

The Contravariancy of Anticipatory Systems

Dimitris Sisiaridis, Michael Heather, Nick Rossiter,
Department of Computer Science and Digital Technologies
Northumbria University, NE1 8ST, UK

dimisis@sch.gr, michael.heather@trinity.cantab.net, nick.rossiter1@btinternet.com
<http://computing.unn.ac.uk/staff/CGNR1/>

Abstract Anticipation is a property of any system and resides in its semantics as a duality of the system itself. The relationship is an adjointness between levels, requiring contravariancy. The intension/extension levels are impredicative in nature but this recursive characteristic can be represented formally in category theory. This paper focuses on the vital role of contravariancy in adjointness, permitting a structured re-ordering of the categories involved. A worked example of a three-level architecture for an information system is provided, illustrating the alternation of intension/extension pairs, the adjointness of two-way functors between each level, the (bi)functors for linking intension to extension and the locally cartesian closed structure of the underlying categories. The dynamic anticipatory aspect of contravariant mapping, relative to static covariant mapping, is highlighted, reinforcing the view that contravariancy underpins anticipation.

Keywords: contravariance, duality, composition, adjointness, intension.

1 Nature of Anticipation

Anticipation is usually described in predicative terms, that is a semantic description forming part of the properties of the system. The focus of attention is often predictive but prediction is only one aspect of predication [12]. Anticipation is classified as of two main types, weak and strong [6]. The weak form occurs where the anticipation is achieved through a model of the system. The strong form occurs where the anticipation resides in the anticipatory system itself. For the strong form anticipation is therefore of the nature of the system and forms part of the Universe, residing in natural processes and relationships [11]. This implies impredication with recursion from higher levels. Likewise from no more than its definition, the Universe consists of entities related one to the other. Thus each entity affects every other. To represent such multi-body interdependence, first-order formalisms are inadequate and a higher order formalism such as category theory is required and at a higher level to investigate the property of existence. A cartesian closure of category theory provides for the relationship between any pair of entities including every possible path between them. This cartesian closure is found in the highest structure possible in category theory: the identity natural transformation designated as the *topos* of Figure 1. This highest level arrow composes structures of categories and functors

and constitutes the full cartesian closed category of a topos. However, the natural arrow is double-headed as a composition of the adjoint functors but with a naturally built-in parity. Although as just explained it may be easier to understand the structure in Figure 1 bottom-up in the way that models are usually built-up, nevertheless process can only exist as a whole from top-down and the full diagram represents a natural occasion or “actual event” as first introduced in the metaphysics of Whitehead [33].

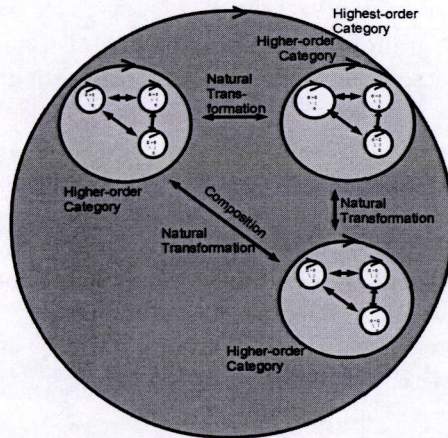


Fig. 1: Natural Transformations of Composing Functors themselves compose in the highest possible category, a Topos

Metaphysics provides access to everything below. Furthermore there may not be a bottom to begin with for the topos does not come naturally equipped with an initial object but one may be selected¹. This structure implies closure only at the top. If the Universe is a topos with a cartesian closed structure and anticipatory systems are featured as part of the Universe then an anticipatory system is locally cartesian closed. This enables us to explore the characteristics of anticipatory systems and we find that because of this structure any system has anticipation. Local cartesian closure is demanded from contravariancy. Anticipatory systems are therefore contravariant. It is this contravariancy that is explored in this paper.

2 Duality and Variance

If every entity is related to every other it follows that the relationship is both ways but not just a simple inverse relationship. This is apparent from the laws of physics

¹This selection is an exercise of the axiom of choice in set theory which as shown by Diaconescu is equivalent to a Boolean world [5] but this introduces a closed-world assumption that reduces an internal Heyting logic to a Boolean one.

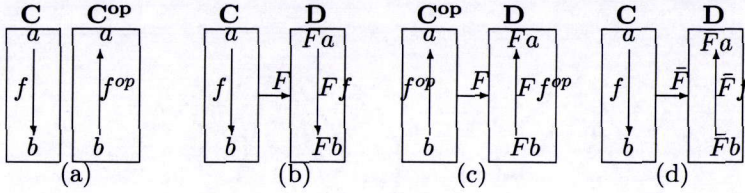


Fig. 2: a) Category C and its opposite C^{op} ; b) Covariant functor $F : C \rightarrow D$; c) Covariant functor $F : C^{op} \rightarrow D$; d) Contravariant functor $\bar{F} : C \rightarrow D$

but Newton's third law with its "equal and opposite" represents only the first order view of classical physics. Indeed to examine the inverse relationship in more detail we need to consider a number of concepts, including duality and variance together with the latter's two alternatives of covariance and contravariance. In category theory duality is concerned with the reversal of the direction of arrows within a category. A category C of objects and arrows between the objects will have a dual C^{op} with arrows reversed. Figure 2(a) shows an example of an opposite (dual) category C^{op} for C where the arrow $f : a \rightarrow b$ has been reversed to $f^{op} : b \rightarrow a$. The whole structure of the Universe of these both-ways relationships is then the limit obtained by considering the total going one way with the total going the opposite way, represented by the product $C^{op} \times C$, which gives rise to the principle of duality throughout the Universe. The fundamental nature of duality is emphasised in Figure 3. Figure 3(a) shows a category with a terminal object, which has a unique arrow on to it from every object in the category. The opposite of this category, in Figure 3(b), shows a typing arrow from the terminal object to every object in the category. This is the simplest example of typing as contravariant. For the terminal object is the identity functor defining the category.

It is to be noted that the arrow in category theory can be usefully interpreted as typing. The objects of Figure 3 could just as well themselves be categories in some higher category or even more powerfully as functors in a yet higher category. The typing arrow may be a natural transformation crossing more than one level. The direction of typing in a set theory approach to typing is usually the other way round with the arrow drawn in the opposite direction. This arises because set theory does not have an inherent concept of covariance-contravariance.

