# Computing Maths for Research Level

The Modelling Type Theory

Iñigo Ximeno Rodríguez

November 30, 2019

**Abstract**

Category and Type Theory, and its natural evolution to Homotopy Type Theory, are essentials on the mathematical language for research level in Computer Science and beyond. I want to show in this pages a brief simplification of the very useful notions of these theories for constructing innovations that helps to open our mind and find new ways for discovering knowledge.

# 1 Introduction

Innovation makes possible what before not, changing nothing innovation does not happen. Technology, especially in the field of computing is changing every day. Since I am a Software Architect, related with computation in health making every day new technologies to gain life time and quality for all, I know very well what innovation is. When I showed for the first time the basis of Modelling Distributed Expert Systems' technique (MoDESt), no one understood anything, except my 6 years old child. That made me think my language was too "childish" so I decided to describe it formally to capture the attention of Data Scientists and Mathematics at research level.

When I made public the pre-print "Innovating Maths for Research Level", only a few hours later, Victor Porton recognise that I was claiming the same essence of the Homotopy Type Theory HoTT[1]. In fact the fundamentals of Type Theory and the Univalent Foundations of HoTT are elementary the same notions I want to describe. After that revelation I decide to reword Innovating Maths for Research Level as Computing Maths for Research Level, with the same aim, exposing the Modelling Type Theory.

In the following pages I will introduce some singular notations and definitions that represents the essence of the mathematical computing science, making it easy and useful for every one in technology research and development. It is not needed special knowledge in maths, only some notions of basic geometry, discrete maths and algebra, but not much more than you learned at High School. In any case you will want to have on hand an scholar lecture like the book of Oscar Levin, Discrete Mathematics:An Open Introduction [2].

Please let me know[1] if you don't understand something or you don't agree with some assertions or if you think it must be improved, telling me why and/or how to.

[1] You can send me an e-mail to researcharea@ixilka.net with the subject "About Computing Maths"

## 2 Basic conventions

Basic conventions are those little wordings and notions that author wants to use in its explanations. They are little licences of authoring a creation but they are not impositions neither inventions, they are justified conventions for naturalize the language, emphasizing concepts, giving them relevance in the exposition, or convening one between different existing notations or notions. The following are the basic conventions that are essentials (and suffice) for exposing the Modelling Type Theory thus it is compulsory to internalize them strongly.

**1.** *Those sets that are more than a simple collection of items are typed sets. Given that it is not easy to find a set that do not represent something commonly a set is a typed set, which we call simply a Type.*

**2.** *A universe is a set of types. In Homotopy Type Theory, to avoid the Russell's paradox*[2]*, a universe can contains other universes but these ones can never contain other types that are not defined in its elements. However, we only need to think in an induced coincident type for a set of types which is their common ancestor.*

**3.** *Due the previous convention every type is a universe, so defining a type we are inducting a common universe for the set of small universes which are its elements.*

**4.** *The simplest type is the void set, and with this it is possible to construct every type. A set with a void set is not a void set, so with these 2 types, being void the notion of false and not void conceptually true, we can construct the type BOOL.*

**5.** *It is denoted by : a definition of a type, as well called evidence. We will say that a is an element of the type A with the expression* $a : A$.

**6.** *Definitions are propositions in the same manner as in propositional logic. Instancing is bringing any proposition to live giving it a name; such as* $null \equiv a : \{\}$ *is any instance of a void set, but it is not a set neither a type, it is simply nothing.*

**7.** *It is possible to instancing a type or universe by enumerating its elements, for example:* $BOOL \equiv \{\{\}, \{\{\}\}\}$ *or substituting them for its types* $BOOL \equiv \{false, true\}$ *where* $false \equiv \{\}$ *and* $true \equiv \{\{\}\}$

**8.** *Types can have morphisms, so the notion of category in Category Theory is a type too, and as in Category Theory that morphisms are called functors. A function is a type with a defined collection of functors between elements of the same type or over other type.*

[2] This paradox is a fundamental fallacy since recursivity is allowed in thinking, even when not always is the best way to formalize ideas.

