Perception, 1992, volume 21, pages 517–544

Serial pattern complexity: irregularity and hierarchy

Peter A van der Helm, Rob J van Lier, Emanuel L J Leeuwenberg
Nijmegen Institute for Cognition and Information (NICI), Psychology Laboratory, University of Nijmegen, PO Box 9104, 6500 HE Nijmegen, The Netherlands
Received 8 March 1991, in revised form 21 November 1991

Abstract. In perception research, various models have been designed for the encoding of, for example, visual patterns, in order to predict the human interpretation of such patterns. Each of these encoding models provides a few coding rules to obtain codes for a pattern, each code expressing regularity and hierarchy in that pattern. Some of these models employ the minimum principle which states that the human interpretation of a pattern is reflected by the simplest code for that pattern, ie the simplest code according to a given complexity metric. In this paper a new complexity metric is proposed. This metric is based on a formal analysis of the concept of regularity. Some conclusions of this analysis are sketched. The new metric does not depend on artifacts of the coding rules. It accounts for the amounts of irregularity and hierarchy as represented in a code of a pattern, such that these two amounts can be added to determine the complexity of a code. An experiment is discussed that shows that the new metric performs significantly better than the metrics used previously. In particular, the new metric predicts more local pattern organizations than the old metrics. This implies that various local pattern organizations do not falsify the minimum principle anymore.

1 Introduction

Regularity is a rather intuitive concept that seems to defy formal description. This aspect becomes relevant when regularity explicitly plays a crucial role, as it does in many formal models of perception (Simon and Kotovsky 1963; Vitz 1966; Vitz and Todd 1969; Leeuwenberg 1969, 1971; Garner 1970; Restle 1970, 1979; Jones and Zamostny 1975; Deutsch and Feroe 1981; Palmer 1983; Leyton 1986a, 1986b). These models are designed to explain the phenomenon that although, in principle, a pattern can be interpreted in many ways, usually one interpretation is preferred (see figure 1 for an example in visual shape perception). This preference is assumed to be guided by the regularity that is present in a pattern. For instance, in figure 1, the usually preferred interpretation is assumed to be induced by the repetition, ie the

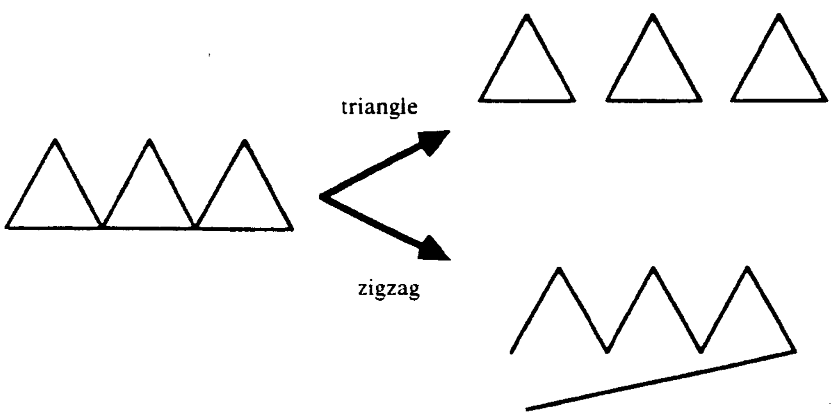


Figure 1. In visual shape perception, a major problem is how to predict the preferred interpretation of a pattern which, in principle, can be interpreted in many ways. Here, two possible interpretations of a line drawing are visualized. Usually, the triangle interpretation is preferred, not the zigzag interpretation.

regular occurrence of the triangle part. This assumption is in line with the minimum principle (Hochberg and McAlister 1953) which, historically, fits in with the tradition of Gestalt psychology (Wertheimer 1912; Kohler 1920; Koffka 1935). The minimum principle states that the preferred interpretation of a pattern is reflected by the simplest description of that pattern. Now, for each model mentioned above, it holds that patterns are described in terms of a formal 'language' designed to express regularity in patterns. Yet, the models show great variety in the formal definition and in the justification of the complexity metric that is used to decide which of the possible descriptions is the simplest one. The fact that, for many of these metrics, the predictions are rather similar (Simon 1972) is, in our view, not very surprising since, for all metrics, it holds more or less that a pattern description is simpler if it expresses a larger amount of regularity in the pattern. However, in each model, regularity is incorporated and discussed only in terms of examples showing only a few kinds of regularity (like repetitions and symmetries). These few kinds of regularity may be relevant in perception, but represent only a choice out of many possible kinds of regularity. This choice may be plausible and may even be empirically supported, yet it is not the same for all models. That is, as long as regularity as such is not specified, one can hardly assign psychological significance to a specific choice concerning pattern descriptions and complexity metrics (Simon 1972). In the present paper, we will propose a solution to this problem, which may be introduced briefly as follows.

In the paper of van der Helm and Leeuwenberg (1991), the intuitive concept of regularity has been formalized within the framework provided by the structural information model of Leeuwenberg (1969, 1971). This model is an encoding model, focusing mainly on visual shape perception. In this model, a pattern can be described, in several ways, by means of coding rules expressing regularity in the pattern. In agreement with the minimum principle and given a complexity metric, the simplest pattern description, or simplest code, is assumed to reflect the preferred interpretation of the pattern. So, the structural information model consists of the minimum principle plus an operationalization of the minimum principle, and the operationalization consists of coding rules plus a complexity metric. In the present paper, we will not compare Leeuwenberg's model with other models [for a review on these topics and for references to related literature, see van der Helm and Leeuwenberg (1991)]. Instead, we will focus primarily on the choice of a psychologically significant complexity metric, on the basis of an analysis of the intuitive concept of regularity. To that end, first we will go into the structural information model in some detail. Second, we will sketch the results of an analysis of regularity. The actual analysis is presented in van der Helm and Leeuwenberg (1991). Third, we will propose a new metric of complexity. This metric stems from the analysis of regularity. Finally, we will discuss an experiment that was designed to test how well preferred interpretations can be predicted by the new complexity metric. In the experiment, subjects had to indicate their preference for one out of three segmentations of patterned sequences of, for example, black and white dots.

2 The structural information model

The encoding of visual patterns, as performed in the structural information model (Leeuwenberg 1969, 1971), proceeds as follows. In order to describe regularity in a pattern, the pattern is first represented by a symbol series. For instance, in figure 2, the contour of the pattern consists of subsequent angles and line segments, each of which is labelled with a symbol. Angles or line segments of equal size are labelled with an identical symbol. These symbols will be called pattern symbols, indicating that a symbol refers only to a pattern part and not to its meaning in, for example, the

Roman alphabet. Now, tracing the contour in a clockwise direction would yield the symbol series 'kalckalckalc', representing the pattern. Note that, in theory, a pattern is not considered to be mapped onto a symbol series. On the contrary, the symbol series has to be such that the pattern can be reconstructed by substituting the actual sizes of angles and line segments for the pattern symbols (cf Leyton 1986a, 1986b). This substitution is called the semantic mapping from the symbol series onto the pattern. The semantic mapping as such may already give rise to several questions, for instance with respect to the number of possible symbol series by which a pattern can be represented. In the present paper, however, we will not deal with these questions. In particular, in the experiment to be discussed in this paper, we chose stimuli for which the semantic mapping does not raise problems.

The actual encoding consists of describing regularity in terms of identical symbols in the symbol series which, because of the semantic mapping, corresponds to the regularity in the pattern. In the structural information model, only three classes of regularities play an essential role, namely iterations, symmetries, and so-called alternations. Each of these three classes is described by means of one coding rule. As we will argue in the next section, these three coding rules are indeed the proper ones to use according to the formalization of regularity as elaborated in the paper of van der Helm and Leeuwenberg (1991). Next, we will proceed by giving the definitions of these coding rules, and the way in which these coding rules can be applied to symbol series.

First, the iteration rule, which can be applied to express that a series contains successive identical symbols, is defined as follows:

$$\mathrm{kkk} \ldots \mathrm{kk} \rightarrow N \times (\mathrm{k}) .$$

The expression on the right hand side is called an I-form, in which N equals the number of symbols 'k' in the series at the left-hand side ($N \geqslant 2$), while (k) is called the I-argument. For instance, according to the iteration rule, the series 'aaaaa' can be encoded as $5 \times (\mathrm{a})$.

Second, the symmetry rule, which can be applied to express that a series contains pairs of identical symbols, nested around a so-called pivot, is defined as follows:

$$\mathrm{k}_1 \mathrm{k}_2 \ldots \mathrm{k}_n \mathrm{p} \mathrm{k}_n \ldots \mathrm{k}_2 \mathrm{k}_1 \rightarrow \mathrm{S}[(\mathrm{k}_1)(\mathrm{k}_2) \ldots (\mathrm{k}_n), (\mathrm{p})] .$$

The expression on the right-hand side is called an S-form, in which (p) is the pivot, and the series $(\mathrm{k}_1)(\mathrm{k}_2) \ldots (\mathrm{k}_n)$ is called the S-argument and consists of elements (k_i),

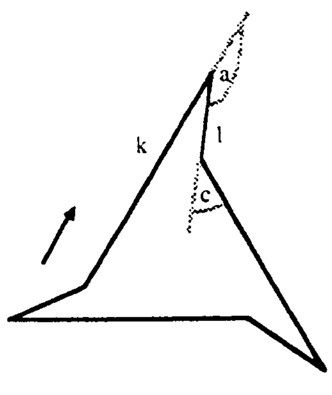


Figure 2. Tracing the contour of the pattern (the arrow indicates the starting point and direction) yields the symbol series 'kalckalckalc'. This symbol series represents the subsequent angles and line segments in the contour so that, by substituting the actual values for the symbols, the pattern can be reconstructed. According to Leeuwenberg's structural information model, interpretations of the pattern can be obtained by encoding the symbol series.

where $i = 1, 2, ..., n$ $(n \geqslant 1)$. For instance, the series 'kapmpak' can be encoded as S[(k)(a)(p), (m)].

Third, there is the alternation rule which can be applied to express that a series contains successive subseries that either all begin or all end identically. Both cases are given in the following definitions:

$$kx_1 kx_2 \ldots kx_n \rightarrow \langle(k)\rangle/\langle(x_1)(x_2)\ldots(x_n)\rangle ,$$

and

$$x_1k\ x_2k \ldots x_nk \rightarrow \langle(x_1)(x_2)\ldots(x_n)\rangle/\langle(k)\rangle .$$

The expression on the right-hand side is called an A-form, in which the series $(x_1)(x_2)$ ... (x_n) is called the A-argument and consists of elements (x_i), where $i = 1,, 2, ..., n$ $(n \geqslant 2)$. For instance, the series 'arasat' can be encoded as ⟨(a)⟩/⟨(r)(s)(t)⟩, and the reversal of that series, ie 'tasara', can be encoded as ⟨(t)(s)(r)⟩/⟨(a)⟩.

In applying the coding rules to a symbol series, one should take notice of the following two aspects. First, in the definition of the coding rules, the symbols are considered to be variables standing for arbitrary subseries (with identical symbols standing for identical subseries). This implies that the coding rules can be applied not just to express the identity of single symbols but, in general, to express the identity of subseries in a series. For instance:

$$\text{ababab} \rightarrow 3\times(\text{ab}) ,$$

and

$$\text{badpqvwpqbad} \rightarrow \text{S}[(\text{bad})(\text{pq}), (\text{vw})] .$$

Note that the parentheses in an I-form, S-form, or A-form (which we shall call an ISA-form) yield an unambiguous notation. Any subseries between parentheses in an ISA-form is called a chunk. It should be noted that the pivot in an S-form is a chunk which, as a limiting case and without distorting the symmetrical structure, may be 'empty'. This implies that, for instance, the series 'abppab' can be encoded into an S-form denoted by S[(ab)(p)].