Duality is a common enough concept in mathematics, philosophy and most of the sciences with some renowned examples like the mind-body dualities [4] and, in anticipatory systems, the duality of incursion [1]. It also appears in other versions of contrast as between the dynamic and the static and between global and local. To capture the full effect and subtleties of opposing views and relationships a single view of duality is needed from variance as process [32, 23]. In our work particular interest is taken in the dual relationship between Intension **INT**, the specification of the type of an entity, and Extension **EXT**, the instances that conform to the type specification. Figure 4 shows the adjointness between the categories **INT** and

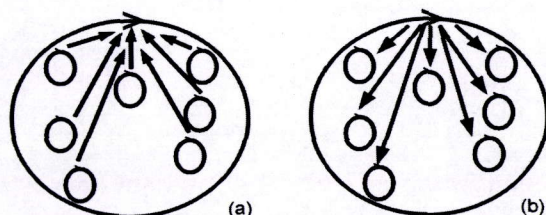


Fig. 3: (a) A Category with a Terminal Object; (b) The Opposite Category with Typing Arrows

EXT. Each arrow has a dual role. F is the contingent arrow of intension and the determinant arrow of extension while G is the contingent arrow of extension and the determinant arrow of intension. T the composition GF is the global ontological and S the composition FG is the local ontology. Each of these compositions may be compared in Figure 5, at the next level up with the contribution they make to their respective identity functors by means of the creative unit of adjunction $\eta : \mathbf{1}_F \rightarrow GF$; and the qualitative co-unit of adjunction $\epsilon : FG \rightarrow \mathbf{1}_G$ (from the interpretation in earlier work [13]).

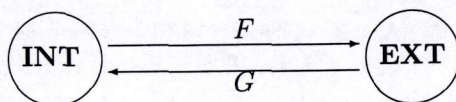


Fig. 4: Adjunction $F \dashv G$ for Functors between Categories **INT** for intension and **EXT** for extension

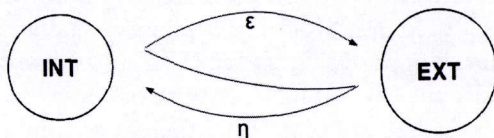


Fig. 5: Adjunction expressed with Natural Transformations η and ϵ

Duality is not a closed Boolean view. Rather it encapsulates opposite orderings within a single (functorial) concept of variancy. These may be conveniently labelled covariant and contravariant but only relative one to the other and not as absolute descriptions. With a covariant functor the arrows are mapped from the source category onto the target category without any reversal of the direction of the arrows.

As shown in Figure 2(b), the covariant functor F takes category \mathbf{C} to \mathbf{D} by mapping the arrow $f : a \rightarrow b$ in \mathbf{C} to $Ff : Fa \rightarrow Fb$ in \mathbf{D} . A further example in Figure 2(c) shows the covariant functor F taking an opposite category \mathbf{C}^{op} to a category \mathbf{D} by mapping the arrow $f^{\text{op}} : b \rightarrow a$ in \mathbf{C}^{op} to $Ff^{\text{op}} : Fb \rightarrow Fa$ in \mathbf{D} . Although a dual category is involved, the functor is covariant as the order of the arrows is not reversed by the functor. With a contravariant functor the arrows in the source category are reversed in direction before being mapped onto the target category. Figure 2(d) shows a contravariant² functor \bar{F} taking category \mathbf{C} to category \mathbf{D} by mapping $f : a \rightarrow b$ in \mathbf{C} on to $\bar{F}f : \bar{F}b \rightarrow \bar{F}a$ in \mathbf{D} . Without a reversal there may be a technical problem, particularly in typing, in performing the mapping of source a on to target $\bar{F}b$ and source b on to target $\bar{F}a$.

In terms of levels, positive and negative at the lowest level of basic set operations correspond to covariant and contravariant respectively at the functorial level and comonad and monad respectively at the top level. At the lowest level paradoxes occur, such as in set theory; these can be avoided by moving to higher levels, including the intension-extension relationship [29]. Also at the lowest level, commutativity, which is assured at higher levels, may not hold. In mathematics, anticommutativity is the property of an operation, such as subtraction, in which swapping the position of any two arguments negates the result. In physics anticommutative operations are commonly used and termed antisymmetric. Cross-product is often described as anticommutative but this holds only in the extension. So for sets of values: $A \times B \not\cong B \times A$. If the labels, that is the intension, of the components of the product are included in the operation as well, then it is commutative with $A \times B \cong B \times A$ as the meaning of A and B on each side is retained.

Systems theory is a case in point where these different views need to be distinguishably integrated [22]. Thus for anticipatory systems, anticipation is an instantaneous, local static instantiation of a dynamic global feature that looks either forward or back³. The natural categories of process as advanced by Whitehead encompass this contravariancy found in reality for which he uses the term 'dipolar'. Thus the process of becoming is dipolar. He refers to the two poles as formal and mental or conceptual ([33] at p.45, [32] at p.74). Whitehead also uses alternative terms 'bipolar, physical and mental' ([33] at p.108, [32] at p.165). Whitehead's term *prehension* (category XII of Explanation), meaning capture, is equivalent to adjointness with the counit ϵ as positive prehension and the unit η as negative prehension

²We use a bar over a functor name to indicate that it is contravariant, following early standard expositions such as Mac Lane [17]

³That is forward and back in an ordering that may be some kind of time. Which is covariant, which is contravariant may be arbitrary but the underlying ordering is paritous, that is the abstract property of parity and does not therefore imply is necessarily reversible as asserted from time to time in classical mathematics [10]. This may be the rationale behind the second law of thermodynamics that may only be relied on to first order as the laws of thermodynamics are statistical in nature and not categorically exact. For a study of process in the thermodynamic context see [3, 18].

([33] at pp.23-24, [32] at p.35). Negative prehension corresponds to contravariance in adjointness. Whitehead's term *concrecence*, corresponding to the operation of a free functor 'snapping', is also relevant to adjointness as it indicates the simultaneity with which the unique adjoint solution is sought. Mac Lane's 'insertion of generators' ([17] p.87), used later in Figure 14, describes the action of the free functor and may be identifiable with Whitehead's *concrecence*.

Contravariance has long been needed in physics, for example in stress-strain relationships. It was brought to the fore in category theory by Lawvere in 1969 as he contended it to be the basic property of proof theory⁴. Lawvere tentatively extended his ideas by examining the duality between *Conceptual* and *Formal* as shown in the Appendix (section 1). However, Lawvere does not seem to be aware of Whitehead's earlier work cited above on formal/physical and mental/conceptual aspects in 1929. Lawvere in developing the idea of a hyperdoctrine, later in his paper, indicates that the mapping between the categories **Formal** and **Conceptual** is by adjoint contravariant functors with the arrows in **Formal** reversed as in the opposite category **Formal**^{op} before being mapped on to the **Conceptual** category as shown in the Appendix (section 2).