**9.** *A functor is an arithmetical or logic proposition that can be applied over elements in an universe transforming them into other universe, hence functors are the constructors of types.*

**10.** *Elements of the same type are fibred when a functor* $fib_A : A \to A$ *namely fibre, for each element can produce other element of the same type. Chaining fibres we define paths and/or loops.*

**11.** *The space of a type is inferred by all the possible paths that can be constructed by chaining fibres, so an space is a set of fibred types. It can be saw as the induced type by combination of its operators which are its fibres.*

**12.** *Getting a subspace selecting by fixing, conditioning or excluding ranges of fibres, we construct categories of types, hence the space of a type is another type induced by a generator type and its fibres, so a set of spaces are categories of a common universe where elements are its type generators.*

**13.** *An* ***Homotopy*** *is a pair of types, where one is a fibred category called* ***groupoid*** *and other is a* ***generator type*** *equipped with at least a pair of functors called* ***funcoid*** *[3] which produce a reflexive space as a fibred space called* ***fibration***.

**14.** *Types inject its functors over its universe inducing its descendent fibrations from the structure of each type, but only the common functors of all its types can be inferred as operators or fibres to use in the construction of functors for new spaces. Note that this notion is essential key to build it self algebra including logic and arithmetic operators.*

**15.** *Given that types can be structured types, a type can be an element on more than one types and elements can be seen as containers referencing the same type. Setting the content with same type we are mirroring elements, however instancing different types with same definition we clone them.*

**16.** *The universal identity type* $Id_{\mathcal{U}}$*, is such that* $Id_{\mathcal{U}} : BOOL$ *giving the "true" element when elements of different type are the same. Regarding to the reflexivity of the identity, we say* $a : A$ *and* $b : B$ *are univalent in the sense that exists a way from* $a$ *to* $b$ *and from* $b$ *to* $a$*, even through other ones. In fact it is possible to say that* $a = b : U$ *and* $(A, B) : U$ *and not necessary* $A = B$.

**17.** *Equipping the type* $BOOL$ *with functors* $\{+, \times\}$ *equivalent to* $\{or, and\}$ *respectively, every type defined will inherit them. This type is the category of Boolean, with is the base of all the boolean algebra, and that functors are funcoids (or operators) that operates over the whole boolean algebra, and over the whole Typed Algebra described here.*

**18.** *Given that from a generator type and a fibred category, it is possible to define an homotopy by its fibres, and with funcoids a fibred space; it is called the Univalent Homotopy to the space from every category can descent. Its indexed categories are functionally the lambda concept, hence we will use* $\lambda$ *symbol to represent the univalent space of all types as categories with no lose of meaning in other theories.*

**19.** *It will be denote the infinite category of all euclidean and undirected complete graphs containing n vertices with the short name of Complete Spatial Graph and with the notation $K_n$. Given that the triangle inequality is a selector over the whole universe of graphs, as well as the condition on completion reduces the side and dimensions of that, $K_n$ is a category over the induced space of complete graphs $K$, where $n$ is the input variable for the constructor or the generator type of the homotopy what produce its subspace or homotopy type.*

**20.** *The notation (kappa) $\kappa_n$, designs implicitly an instance (any evidence in the homotopy) of a complete graph such that $\kappa_n : K_n$.*

**21.** *The notion of "edge" of a graph will be sometimes called "trace", such that both of them are an indivisible part of a path, tracing a path with traces instead of with edges will consolidate the concept widely to gain relevance.*

**22.** *The notion of vertex on an euclidean graph will be naturalized as point, usually belonging to the same plane if it is not explicitly said otherwise. The notion of vertex will be reserved to those points that are not [geometrically] enclosed by any loop of any fibred subspace.*

**23.** *We define an isomorphism of an Spatial Graph between elements of an instance of $\kappa_n$ like a loop passing from all them, fibred by a functor through an identity element into a subspace-manifold, called Isomorphic-K-Space $I_{\kappa_n}$, such as $Is : \kappa_n \to I_{\kappa_n} \to \kappa_n$. This functor transform each element of the same isomorphism to same element in the Isomorphic-SG-Space, when its vertices converge over the same trace-relationship, transforming traces and points as homotopy types. It means that traces are any path between its end-points and points can be moved to anywhere.*

**24.** *Child types, elements, loops or paths, of big types are designed with lowercase characters, but type instance (as functions or categories instances) will be denoted by uppercase (usually Greek) characters. For example, it will be said: Let $\Delta$ the result of an instance of the function $\Lambda$, we define $\Delta \equiv \Lambda(\kappa)\Big|\kappa : K_n$.*