Second, the subseries inside a chunk in an ISA-form can be encoded just like any symbol series, for instance:

$$\text{bapabapa} \rightarrow 2\times(\text{bapa}) \rightarrow 2\times(\text{bS}[(\text{a}), (\text{p})]) ,$$

and

$$\text{aabppaab} \rightarrow \text{S}[(\text{aab})(\text{p})] \rightarrow \text{S}[(2\times(\text{a})\text{b})(\text{p})] .$$

In such cases, the ISA-forms are said to be hierarchically nested. Similarly, but less trivially, a hierarchical nesting of ISA-forms may result in the following. Whereas an I-argument consists of only one chunk, an S-argument or A-argument is, in general, a series consisting of several chunks. Such a chunk series can be encoded the same way as a symbol series. Consider, for instance, the following encoding:

$$\text{ababbaba} \rightarrow \text{S}[(\text{a})(\text{b})(\text{a})(\text{b})] \rightarrow \text{S}[2\times((\text{a})(\text{b}))] .$$

In this example, the S-argument (a)(b)(a)(b) constitutes a chunk series 'xyxy' in which the chunk 'x' = (a) and the chunk 'y' = (b). Just like a symbol series, the chunk series 'xyxy' can be encoded by applying the coding rules. This yields, for example, the code 2×(xy) which, by substituting (a) for 'x' and (b) for 'y', yields the code 2×((a)(b)), as given in the example. Similarly, the argument of an A-form can be encoded. Whereas the symbol series is said to represent the lowest hierarchical level, an I-argument, S-argument, or A-argument is said to represent a higher hierarchical level, and the encoding of an S-argument or an A-argument leads to still higher hierarchical levels.

Now that we have considered the way in which the coding rules can be used to encode symbol series into codes, we can go into the meaning of the codes in more detail. First of all, by expressing identities in a series, a code provides a description of regularity in that series. However, this is not the ultimate meaning of a code. The ultimate meaning of a code is constituted by the fact that a code provides a means of obtaining a classification and an organization of the series, which together reflect an interpretation of the pattern that is represented by the series. This may be illustrated as follows.

First, for the four-symbol series 'aaba', the code 2 × (a)ba expresses just the identity of the first and second symbols, which corresponds to all the identity in a four-symbol series like 'ppqr'. Thus, 'ppqr' can be seen as a representative of the class of symbol series to which 'aaba' belongs according to the code 2 × (a)ba (cf Collard and Buffart 1983). In general, such a class representative can be found as follows. First, replace all pattern symbols in the code by arbitrary but different symbols; then, decode the code. For instance:

aaba → 2 × (a)ba → 2 × (p)qr → ppqr .

Note that, according to another code, the symbol series 'aaba' belongs to another class. For instance:

aaba → aS[(a), (b)] → pS[(q), (r)] → pqrq .

According to this classification, the second and fourth symbols are taken as being identical. Whether 'aaba' is encoded and classified in the first way or in the second way is determined by some complexity metric. That is, in agreement with the minimum principle, the humanly preferred interpretation of a pattern is assumed to be reflected by the simplest code for the pattern. We will discuss complexity metrics in a later section.

Second, a code not only prescribes a classification, but also an organization of a pattern. For the series 'ababab', the code 3 × (ab) expresses that 'ababab' is similar to 'yyy' where 'y' = 'ab'. So, the code can be said to induce the organization (ab)(ab)(ab) in the series, ie a partitioning of the series into chunks. Such organizing in terms of chunks will be called a chunking (cf Geissler et al 1978). In general, the chunking induced by an ISA-form can be found by decoding the ISA-form without removing the parentheses in the ISA-form, eg:

badpqvwpqbad → S[(bad)(pq)), (vw)] → (bad)(pq)(vw)(pq)(bad),

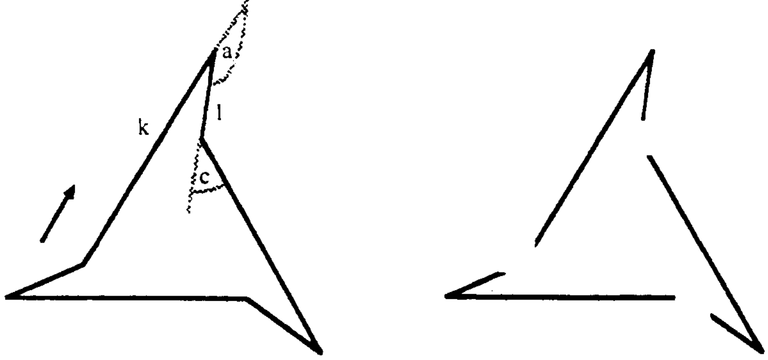
and

hkghkpq → ⟨(hk)⟩/⟨(g)(pq)⟩ → (hk)(g)(hk)(pq) .

As illustrated in figure 3, the chunking of a symbol series reflects an organization in the pattern that is represented by the symbol series. Note that, perceptually, 'ababab' does indeed seem to consist of the three repetitions of (ab) in line with the chunking (ab)(ab)(ab) as induced by the code 3 × (ab), but that 'abababpqqp' seems to consist of the parts 'ababab' and 'pqqp' which does not imply a chunking in the above sense. Yet the latter organization is relevant too (Leeuwenberg and van der Helm 1991) and will be called clustering. One aspect of clustering is that each ISA-form is considered to induce a cluster containing all the chunks needed to construct that ISA-form. For instance, for (sub)series 'ababab', the I-form 3 × (ab) groups the three chunks (ab) into one cluster ((ab)(ab)(ab)).

Another aspect of clustering is related to the intrinsic character of a regularity structure, and has consequences for S-forms and A-forms only. (I-forms are too simple to show such an extra aspect.) First, we will discuss the S-forms. We stated that an S-form indicates that a symbol series contains pairs of identical subseries nested around a pivot. For instance, the series 'abcpbca' can be encoded into the S-form S[(a)(bc), (p)], inducing the chunking (a)(bc)(p)(bc)(a). Now, with respect to that chunking, the S-form can also be said to induce a tri-partitioning into the S-argument (a)(bc), the pivot-chunk (p), and the reversed S-argument (bc)(a). Therefore, the S-form will be said to induce the grouping of the chunks in the S-argument into one cluster ((a)(bc)), as well as the grouping of the chunks in the reversed S-argument into one cluster ((bc)(a)). This clustering can be indicated in the chunk series by ((a)(bc))(p)((bc)(a)). We also stated that an A-form expresses that a symbol series contains successive subseries which either all begin or all end identically. Let us focus on the 'all-begin-identically'-case. (The 'all-end-identically'-case is completely analogous.) For instance, the series 'abpqabrsabt' can be encoded by the A-form ⟨(ab)⟩/⟨(pq)(rs)(t)⟩, which induces the chunking (ab)(pq)(ab)(rs)(ab)(t). Now, in that chunking, each of the above-mentioned 'successive subseries' is chunked into a pair of chunks called an A-pair. So, each A-pair constitutes a unit that is characteristic for the regularity described by the A-form. Moreover, as we will see later on, the A-pairs are essential for understanding the hierarchical character of the A-rule. Therefore, the A-form above will be said to induce a clustering by grouping each A-pair into a cluster, which can be indicated in the chunk series by ((ab)(pq))((ab)(rs))((ab)(t)). So, in summary, besides a chunking, an ISA-form also induces a clustering based on that chunking: each ISA-form induces one cluster consisting of all chunks in that chunking; an S-form induces two further clusters, namely the S-argument and the reversed S-argument; and an A-form induces further clusters by grouping each A-pair into a cluster.

The perceptual organization of a pattern, as induced by a code of the pattern, will be said to consist of the combination of the chunking and the clustering induced by that code. Furthermore, the dominant segmentation in a perceptual organization of a pattern is said to be the segmentation that consists of at least two, maximally sized, segments (chunks or clusters) present in that perceptual organization. As an illustration, we reconsider the pattern in figure 1. The pattern in figure 1 can be represented by a symbol series in several ways; in this example, we will focus on the two most relevant representations (see figure 4). Remember that a symbol series represents a pattern, if that pattern can be reconstructed from the symbol series by substituting the



(a) (b)

Figure 3. If the pattern in (a) is represented by the symbol series 'kalckalckalc' then, according to the structural information model, this symbol series can be encoded into 3×(kalc). This code induces, in the symbol series, the organization (kalc)(kalc)(kalc) which, in the pattern, corresponds to the organization as shown in (b).

actual sizes of angles and line segments for the pattern symbols. One way to represent the pattern, is by means of the symbol series 'kakakakkakakakkakakak' (see figure 4a). For all of the complexity metrics to be discussed later on, the simplest code for this symbol series is 3×(k3×(ak)). This code yields the chunking (kakakak)(kakakak)(kakakak) as the dominant segmentation, corresponding to the triangle interpretation of the pattern. Another way to represent the pattern is by means of the symbol series 'kakbkakbkakbcl' (see figure 4b). The simplest code of this symbol series is 3×(⟨(k)⟩/⟨(a)(b)⟩)cl. This code yields the clustering (kakbkakbkakb)cl as the dominant segmentation, corresponding to the zigzag interpretation of the pattern. Now, the code that reflects the triangle interpretation is simpler than the code that reflects the zigzag interpretation and, indeed, the triangle interpretation is usually preferred.

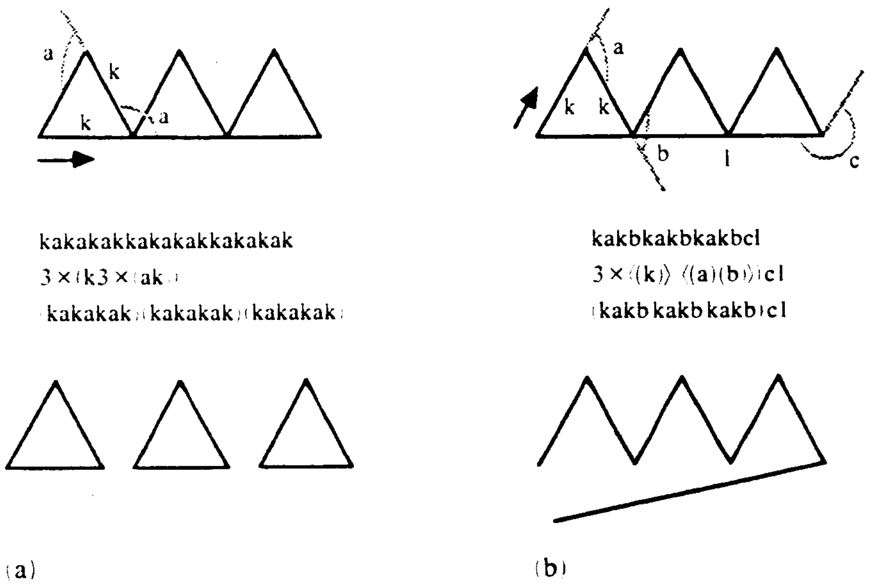


Figure 4. The pattern in figure 1 can be represented in two ways by means of the symbol series shown in (a) and (b). For each symbol series, the simplest code is shown with the induced dominant segmentation in the symbol series and, correspondingly, in the pattern. The code in (a) is simpler than the code in (b) and, indeed, reflects the usually preferred triangle interpretation.

3 The concept of regularity

In the paper of van der Helm and Leeuwenberg (1991), a formalization of the intuitive concept of regularity has been given within the framework provided by the structural information model. In this section, we will summarize this formalization in a nonformal way, ie we will, by means of examples, give a gist of the model in order to put the new complexity metric in perspective. The formalization is twofold: first, the intrinsic character of regularity is specified by the formal notion of holographic regularity, and second, the way in which cases of regularity can be related hierarchically is specified by the formal notion of transparent hierarchy. In this section, these two formal notions will be discussed successively, after which several psychologically relevant implications will be discussed.

3.1 *Holographic regularity*

As we have seen, the structural information model represents a visual pattern by means of a symbol series in which the symbols refer to pattern parts (the semantic mapping). Therefore, an interpretation of the pattern is considered to be reflected in the classification and organization of the symbol series which, because of the semantic mapping, corresponds to a classification and an organization of the pattern. By using coding rules, the symbol series is classified and organized on the basis of a hierarchical description of regularity in the symbol series. Note that the symbol series only represents information about the order and the identity of pattern parts. So, in this context, the formalization of regularity can only be based on arrangements of identical symbols in a symbol series. Now, the formalization proceeds as follows.