Ultimate contravariance is a three-level structure of arrows that is sufficient to provide complete closure with internal contravariant logic providing a generalisation of negation. Further levels are redundant as at the top level an arrow from one natural transformation to another gives a composition of the natural transformations, not a new level [21]. Contravariance across levels provides more sophisticated reversals such as to be found in the intricacies of reverse engineering. The structure of a cartesian closed category is entirely given by adjointness. Indeed Lawvere's structure of a hyperdoctrine is an adjointness between cartesian closed categories, including quantification and logic. Such ultimate contravariance is to be found in the universal adjointness between any pair of functors contravariant one to the other to provide both the quantitative and qualitative semantics of intension-extension logic as shown in Figure 6 with the adjunctions $\exists \dashv \Delta$ and $\Delta \dashv \forall$, that is Δ the diagonal functor is right adjoint to \exists and left adjoint to \forall .

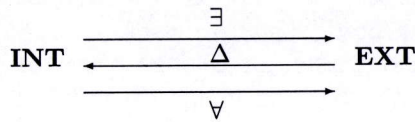


Fig. 6: Adjointness $\exists \dashv \Delta \dashv \forall$ for Functors between Categories **INT** for intension and **EXT** for extension

For application it is necessary to relate Lawvere's terminology to Whitehead's metaphysics and to look in more detail at the arrows in the underlying categories.

⁴His 1969 paper [14] was republished in 2006 [15] with a commentary.

As the *Formal* is the definition of the *Conceptual*, his terms *Formal* and *Conceptual* appear to correspond to *Intension* and *Extension* respectively in metaphysics as used in our work [24]⁵. Mac Lane [17] provides the classical set theoretic treatment for contravariance as shown in the extracts also in the Appendix (section 3). The inversion of the order of composition is a key property for metaphysics, which needs to be considered in more detail with an example. Mac Lane's contravariant functor (see Appendix, section 3) labelled \bar{S} inverts the arrow in the target category and maps the result onto the source category. For general purposes Mac Lane indicates ([17] p.34) that it is much more convenient to represent a contravariant functor \bar{S} on \mathbf{C} to \mathbf{B} as a covariant functor $S : \mathbf{C}^{\text{op}} \rightarrow \mathbf{B}$. This notation of Mac Lane is used throughout this paper for the convenience of comparison with standard texts⁶.

In ontologies for information systems the terms *Formal* and *Descriptive* have been used for intension and extension respectively as by Poli [19] who stated: "descriptive ontology concerns the collection of ... *prima facie* information either in some specific domain of analysis or in general. Formal ontology distils, filters, codifies and organizes the results of descriptive ontology (in either its local or global setting)".

3 Composition in the Intension-Extension Relationship

In any universe of discourse the intension-extension relationship governs entities and their instances or values in the context of name and type. In an extension category **EXT** the relationship between *value* and *name* is of the form⁷:

$$e : \text{value} \rightarrow \text{name}$$

The dual $e^{-1} : \text{name} \rightarrow \text{value}$ does not exist as an arrow with a unique solution (a function in set theory) as there are potentially many values associated with a name. e is therefore not monic, that is not left-cancellable. e also may not be epic, that is right-cancellable, as every name may not be assigned a value. e cannot be isomorphic as it is neither monic nor epic.

In an intension category **INT** the relationship between *name* and *type* is of the form:

$$i : \text{name} \rightarrow \text{type}$$

as it is N:1, with each name having an associated single type. The dual $i^{-1} : \text{type} \rightarrow \text{name}$ does not exist as an arrow with a unique solution as there are potentially many names associated with a type. i is therefore not monic and also may not be epic as every type may not be assigned a name. i cannot therefore be isomorphic.

⁵In earlier years we used syntax, semantics and pragmatics as our three levels. Syntax, giving the rules and principles may be considered as intension; semantics, expressing the meaning within the rules, can be regarded as extension; and pragmatics [28], representing the use of language in a context, often social using semiotics, as a further extension [20].

⁶Pure category theory gives a simpler representation by the operation of *Occam's razor*.

⁷For example $\text{student_id} \rightarrow \text{student_name}$ is N:1 (many to 1) with each student_id being associated with a single student_name but each student_name may be associated with many student_id .

We have therefore two categories **INT** and **EXT** holding the respective arrows

$$i : \text{name} \longrightarrow \text{type}, \quad e : \text{value} \longrightarrow \text{name}$$

It is worth first considering the composition of these arrows over the common object *name*. If the arrows were within a single category it would then be possible to compose *i* with *e* conventionally, giving

$$i \circ e : \text{value} \longrightarrow \text{name} \longrightarrow \text{type}$$

However, if the arrows are in different categories this composition is not available and we rely on a functor to perform the composition. This composition relies on mapping one arrow in the source category onto another compatible arrow in the target category. A covariant functorial mapping $H : \mathbf{INT} \longrightarrow \mathbf{EXT}$ would map $H(\text{name})$ onto *value* in the extension and $H(\text{type})$ onto *name* in the extension. This mapping fails to match *name* in **EXT** with that in **INT** as the common attribute *name* is codomain in the extension and domain in the intension.

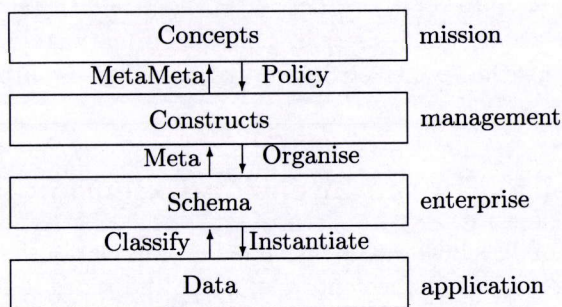


Fig. 7: An Informal Three-level Architecture for an Information System

A contravariant mapping $\bar{H} : \mathbf{INT} \longrightarrow \mathbf{EXT}$ overcomes this problem. \bar{H} reverses the direction of arrows in the intension, taking $\bar{H}(\text{type})$ onto *value* in the extension and $\bar{H}(\text{name})$ onto *name* in the extension. So *value* is related to *type* in the context of a common *name*. It is noteworthy that contravariant functorial mapping facilitates composition through functors that would be natural were all the arrows to be held within a single category. That is the composition across categories in a contravariant manner corresponds to conventional composition within a category. Both forms of composition are horizontal, that is pointwise composition within a category and from category to category. An alternative form of composition, vertical, of natural transformations is performed by juxtaposition using the rules of the functorial calculus of Godement [9]. Both types of composition produce equivalent results and both are involved in the three-level architecture described later.