**25.** *As it was stated previously functors are constructors of types, but usually we define constructors for many types from a single type. To distinguish constructors, even fibrations, it is recommendable to use a simplified lambda notation with dot. Using a dot following an instance of a type it will instance a new type. If we have define the functor "len" for the category circles $C$, it is possible to create a new type or a subspace of "circles" using dot notation in an expression, e.g.:*

$$C_r \equiv \{o : C\Big|o.len = r2\pi, r : N^*\} \tag{1}$$

**26.** *When a category is a fibration we can use a sub-index like in indexed families ($q : C_4$) to referring or instancing one, but when a category extends other it is more likely to use the category notation; preceding the generator type* *(or input) to the source category, followed by propositions that computes this new category. For example:*

$$\kappa_n \equiv \upsilon K \Big| \upsilon = \{p \in \mathbb{R}^2\}, |\upsilon| = n \tag{2}$$

*Note that $\equiv$ can be read like "as" denoting it is a declaration or an assignation, however for evaluating an expression of identity, as an operator, the symbol "=" will be used.*

**27.** *A subset of a fibred category or space it is referring to a subspace or a derived type with the same fibres and functors with the same elements except one or more. A subset of a fibred category or space is a fibration over the same space. When it is obvious that a subset is over a fibration we can use the symbology of Set Theory like $\subset$ or $\in$ to define a subspace to disambiguate the notion of evidence noted by ":". In that sense we say $\kappa_7 \in K_n$ to define the subspace of all Complete Spatial Graphs with 7 points, and $k : \kappa_7$ is an element in this space. Note that this quite difference in expressing $\{k : \kappa_7\} \subset \kappa_7$ gives more context of the idea and it will be necessary sometimes for defining types.*

With no impairment of the Modal Type Theory, this notions, and specially the ability of "modelling" fibrations trough fibred categories as spaces of homotopy manifolds is an innovation that makes possible to compliance computing notions with the univalent foundations of mathematics.

It is important to note that every logic and arithmetic operator, is a functor as a fibre over the elements of the same universe. Functors are operators between operands, and they correspond usually to the common operators, but although all are derived from the basic operator of addition, we can define new inexistent ones to make possible every sort of categories with homotopy types. We will define a few to show this conception in the next chapter, but it is expected that the reader can figure-out by it self the most.

# 3 First level notions

First level operators are applicable functors over descendent types of a universes. They are functors as logics expression that are inherited in all objects which uses them and produces the basic algebra in the foundations of maths. Sometimes we don't need to know how they work internally, they only represents a morphisms as a logic convention. The addition (sum) is an example of this logic convention, we really don't need how sum works internally for integers. We can define the "sum" functor to the whole universe of numbers defining it at the essential type $1 \equiv a : \{\{\}\}$ and declaring it verbally simply as : "the type 1 have a operator sum (+) which produces a result as addition of elements". Every type which uses this to build its structure will have this operator over its elements, even when its archetypes could differ. For example numbers will differ in archetyping same structure in different bases, for example:

$$\begin{aligned} (10 : REAL_2) &\equiv \{1, 1\} \\ (2 : REAL_{10}) &\equiv \{1, 1\} \end{aligned} \tag{3}$$

However, in every different category the sum functors produces internally the same:

$$\begin{aligned} (1.sum(2) : REAL_2) &\equiv \{1, 1, 1\} \\ (1.sum(2) : REAL_{10}) &\equiv \{1, 1, 1\} \end{aligned} \tag{4}$$

We can figure-out how internally works for each pair of numbers so we can assume simply that it works as expected. However in some cases it will be necessary to demonstrate how it works, then we have to show its inner proposition. In the case of the sum it can be stated using the lambda trick as simply as:

$$sum : ((a, b) : \lambda(1)) \to X \Big| X \equiv \{\{a_i : a\}, \{b_i : b\}\} \tag{5}$$

The type $\lambda$ represent all the possible universes derived for types. Giving an input type it results in the space of all possible types derived from that, 1 in this case. This sentence states that sum is a functor applicable to every pair of types inherited from type 1 resulting in a new type X such that X is composed by the elements of the input (a,b) types[3].