First, any arrangement of identical symbols in a series is formally called a case of regularity, no matter whether or not it reflects something intuitively regular. That is, we start with all theoretically possible cases, and eg a repetition of a specific something a specific number of times (such as nine times the symbol 'p') is just one of these cases. So, the arrangement of identical symbols in a series is formally called a case of regularity, not only for series like 'aaaa' and 'abccba', but also for series like 'kpfzpkf' and 'kyppzfkf'. Now, for example, in the series 'yy', the arrangement of identical symbols is formally denoted by the expression (1) = (2), which simply indicates that the first symbol is identical to the second symbol. The expression (1) = (2) is called an identity. Note that the same identity denotes the arrangement of identical chunks in, for example, the chunk series (ab)(ab). A set of identities is called an identity structure, if the set meets some formal requirements. One of the requirements is that the identities in the set are ordered. For instance, the case of regularity reflected by the arrangement of identical symbols in the series 'aaaa', is formally denoted by the identity structure {(1) = (2), (2) = (3), (3) = (4)} which consists of three identities in the given order.

Second, all different cases of regularity are categorized into kinds of regularity, which may be illustrated as follows. For the series 'abab', the arrangement of identical symbols can be denoted by, among others, the identity structure {(1 2) = (3 4)} which indicates that the two subseries 'ab' are identical. In this identity structure, each of the two subseries 'ab' is taken as one unit, just like when the series 'abab' would be chunked into the chunk series (ab)(ab). In other words, the identity structure expresses that 'abab' is the same as 'yy' under the substitution of 'ab' for 'y'. Now, observe that, in the series 'yy' or (ab)(ab), the identity structure {(1) = (2)} actually expresses the same as the identity structure {(1 2) = (3 4)} in the symbol series 'abab', namely the identity of the two subseries 'ab'. Therefore, although these two identity structures describe different cases of regularity, they are said to describe the same kind of regularity (namely, in words, a repetition of two times an arbitrary something). Note that, formally, two times an arbitrary something is a different kind of regularity from that of three times an arbitrary something. In summary, a repetition of a specific something a specific number of times is a case of regularity, and a repetition of an arbitrary something a specific number of times is a kind of regularity. As we will argue next, repetition in general (an arbitrary something an arbitrary number of times), is a case of holographic regularity.

Above, we saw that an identity structure reflects a property of a series, namely identity of elements in that series. Now, holographic regularity specifies a possible property of identity structures. Metaphorically, the notion of holographic regularity may be illustrated by a jigsaw puzzle that is to represent a landscape consisting of a green lawn and a sky with clouds. Each green piece in the disordered set of pieces can be classified as a lawn-piece, since such a piece shows the holographic property of having the same color as the entire lawn. In our view, such a holographic property applies quite well to the intuitive concept of regularity. For instance, a repetition of something is a repetition no matter whether the number of times that the something is repeated is large (analogous to the jigsaw lawn) or small (analogous to the jigsaw piece). The formal notion of holographic regularity has been elaborated as follows.

An identity structure can be seen as a chain of identities because it is an ordered set. This implies that one can define a substructure of an identity structure as an ordered subset of successive identities in the identity structure or, in other words, as a subchain. For instance, the earlier-mentioned identity structure {(1) = (2), (2) = (3), (3) = (4)} has five substructures, namely {(1) = (2), (2) = (3)}, {(2) = (3), (3) = (4)}, {(1) = (2)}, {(2) = (3)}, and {(3) = (4)} (see also figures 5 and 7). Analogous to the jigsaw puzzle and expressed simply, an identity structure is said to describe a holographic kind of regularity if its substructures all describe the same kind of regularity (see also figure 5).

Observe the recursive character of the notion of holographic regularity; if an identity structure describes a holographic kind of regularity, then any of its substructures describes a holographic kind of regularity too, since all substructures of the identity structure (including the substructures of a substructure) describe the same kind of regularity. This recursive character is essential for elaborating the formal implications of the notion of holographic regularity. Without going into details, these formal implications can be summarized as follows [for the details, see van der Helm and Leeuwenberg (1991)].

Instead of considering all possible identity structures (an infinite number), it is more suitable and suffices to consider only all identity structures consisting of precisely three identities (also an infinite number). One can prove, first, that the infinite number of different cases of regularity, as described by these identity structures consisting of three identities, can be categorized into precisely 648 different kinds of regularity and, second, that only 20 of these 648 kinds of regularity are holographic. Then, because of the recursive character of holographic regularity, one can prove two further things. First, for each of those 20 holographic kinds of regularity, a representative identity structure (consisting, as before, of three identities) can be generalized uniquely into an identity structure consisting of an arbitrary number of identities (analogous to simply increasing the number of times in a repetition). Each of these 20 generalized identity structures is called a case of holographic regularity. Second, one can prove that these 20 cases constitute all possible cases of holographic regularity. Figure 6 shows some of these. Finally, ignoring syntactical variations in the definitions of specific coding rules (since such variations are irrelevant with respect to the meaning of coding rules), one can prove that each of the

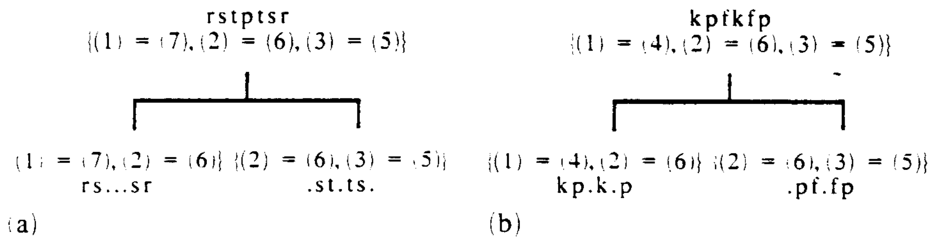


Figure 5. Holographic regularity. In (a), the arrangement of identical symbols in the series 'rstptsr' is denoted by an identity structure consisting of three identities. Below this identity structure are shown its two substructures consisting of two identities each. Each of these substructures is visualized by substituting, in 'rstptsr', a dot for those symbols to which the substructure does not apply. In (b) exactly the same procedure has been followed for the series 'kpfkfp'. In (a), both substructures reflect the same arrangement of identical symbols. That is, the two symbols 'r' and the two symbols 's', in the visualization of the first substructure, are arranged in the same way as the two symbols 's' and the two symbols 't' in the visualization of the second substructure. In other words, both substructures describe the same kind of regularity, which is precisely the reason that the identity structure shown at the top is said to describe a holographic kind of regularity. In contrast, in (b), the two substructures clearly do not reflect the same arrangement, ie do not describe the same kind of regularity, so that the identity structure shown at the top does not describe a holographic kind of regularity.

20 cases of holographic regularity can be described by precisely four different coding rules. So, the final result is that only precisely eighty so-called holographic coding rules exist. Among these eighty coding rules are the ISA-rules employed in the structural information model. Figure 7 shows, for the iteration rule, a scheme that typically holds for any holographic coding rule.

Now, if holographic regularity is accepted as being relevant and if eighty holographic coding rules exist, one might wonder why the structural information model employs only the ISA-rules. Well, the answer lies in the fact that, above, only the intrinsic character of regularity has been dealt with. So far, nothing has been said about the way cases of regularity can be related hierarchically. The latter aspect is, in the formalization, specified by the formal notion of transparent hierarchy. This notion is, even more than the notion of holographic regularity, relevant with respect to the complexity metric to be proposed, and will be discussed next.

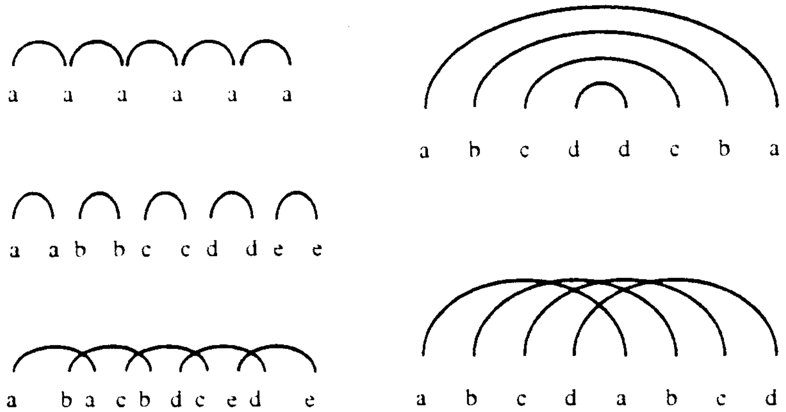


Figure 6. Five characteristic visualizations of holographic regularity. In the five prototypical symbol series, each identity relationship between two symbols is visualized by an arc. For each series, the set of arcs shows a regular ordering. The holographic property of this ordering is reflected by the fact that the first or the last arc in each set can be removed without distorting the regular ordering in the set of arcs.

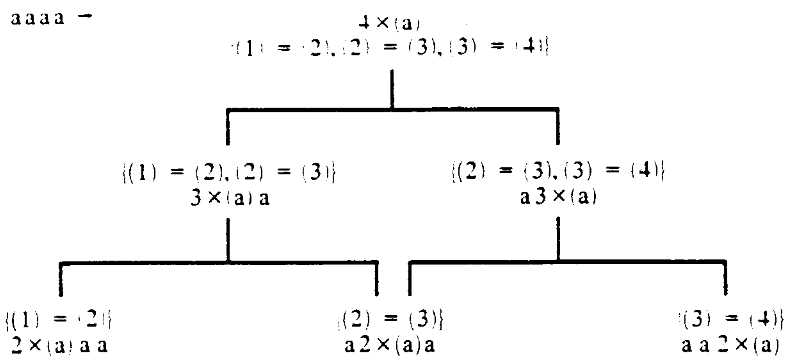


Figure 7. The holographic iteration rule. The series 'aaaa' can be encoded into the I-form shown at the top, expressing the identity structure (chain) consisting of three identities. For each substructure (subchain) of this identity structure, an I-form exists such that it expresses that substructure. This indicates that the iteration rule is a holographic coding rule.

3.2 *Transparent hierarchy*

Metaphorically, transparent hierarchy may be illustrated by the hierarchical structure of an industrial organization. In such an organization, the position of the manager may rely on that structure, but the manager can also be approached independently of the other employees. The analogue in terms of symbol series may be illustrated as follows.

In encoding models like the structural information model, the coding rules yield hierarchical descriptions of regularity in a symbol series. For instance, as we saw before, the series 'ababbaba' can be encoded into the S-form S[(a)(b)(a)(b)]. In this S-form, the S-argument (a)(b)(a)(b) is said to represent a higher hierarchical level and can be encoded into the I-form 2×((a)(b)). The latter code can be nested in the S-form, yielding the hierarchical code S[2×((a)(b))]. Note that the S-form describes regularity in the symbol series, but that the I-form describes regularity in the S-argument at a higher hierarchical level. So, one might wonder what the meaning of the I-form is with respect to the description of regularity in the symbol series 'ababbaba'. Now, observe that the I-form 2×((a)(b)) in the S-argument (a)(b)(a)(b) describes a kind of regularity (a repetition of two times something) that is also described by the I-form 2×(ab) in the subseries 'abab' of the symbol series 'ababbaba'. This example illustrates a general characteristic of the S-rule, namely that any kind of regularity in the argument of an S-form corresponds unambiguously to the same kind of regularity in the symbol series. So, in an almost visual sense, any S-form is 'transparent', ie regularity in the symbol series can be 'seen through' the S-form. And indeed, because of this general characteristic, the S-rule is called a transparent coding rule. For the example above, this implies that the hierarchical code S[2×((a)(b))] can be seen as indicating that the S-form S[(a)(b)(a)(b)] and the I-form 2×(ab) can be related hierarchically. Inversely, this implies that, instead of the I-form, 2×((a)(b)), in the S-argument, one could just as well consider the I-form, 2×(ab), in the symbol series. Thus, analogous to the manager in the metaphor, the higher-level I-form can be considered independently of the lower-level S-form. Moreover, note that, in the example above, the I-form 2×(ab) induces the chunking (ab)(ab) in the subseries 'abab' of the symbol series 'ababbaba'. This chunking can be superimposed on the chunking (a)(b)(a)(b)(b)(a)(b)(a) induced by the S-form in the symbol series, to yield the hierarchical chunking ((a)(b))((a)(b))(b)(a)(b)(a) in the symbol series (see figure 8). Such a hierarchical chunking can be assigned unambiguously to any hierarchical code obtained by means of transparent coding rules and is, therefore, called a transparent hierarchy. So, a hierarchical code obtained by means of transparent coding rules indicates how different cases of regularity in a symbol series can be related hierarchically, and the resulting hierarchical code induces a hierarchical chunking in the symbol series.

	level 1:	a b	a b	b a	b a
S[(a)(b)(a)(b)]	↓				
	level 2:	(a)(b)	(a)(b)	(b)(a)	(b)(a)
S[2×((a)(b))]	↓				
	level 3:	((a)(b))	((a)(b))	(b)(a)	(b)(a)

Figure 8. The transparent symmetry rule. The S-form S[(a)(b)(a)(b)] induces a chunking in the level 1 symbol series, represented by the level 2 series. The I-form, 2×((a)(b)), in the S-argument corresponds unambiguously to the I-form, 2×(ab), in the level 1 series, inducing the chunking (ab)(ab) which can be superimposed on the level 2 series, and leading to the hierarchical chunking represented in level 3.