4 Worked Example of Three-level Architecture

As an example of a three-level architecture with alternating intension-extension pairs, we first consider the informal diagram in Figure 7 showing the concepts,

constructs, schema and data, making up a complete information system such as the universe.

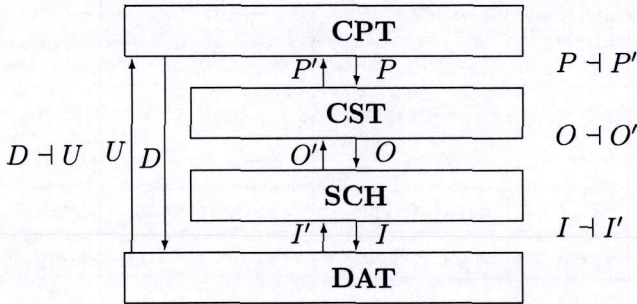


Fig. 8: A Categorical Three-level Architecture for an Information System

Level	Left Adjoint Functor	Right Adjoint Functor	Source Category	Target Category
$P \dashv P'$	P Policy	P' MetaMeta	CPT Concepts	CST Constructs
$O \dashv O'$	O Organise	O' Meta	CST Constructs	SCH Schema
$I \dashv I'$	I Instantiate	I' Classify	SCH Schema	DAT Data

Fig. 9: Adjoint Functors: The Three Levels

The diagram comprises four data structures, in top-down order of Concepts, Constructs, Schema and Data, from the broadest data abstractions [2], through constructions used in design and available schema types down to the actual data values themselves [7]. Between each data structure there is a two-way mapping, which can be readily identified in terms of information system processes. For instance *Instantiate* connects a typed name with a named data value and its dual *Classify* connects a named data value with a typed name; *Organise* connects an available construction with a typed name and its dual *Meta* connects a typed name with an available construction; *Policy* connects a data abstraction with an available construction and its dual *MetaMeta* connects an available construction to a data abstraction. The data structures might be described as mission, management, enterprise and application respectively from the top but this is an over-simplified view as the structures have no meaning on their own: it is the two-way mappings which facilitate the functionality. Indeed this is why the whole is regarded as a three-level architecture as there are three levels of mapping.

Level	Right Adjoint Functor (contravariant)	Target Category (source at bottom)	Target Arrow (source at bottom)	Relational Database Property	Relational Database Aggregation
$P \dashv P'$	P'	CPT	name \rightarrow type ↗ ↘	attr \rightarrow prop ↗ ↘	tab \rightarrow aggr ↗ ↘
$O \dashv O'$	O'	CST	value \rightarrow name ↗ ↘	reg_no \rightarrow attr ↗ ↘	birth_t \rightarrow tab ↗ ↘
$I \dashv I'$	I'	SCH	name \rightarrow type ↗ ↘	car_reg \rightarrow reg_no ↗ ↘	birth_r \rightarrow birth_t ↗ ↘
		DAT	value \rightarrow name ↗ ↘	'x123y' \rightarrow car_reg ↗ ↘	<'Smith', 25 Mar 1980, 'Torquay'> \rightarrow birth_r

Fig. 10: Examples of Contravariant Mapping in the Upwards Direction in the Three-Level Architecture.

The arrows for each functor indicate the contravariant mapping with the domain and codomain of each source category arrow mapped onto the codomain and domain respectively of each target category arrow.

Abbreviations: attr = attribute, prop = property, aggr = aggregation, tab = table, birth_r = birth record, birth_t = birth type.

The informal diagram in Figure 7 can be readily converted to the formal categorial diagram of Figure 8 by replacing each data structure with a category and each two-way mapping with a pair of functors, one dual to the other. The two-way mappings may be adjoint in which case we write for example $P \dashv P'$ indicating that the free functor P is left adjoint to the underlying functor P' and P' is right adjoint to P . The functors can be composed, both upwards with $U = P'O'I'$ and downwards with $D = IOP$. U relates a data value to a data abstraction; D relates a data abstraction to a data value. Adjoints can be composed naturally so we can write $D \dashv U$ if the individual levels are adjoint. More detail on the three pairs of adjoint functors comprising the architecture is given in Figure 9 and the question of adjointness is further discussed later.

The nature of the two-way mapping is of particular interest. The relationship between the categories **DAT** and **SCH** is the intension-extension mapping developed earlier. So **DAT** contains arrows of the form $e : \text{value} \rightarrow \text{name}$ and **SCH** of the form $i : \text{name} \rightarrow \text{type}$. The mapping in both of the functors I and I' is therefore contravariant with the arrows in the source category reversed before mapping onto the target category. We can also see that the relationship between the two categories higher in the architecture **CST** and **CPT** is an intension-extension mapping with **CST** containing arrows of the form $e : \text{value} \rightarrow \text{name}$, and **CPT** of the form $i : \text{name} \rightarrow \text{type}$. In **CST** we relate the value for a construction to the name of an abstraction through the arrow $e : \text{value} \rightarrow \text{name}$ and in **CPT** we relate the name

of the abstraction to a type of construction through the arrow $i : \text{name} \rightarrow \text{type}$. As with the mapping in the lower part of the architecture the mapping between **CST** and **CPT** is through contravariant functors P and P' .

The table in Figure 10 gives two examples of the three-level architecture, showing the mapping for a relational database of property and of aggregation. The mapping is in the upwards direction from **DAT** to **CPT** through the three contravariant functors I' , O' and P' , in turn. The arrows shown for the functors indicate the contravariant nature of the mapping with domain onto codomain and codomain onto domain. All the structures shown are target except for **DAT**, which is the source at the lowest level. For a worked example, the arrow with the record (3-tuple) as source of $\langle \text{'Smith'}, 25 \text{ Mar } 1980, \text{'Torquay'} \rangle \rightarrow \text{birth_rec}$ is mapped contravariantly by the functor I' onto $\text{birth_rec} \rightarrow \text{birth_type}$ to compose the 3-tuple with birth_type ; the arrow $\text{birth_rec} \rightarrow \text{birth_type}$ is then mapped contravariantly by the functor O' onto $\text{birth_type} \rightarrow \text{table}$ to compose birth_rec with table ; finally the arrow $\text{birth_type} \rightarrow \text{table}$ is mapped contravariantly by the functor P' onto $\text{table} \rightarrow \text{aggregation}$ to compose birth.type with aggregation. Overall across the three levels we can see that the 3-tuple is related to the data abstraction of aggregation.