We will want to know other definition tricks such as the infinite number

$$INF \equiv \{a : \lambda(1) \Big| a : 1)\} \tag{6}$$

[3] Note that parenthesis worths is definitions

and the equality comparer "equal"

$$equal : ((a, c) : \lambda(1)) \to b : BOOL \Big| (b \equiv \{a_i : a\} \to \{b_i : b\} \wedge \{b_i : b\} \to \{a_i : a\}) \tag{7}$$

and lower or equals "lowereq"

$$lowereq : ((a, c) : \lambda(1)) \to b : BOOL \Big| (b \equiv \{a_i : a\} \to \{b_i : b\}) \tag{8}$$

It shows enough that every operator can be syntactically constructed, in addition of the enumerating distinct elements with "," using only the primitive keywords of set $\{\}$, as $\equiv$, inherit of ":", morphism $\to$, such that $\Big|$ and with the notion of induction for inferring types as the universal category lambda $\lambda$. In fact the first level notions may include more inferred types like lambda category as powerful foundational notions in an any proceeding for research level.

Categories are a very wide and abstract concept in maths, and sometimes confusing but they are essentials to manage huge notions in mind. Categories are fundamentals objects in mathematical language as types definitions or archetypes so they must be stated as first level notions. Categories are the abstraction of a collection of objects equipped with some morphisms named functors that transform the type in other preserving its elements structure and behaviour; many times work as selectors over the same upper space. It is basically the concept of a primitive type and the basic structure of functions, or indexed families.

Functions are as well a kind of types that must be defined as basic elements. They maps elements of a collection to others on other collection, they represent categorizations over types but the behaviour of the function (its functors) must be explicitly defined by arithmetic or logic propositions. We can only declare a category using the classical notation for functions. For example, $f : X \to Y$, when X and Y are sets of objects of the same type/category $Z$, denotes uniquely that exists a collection of morphisms over the elements of the universe of $Z$. This definition is ambiguous and reports nothing to construct well formed ideas; it is needed to define a category with its logic or arithmetic behaviour to work with them.

The next assertion can be checked to internalize the notion.

**28.** *The category of all Complete Spatial Graphs K having a functor $\Phi : (\kappa_n : \lambda(K)) \to TRACE$ and being $t$ an evidence of the type $TRACE$, satisfies the following:*

$$t \notin \Phi(\kappa_n) \to t \notin \Phi(\Phi(\kappa_n)) \tag{9}$$

*Note that the elements of the type TRACE are a sort (or class) of Complete Spatial Graphs ($K_2$), and $\kappa_n$ is an instance of the Complete Spatial Graph (K)* *category where its constructor needs only a set of points, for example:*

$$\kappa_n \equiv \upsilon K \Big| \upsilon = \{p \in \mathbb{R}^2\}, |\upsilon| = n \tag{10}$$

This simple equation, is an example of the powerful of Modelling Types for using them in first level order logic. It defines the principles and/or maximas, the basic pieces of the knowledge to construct methods, lemmas, and theorems as corollaries, which we will reach in the next level of reasoning.

# 4 Descent level notions

Gluing abstractions well defined as categories we can construct hypothesis from observations (experiments) or playing with abstractions to conjecturing a possible new knowledge (innovations). Experiments and Innovations are the fuel (or energy) that propel the engine of research.

Fibred categories are those that can be used to implements others, inhering or injecting dependencies. It is the algebra to produce the level of descent theories; expressions as formulas that gives the conceptions behind experiments or innovations.

It is wrong to say that conceptions are wrong. The power of fibred categories is the conception what you can construct with them. It is impossible to construct erroneous conceptions with fibred categories, they are spaces of inheritable types, if they can be defined they are not fallacies. They could explain something or not, they can be useful to predict or to explore new knowledge, but well-defined spaces exist, they can not be wrong.

**29.** *We can define the category Hamilton Paths as a manifold of an Spatial Graphs as the set of all paths that passes through all elements of the points set. Equipped with a functor $\Psi$, providing no proper values for the instance, this category returns the shortest one of all possibles, however the notation $\Psi_{x,y}$ returns the shortest Hamilton's Path between two points $x, y$ and providing only one $\Psi_x$ the Shortest starting or ending in the point $x$.*

This a simple example of how easy can be abstracted complexity of notions. With them we can now think without the formulation behind these important extracts of knowledge. Given that Spatial Graphs and Hamilton Paths have the same kind of objects it is possible to apply arithmetics and logic propositions, hence they are fibred categories. Furthermore we can reduce and compact our wording to define, explain conclusions, observations or proofing something, for example:

**30.** *The union of all $\Psi$ for every sub-graph in an any graph is the same graph, hence:*

$$t \in \Phi(\kappa_n) \iff \exists(t \in \Phi(\kappa)) \Big| \kappa \subset \kappa_n \tag{11}$$

*What can be a powerful conception, even enough, to assert in a syllogism.*

## 5 Research level conceptions

No scientist thinks with formulas in his head, they must first have in their mind the course of reasoning, which must be able to be expressed in simple words. Calculations and formulas come later, said Albert Einstein (or something similar) and it is very true.