From the example just given, one may get the impression that, for coding rules, transparency is only a plausible requirement and may even be rather trivial. This may be the case for the S-rule but, for coding rules in general, transparency is far from trivial. This may be illustrated by means of the so-called M-rule which is one of the eighty holographic coding rules mentioned earlier, and is defined as follows:

$$k_1 x_1 k_1 \; k_2 x_2 k_2 \; ... \; k_n x_n k_n \; \rightarrow \; M[(k_1)(k_2) ... (k_n), (x_1)(x_2) .. (x_n)] \, .$$

The M-rule can be applied to express that a series contains successive subseries which each begin and end identically. Now, consider the symbol series 'arabsb ayabzb' which, by means of the M-rule, can be encoded into the M-form M[(a)(b)(a)(b), (r)(s)(y)(z)]. The first argument of this M-form, (a)(b)(a)(b), can be encoded into the I-form $2 \times ((a)(b))$. However, this I-form describes, in the first M-argument, a kind of regularity (a repetition of something twice, ie two successive identical subseries) which does not occur in the symbol series itself. Therefore, the M-rule is not a transparent coding rule. In fact, one finds in this way that only nine of the eighty holographic coding rules are transparent coding rules, among which are the ISA-rules employed in the structural information model.

Since the transparency of the ISA-rules is important with respect to the new complexity metric, we will now go into the transparency of the I-rule and the A-rule in some detail. The I-rule is transparent 'by default', since the I-argument of an I-form consists of only one chunk so that, at the higher hierarchical level, no regularity can be described. The transparency of the A-rule, however, is not that trivial. We saw that the A-rule can be applied to express that a series contains successive subseries which either all begin or all end identically, as follows:

$$\mathrm{k}x_1\,\mathrm{k}x_2 \ldots \mathrm{k}x_n \rightarrow \langle(\mathrm{k})\rangle\,\langle(x_1)(x_2)\ldots(x_n)\rangle$$

$$x_1\mathrm{k}\,x_2\mathrm{k} \ldots x_n\mathrm{k} \rightarrow \langle(x_1)(x_2)\ldots(x_n)\rangle\,\langle(\mathrm{k})\rangle\,.$$

Clearly, the two cases are very similar, and we will discuss only the transparency for the first case. Suppose the symbol series 'kxkykykz' is encoded into the A-form ⟨(k)⟩ ⟨(x)(y)(y)(z)⟩. Then the subseries (y)(y) in the A-argument can be encoded into the I-form $2 \times ((y))$, yielding the hierarchical code ⟨(k)⟩ ⟨(x)$2 \times ((y))$(z)⟩. Now, observe that the I-form, $2 \times ((y))$, in the A-argument does not correspond to an I-form, $2 \times (y)$, in the symbol series 'kxkykykz'. Yet the regularity described by the I-form as in the A-argument does correspond unambiguously to the same kind of regularity in the symbol series. That is, the I-form in the A-argument corresponds unambiguously to the I-form, $2 \times (\mathrm{ky})$, in the symbol series. Note that, in the latter I-form, the I-argument (ky) corresponds to an A-pair cluster as discussed before. Similarly, in general (see definition above), for an A-form ⟨(k)⟩ ⟨$(x_1) \ldots (x_n)$⟩, any regularity in the argument $(x_1)(x_2) \ldots (x_n)$ corresponds unambiguously to the same kind of regularity in the series $(\mathrm{k}x_1)(\mathrm{k}x_2) \ldots (\mathrm{k}x_n)$, consisting of A-pair clusters. Clearly, any kind of regularity in this cluster series corresponds unambiguously to the same kind of regularity in the symbol series '$\mathrm{k}x_1\,\mathrm{k}x_2 \ldots \mathrm{k}x_n$', thus showing that the A-rule is indeed a transparent coding rule. Furthermore, in the example above, the A-form ⟨(k)⟩ ⟨(x)(y)(y)(z)⟩ induces the chunking (k)(x)(k)(y)(k)(y)(k)(z) in the symbol series, while the I-form $2 \times (\mathrm{ky})$ induces the chunking (ky)(ky) in the subseries 'kyky' of the symbol series. Clearly, the latter chunking can be superimposed on the former

level 1: k x k y k x

⟨(k)⟩ ⟨(x)(y)(x)⟩ ↓

level 2: (k)(x) (k)(y) (k)(x)

⟨(k)⟩ ⟨S[((x)), ((y))]⟩ ↓

level 3: ((k)(x)) ((k)(y)) ((k)(x))

Figure 9. The transparent Alternation rule. The A-form ⟨(k)⟩ ⟨(x)(y)(x)⟩ induces a chunking in the level 1 series, represented by the level 2 series. The S-form S[((x)), ((y))] in the second argument of the A-form corresponds unambiguously to the S-form S[(kx), (ky)] in the level 1 series, inducing the chunking (kx)(ky)(kx) which can be superimposed on the level 2 series, leading to the hierarchical chunking represented at level 3.

chunking, yielding the hierarchical chunking (k)(x)((k)(y))((k)(y))(k)(z) (see also figure 9). So, in order to understand the hierarchical character of the A-rule, one should replace each chunk in the A-argument by the related A-pair, in order to obtain the chunking at the higher hierarchical level.

3.3 *Implications of formal regularity*

We have given an overview of the formalization of the intuitive concept of regularity, specified by the formal notions of holographic regularity and transparent hierarchy. Holographic regularity applies to the intrinsic character of regularity, and reflects the fact that, for example, an arbitrary repetition belongs to the set of all repetitions. Transparent hierarchy applies to the way different cases of regularity can be related hierarchically, and implies that the hierarchical character of codes is not just a syntactical artifact of the coding language, but a psychologically meaningful aspect of the description of regularity. The result of the formalization is that only eighty coding rules are such that they describe holographic regularity, and that only nine of these eighty coding rules are such that they describe transparent hierarchy too. Among these nine transparent holographic coding rules are the ISA-rules employed in the structural information model. Actually, the other transparent holographic coding rules are superfluous in the sense that they describe only identities that can also be described by the ISA-rules, whereas the ISA-rules can describe other identities as well. This implies that, according to the formalization, a set of appropiate coding rules may consist of just the ISA-rules. This result as such is not very surprising since the kinds of regularity described by the ISA-rules are widely accepted as being relevant in perception and have been emphasized by various scientists (eg Wertheimer 1923; Koffka 1935; Palmer 1977). The way this result has been obtained is what matters here, because the ISA-rules now have a unique formal status which is psychologically relevant. The psychological plausibility is supported by several further implications of the formalization, as we will argue next.

If holographic coding rules are used to extract pattern information (regularity) from a symbol series that represents a pattern, then the extraction is very easy, ie the pattern information to be extracted is very accessible. This may be illustrated by the holographic I-rule, for which I-forms can be constructed in a simple stepwise fashion, starting with any single identity and with each step adding just one identity, eg as follows (see also figure 7):

$$\text{aaaa} \rightarrow 2\times(\text{a})\text{aa} \rightarrow 3\times(\text{a})\text{a} \rightarrow 4\times(\text{a}).$$

So, for the I-forms, the construction proceeds from one I-form towards another I-form, each step enlarging the expressed identity structure by one identity. Clearly, this is possible because of the holographic property that any iteration belongs to the collection of all iterations. Thus, complex combinations of identities do not have to be matched, since the construction involves 'atomic' steps of one identity at a time.

Maybe even more than the notion of holographic regularity, the notion of transparent hierarchy results in the accessibility of pattern information as contained in a symbol series. As we saw before, the transparency of coding rules implies that regularity at a higher hierarchical level in a code of the series corresponds unambiguously to the same kind of regularity in the series itself. Since the classification and the organization of the series is based on the described regularity, the transparency implies that higher cognitive levels will deal with (or have access to) basic pattern information itself. That is, the information that is considered to be passed on to higher cognitive levels is not, as with an artifact of the employed model, based on just regularity at a higher hierarchical level in a code, but on regularity in the pattern.

Moreover, the notion of transparent hierarchy yields the possibility of a largely parallel encoding process, as follows. As we saw before, the hierarchical code S[2 × ((a)(b))] can be seen as indicating that the S-form S[(a)(b)(a)(b)] and the I-form 2 × (ab) can be related hierarchically. This implies that the I-form and the S-form can be constructed independently of each other, and then tested for hierarchical compatibility. This shows that, in general, all single ISA-forms in a symbol series can be constructed in parallel after which they can be tested, in parallel, in pairs for hierarchical compatibilty. Thus, on the one hand, the encoding may result in a hierarchy consisting of sequentially ordered hierarchical levels, but, on the other hand, this hierarchy does not have to be established in a strictly sequential way. This illustrates that, in our view, the notion of hierarchy should not be based on some process model as such, as in, for example, the so-called hierarchical sequential search model of Simon and Feigenbaum (1964), or as in other so-called top-down models that involve reasoning or unconscious inferences (von Helmholtz 1909/1962; Neisser 1966; Gregory 1972; Rock 1983). Nor should hierarchy be seen, as suggested in Buffart (1987), as a property of a single pattern representation in which several different hierarchical levels can be distinguished by decomposing that representation [see also van der Helm (1988)]. On the contrary, hierarchy should rather be seen as a relation between several different representations of the same pattern. That is, several different ISA-forms, obtained in parallel, may be related hierarchically in order to compose a hierarchy consisting of several different hierarchical levels. In the latter sense, the notion of hierarchy agrees with a bottom-up extraction of gradually more structured information. That is, starting from the single identities in a 'raw' pattern registration, regularity is described first and then different kinds of regularity are related hierarchically, resulting in a classification and an organization which can be 'embedded' in stored knowledge structures.

Another implication of the formalization is related to the problem that the minimum principle seems to require an unrealistic search for simplest codes since, for an arbitrary symbol series, the number of possible codes is combinatorially explosive (cf Hatfield and Epstein 1985). This problem does not depend on the exact complexity metric that is used. For instance, if a series can be encoded entirely into one S-form (or A-form) then, in general, the series can be encoded entirely into an exponential number of S-forms (or A-forms). Because of the notions of holographic regularity and transparent hierarchy, however, these S-forms (or A-forms) can all be stored and encoded simultaneously, as if it were just one S-form (or A-form). Thus, the search for the simplest S-form (or A-form) becomes realistic, since it does not involve an explosive amount of storage space or processing time. For detailed information on this problem and its solution, we refer to van der Helm and Leeuwenberg (1986, 1991) and van der Helm (1988).

The discussion in this subsection shows that the formalization of regularity not only provides a psychological basis for the choice of appropriate coding rules, but also has further implications that are psychologically relevant. The implication that is most relevant in the present paper is the fact that the formalization paves the way for a new complexity metric. This implication will be discussed next.

4 The new complexity metric

In this section, we will discuss and evaluate complexity metrics on the basis of the analysis in the previous section. First, in order to clarify the need for a new metric, we consider three metrics that have been used more or less frequently in earlier research on the structural information model. Then we introduce the new metric.