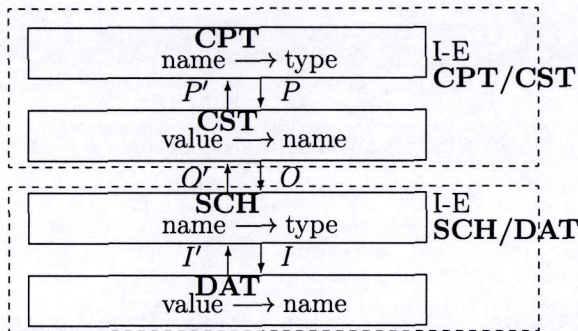


Fig. 11: Defining the Three Levels with Two Intension-Extension (I-E) Pairs and the Covariant Functors O and O'

It is indeed possible to reduce the diagram in Figure 8 to a single outer intension-extension relationship as shown in Figure 11. The upper level **CPT/CST** and the lower level **SCH/DAT** both become of the form internally $\text{value} \rightarrow \text{name} \rightarrow \text{type}$ whether by contravariant composition of the functors in an upwards direction I' and P' respectively or by contravariant composition of the functors in a downwards direction P and I respectively. This is because any other composition attempts to construct arrows which are not functions. It is interesting that the functors O' and O , in this case connecting the two intension-extension pairs, are covariant, while the functors dealing with each level at a time are contravariant. It is the difference between the static, or precompilation, of the covariant and the dynamic of the contravariant: anticipation requires the dynamic, contravariant approach.

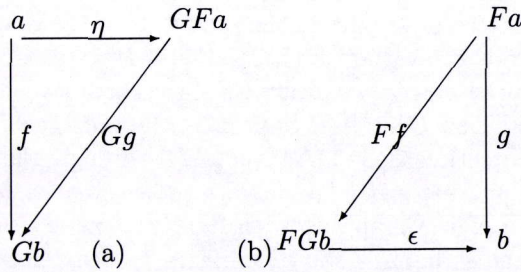


Fig. 12: Roles in Adjointness of a) η , the unit and b) ϵ , the counit of adjointness respectively

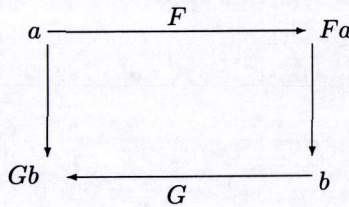


Fig. 13: Correspondence between the Arrows in Category **B** of $Fa \rightarrow b$ and in Category **A** of $a \rightarrow Gb$

4.1 Units and Counits of Adjunction in the Three Levels

Adjointness, in the general case⁸, is specified as a 4-tuple $\langle F, G, \eta, \epsilon \rangle$ where F is the free functor $F : \mathbf{A} \rightarrow \mathbf{B}$ mapping from category **A** to **B**, G , the dual of F , the underlying functor $G : \mathbf{B} \rightarrow \mathbf{A}$, η the unit of adjunction $\eta : 1_a \rightarrow GFa$ and ϵ the counit of adjunction $\epsilon : FGb \rightarrow 1_b$. η measures the difference between the identity of a in **A** and the result from applying F and G in turn to a . ϵ measures the difference between the result from applying G and F in turn to b and the identity of b in **B**. Both η and ϵ are natural transformations, comparing identities with functors.

Adjointness may not hold for every pair of dual functors. Doubt is greater for artificial mappings, as found in pure mathematics. It could be argued that natural physical systems will always exhibit adjointness because of their underlying stability. Information systems are an attempt to represent natural systems but may

⁸We are here using the set theoretic language for category theory as used by Mac Lane, see sections 3 and 4 in Appendix below.

be imperfect so it cannot be assumed that all their dual mappings are adjoint. The first condition is that it is necessary for both the diagrams in Figure 12 to commute, that is $Gg \circ \eta = f$ and $\epsilon \circ Ff = g$. Secondly, there is a correspondence, giving a unique solution, between the arrows in category **B** of $Fa \rightarrow b$ and in category **A** of $a \rightarrow Gb$, as shown in the diagram in Figure 13. The correspondence is a 1:1 mapping between $\text{hom}(Fa, b)$ and $\text{hom}(a, Gb)$ where *hom* represents a collection of arrows between the two objects. This correspondence, termed Φ , must be a natural isomorphism, that is a natural transformation that is invertible, between the two functors:

$$\text{hom}(F-, -) : \mathbf{A}^{\text{op}} \times \mathbf{B} \rightarrow \mathbf{Cat}; \quad \text{hom}(-, G-) : \mathbf{A}^{\text{op}} \times \mathbf{B} \rightarrow \mathbf{Cat}$$

Cat is a general category. The source of both functors⁹ $\text{hom}(F-, -)$ and $\text{hom}(-, G-)$ is the product of the opposite category of **A**, that is \mathbf{A}^{op} , and the category **B**. The reason why \mathbf{A}^{op} is used instead of **A** is because *hom* is a bifunctor, with two categories as its argument, contravariant in **A** and covariant in **B** ([17] pp.36-39). That one is contravariant and the other covariant is a natural consequence of the manner in which commutative diagrams are constructed to prove that the construction is a bifunctor. In particular, in an analogous manner to that shown earlier for composition of intension and extension, composition can only be achieved in the bifunctor if the opposite of one category is used as an argument. So adjointness is basically a contravariant operation in **A**, emphasising how contravariance underpins higher-order relationships. Mac Lane [17] uses the bifunctor in his definition of adjointness, as shown in the Appendix (section 4).

Figure 14 shows the details of the adjunctions, with respect to the bifunctor and the unit and counit of adjunction, for the three levels in our architecture. The bifunctor is the product of the two categories involved in the adjunction, a functor from a product in the category of limits. The first argument, as in Mac Lane's nomenclature for a bifunctor of $\mathbf{X}^{\text{op}} \times \mathbf{A}$, is taken as its dual to permit contravariant composition. So $\mathbf{CPT}^{\text{op}} \times \mathbf{CST}$ is the bifunctor for the first row where \mathbf{CPT}^{op} , the opposite of **CPT**, is the first functor argument and **CST** the second argument. The first three rows of the table show the basic adjoints $P \dashv P'$, $O \dashv O'$ and $I \dashv I'$ as already described in Figure 9. The first argument of the bifunctor is the intension and the second argument the extension. For multi-level purposes such as interoperability and enforcing security policies, it is often necessary to compose the levels. This can be done naturally with the table also showing the horizontal composition of the adjoints $OP \dashv P'O'$ and $IO \dashv O'I'$. The last row shows the horizontal composition of the adjoints across all three levels $IOP \dashv P'O'I'$. The unit and counit of adjunction become more complex as additional functors are composed, including the need for some vertical composition¹⁰. The phrase 'insertion of generators' ([17] p.87)

⁹The symbol $_$ means any object.