In fact when we think deeply on something we produce the "movie" of the objects "playing" between them. At the research level we are imagining those conceptions and categories of objects doing something. Researchers, physic theorist and software architects[4] have similar jobs, imagining things in movement. That's a kind of creativity what makes the difference between a scientist who know a lot of maths to apply them rigorously as a technique (engineers) and the scientist who is a researcher; one knows a lot of useful formalism that can apply to improve or to guarantee constructions, and researcher makes the formalisms and conceptions, and with them constructs new ways for a new knowledge, methods and/or techniques.

When a new conception, or its conclusion or results, is useful we said it is a method, theory or theorem. This is the hat-trick of the researcher; to achieve a new useful conception often needs to define its fibred categories (multi-inherited type classes) as well as the categories participating on the storyboard. Proofing they work as a system or mechanism we reveal the essence of the model or theory (through a theorem) as the method of its source code.

Using previous abstractions is easy to state lemmas or theorems. For example, with the conceptions of Shorthest Hamilton's Path and categories of Spatial Graph and Traces we can proof a useful notion as a lemma as follows:

[4] As well as screenwriters and others who have to plan or prevent correlative interactions and issues

**31.** *Euler Graphs embed Hamiltonians*

*Proof.* As a Hamilton's Path is a sub-graph of a minimal Euler Graph, it is evident that for every $\Psi_{x,y}$ in all $\kappa_n$ adding some traces it is possible to obtain an Euler Graph, what proofs that some Euler Graphs contains at least one Hamilton Path and given that every Hamilton Path is an Euler path it concludes that the space of Euler Graphs embeds all possible Hamiltonian Graphs. $\square$

Proofing the same with low level arithmetical objects is excessively cumbersome and will not provide the same intuitionist notion. This may seem self-evident to said that every graph and every Hamilton Path is a subgraph of an Euler Graph given that every Complete Graph with a pair number of points is an Euler Graph, but it is a very different assertion[5] and it will need to proof additionally that Complete Graphs with a pair number of points are all Euler Graphs. Furthermore in that case the relevance of this assertion is that the universe of Euler Graphs is embedding the space of Hamiltonian Graphs which could be necessary in developing a reasoning.

[5] The constructor type for graphs is bounding the number of points of its descendent categories so we can't add points to Hamiltonians if we want to check they are embedded.

## 6 From the theory to the theorem

Theories are answers for questions, which follow a logic thread. They are not more than answers with some sense, even when for the answering you need inventions no one know why they are or no one have ever seen, expecting they give always the correct answer or fits the model what it represents. But "always" is a reserved word in science, it is difficult to be sure if an answer is fitting exactly always a question.

If someone asks me for the speed on which apples come down to the ground from its branch in the moon, I can say that it can be calculated knowing the radius of the moon $R$, the mass of the moon $M$ and the distance of the apple to the ground $d$, using this simple equation:

$$V = \sqrt{\frac{2GMd}{(R+d)^2}} \tag{12}$$

As Karl Popper[5] putted forward in 1934 in its primer publish "Logik der Forchung", and what before has been extensively accepted, only when an answer gives the chance to be refused by an empirical test or by a counterexample, it deserves to be considered a theory.

In fact, the equation 12 can be elevated as a theory given that it is possible to test the result with the speed of the apple when crashes to the ground. But, the theory must explicitly but not uniquely define (open-close concept) the method of refusing the exactitude of the answer it gives, what leads to assert:

**32.** ***A theory must be falsifiable and propose an infallible method that can refuse it and must be open to any other refusing method.***

For example I can propose a method using a swish clock and a digital tape measure, to calculate the speed of the apple; and if one (and only one but reliable one) of the results does not fits the result dividing the distance by the time measured, then my theory is wrong. But although the test is possible, and the inputs for the test are enough, the proposal is incorrect. It will drop false negatives, because the speed we are calculating with this formula is only good enough for short distances; this formulation (distance / time) will give the average speed, but not the final speed of the apple. If the method to refuse the hypothesis I propose is not enough reliable and/or can give false negatives, I have to propose other even when it needs new inventions. In that case, being generalist, I can propose that with a precise speedometer the result can be tested.

