# CHAPTER 5

## The meaning of superstructure games

### 5.1. Introduction.

We have separated out the category theory of suoperators from the meaning of superstructure games given here. Non-suoperator category theory is generally thought to be associative, although exponentiation is not of course associative. Superstructures are inherently multiplicatively nonassociative from the beginning, as are thus represented by zargon algebras discussed in chapter 4.

This chapter goes beyond extensions of categories in the ideas of sutoposes, superstructural ideas of a set, comma categories and comma superstructures, Kan extensions, limits, colimits and discussion of choice, to interact these ideas with general descriptions of games.

Games have ethical implementations, but the left and right hand parts of a game are not defined ethically, but by *overwhelming*. We consider cognitive games (in Kogito) and physical games (in Fizyk). We discuss Fizyk games for zargon algebras.

Finally, we introduce intuition as a Kogito game involving insight on the left and delusion on the right. The control or boundary wall in Kogito is called the Kampf wall. In Fizyk, it is a tribble (a type of neutrino).

Insight back-propagates to Reason, a system of axioms, truth deduction interaction threads and end theorems. We have a general sequence

Reason → Insight → ⋯ → Hyperinsight

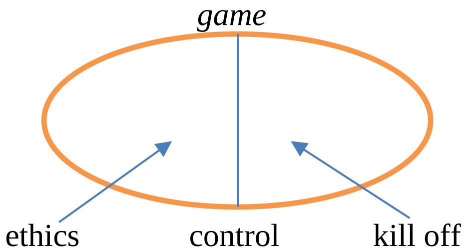
and generally

Intuition → ⋯ → Hyperintuition.

### 5.2. Making breakfast – ethical game theory at work.

We now subvert our development by starting with an example. This is very unmathematical and we like it.

Here is a game, like an egg with a line down the middle.




All games (well, the control may have disappeared) are like this. They are everywhere. My brother, Martin, said it wouldn't work for breakfast. So I said, OK, let's try an experiment. We make breakfast with toast, olive oil spread and cheese, and coffee.

So I took out two slices of bread from a cupboard, and put them in the toaster. The toaster is a *control* structure in this part of the breakfast game.

I put down the toaster control, and it popped up again. I did this twice. The toaster was refusing my need to make toast. Rather than hit the toaster until it correctly functions, the *Triviality Up* principle (see the **Contents** page of the website for further information on this) says it is most pleasant to do the easy option; obey the toaster and make bread and butter with cheese on top, so I did. Triviality Up is the easiest way to clear a general aim, in this case: make breakfast. It is however serial (not *cereal*). For any two options in a triviality tree, choose the easiest. Evaluate what is easy in the binary tree downwards so all trivial tasks are cleared with the most trivial first. This includes all distractions, like scratching your nose. If you have missed something simpler, choose it. This clears all tasks in the aim (making breakfast), it requires minimal computing power, so it is relaxing, and it is the equivalent optimal path for solving the algorthithm if fixed.

The alternative is *Problem Descent*. We said in the Contents page, don't use it, but that is not quite the case. For problem descent, you evaluate the entire problem tree globally, so it is a parallelly obtained solution if you have got that available. For problem descent, if you have Galactic computers and Donald Trump to help you, and you have missed out a peanut in error in the solution, it might be significant. So you have to solve the whole problem again, if you are lucky and have enough memory to evaluate the tree. This does not mean you cannot use Problem Descent, but that it is very probably unpleasant. It is true you can probably optimise the solution from the global answer, especially if you have plenty of experience in trying. Then if you have parallel processing the answer is faster than Triviality Up, at least if you are learning from many examples.

This shows that Problem Descent, if evolved through many trials, is probably the most evolutionary advanced method, despite being so painful. Of course we could try say, 50/50, between the two methods and maximally learn in the easiest way, perhaps.

So if we had not chosen the easiest answer, we could have chosen Problem Descent, and eventually we would have evolved our learning to know the reason why the toaster was not working. The switch was not on, as my brother Martin later mentioned.

The problem with the coffee, was that it was almost cold, and had been previously part-drunk. The question was: How do I make ethical coffee in the coffee part of the breakfast game? If we look at the game diagram, ethical coffee means we do not kill it off and start again, but supply its need by making it warm again so it can be drunk. So we warm the coffee in the microwave. But how?

If we want to be nice to the coffee and not warm it up too much, then we can estimate half the heat, test the coffee, and from that warm it up to the right amount afterwards. That way we do not overheat the coffee, so we have to wait for it to cool down. In fact, half was the right estimate, so there was only one try.

That seems to be it, but it isn't. We have not worked out why we did not find out that the toaster was not plugged in. Ethical game theory gives us advice. The control mechanism was not exercising power. If we look at the **Global Embezzlements** section of the website, we see the need, control and murder parts of the game have currencies (and they have logics!). The currency of the control section is usually money, which is a valuation of the freedom to use controlled objects. In this case the currency is electric power. If we give the control object, the toaster, electric power, it can then supply our need to toast the bread! We could have worked this out directly. Supplying electricity is the ethical method. The murder method is to supply the money to buy a new toaster!