¹⁰The symbol \bullet indicates vertical composition as distinct from horizontal composition, indicated by \circ , which is normally, as here, omitted altogether. The bars, as elsewhere in this paper, indicate

is used in the sense of the free functor and is equivalent to Whitehead's concreteness, as discussed earlier. The final row shows how contravariant composition is achieved across all three levels. The adjointness $IOP \dashv P'O'I'$ can be represented by the functor category $\mathbf{DAT}^{\mathbf{CPT}}$, that is the exponential from \mathbf{CPT} to \mathbf{DAT} , providing the ultimate closure in the three-level architecture.

4.2 Structure of the Categories in the Three-level Architecture

Information system structures are more complex internally than those described in Section 3, where the properties of value, name and type are considered. For the \mathbf{DAT} category in particular we need to hold further details such as relationships within a category, roles, cardinalities, virtual attributes and alternative identifiers. Part of the work by Sisiaridis [27] addressed this problem. Earlier work in type theory by Robert Seely [26] presented a proof that the locally cartesian closed category (LCCC), employed as the basic higher-order category in Figure 1, and the category \mathbf{ML} of syntactically presented Martin-Löf type theories (with Π (\forall), Σ , and extensional identity types) are equivalent. Seely's work was based upon earlier work on adjointness by Lawvere [14] who developed the hyperdoctrine concept, described earlier in section 2, which was to form the basis of LCCC. Later work in computing science [30], such as compiler formalisation, has continued to follow that LCCC path.

We analysed the alternatives and concurred with the suitability of the LCCC approach for type systems in information systems. In particular for LCCC, in the form of pullbacks or comma categories, we can explore the relationships in typed systems, as representations of logical structures that handle effectively the physical world. As an example the LCCC of Figure 15(a) is proposed as an appropriate construction for the extension category \mathbf{EXT} such as \mathbf{DAT} . The category is basically a pullback with $N \times_C V$ as the limit, C as the colimit and the required adjointness as $\exists \dashv \Delta$ and $\Delta \dashv \forall$, where N is the name and V the value. The first adjoint always holds but the second, which is non-trivial, must hold for the category to be LCCC. This second adjoint requires all the possible paired values to be found in C so is fundamental to our purpose. Typing C appropriately as below means this adjoint should hold in an information system. The limit $N \times_C V$ enables all pairs of $\langle \text{name}, \text{value} \rangle$ that occur in the context of C to be represented; the colimit C holds all permissible names and values, at least conceptually, together with other information to be held in the category as itemised above. The intension category \mathbf{INT} , for example \mathbf{SCH} , shown in Figure 15(b) has $T \times_{C'} N$ as the limit, C' as the colimit and the required adjointness as $\exists \dashv \Delta$ and $\Delta \dashv \forall$, where N is the name and T the type.

contravariant mapping as the proof involves a composite isomorphism with *hom*([17] pp.103-104).

Level	Bifunctor (intension ^{OP} extension) ×	Unit of Adjunction	Counit of Adjunction
$P \dashv P'$	CPT^{OP} × CST	insertion of generators, for new construct from concept $\eta_{cpt} : 1_{cpt} \rightarrow P'P(cpt)$	verifier, for new construct $\epsilon_{cst} : PP'(cst) \rightarrow 1_{cst}$
$O \dashv O'$	CST^{OP} × SCH	insertion of generators, for new schema type from construct $\eta_{cst} : 1_{cst} \rightarrow O'O(cst)$	verifier, for new schema type $\epsilon_{sch} : OO'(sch) \rightarrow 1_{sch}$
$I \dashv I'$	SCH^{OP} × DAT	insertion of generators, for new data item from schema type $\eta_{sch} : 1_{sch} \rightarrow I'I(sch)$	verifier, for new data item $\epsilon_{dat} : II'(dat) \rightarrow 1_{dat}$
$OP \dashv P'O'$	(CPT^{OP} × CST) × (CST^{OP} × SCH) → (CPT^{OP} × SCH)	insertion of generators, for new schema type from concept $P'\bar{\eta}_{cst}P \bullet \eta_{cpt} : 1_{cpt} \rightarrow P'O'OP(cpt)$	verifier, for new schema type $\bar{\epsilon}_{sch} \bullet O\epsilon_{cst}O' : OPP'O'(sch) \rightarrow 1_{sch}$
$IO \dashv O'I'$	(CST^{OP} × SCH) × (SCH^{OP} × DAT) → (CST^{OP} × DAT)	insertion of generators, for new data item from construct $O'\bar{\eta}_{sch}O \bullet \eta_{cst} : 1_{cst} \rightarrow O'I'IO(cst)$	verifier, for new data item $\bar{\epsilon}_{dat} \bullet I\bar{\epsilon}_{sch}I' : IOO'I'(dat) \rightarrow 1_{dat}$
$IOP \dashv P'O'I'$	(CPT^{OP} × CST) × (CST^{OP} × DAT) → (CPT^{OP} × DAT)	insertion of generators, for new data item from concept $P'O'\bar{\eta}_{sch}OP \bullet P'\bar{\eta}_{cst}P \bullet \eta_{cpt} : 1_{cpt} \rightarrow P'O'I'IOP(cpt)$	verifier, for new data item $\bar{\epsilon}_{dat} \bullet I\bar{\epsilon}_{sch}I' \bullet IO\epsilon_{cst}O'I' : IOPP'O'I'(dat) \rightarrow 1_{dat}$

Fig. 14: Details of Adjunctions, Simple and Composed, in the Three-level Architecture

The arrows $N \rightarrow T$ and $V \rightarrow N$ correspond respectively to name \rightarrow type and value \rightarrow name, as considered earlier in Section 3. We can map category **INT** onto **EXT** through a contravariant functor, for example $I : \mathbf{SCH} \rightarrow \mathbf{DAT}$. Similarly we can map category **EXT** onto **INT** through a contravariant functor, for example $P' : \mathbf{CST} \rightarrow \mathbf{CPT}$. Our whole information system will use such diagrams as building blocks for constructing a topos as in Figure 1, employing higher-order logic for transforming and composing the various categories into one coherent information system.