Saying that the calculation result can be tested with an "speedometer", and it exists one approved by the scientific community, when it is not precise enough theory could be refused. And in fact it could, even when it is very correct arithmetically and in most cases it gives an exact enough result, because it is not exactly the good **always** if Einstein's Relativity is not wrong. Relative masses change with speed so, furthermore other precisions, the values of the formula are changing in each infinitesimal change of space/time.

However, precision of the tools for the experiments sometimes are not enough for proof neither for refuse a theory, scientist are not inquisitors and science community must not be the inquisition. Newton's theory of gravitation remains a theory despite Einstein's Relativity one. New theories can precise more than previous but a theory can not be refused by little imprecisions, hence:

**33.** ***A theory can never refuse other even when it provides a better accuracy.***

This words are very important to stack in our mind when answering questions. We are not going against the knowledge of what a theory represents, we must make it grow up. One never need to say more than:

**34.** ***A theory is a good enough answer.***

But when we have the answer to the relevant question on the subject we are researching, and we have stated a claim as a corollary through a syllogism, sometimes is possible (not always is) to make a theorem proofing that.

A theorem is an assertion that reveals a new non-trivial knowledge stating that a theory always is. Sometimes in research level articles, a minor result proof like lemmas are stated as theorems but it is an abuse of worthing new conceptions. A theorem must be a corollary (of some premises like existing or new theorems, lemmas or conceptions) what serves to demonstrate a theory even when before was demonstrated by other ones.

I could say the [31] is a new theorem about Hamiltonian Graphs, and probably it is more important than I think at the moment, but on the subject I am showing is only a solid proofed argument, and probably a necessary evidence in the final proof but not a theorem; there is not a good enough answer behind an important matter, hence it is important to have in mind the following sentence:

**35.** ***A theorem demonstrates the knowledge source of a model***

Pythagoras' Theorem[6] reveals the nature of the geometric conception. It was not known as a theorem at the beginning, it was very well known by Indian and Babylonians centuries before but was never proofed as a corollary of a theory. Capturing the essence of the geometry by using the simple conception of a triangle, Euclides gave it the relevance as a theorem in the first research manuscript of the history of maths, "Elements".

# 7 Proofing Innovations

Of course new conceptions must be proofed they are valid abstractions on something. If we stablish as a new conception that each apple has a bug we can use this conception to assert that the number of apples are a lower bound of the population of bugs as new lemma that can be useful in a demonstration. If conceptions are not valid we can construct theories with them but no one will wanted to give them credibility. Obviously no one wants to believe that every apple in the whole world has at least one bug inside such as our experience contradict that.

Otherwise, if I create the new conception "Negative Energy" suggesting the existence of a subatomic undetectable particle $E^+$ and other one $E^-$, and given the equivalence of energy and mater $E = Mc^2$ and given that speed of light is constant and positive then exists "Negative Mater" as well, and there is no reliable observations against that. It is a theoretical hypothesis that can be elevated as theory giving a proof of falsifiability, such as saying that for every particle with mass exists an antiparticle with negative mass. So if we find a particle with mass that can not have an antiparticle pair, this theory is not correct. Probably using only a new not-proofed conception, the theory can be accepted as a conjecture[6], but if we use more than one we have the risk of falling in a circular reasoning or something more imbricated.
Proofing innovations, even when they are a huge, there is no problem on using them for a new conception or proofing a theorem, but sometimes it is hard to introduce a lot of new innovations to whom have being working for years with the same handful of solid notions, but it is not your fault.

They probably said you are wrong because you are not managing the notions that they have been using for years, omitting to say -with no success-, and not giving any counterexample and ridiculing your conceptions even when they are solid evidence or proofs. That's exactly the reason they never innovate at all anything and if you believe their lies you will not as well.

When someone explodes against your conceptions you are doing something right. If I think your work has no sense or is fully incorrect I will try to show you where is your fault, but it is not a question of faith neither of verdicts. Trying to make you give up with discredits, detracting or undermining your work with acrimony is a sign of fear that you are correct, so go ahead, **proofing innovations you are in the good way**.

[6] A conjecture is a theory that could be proofed but still is not.

### 7.1 The Geometric Method

If a problem can be drawn geometrically, such as a graph or as a manifold, geometrics can help to proof your conceptions[7].