4.1 I_{old}-*load*

In many studies concerning the structural information model, the complexity of a code has been measured by means of the I_{old}-load (Leeuwenberg 1971). This load was meant to reflect the amount of memory space needed to represent a code, ie the preferred pattern interpretation is assumed to be reflected by the code that requires a minimum of storage space. Therefore, an assumption has been made with respect to the relation between the syntactical components in a code and the memory space needed to represent the code, as follows. First, the encoding of a series yields a reduction of the series into a code such that not all of the pattern symbols in the series, but only those in the code, need to be represented in memory. Second, the means of establishing the reduction have to be taken into account as well. These means consist of the ISA-forms, and are taken into account as follows. For decoding a stored I-form like $5\times(\mathrm{ab})$ into the series 'ababababab', the numeric value '5' has to be known and is, therefore, assumed to be represented in just as much memory space as each of the two pattern symbols, so that this I-form requires three units of memory space. In decoding a stored S-form like $S[(a)(b)(c)]$ into the series 'abccba', the S-argument has to be reversed in order to produce the second half of the series; this reversal operation is assumed to be represented in, again, just as much memory space as each of the three pattern symbols, so that this S-form requires four units of memory space. For decoding a stored A-form like $\langle(a)\rangle/\langle(r)(s)(t)\rangle$ into the series 'arasat', the number of times that the part 'a' has to be repeated does not have to be stored since this number is implicitly given by the number of elements in the A-argument $(r)(s)(t)$; therefore, the storage of this A-form requires only four units of memory space, ie only for the four pattern symbols. So, for an arbitrary code, one has to count the pattern symbols, the I-forms, and the S-forms in the code to determine the I_{old}-load of the code, ie the complexity of the code in terms of storage space.

The structural information model has gained empirical support by using the I_{old}-load as a complexity metric. However, the I_{old}-load gives rise to several conceptual problems. First, the pattern symbols, the numeric value in an I-form, and the reversal operation for an S-form, are given equal value but are in fact incomparable entities, or at least very different entities. For example, why not assign the value of the numeric value and of the reversal operation as being equal to two or more pattern symbols? Second, the reversal operation for an S-form is counted, but not the iteration operation for an I-form, nor the alternation operation of an A-form. These conceptual problems indicate that the I_{old}-load is based on a dubious, and in our view unacceptable, assumption concerning the relation between syntactical components in a code and the memory space needed to represent the code. That is, in our view [see also Hatfield and Epstein (1985)] this assumption depends too much on artifacts of the encoding language. Moreover, we think that the semantical implication of a code (ie an interpretation) should not be judged on the basis of the required memory space but, psychologically more meaningfully, on the basis of the semantical content of the code (ie the description of regularity). The conceptual problems with respect to the I_{old}-load have been recognized in research concerning the structural information model; the following complexity metric was devised to overcome these problems.

4.2 *P-load*

In order to determine the *P*-load of a code, one simply counts the number of pattern symbols in that code (cf Collard and Buffart 1983). For instance, for the code $3\times(\mathrm{ab})$, the value of the *P*-load, is $P = 2$, because it contains only two pattern symbols, 'a' and 'b'. This metric implies a judgement of the resulting classification, as follows. For instance, as we saw in the second section, if the series 'aaaaaa' is encoded into the I-form $3\times(\mathrm{aa})$, for which $P = 2$, then the series is considered to be classified into the class of series that can be represented by, for example, the series 'yzyzyz'. Occasionally, such a class representative has been called an abstract code (cf Collard and Buffart 1983). Now note that, in the example above, the I-form $3\times(aa)$ described regularity in the symbol series 'aaaaaa' by expressing the identity of the elements contained in the abstract code 'yzyzyz'. Therefore, the residual nonidentity of elements in the abstract code can be regarded as the irregularity in the symbol series (at least, according to the I-form). Now, the *P*-load of a code equals the number of different elements in the corresponding abstract code and, therefore, implies a judgement of the resulting classification by measuring irregularity: the more different elements, the more irregularity, the more complex.

So, the *P*-load applies to the output of the encoding model, ie it does not depend on artifacts of the encoding language. Therefore, conceptually, the *P*-load seems better than the I_{old}-load. Yet the *P*-load has not been used frequently because it yields worse predictions than does the I_{old}-load. A possible explanation for this leads, as follows, to a third metric.

4.3 I_{A}-*load*

On the one hand, the *P*-load accounts only for the resulting classification, not for the resulting organization. Since both the classification and the organization are part of the output, one could say that the *P*-load is incomplete so that the *P*-load still gives rise to a conceptual problem. On the other hand, the I_{old}-load can be seen as accounting for not only the resulting classification but also, although only to some extent, for the resulting organization, as follows. First, note that the I_{old}-load of a code equals the *P*-load of the code plus the number of I-forms and S-forms in the code. That is, the I_{old}-load can be seen as accounting for the classification in the same way as does the *P*-load. Now suppose that, instead of only the I-forms and S-forms, the A-forms in a code are also counted. Let us call this adapted metric the I_{A}-load. Now, recall that the argument of an ISA-form represents a higher hierarchical level (in the code as well as, as we saw before, in the resulting organization). That is, each ISA-form in a code yields a transition to a higher hierarchical level. Thus, counting all ISA-forms in a code corresponds to counting all hierarchical levels. In this way the I_{A}-load can be seen as taking into account the resulting organization. So, similarly, the I_{old}-load can be said to account for the resulting organization (though only to some extent since the A-forms are not counted), which might explain why it yields better predictions than does the *P*-load. For the same reason, we expect (and hypothesize in the experiment to be discussed) that the I_{A}-load, which also counts the A-forms, should yield better predictions than the I_{old}-load.

However, this way of accounting for the resulting organization gives rise, as before, to a conceptual problem. Namely, hierarchical levels and pattern symbols are valued equally but are, in fact, incomparable entities. That is, whereas the number of pattern symbols refers directly to something (namely, irregularity) that is present in a symbol series, the number of hierarchical levels refers only to the presence of those levels as such, ie not to something that is present at those levels. The new complexity metric does not show such a conceptual problem, as we will see next.

4.4 I_{new}-*load*

To a large extent, the new complexity metric, the I_{new}-load, is based on the concept of transparent hierarchy, as discussed in section 3.2. In the present study, we have seen that if a symbol series is encoded by means of the ISA-rules, then the resulting code unambiguously induces a transparent hierarchy, ie a hierarchical chunking, in the symbol series (see figures 8 and 9). Now, imposing the same hierarchical chunking on the corresponding abstract code (as defined when discussing the *P*-load), yields an organized class representative which reflects both the resulting classification and the resulting organization of the symbol series. For instance, if the symbol series 'abcabcab' is encoded into the S-form S[(ab)(c), (ab)], then the classification is represented by the abstract code 'xztpqtxz', whereas the organization is represented by the chunk series (ab)(c)(ab)(c)(ab). Imposing this same chunking on the abstract code yields the expression (xz)(t)(pq)(t)(xz). Such an expression will be said to represent an abstract chunking. Note that, in general, such an abstract chunking exists only for codes obtained by means of transparent coding rules.

Now, in the new complexity metric, the I_{new}-load, all the different elements over all the hierarchical levels in such an abstract chunking are counted. For instance, for the S-form above, $I_{new} = 7$, since the abstract chunking contains the following seven different elements: 'x', 'z', '(xz)', 't', 'p', 'q' and '(pq)'. Note that the lower-level element 't' is not different from the higher-level element '(t)' since, in this case, the parentheses merely indicate a chunk that equals one symbol. That is, in addition to the different single symbols, only the different chunks consisting of at least two symbols or chunks are counted. This may be illustrated by two further examples. In figure 8, we saw that the encoding of the symbol series 'ababbaba' into the code S[2 × ((a)(b))] yields a transparent hierarchy which is represented by the expression ((a)(b))((a)(b))(b)(a)(b)(a). Since the code characterizes all the identity of elements in the symbol series, the latter expression also represents the abstract chunking. This abstract chunking contains three different elements, namely, 'a', 'b', and '((a)(b))', so that the code has an I_{new}-load value of $I_{new} = 3$. Furthermore, in figure 9, we saw that the encoding of the symbol series 'kxkykx' into the code ⟨(k)⟩ ⟨S[((x)), ((y))]⟩ yields a transparent hierarchy which is represented by the expression ((k)(x))((k)(y))((k)(x)). Again, the code characterizes all the identity of elements in the symbol series, so that the latter expression also represents the abstract chunking. This abstract chunking contains five different elements, namely, 'k', 'x', '((k)(x))', 'y', and '((k)(y))', so that the code has an I_{new}-load value of $I_{new} = 5$.

Now that we have introduced the I_{new}-load, we first of all want to emphasize that this complexity metric does not depend on artifacts of the coding language because it is based solely on the output, ie on the classification and organization. Therefore, it does not show the conceptual problems connected with the I_{old}-load. A better account is now given of the hierarchical structure by counting not just hierarchical levels but the different elements at those levels. Counting the different elements at some level corresponds, as we saw when discussing the P-load, to measuring the irregularity at that level. In other words, the I_{new}-load can be said to account for the hierarchical structure by quantifying its contribution to pattern complexity in terms of the irregularity at higher levels. In this way, the I_{new}-load can be said to measure pattern complexity adding irregularity and hierarchy.

In section 5, we will discuss an experiment that has been designed to test the new complexity metric by considering the dominant segmentation in the perceptual organization as predicted on the basis of the simplest code (as discussed before). We argued that one has to take notice of not only the chunking but also the clustering as part of the perceptual organization of a pattern. That is, for instance, the series 'abab' is chunked into (ab)(ab) in order to describe the iteration regularity by means of the I-form 2 × (ab), which implies that the entire series 'abab' is clustered into one regularity structure which is represented by the cluster (abab). Note that clustering is not taken into account in the new complexity metric. Whereas chunks are needed to construct ISA-forms, clusters are 'only' a consequence of ISA-forms. Therefore, clustering may be relevant in experimental settings but is not relevant with respect to the complexity of ISA-forms.

5 Experiment

The following experiment has been designed to test the structural information model by using the new metric of complexity, ie the I_{new}-load as discussed in the previous section. This test comprises a comparison, with respect to preferred pattern segmentations, between the new metric and the other metrics discussed in section 4, ie the P-load, the I_{old}-load, and the I_{A}-load. Each pattern, as used in the experiment, is a patterned sequence in which the elements are drawn from one of three different sets of graphic symbols (see figures 10 and 11). For each pattern, subjects were asked to indicate the preferred pattern segmentation. In each trial, subjects were forced to choose between two given pattern segmentations. Each of the two segmentations was the dominant segmentation in the perceptual organization as induced by a code obtained by means of the ISA-rules. So, given one of the complexity metrics, one could check in each trial which of the two codes was the simpler one and whether or not that code indeed induces the preferred segmentation.Thus, one can investigate which of the complexity metrics performs best.

Patterned sequences were chosen because a straightforward semantic mapping can be assumed between a (serial) pattern code and the pattern. That is, still existing unclarities with respect to the semantic mapping, eg for (nonserial) two-dimensional patterns, or (serial) auditory patterns, are excluded. The specific stimuli were selected such that the various metrics mostly yield different predictions (except for the P-load which, as will be clear, is used as a sort of 'baseline').

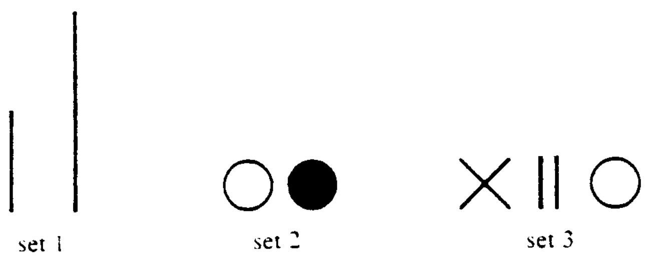


Figure 10. The three sets of graphic symbols used in the experiment. To construct a stimulus, graphic symbols from one set are drawn and composed into a patterned sequence. The graphic symbols within a set are very distinctive phenomenally (short versus long; white versus black; crossed versus parallel versus circular), so that they can be considered to be basic pattern elements.

