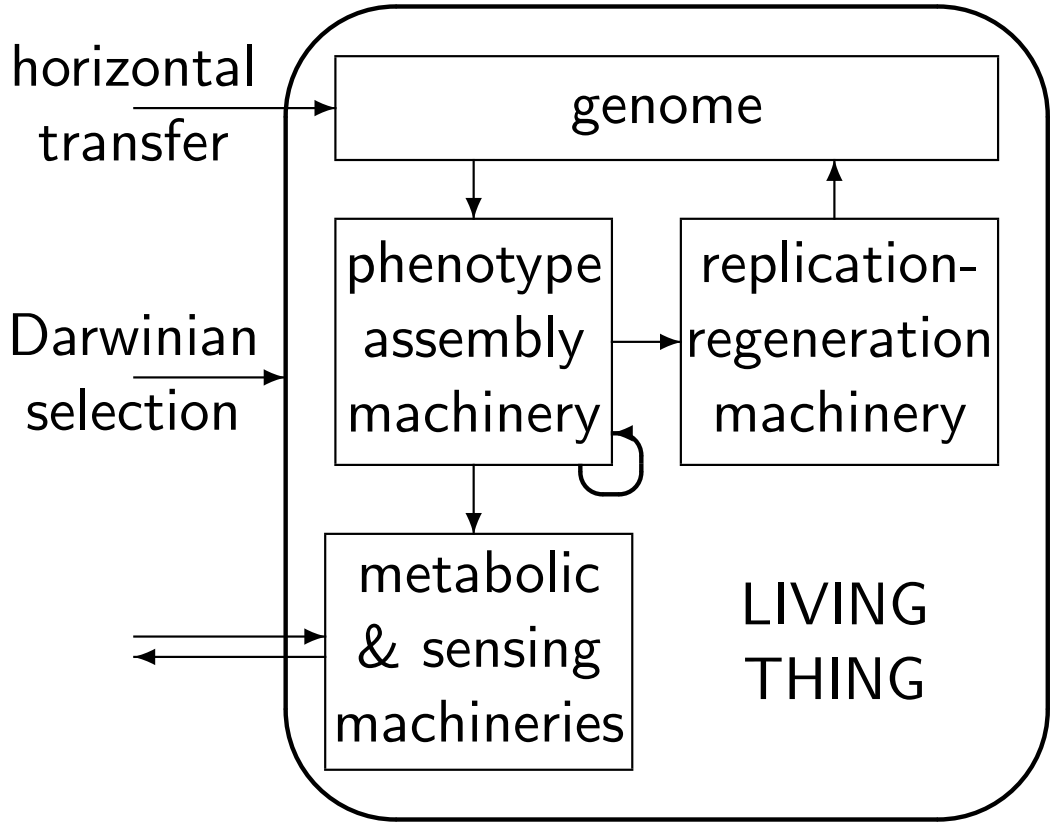

Figure: A living thing modelled as interwoven extended semantic feedback loops. The information borne by the genome instructs the assembly of all phenotypic machineries, one of which in turn performs its replication and regeneration. Horizontal transfer may increase the information quantity it bears. The phenotype assembly machinery controls the assembly of all phenotypic machineries, including itself. The living thing as a whole is subjected to the Darwinian selection.

Possible origin of semantic feedback loops

How the instructions borne by the genome are implemented depends on the assembly machinery in its present state. Can we understand how the semantic feedback loops came into existence? We may think that, within a mixture of molecules, some of which having memory, i.e., able to bear symbolic sequences like modern DNA or RNA, others with enzymatic properties like modern proteins (and maybe certain having both abilities), a rudimentary semantic feedback loop as described above has been assembled by chance. Then, once assembled it remains closed and thus conserves itself, at variance with fleeting structures which appear and disappear at random. Further lengthening of the memory may result in specifying other loops interwoven with the first one. The probability that an initial system of semantic feedback loops is assembled may be extremely small. This event can however occur sooner or later if its probability is not strictly zero.

Possible evolution of semantic feedback loops

Maybe a rudimentary semantic specifity and a rudimentary error correction ability were enough for initiating the whole evolution process provided the genome length increases since it can then entail both an increase of information quantity, hence of semantic specificity, and of redundancy. At this early stage, of course, the genomic error correction system was much less efficient than it is in the present, and similarly the enzymatic specificity of the proteins was much less. The Darwinian selection can progressively improve both if it operates on increasingly long genomes.

Only the genomes which are the most effectively improved as regards both the semantic specificity and the error correction ability will survive the Darwinian selection. Then the high error correction ability and the sharp enzymatic specificity of modern genomes result from evolution.

Objects cannot be conserved, information can be

It should be realized that, contrary to our intuition, *conservation* of an object is not the rule but the exception. This is especially true at the geological time scale. As stated by the second law of thermodynamics, the rule is indeed the object's *degradation*. Conserving an object actually needs an *active* approach involving information. We met such an approach when explaining the genome conservation over time intervals at the geological scale by means of an error-correcting code. Even during the much shorter life time of an individual, the stability of living structures is ensured only by means of semantic feedback loops.

Strictly speaking, then, it is not a particular physical object which is conserved by such means. As regards living things, what can indeed be conserved is an *information*, i.e., a *physically inscribed abstraction* which represents its composition and enables its re-assembly as frequently as needed (see *The Delphic boat* by Antoine Danchin).

Conclusion

The information borne by the genome instructs the assembly of all phenotypic structures by the agency of semantic rules. Its conservation is ensured by a system of nested error-correcting soft codes which makes it resilient to symbol errors.

As specified by the genome, the phenotypic structures involve interwoven semantic feedback loops which, once assembled, remain locked thus ensuring their own conservation, which entails that of the semantic rules they implement. The locking of these structures does not prevent the genome lengthening which may specify new structures and bring more redundancy, hence can result in further evolution. Any new semantic feedback loop originates a new nested soft code, hence a new organic code in Barbieri's meaning.

Thus, the genomic error-correcting code and the semantic feedback loops, acting together, make life resilient to the intrinsic trend of the physical world towards disorder.

Thanks

The influence of Marcello Barbieri is gratefully acknowledged. The convergence between my views and his is all the more interesting since our backgrounds are quite different. Mine is communication engineering.

Thank you for your attention.

If you wish to receive the .pdf file of this presentation, please give me your email address.