Over the Euclidean Geometry (and other Groups of Lie), specially on plane, it is possible to proof some hypothesis graphically what gives visual evidence of the dynamic of the problem. Example:

**36.** *Given "n" elements more than 2, initially at the same point, if each one of them moves from the initial point linearly at the same speed "c" in different directions, only the same one will be always the farthest from the others.*

*Proof.* Given that elements were initially at the same point, and all of them moves at the same speed, at this instant they are at the same distance from the initial point, so they are any where in a circle instance. It is easy to see that

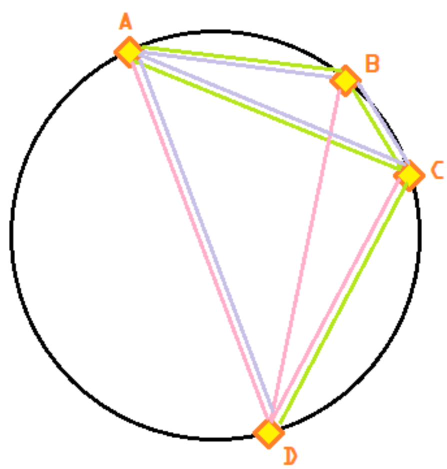


Figure 1: A 4-Expansion Graph

at any time, the same point D, will be the farthest from the others because the addition of the distance from D to A, B and C is greater than the addition of the distances from A, B or C. □

One can extends the proof showing properties of the circle, trigonometry and a huge of arithmetics explanations, but it is not needed actually, an image worths more than thousands words. It is expected that the reader must have the interest enough to conclude by its self the reasons of the proof. Unnecessary explanations can make boring what is evidently beautiful and simple.

---

[7] It is an interesting point of view in the Schoenfield theory which Aurea Grané follows in her jobs ([7] and [8]).

A hard problem for resolving connectivity in some layers of Distributed Systems can use this innovation. Dynamic Neural Persistence Networks are not fully directly connected, they are partially connected and logically structured by Transmission Control Protocol using in exclusive one port once. They shares with all the computers its local data with a level of redundancy to prevent lost of information when a computer goes down on the network. When a connection between two computers is lost, the weights (or lengths) of each path from each computer to each other is different, so every one must to recalculate paths, and the result of one depends on the result of the others.

The lemma showed in [36] gives the conception of **farthest point in a n-Expansion graph**, assuring that always exists one and that can be used to discard them from the primary layer until the number of connections assure the completion of data. Furthermore we can study the dynamical of this Graphs category and transforms other proper values in polar ones. For example: if we know that a packet reception from a "centred" source to other 4 elements happens in different times, then to know the best performance structure-connection we can draw the 4 elements in different angles to draw a 4-Expansion graph like in figure $\langle 1 \rangle$ to transpose the connectivity between them.

It is not the purpose of this essay to show the method of the Dynamic Neural Persistence Networks Model but it is to notice the importance of geometry as method of proofing, representing problems as well as the certainty of new conceptions. Probably other conception, even existent one, could bring us to the same destiny, but making your own way probably you will arrive where others not.

November 30, 2019
Iñigo Ximeno Rodríguez
Innovating Maths for Research Level

# References

[1] T. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[2] O. Levin, "Discrete mathematics." [http://discretetext.oscarlevin.com/].

[3] V. Porton, *Algebraic General Topology*, vol. 1. Republished from INFRA-M, 1 ed., 2019.

[4] D. Papadimitriou, "Introduction to functors in c♯." [https://medium.com/@dimpapadim3/introduction-to-functors-in-c-1e6b434c85ae], Jan. 2018.

[5] W. contributors, "Karl popper." [https://en.wikipedia.org/w/index.php?title=Karl_Popper&oldid=923143905], Nov. 2019.

[6] W. contributors, "Pythagoras." [https://en.wikipedia.org/w/index.php?title=Pythagoras&oldid=921064319], Nov. 2019.

[7] A. P. Albarrán, I. and A. Grané, "Profile identification via weighted related metric scaling: An application to dependent spanish children.," *Journal of the Royal Statistical Society, Series A- Statistics in Society*, vol. 178, pp. 1–26, 2015.

[8] A. Grané and R. Romera, "On visualizing mixed-type data: A joint metric approach to profile construction and outlier detection," *Sociological Methods & Research*, vol. 47, no. 2, pp. 207–239, 2018.