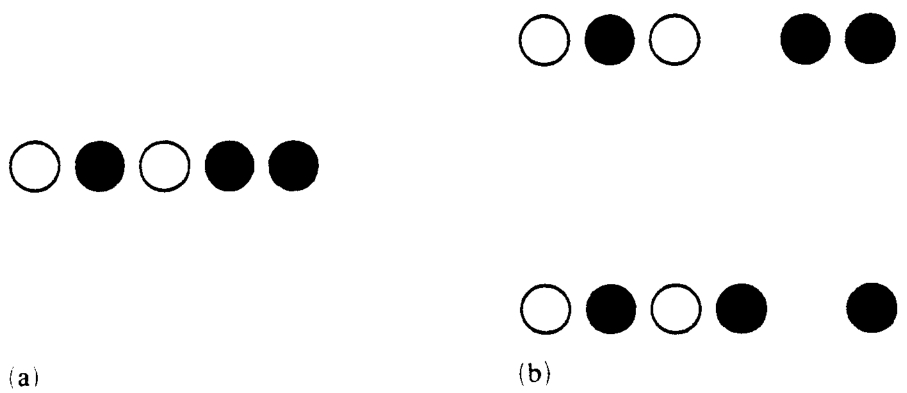


Figure 11. A target (a) and response alternatives (b) used in the experiment. The pattern sequence at the left was constructed by assigning graphic symbols from set 2 in figure 10 to the symbols in the symbol series 'ababb'. In the segmentations (shown in b) a large space between two graphic symbols indicates a border between two segments. Subjects were asked to indicate which of the two segmentations they prefer as partitioning of the target into coherent parts.

5.1 *Hypothesis*

In line with the minimum principle we predict that the simpler the code of a pattern, the greater the preference for the segmentation induced by that code. Now, for some complexity metric M, let G_M be the goodness of M, ie, the amount of preferred pattern segmentations that M predicts correctly. Then, on the basis of the theoretical analysis in the previous sections, we hypothesize that:

$$G_{P\text{-load}} < G_{I_{old}\text{-load}} < G_{I_{\Lambda}\text{-load}} < G_{I_{new}\text{-load}} \, .$$

A brief explanation of this is as follows: in the previous section, we argued that the complexity of a pattern code is constituted by the irregularity and the hierarchy represented in that code. Schematically, the four metrics count the following code components:

P-load —pattern symbols,
I_{old}-load —pattern symbols plus I-forms and S-forms,
I_{Λ}-load —pattern symbols plus ISA-forms,
I_{new}-load—pattern symbols plus chunks .

By counting the pattern symbols, the four metrics all account in the same way for irregularity, so that differences between the metrics have to be sought in the way of accounting for hierarchy. According to the theoretical analysis and the hypothesis, the successive metrics account for hierarchy in a better way: the P-load does not take hierarchy into account: the I_{old}-load counts some of the higher hierarchical levels in a code: the I_{Λ}-load counts all higher hierarchical levels in a code; and the I_{new}-load counts hierarchy in terms of irregularity at higher hierarchical levels. Note that the main issue will be to contrast the I_{new}-load with the I_{old}-load, the most frequently used metric in earlier empirical research on the structural information model. The less frequently used P-load and I_{Λ}-load are considered mainly to obtain a more detailed view of the adequateness of the theoretical analysis in the previous sections.

5.2 *Method*

5.2.1 *Subjects.* Thirty-one undergraduates received course credit to participate in the experiment.

5.2.2 *Materials.* To construct the patterns for the experiment, ie the patterned sequences of graphic symbols, three different sets of graphic symbols were used (see figure 10). For each pattern, graphic symbols were drawn from one set of such symbols to construct a sequence (see figure 11a). In each set, the graphic symbols are 'semantically independent', ie are very distinct phenomenally (set 1: short versus long; set 2: white versus black; set 3: cross versus parallel versus circular). This ensures that subsequent graphic symbols in a pattern will not be grouped together because of the 'visual distance' (cf Tversky and Gati 1982), such as for a regular increase of grey-value, or because they constitute, for example, a complementary pair of parentheses. This, together with the symmetry in each graphic symbol, implies that the symbols can be considered as 'primitives', ie, as basic pattern elements which are not sensitive to biases that are irrelevant with respect to the goal of this experiment. This ensures that a straightforward semantic mapping can be assumed between such a patterned sequence and a symbol series as used in the structural information model. For instance, the pattern in figure 11a can be represented by the symbol series 'ababb'. This series can be encoded by the ISA-rules, to yield the code S[(a), (b)] 2×(b) or the code 2×(ab)b. The former code is the simplest code according to the I_{new}-load, implying that (aba)(bb) is the dominant segmentation, whereas the latter code is the simplest code according to the I_{old}-load, and implies that (abab)(b) is the dominant segmentation. These two segmentations of the symbol series can be mapped straightforwardly onto two segmentations of the patterned sequence of graphic symbols (see figure 11b in which the large space between two of the graphic symbols stands for the border between the two segments). Thus, each stimulus can be constructed from a patterned sequence of graphic symbols (the target) and two relevant segmentations (response alternatives) out of which subjects can choose the preferred segmentation. These preferences can be checked to determine which complexity metric correctly predict them.

The total stimulus set consisted of 240 stimuli, derived from forty different symbol series like the above-mentioned 'ababb' (see also below). Among these forty series, twenty-four contained two different symbols, and sixteen contained three different symbols. The length of the series varied from five up to eighteen symbols. For each of the forty series, three different segmentations were considered. For example, for 'ababb' the two segmentations mentioned above, plus the segmentation (ab)(abb), as induced by the code $\langle(a)\rangle\langle(b)(2\times(b))\rangle$ were considered. Out of these three segmentations, the three different pairs of segmentations were each used in a stimulus. This yields $40\times 3 = 120$ 'symbolic' stimuli, ie stimuli in terms of symbols used in the structural information model and not yet in terms of graphic symbols used in the experiment. Furthermore, from each of the 120 symbolic stimuli, one additional symbolic stimulus was derived by reversing the order of the symbols both in the symbol series and in the two segmentations, eg for the series 'ababb', the reversal of the series is 'bbaba', and the reversal of the segmentation (ab)(abb) is (bba)(ba). Note that the latter segmentation is induced by the 'reversed' A-form $\langle(2\times(b))(b)\rangle/\langle(a)\rangle$, and that any code can be reversed in a similar way. Thus, one gets a total of $120\times 2 = 240$ symbolic stimuli.

During the experiment, each of the 240 symbolic stimuli was assigned randomly to one of the three graphic symbol sets. (Clearly, symbolic stimuli containing three different symbols can only be assigned to set 3 in figure 10.) Then, each of the different symbols in the symbolic stimulus was assigned randomly to one of the graphic symbols in that set. Thus, one obtains the actual stimulus set consisting of 240 stimuli. In order to cancel out any residual bias with respect to the graphic symbols, this random transformation of the symbolic stimuli into real stimuli was performed for each subject individually. The transformation was performed by a computer program that was developed to run the experiment and to present the stimuli on a monitor.

For selecting the forty symbol series from which the 240 stimuli were derived, and for selecting the three segmentations for each of the forty series, three criteria were used:

(i) For each symbol series, the three codes that induce the three segmentations should have the same value for P but different complexities according to each of the other three metrics such that, for each of these three metrics, the simplest code is included. That is, the P-load was used as a baseline because the P-load takes irregularity into account in the same way as do the other metrics but does not account for hierarchy, whereas the other three metrics precisely differ in how they take into account hierarchy. In this way the experimental results can be related to the differences between the metrics.

(ii) The forty symbol series should be balanced with respect to the length of a subseries that is covered by a specific ISA-rule. For instance, the series 'ababb' shows an iteration of the subseries 'ab' and a partly overlapping iteration of subseries 'b'. The preference for one of these iterations may depend on how prominent such an iteration is in a series. Therefore, also included were series such as 'abababab' and 'ababbbb', in which one of the iterations is more redundantly present.

(iii) For each of the forty symbol series, each ISA-rule should be used in at least one of the three codes, so that the results indeed apply to the entire encoding model.

These criteria are hard to meet simultaneously; thus they are met to a large extent for the entire stimulus set but not for every stimulus. This holds in particular for the requirement for different complexities for each of the three codes, as given in the first criterion. That is, in the set of 240 stimuli two subsets can be distinguished for each metric. One subset contains the stimuli in which the two segmentations are induced by equally complex codes so that the metric predicts ambiguity, ie no preference for one of the two segmentations. The other subset contains the stimuli for which the metric predicts nonambiguity, ie a preference for one of the two segmentations since the two codes differ with respect to complexity. In the analysis of the results, these two subsets are considered separately. In figure 12, the sizes of the two subsets are shown for each of the metrics, except for the *P*-load. As mentioned above, the *P*-load is used as the 'baseline', and predicts ambiguity for almost all stimuli. Furthermore, the *P*-load was already known to be inadequate, so that the experimental results are not very interesting with respect to the *P*-load (unless, of course, the results showed that most of the stimuli are indeed ambiguous but, as will be clear, this is not the case). Therefore, the *P*-load is not only omitted in figure 12 but also in the analysis of the results. To conclude this subsection on stimulus selection, some examples of the symbol series, as used in the experiment, are given in figure 13. Each symbol series is encoded in three different ways. Each pair out of the three segmentations for one series represents the response alternatives for the stimulus.

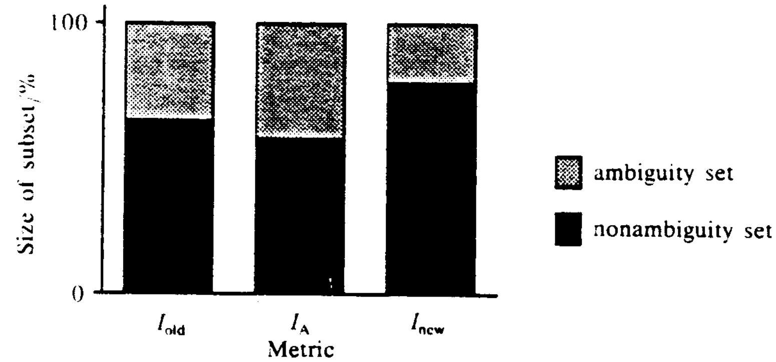


Figure 12. For each complexity metric, the sizes of the two subsets (expressed as a percentage of the total 240 stimuli) that contain the stimuli for which that metric predicts nonambiguity (ie preference for one of the two response alternatives) and ambiguity (ie no preference), respectively. In the analysis of the results, these subsets are dealt with separately.

Series	Code	Segmentation	P	I_{old}	I_{A}	I_{new}
ababb	2×(ab)b	(abab)(b)	3	4	4	4
	S[(a),(b)]2×(b)	(aba)(bb)	3	5	5	3
	⟨(a)⟩/⟨(b)(2×(b))⟩	(ab)(abb)	3	4	5	4
aabaabb	2×(2×(a)b)b	(aabaab)(b)	3	5	5	4
	S[2×((a)),(b)]2×(b)	(aabaa)(bb)	3	6	6	3
	⟨(2×(a))⟩/⟨(b)(2×(b))⟩	(aab)(aabb)	3	5	6	5
abcabcc	2×(abc)c	(abcabc)(c)	4	5	5	5
	S[(ab),(c)]2×(c)	(abcab)(cc)	4	6	6	5
	⟨(ab)⟩/⟨(c)(2×(c))⟩	(abc)(abcc)	4	5	6	6

Figure 13. Three examples of symbol series used in the experiment. Each pair out of the three segmentations for one series represents the response alternatives. The values for the four metrics are given.

5.3 *Procedure*

All subjects were tested individually. As the total duration of the experiment was about 1.5 h, there was a break halfway through the experiment. All subjects were given the same instructions, projected onto a monitor. Each stimulus was presented as follows. First, only the target (a patterned sequence of graphic symbols) was presented in the middle of the left-hand side of the screen. After 7 s, the response alternatives (two segmentations of the target) were presented in addition, at the right-hand side of the screen, one in the upper half and one in the lower half of the screen (as in figure 11). The task was "to decide how you would partition the pattern into coherent parts" (during the 7 s), and then "to select, from the two partitionings, the partitioning that resembles your own partitioning most closely". Pilot investigations showed that subjects had a clear preference within a period of 7 s. Subjects responded by pressing one of two buttons on the table in front of them, the buttons corresponding in position to the positions of the response alternatives on the screen (top versus bottom).