Further Work to be reported includes the applicability of the three-level architecture to areas such as security in information systems, where anticipation is of the essence [27]. The construction of the topos of Figure 1 in more detail, analogous to the work in this paper on the three-level architecture, is also required.

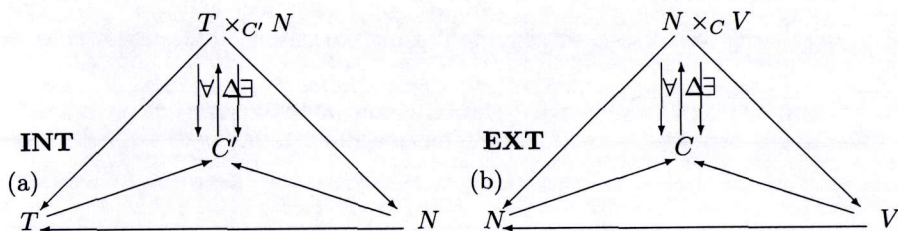


Fig. 15: Locally Cartesian Closed Categories: (a) **INT** for the Intension $N \rightarrow T$; (b) **EXT** for the Extension $V \rightarrow N$

5 Contravariance and Anticipation

Earlier we said that “If the Universe is a topos with a cartesian closed structure and anticipatory systems are featured as part of the Universe then an anticipatory system is locally cartesian closed”. We have shown that indeed the underlying categories are locally cartesian closed but in an anticipatory system all categories must be connected either as objects in a topos or by dual functors to give two-way mapping. The dual functors must be adjoint to provide a structured re-ordering of the categories involved. Such adjointness by definition requires contravariant composition across all functors. We have constructed a three-level architecture with three pairs of adjoint functors connecting four alternate intension/extension categories. The architecture can be used to illustrate for an information system, such as the Universe, how contravariant mapping underpins anticipation. Contravariant mapping enables all traversals of the levels to be dynamic on demand, as required for anticipation. Indeed when the mappings are made covariant through precompilation of functors, the resulting structure is static and non-anticipatory. That anticipatory systems are inherently contravariant is therefore our main finding.

APPENDIX

Early Development of Semantics as Contravariancy in Category Theory

1. Lawvere's Introduction to Conceptual and Formal

Lawvere introduced the principle of semantics as contravariance in classical category theory in 1969 [14, 15] in a development from sheaf theory with the addition of existential quantification in the following manner:

That pursuit of exact knowledge which we call mathematics seems to involve in an essential way two dual aspects, which we may call the Formal and the Conceptual. For example, we manipulate algebraically a polynomial equation and visualize geometrically the corresponding curve. Or we concentrate in one moment on the deduction of theorems from the axioms of group theory, and in the next consider the classes of actual groups to which the theorems refer. Thus the Conceptual is in a certain sense the subject matter of the Formal. ([15] at p.3)

Although Lawvere seems to use 'exact knowledge' to describe the general formal nature of mathematics, the phrase is a technical term in this context because of the exactness of the adjoint relationship so that early categorists (compare Freyd [8]) refer to left and right exactness for the categories here **INT** and **EXT** respectively. Nevertheless examples given by Lawvere in his extract including polynomial equations and the axioms of group theory suggest he has only the perspective of finitary predicative mathematics in mind.

2. Lawvere's Relationship between the Conceptual and the Formal

However Lawvere has in mind a formal/conceptual setting "with categories of a general sort":

There is then a further adjoint situation
$$\text{Formal} \rightleftarrows \text{Theories}$$
describing the presentation of the invariant theories by means of the formalized languages appropriate to the species. Composing these two adjoint situations, and tentatively identifying the Conceptual with categories of the general sort ($\text{Cat}, [\text{Sets}^I]$), we arrive at a family of adjoint situations
$$\text{Formal}^{op} \rightleftarrows \text{Conceptual}$$
(one for each species of theory) which one may reasonably hope constitute the fragments of a precise description of the duality with which we began our discussion.
([15] at p.15)

This description characterises his idea of hyperdoctrine ([15] pp.10-14) but the reductionism from categories of a general sort to the category of sets, *viz* ($\text{Cat}, [\text{Sets}^I]$) presupposes some theorem of representation relying on the axiom of reducibility [25].

3. Mac Lane's Contravariance and Opposites

Mac Lane on the other hand in his seminal text in 1972 [16] gives a straightforward algebraic description of contravariancy:

Consider a functor $S : \mathbf{C}^{op} \rightarrow \mathbf{B}$. By the definition of a functor, it assigns to each object $c \in \mathbf{C}^{op}$ an object Sc of \mathbf{B} and to each arrow $f^{op} : b \rightarrow a$ of \mathbf{C}^{op} an arrow $Sf^{op} : Sb \rightarrow Sa$ of \mathbf{B} , with $S(f^{op}g^{op}) = (Sf^{op})(Sg^{op})$ whenever $f^{op}g^{op}$ is defined. The functor S so described may be expressed directly in terms of the original category \mathbf{C} if we write $\bar{S}f$ for Sf^{op} ; then \bar{S} is a *contravariant functor* on \mathbf{C} to \mathbf{B} , which assigns to each object $c \in \mathbf{C}$ an object $\bar{S}c \in \mathbf{B}$ and to each arrow $f : a \rightarrow b$ an arrow $\bar{S}f : \bar{S}b \rightarrow \bar{S}a$ (in the *opposite* direction), all in such a way that

$$\bar{S}(1_c) = 1_{\bar{S}c}, \quad \bar{S}(fg) = (\bar{S}g)(\bar{S}f)$$

the latter whenever the composite fg is defined in \mathbf{C} . Note that the arrow function \bar{S} of a contravariant functor inverts the order of composition. ([17] at p.33)

Note that S is an *ad hoc* functor not the standard composition FG as used by us above. Mac Lane's notation is also a little confusing as he uses b as an object of the category \mathbf{A} not B as in the standard convention. The use of the equality sign '=' is also ambiguous in this context involving quantitative and qualitative conceptual knowledge [31] in adjunction.