## 5.3. The meaning and ideas of superstructure theory.

There is a programme in mathematics to replace it by generalised transformations operating on generalised objects, such as groups, known as *category theory*, or according to some, *abstract nonsense*. Transformations arising in this theory are known as *morphisms*, but in the case of morphisms these are associative mappings, that is, for three morphisms r, s and t

$$r(st) = (rs)t.$$

To distinguish a superstructure and a category, we introduce some modifications to names.

**Definition 5.3.1.** A *superstructure*, which is nonassociative, consists of suobjects or sunodes a, b, c, ..., suarrows f, g, h, ..., and two operations

*Domain*, which provides each arrow f with an object a = dom f

*Codomain*, providing each arrow f with an object b = cod f.

These satisfy

*Identity*, which assigns a suarrow $id_a = 1_a$ to each suobject a,

*Composition*, assigning to each pair of suarrows <g, f> with dom g = cod f, a suarrow gf, called their composite,

with the property

*Unit law*. For all suarrows f, a → b and g, b → c

$$f\_1_b = f \text{ and } 1_b\_g = g,$$

so that the identity suarrow $1_b$ for an object b acts as an identity operation for composition.

If it unclear from the context whether, say an identity, belongs to a superstructure or a category, we may prefix a name describing an aspect of a superstructure by su, otherwise retaining the name used for a category.

For mathematical structures derived from previously known ones, but dropping the associative rule, we prefix them by su. So we use the superstructure **suGrp** for sugroups, the superstructure **suAb** of all suabelian groups, **suMon** for the superstructure of sumonoids, etc.

We will generalise this idea to include morphisms which are not associative. We will define a generalised category with nonassociative morphisms as a *superstructure*.

## 5.4. Toposes and sutoposes.

Sets, which are particular types of mathematical objects, can be given a general categorical description in terms of mappings, called a topos, and suitable toposes can replace the category of sets as a foundation for mathematics. They were originally introduced from the consideration of sheaves, which are general sets having local differential structure defined in terms of open sets, and were used by Grothendieck in an attack on the Weil conjectures. They were later put on an even more general categorical footing by Lawvere and Tierney.

**Definition 5.4.1.** A *presheaf* on a category C is a functor F: $C^{op}$ → **mSet**.

A *sheaf*, usually denoted by F from the French 'faisceau' with the same meaning, describes a class of functions on a topological space X. These may be continuous or differentiable. The sheaf is described locally in open neighbourhoods of each point x ∊ X in terms of functions $F_x$ defined in these neighbourhoods. For all x, the sets $F_x$ can be pasted together, so that $F_x$ varies with variations of x in a continuous or differentiable way.

For example, if we were to choose continuity as the criterion, then this pasting together of open neighbourhoods U in X can be defined locally by the properties of functions f: $U \rightarrow X$

(i) (*Locality*) If f: $U \rightarrow X$ is continuous and $V \subset U$ is open, then the function f restricted to V is continuous.

(ii) (*Gluing*) If U is covered by open sets $U_i$, and the functions $f_i$: $U_i \rightarrow X$ are continuous for all $i \in I$, where I is an index set, then there is at most one continuous f: $U \rightarrow X$ with restrictions to $U_i$ for all i. This f exists if and only if the overlaps match, so that for every $x \in U_i \cap U_j$, $f_i(x) = f_j(x)$.

Let C(U) be the function which assigns each open $U \subset X$ the set of all continuous functions on U. We have seen that if inclusion is covariant, then restriction is contravariant. If $\mathcal{O}(X)$ is the category with objects all open sets U of X, and arrows $V \rightarrow U$ the inclusions $V \subset U$, then this means that the assignments

$$U \rightarrow C(U), \text{ and } \{V \subset U\} \rightarrow \{C(U) \rightarrow C(V)\} \text{ by restriction} \qquad (10)$$

define a functor C: $\mathcal{O}(X)^{op} \rightarrow \mathbf{mSet}$.

We can rephrase this as saying that a sheaf is a presheaf on a topological space satisfying the locality rule (i) and the gluing rule (ii).

Concerning property (ii), for an open covering $U = \cup U_i$ and an i indexed family of functions $f_i$: $U_i \rightarrow X$, then i is a member of the product set $\prod_i C(U_i)$, whilst the assignments $\{f_i\} \rightarrow \{f_i\}$ restricted to $U_i \cap U_j$ and $\{f_j\} \rightarrow \{f_j\}$ to $U_i \cap U_j$ define two maps, p and q, of I indexed sets to $\{I \times I\}$ indexed sets, given in the equaliser diagram

$$C(U) \overset{e}{\rightarrow} \prod_i C(U_i) \underset{q}{\overset{p}{\rightrightarrows}} \prod_{i,\,j} C(U_i \cap U_j). \qquad (11)$$

Here e is the universal map which is the equaliser of the maps p and q.

The axiom of consistent choice holds for sets in mZFC (the consistency requirement here is sometimes described by the statement that the sets are well-pointed) and the natural numbers given by the Peano axioms satisfy their properties. Additional to these items is the topos idea.