First, 10 trials were presented so as to get acquainted with the task, then the actual experiment began by the 240 stimuli being presented in a random order. As mentioned before, the random transformation of the 240 symbolic stimuli into real stimuli was performed for each subject individually. The position of each response alternative (top or bottom) was randomized too. In the response alternatives, the larger spaces between the segments were made such that the visual angle for both response alternatives was always equal (as in figure 11), independent of the number of segments. The computer registered not only each response, but also each response time, this being the time between the onset of the response alternatives and pressing the button.

5.4 *Results*

Each of the thirty-one subjects performed all 240 trials, so that the total number of trials was $31 \times 240 = 7440$ trials. As mentioned before, the P-load is not considered and, for each of the other three metrics, the nonambiguity set and the ambiguity set are considered separately (see figure 12). Since the subjects performed a forced-choice task, the metrics cannot be tested for correct predictions for stimuli in the respective ambiguity sets. Only if, experimentally, stimuli in an ambiguity set appear to be significantly nonambiguous, can the respective metric be said to have predicted falsely. The latter cases will be dealt with together with the false predictions for stimuli in the nonambiguity sets. First, the correct and false predictions for stimuli in only the nonambiguity sets are considered.

Figure 14 shows, for each of the three nonambiguity sets, a histogram of the raw data. In each histogram, the bars represent disjunct stimulus subsets, together constituting the entire nonambiguity set. The height of a bar represents the size of the subset, and the position on the horizontal axis represents the number of subjects for which the respective metric correctly predicted the responses to the stimuli in that subset. For instance, the I_{new}-histogram applies to a nonambiguity set containing 186 of the 240 stimuli. From that histogram, one can read, for example, that, for one subset of 15 of the 186 stimuli, the I_{new}-load correctly predicted the responses of twenty of the thirty-one subjects and that, for another (disjunct) subset of 15 of the 186 stimuli, the I_{new}-load gave the correct predictions for twenty-four of the thirty-one subjects. So, from figure 14, one readily sees that the I_{A}-load roughly performs better than the I_{old}-load since, in the I_{A}-histogram, larger subsets tend to be related to a larger number of subjects, ie more of the responses were predicted correctly. In the same way, one readily sees that the I_{new}-load roughly performs better than the I_{A}-load.

So, at first glance, figure 14 confirms the hypothesis. For a second glance, more detailed statistics are presented next.

First, for each metric, the predictions were compared to choice at chance level ($p = 0.5$). The results of the t-test performed with the data (pooled across subjects) in table 1, show that the I_{old}-load scores are highly significantly false(!) ($t_{30} = 7.785$, $p < 0.001$), that the I_A-load scores are just significantly correct ($t_{30} = 2.088$, $p < 0.05$), and that the I_{new}-load scores are highly significantly correct ($t_{30} = 7.736$, $p < 0.001$). Furthermore, within subjects, and again compared to choice at chance level, the I_{old}-load scored significantly ($p < 0.05$) more correct predictions for only one subject, the I_A-load scored significantly more correct predictions for fifteen subjects, and the I_{new}-load scored significantly more for twenty-four subjects.

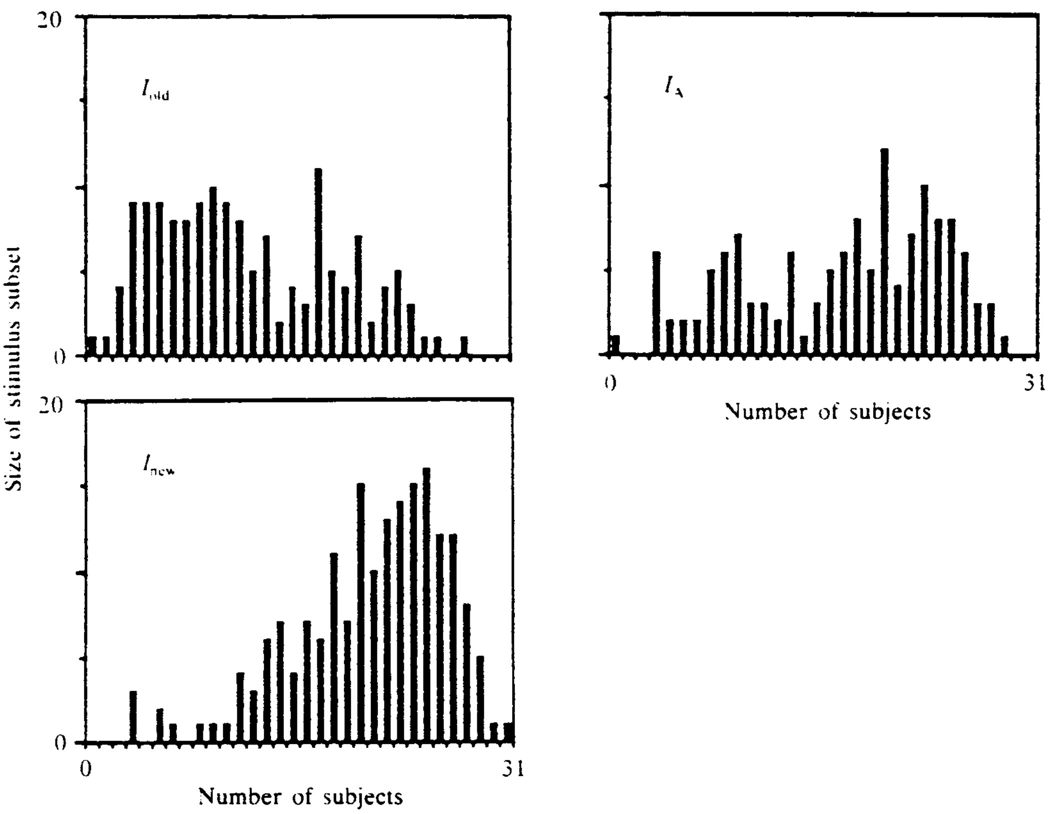


Figure 14. Histograms of the raw data for each metric on its nonambiguity set. In each histogram the bars represent disjunct stimulus subsets, together constituting the entire non-ambiguity set. For each subset the histogram shows the size of the stimulus subset and the number of subjects for which the respective metric correctly predicted the responses to the stimuli in that subset. In each histogram, the eleven leftmost horizontal positions show the subsets for which the respective metric prediction was significantly false, and the eleven rightmost horizontal positions show the subsets for which the respective metric prediction was significantly correct.

Table 1. The means (±SD) and percentages of responses all predicted correctly by each metric for stimuli in the respective nonambiguity set.

Metric	Size of set	Responses predicted correctly	
		mean ± SD	%
I_{old}	150	56.0 ± 13.6	37.3
I_A	136	74.6 ± 17.6	54.9
I_{new}	186	124.4 ± 22.7	66.9

Second, the metrics were compared by considering the respective proportions of significantly ($\alpha = 0.05$) false and correct predictions (pooled across subjects); these results are presented in figure 15. All differences between the proportions are highly significant, both for the significantly false predictions (I_A compared with I_{old}: $z = 4.564$, $p < 0.001$; I_{new} compared with I_A: $z = 5.253$, $p < 0.001$) as well as for the significantly correct predictions (I_A compared with I_{old}: $z = 5.071$, $p < 0.001$; I_{new} compared with I_A: $z = 3.682$, $p < 0.005$). Note that this analysis may seem to be tricky because of dependencies between the three nonambiguity sets. However, for the given stimulus set, it is clear a priori that a more complex analysis to account for those dependencies would yield an even stronger effect. The same argument applies to the next test.

Third, in figure 16, the proportions of significantly false predictions for stimuli in the respective nonambiguity sets are plotted again, but now together with the proportions of significantly false predictions for stimuli in the respective ambiguity sets. The differences between the latter proportions (ie for just the ambiguity sets) are not significant, but the differences between the summed proportions (ambiguity set plus nonambiguity set) are highly significant (I_A compared with I_{old}: $z = 3.776$, $p < 0.005$; I_{new} compared with I_A: $z = 5.512$, $p < 0.001$).

Fourth and finally, each metric was tested for a possible differentiation in response time between responses predicted correctly and responses predicted falsely (which can

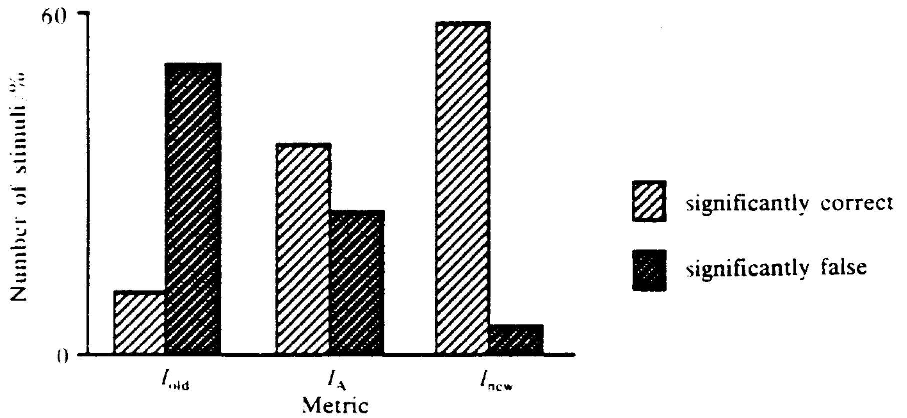


Figure 15. The proportions of significantly false and significantly correct predictions for the stimuli in the respective nonambiguity set for each metric. All differences between proportions are significant.

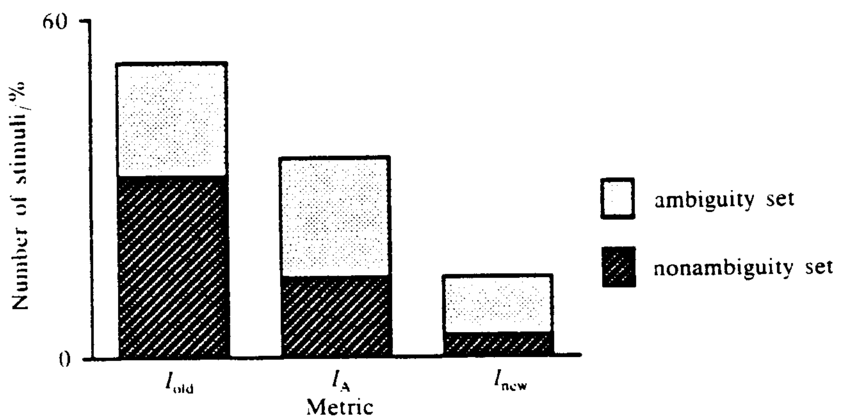


Figure 16. The proportions of significantly false predictions for the entire stimulus set, ie for both the ambiguity set and the nonambiguity set, for each metric. For just the ambiguity sets, the differences between the proportions are not significant, but for the entire stimulus set the differences are significant.

be done only for the nonambiguity sets). Figure 17 shows, for each metric, the average response times over all responses predicted correctly and over all responses predicted falsely; for the I_{old}-load, 'false' response times tend to be shorter than 'correct' response times, whereas the other metrics yield an opposite tendency. A MANOVA was performed to check whether these tendencies might be influenced by the number of different elements in a target (two or three symbols) or by the total number of elements in a target (ie target 'length' which varied from five to eighteen symbols). The factor Length was set to four levels, and the factor Symbols was set to two levels. Then, for the I_{new}-load only, the factor Prediction (two levels: false or correct) yields a significant differentiation in response times (MANOVA over $4 \times 2 \times 2 = 16$ cells; five subjects were rejected because of empty cells): $F_{1,25} = 5.88$, $p < 0.05$). The factor, Prediction, does not interact with the factor Length nor with the interaction term Length × Symbols.