4. Mac Lane's Adjunctions

The description of adjointness by Mac Lane is also from the perspective of finitary mathematics:

Let A and X be categories. An adjunction from X to A is a triple $\langle F, G, \phi \rangle : X \rightarrow A$ where F and G are functors

$$X \xrightleftharpoons[\bar{G}]{F} A,$$

while ϕ is a function which assigns to each pair of objects $x \in X, a \in A$ a bijection of sets

$$\phi = \phi_{x,a} : A(Fx, a) \cong X(x, Ga)$$

which is natural in x and a .

Here the left hand side $A(Fx, a)$ is the bifunctor

$$\mathbf{X}^{op} \times \mathbf{A} \xrightarrow{F^{op} \times Id} \mathbf{A}^{op} \times \mathbf{A} \xrightarrow{\text{hom}} \mathbf{Set}$$

which sends each pair of objects $\langle x, a \rangle$ to the hom-set $A(Fx, a)$, and the right hand side is a similar bifunctor $\mathbf{X}^{op} \times \mathbf{A} \rightarrow \mathbf{Set}$. ([17] at p.80)

The set theoretic concepts like '=', 'triple', 'function' and 'bijection' can be better represented by higher-order arrows in pure category theory.

References

- [1] Antippa, A F, & Dubois, D M, The Dual Incurive System of the Discrete Harmonic Oscillator, CASYS'05, Dubois, D M, (ed.), AIP Conf Proc **839** 11-64 (2006).

- [2] Codd, E F, Extending the Database Relational Model to Capture More Meaning, *ACM TODS* **4**(4) 397-434 (1979).
- [3] Dean, Malcolm, Opposing Entropy: Recent Advances in Cognitive Thermodynamics, Working Paper 10.0, JETC-10, Copenhagen, June 22-24 (2009).
- [4] Descartes, R, *Meditations on First Philosophy*, 1641, republished in: *The Philosophical Writings of René Descartes*, Cottingham, J, Stoothoff, R, & Murdoch, D, (trans), Cambridge **2** 1-62 (1984).
- [5] Diaconescu, Radu, Axiom of Choice and Complementation, *Proc AMS* **51** 176-178 (1975).
- [6] Dubois, D M, Mathematical Foundation of Discrete and Functional Systems with Strong and Weak Anticipation, in: M. V. Butz, O. Sigaud, and P. Gérard (Eds.), *Anticipatory Behavior in Adaptive Learning Systems LNCS* **2684** 107-125 (2003).
- [7] Elmasri, R, & Navathe, S, *Fundamentals of Database Systems*, 6th ed., Addison-Wesley (2011).
- [8] Freyd, P J, & Scedrov, A, *Categories, Allegories*, **39**, North-Holland Mathematical Library (1990).
- [9] Godement, R, *Topologie Algébrique et Théorie des Faisceaux*, Hermann, Paris (1958).
- [10] Hartle, J B, & Hawking, S W, Wave function of the Universe, *Physical Review D* **28**(12) 2960 (1983).
- [11] Heather, M, Rossiter, N, & Sisiaridis, D, The Semantics of Jitter in Anticipating Time Itself within Nano-Technology, *CASYS'07*, Dubois, D M, (ed.), *AIP Conf Proc*, **1051** 326-337, Liege (2008).
- [12] Heather, M.A. & Rossiter, B N, Anticipation as Prediction in the Predication of Data Types, *IJCAS* **20** 333-346 (2009).
- [13] Heather, Michael, & Rossiter, Nick, The Process Category of Reality, *ANPA* **31**, 224-262, Cambridge (2011).
- [14] Lawvere, F W, Adjointness in Foundations, *Dialectica*, **23** 281-296 (1969).
- [15] Lawvere, F W, Author's commentary on [14], *Reprints in Theory and Applications of Categories* **16** 1-16 <http://www.tac.mta.ca/tac/reprints/articles/16/tr16.pdf> (2006).

- [16] Mac Lane, S, *Categories for the Working Mathematician*, 1st ed, Springer-Verlag, New York (1972).
- [17] Mac Lane, S, *Categories for the Working Mathematician*, 2nd ed, Springer-Verlag, New York (1998).
- [18] Merali, Zeeya, Quantum Computing: The Power of Discord, *Nature* **474** 24-26 (2011).
- [19] Poli, Roberto, Ontological Methodology, *International Journal of Human-Computer Studies* **56**(6) 639-664 (2002).
- [20] Rossiter, B.N, & Heather, M A, *Toward the Object-Oriented Textbase*. Computing Laboratory, University of Newcastle upon Tyne, Technical Report **297** 49pp (1989).
- [21] Rossiter, N, Nelson, D A, & Heather, M A, Formalizing Types with Ultimate Closure for Middleware Tools in Information Systems Engineering, 5th Int Conf Enterprise Information Systems (ICEIS), Angers, France 23-26 April 366-373 (2003).
- [22] Rossiter, Nick, & Heather, Michael, Free and Open Systems Theory, EMCSR-2006, Cybernetics and Systems, 18th European Meeting on Cybernetics and Systems Research, University of Vienna, 18-21 April 2006, Trappl, R, (ed) 1 27-32 (2006).
- [23] Rossiter, N, Heather, M, & Sisiaridis, D, Process as a World Transaction, ANPA 27, 122-157 Cambridge (2006).
- [24] Rossiter, Nick, & Heather, Michael, The Natural Metaphysics of Computing Anticipatory Systems, *IJCAS* **23** 3-20 (2011).
- [25] Russell, Bertrand, Mathematical Logic as based on the Theory of Types, *Amer J Mathematics* **30**(3) 222-262. (1908).
- [26] Seely, R A G, Locally Cartesian Closed Categories and Type Theory, *Math Proc Cambridge Phil Soc*, **95** 33-48 (1984).
- [27] Sisiaridis, D, A Holistic Approach to Security across Distributed Information Systems, Ph.D. Thesis, Northumbria University, Newcastle upon Tyne (2010).
- [28] Szabó, Z G, ed. *Semantics versus Pragmatics*, Oxford 72-110 (2005).
- [29] Tucker, D, & Thomas, R H, Paradoxes of Intensionality, *Review of Symbolic Logic* **4**(3) 394-411 (2011).
- [30] Walters, R F C, *Categories and Computer Science*, Cambridge (1992).

- [31] Whitehead, Alfred North, *The Principle of Relativity with Applications to Physical Science*, Cambridge (1922).
- [32] Whitehead, Alfred North, *Process and Reality: An Essay in Cosmology*, Macmillan, New York (1929).
- [33] Whitehead, Alfred North, corrected edition of [32], edd D R Griffin, D W. Sherburne, New York, The Free Press (1978).