In order to define the notion of a topos, we need to define in categorical terms how we assign the values true and false to statements, and their mappings to sets or toposes. So that we can be sufficiently general, and be able to describe branched spaces in terms of probability logic and the multivalued logics of volume II, we now proceed to give an account of this.

The simplest example is when there are only two values, as in Boolean logic with true and false. Then the *characteristic function* of a subset $S \subseteq X$ is a function $\varphi_S$: $X \rightarrow \{0, 1\}$ on X with values

$$\varphi_S(x) = 0 \text{ if } x \in S, \text{ otherwise } \varphi_S(x) = 1 \text{ if } x \notin S. \qquad (12)$$

This may be expressed in the diagram

$$\begin{array}{ccc} S & \rightarrow & 1 \\ m\downarrow & & \downarrow t \\ X & \overset{\varphi}{\rightarrow} & \Gamma \end{array} \qquad (13)$$

where the top horizontal map is unique map to the object 1, m and t are monomorphisms, and the set Γ contains in this instance the values {0, 1}.

To generalise, the set Γ could contain, say {0, 1, 2}, etc., or be any arbitrary set. The map t is the *subobject classifier* for the objects in this category.

A *topos* has at least the following properties

(i) It has a subobject classifier.
(ii) It is always possible to form from it a finite number of Cartesian products, equivalently of ordered sequences, or ‘finite limits’
(iii) The category in which it resides is described by hom-sets (which are associative).

With successive restriction, let Y be a zargon subox (or real or transnatural) ladder number.
A *sutopos* has at least the following properties

(i) It has a subobject classifier ∊ Y.
(ii) It is always possible to form from it a number y ∊ Y of Cartesian products, equivalently of ordered sequences, or ‘y limits’
(iii) The category in which it resides has the number of its objects and arrows described by superstructures (which are nonassociative) with coefficients in Y.

## 5.5. Comma categories and superstructures.

A *comma category* (a special case being a *slice category*) is a construction in category theory. It provides another way of looking at morphisms: instead of simply relating objects of a category to one another, morphisms become objects in their own right.

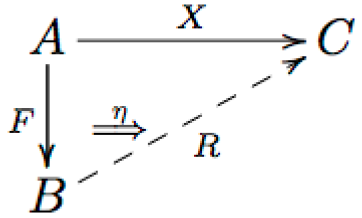
## 5.6. Kan extensions.

*Kan extensions* are universal constructs in category theory. They are closely related to adjoints, but are also related to limits and ends. They are named after Daniel M. Kan, who constructed certain (Kan) extensions using limits in 1960.

Integration and differentiation of ordered groups can be introduced as Kan extensions. Since we can differentiate and integrate exponentials, it follows that we can perform the same operations on the number of combinations of arrows, or even, it turns out, on the arrows directly, via mappings from the combinations of arrows to the arrows themselves. This theory is within category theory and uses hom-sets rather than general exponentials.

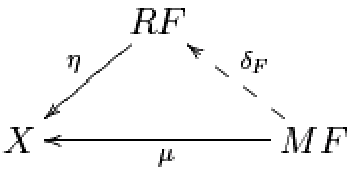
Kan extensions generalize the notion of extending a function defined on a subset to a function defined on the whole set. The definition, not surprisingly, is at a high level of abstraction. When specialised to posets it becomes a type of question on constrained optimisation.

A Kan extension proceeds from the data of three categories and two functors, and comes in two varieties: the *left* Kan extension and the *right* Kan extension. It amounts to finding the dashed arrow and the 2-cell in the following diagram:




(The natural transformation in the above depiction of the right Kan extension points to the functors. However, it should be interpreted as an arrow to the functor from the composed functor.)

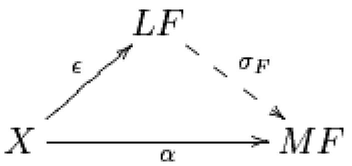
Formally, the *right Kan extension of* along consists of a functor and a natural transformation which is couniversal with respect to the specification, in the sense that for any functor and natural transformation , a unique natural transformation is defined and fits into a commutative diagram




(where is the natural transformation with for any object of ).

The functor *R* is often written .

As with the other universal constructs in category theory, the left version of the Kan extension is dual to the right one and is obtained by replacing all categories by their opposites. The effect of this on the description above is merely to reverse the direction of the natural transformations. This gives rise to the alternate description: the *left Kan extension of* *along* consists of a functor and a natural transformation which are universal with respect to this specification, in the sense that for any other functor and natural transformation , a unique natural transformation exists and fits into a commutative diagram:




(where is the natural transformation with for any object of ).

The functor *L* is often written .

The use of the word *the* (as in the left Kan extension) is justified by the fact that, as with all universal constructions, if the object defined exists, then it is unique up to unique isomorphism. In this case, that means that (for left Kan extensions) if are two left Kan extensions of along , and are the corresponding transformations, then there exists a unique *isomorphism* of functors such that the second diagram above commutes. Likewise for right Kan extensions.

## 5.7. Limits, colimits and choice.

## 5.8. Pure simple games and the definition of overwhelming.

## 5.9. Semantics in Kogito and Fizyk.

## 5.10. Zargon games.

## 5.10. Hyperintution.