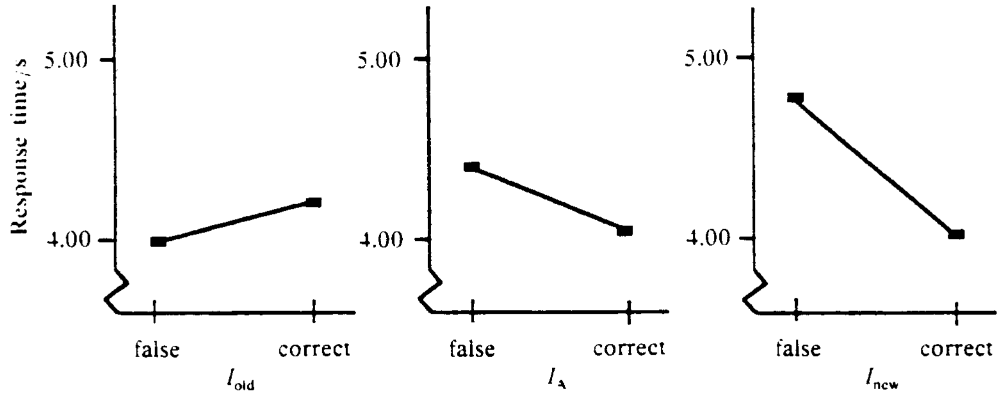


Figure 17. The average response times over all responses predicted falsely and over all responses predicted correctly (only for the respective nonambiguity sets) for each metric.

5.5 Discussion

The experimental results confirm the hypothesis. That is, the results not only confirm that the I_{new}-load is significantly better than both the I_A-load and the I_{old}-load, but also that the I_A-load is already significantly better than the I_{old}-load. The P-load scored 58.2% significantly false predictions for the stimuli in its ambiguity set which, as indicated before, almost equals the entire stimulus set. This implies that the P-load scored worst of all (see also figure 16), and supports the earlier mentioned reasons for excluding the P-load from the analysis. For the three metrics considered in the analysis, the ordering in the goodness of the metrics is significant with respect to both the number of significantly correct predictions for the respectively nonambiguity sets, as well as the number of significantly false predictions for the entire stimulus set (see figures 15 and 16).

The superiority of the I_{new}-load is supported further by its differentiation in response times between falsely and correctly predicted responses for stimuli in its nonambiguity set (see figure 17). In general, we assume that such a differentiation takes place for a good predictor. That is, if subjects have more doubt about their preference, then an increase in response time results. So, inversely, if such an increase occurs consistently in cases of false predictions, then the predictions may have been false but are apparently still good enough to 'compete' with the actual responses. In the present experiment, the MANOVA on the response times showed a significant increase in response time on falsely predicted responses for the I_{new}-load only. So, adopting the assumption above, the response time results support the I_{new}-load only.

The goodness of the I_{new}-load may seem to be undermined somewhat by the fact that this metric scored 'only' 57.5% significantly correct predictions for stimuli in its nonambiguity set (see figure 15). However, as mentioned before, the selected stimuli had to be rather critical with respect to the metrics. For an arbitrary stimulus set, the predictions of the metrics are not very different, and the metrics score much better, as shown in earlier experimental work on the structural information model. Moreover, the main issue in this experiment was to compare the I_{new}-load with the I_{old}-load which scored the five times smaller percentage of 11.3% significantly correct predictions (see figure 15).

One further remark in connection with the stimulus selection is that, as mentioned before, one of the criteria for selecting a target was that all the ISA-rules should be used to generate the three possible response alternatives, so that the results indeed apply to the entire encoding model. Now, for all stimuli (target plus two response alternatives) in which one of the response alternatives is based mainly on the I-rule, the number of responses predicted correctly by the I_{new}-load is about two times as high as the number of responses predicted falsely. Exactly the same holds for the S-rule, and also for the A-rule. So, the I_{new}-load 'treats' the ISA-rules in an equivalent way, which gives further confidence in the correctness of this metric.

Finally, the confirmation of the hypothesis implies strong support for the theoretical analysis of regularity and hierarchy, as elaborated in van der Helm and Leeuwenberg (1991). As argued in the previous sections, that analysis leads directly to the ordering of the metrics according to goodness, that is now confirmed experimentally. So, the I_{new}-load is not only a good metric, but it also has a firm theoretical basis which is supported by the experimental results.

6 Summary and conclusion

In this paper, we introduced a new metric to measure the complexity of serial patterns. Such a complexity metric is needed in pattern encoding models, such as Leeuwenberg's (1969, 1971) structural information model. Such models employ several coding rules to describe possible structures of a pattern, and adopt the minimum principle by assuming that the simplest code of a pattern reflects the humanly preferred interpretation of that pattern. In brief, pattern interpretations are given in terms of pattern parts, ie a pattern segmentation (parts plus relations between parts) represents the pattern structure according to an interpretation, and the simplest code of a pattern is a code that describes the largest amount of regularity in that pattern. In such encoding models, the employed complexity metrics are generally just 'good guesses', formulated in terms of the encoding syntax and lacking an intrinsic psychological justification, whereas regularity is discussed merely in an intuitive sense.

The new complexity metric introduced in this paper, however, is based on a strictly formal analysis of regularity and hierarchy in patterns, as elaborated in the paper of van der Helm and Leeuwenberg (1991). This formal analysis resulted in the notions of holographic regularity and of transparent hierarchy. The notion of transparent hierarchy applies to the conditions under which different but (partly) overlapping regularity structures can be combined hierarchically. The notion of holographic regularity represents a formalization of the intuitive notion of regularity, and results in a restricted number of basic kinds of regularity. Assuming that holographic regularity and transparent hierarchy are psychologically relevant notions, only three coding rules are needed to describe pattern structures, namely the iteration rule, the symmetry rule, and the alternation rule, as used in Leeuwenberg's structural information model. The kinds of regularity described by these three coding rules were already widely accepted (intuitively) as being psychologically relevant but now appear to have a unique formal status too. That is, in a strictly formal way, precisely these kinds of regularity result from the notions of holographic regularity and transparent hierarchy. This implies that these two notions can be seen as the underlying psychological basis of the description of pattern structures.

The plausibility of the formal analysis is supported further by two facts. First, it allows for a largely parallel encoding process. This supports the analysis since the fascinating speed with which the human perceptual system processes patterns is generally thought to result from parallel processing. Second, it allows for the elimination of combinatorial explosions. That is, it enables the selection of a simplest code out of the exponential number of possible codes, by taking into account all codes but without generating every code separately. This also implies support of the analyses since it shows that the minimum principle is realistic in the sense that it does not require an unrealistic search for simplest codes.

All in all, the formal analysis constitutes a firm basis for further research. In the present paper, we elaborated the fact that the formal analysis enables a detailed investigation into the way in which pattern complexity is quantified by means of the complexity metrics used in earlier research. This investigation resulted in proposing a new complexity metric which, according to the formal analysis, should be superior to the metrics used before. That is, the new metric quantifies complexity by taking into account the irregularity in a code in the same way as in the other metrics, but it accounts for the hierarchy in a code in a better way. In particular, this improved account of hierarchy is relevant with respect to so-called local-effect cases, ie cases in which (with the other metrics) an a priori pattern partitioning has to be assumed in order for the simplest code to reflect the preferred interpretation. This requirement contradicts the minimum principle. With the new metric, simplest codes tend to represent less hierarchically organized pattern structures and, therefore, tend to look like codes obtained by encoding pattern parts separately. This suggests that many local effects may 'disappear' with the new metric, since it is not necessary to assume the proper pattern segmentation a priori because it follows directly from the simplest code.

The experiment discussed in this paper shows that the new complexity metric is indeed significantly better than the metrics used before. Moreover, and equally important, the experiment significantly supports the ordering of the metrics with respect to goodness, as was hypothesized on the basis of the formal analysis. This implies that we may conclude that the new metric is not merely a 'better guess', but a plausible choice based on a formal analysis which is supported by experimental results.

References

Buffart H F J M, 1987 "Seven minus two and structural operations" in *Progress in Mathematical Psychology* volume 1, Eds E E Roskam, R Suck (Amsterdam: North-Holland) pp 117 - 119

Collard R F A, Buffart H F J M, 1983 "Minimization of structural information: A set-theoretical approach" *Pattern Recognition* **16** 231 - 242

Deutsch D, Feroe J, 1981 "The internal representation of pitch sequences in tonal music" *Psychological Review* **88** 503 - 522

Garner W, 1970 "Good patterns have few alternatives" *American Scientist* **58** 34 - 42

Geissler H-G, Klix F, Scheidereiter U, 1978 "Visual recognition of serial structure: Evidence of a two-stage scanning model" in *Formal Theories of Visual Perception* Eds E L J Leeuwenberg, H F J M Buffart (New York: John Wiley) pp 299 - 314

Gregory R L, 1972 "Cognitive contours" *Nature (London)* **238** 51 - 52

Hatfield G C, Epstein W, 1985 "The status of the minimum principle in the theoretical analysis of visual perception" *Psychological Bulletin* **97** 155 - 186

Helm P A van der, 1988 *Accessibility and Simplicity of Visual Structures* PhD thesis, University of Nijmegen, Nijmegen, The Netherlands

Helm P A van der, Leeuwenberg E L J, 1986 "Avoiding explosive search in automatic selection of simplest pattern codes" *Pattern Recognition* **19** 181 - 191

Helm P A van der, Leeuwenberg E L J, 1991 "Accessibility, a criterion for regularity and hierarchy in visual pattern codes" *Journal of Mathematical Psychology* **35** 151-213
Helmholtz H von, 1909/1962 *Handbook of Physiological Optics* (New York: Dover, 1962); English translation by J P C Southall for the Optical Society of America (1924) from the 3rd German edition of *Handbuch der physiologischen Optik* (Hamburg: Voss, 1909)
Hochberg J E, McAlister E, 1953 "A quantitative approach to figural 'goodness'" *Journal of Experimental Psychology* **46** 361-364
Jones M, Zamostny K, 1975 "Memory and rule structure in the prediction of serial patterns" *Journal of Experimental Psychology: Human Learning and Memory* **104** 295-306
Koffka K, 1935 *Principles of Gestalt Psychology* (fifth edition 1962, London: Routledge and Kegan Paul)
Kohler W, 1920 *Die physischen Gestalten in Ruhe und im stationären Zustand* (Braunschweig: Vieweg)
Leeuwenberg E L J, 1969 "Quantitative specification of information in sequential patterns" *Psychological Review* **76** 216-220
Leeuwenberg E L J, 1971 "A perceptual coding language for visual and auditory patterns" *American Journal of Psychology* **84** 307-349
Leeuwenberg E L J, Helm P A van der, 1991 "Unity and variety in visual form" *Perception* **20** 595-622
Leyton M, 1986a "A theory of information structure 1. General principles" *Journal of Mathematical Psychology* **30** 103-160
Leyton M, 1986b "A theory of information structure 2. A theory of perceptual organization" *Journal of Mathematical Psychology* **30** 257-305
Neisser U, 1966 *Cognitive Psychology* (New York: Appleton-Century-Crofts)
Palmer S, 1977 "Hierarchical structure in perceptual representation" *Cognitive Psychology* **9** 441-474
Palmer S, 1983 "The psychology of perceptual organization: a transformational approach" in *Human and Machine Vision* Eds J Beck, B Hope, A Rosenfeld (New York: Academic Press) pp 269-337
Restle F, 1970 "Theory of serial pattern learning: Structural trees" *Psychological Review* **77** 481-495
Restle F, 1979 "Coding theory of the perception of motion configurations" *Psychological Review* **86** 1-24
Rock I, 1983 *The Logic of Perception* (Cambridge, MA: MIT Press)
Simon H A, 1972 "Complexity and the representation of patterned sequences of symbols" *Psychological Review* **79** 369-382
Simon H A, Feigenbaum E A, 1964 "An information processing theory of some effects of similarity, familiarization, and meaningfulness in verbal learning" *Journal of Verbal Learning and Verbal Behavior* **3** 385-396
Simon H A, Kotovsky K, 1963 "Human acquisition of concepts for sequential patterns" *Psychological Review* **70** 534-546
Tversky A, Gati I, 1982 "Similarity, separability, and the triangle inequality" *Psychological Review* **5** 125-128
Vitz P C, 1966 "Preference for different amounts of visual complexity" *Behavioral Science* **11** 105-114
Vitz P C, Todd R C, 1969 "A coded element model of the perceptual processing of sequential stimuli" *Psychological Review* **76** 433-449
Wertheimer M, 1912 "Experimentelle Studien über das Sehen von Bewegung" *Zeitschrift für Psychologie* **12** 161-265
Wertheimer M, 1923 "Untersuchungen zur Lehre von der Gestalt" *Psychologische Forschung* **4** 301-350