Completeness, Translation and Logicality

María Manzano and Luis Urtubey
University of Salamanca (Spain) - University of Córdoba (Argentina)

February 24, 2013

Abstract

This article is an extended promenade strolling along the winding roads of completeness, translations and logicality, looking for the places where they converge. We have examined the *Summa Logicae* tree: striving for a new perspective, tasting its fruits, resting on its branches, analyzing the soil where it grows and probing its roots.

1 Introduction

Completeness A logic comprises at least three different things: a class of structures, a formal language to describe these structures, and a satisfaction relation that determines when a formula of the language is true with respect to a given structure. A calculus might be added.

Formal derivability and semantical consequence are, from a general point of view, relations established in the set of formulas of the language. Proof and truth are defined by independent methods and it is not trivial, but necessary, to prove that they are extensionally the same. This is the content of the *completeness* theorem when this property is predicated of a calculus. *Strong completeness* of a calculus establishes its sufficiency for capturing the logical consequence; namely, whenever a sentence follows logically from a set of hypothesis, there is a proof of this sentence in the calculus. While the weak completeness says that we have proofs for all validities. In a complete logic the expressive power of the language and its computational strength are well balanced. Thus the question on completeness is the question on the equilibrium of the basic components of a logic: its semantics and its logical calculus.

Sometimes, we just say that a *logic is complete*. In this case we are only concerned with the computational complexity of the set of logical truths. We do not want to analyze the specific procedure used for listing the truths of that logic; all we need to know is that they are recursively enumerable. In principle, it would not be necessary to have a deductive calculus for the logic; any recursive procedure able to generate logical truths will do. Strong completeness depends on *compactness* results, because for a calculus to be strongly complete we need a guarantee that in each case where a sentence follows logically from a set of hypotheses there will be a finite subset of hypotheses from which to derive such a conclusion, since deductions are finite. As we know,

Strong completess = weak completeness + compactness

Translations Nowadays, the proliferation of logical systems used in mathematics, computer science, philosophy and linguistics makes it a pressing issue to study as well the relationships between them and their possible translations into one another. Translations between logics have been formulated as an ambitious paradigm whose tools would serve for handling the *existing* multiplicity of logics.

1. From a *proof-theoretical* point of view, your style of comparing logics will rest upon morphisms between calculi. They emerge the "labelled deductive systems" of Gabbay [12].
2. From a *model-theoretical perspective*, you will presumably compare logics by defining morphisms between the structures those logics try to describe. As in the "correspondence theory" of van Benthem [3].
3. From a suprastructural point of view, you define morphisms between categories. Among the most abstract approaches to logic we should highlight the "general logics" of Meseguer [30].

It is worth noting that in the three cases you ended up with classical logic. The paradigm of logical translation assumed in this paper belongs to the model-theoretic tradition mentioned in item 2 and the tarjet logic is Many-Sorted Logic (MSL). More on this paradigm is said in section 2 (where the completeness of MSL is explained), whereas section 3 develop our own perspective on translations. The questions we pose now are more abstract: *What is a logical system?, how can we compare logics?, how can we transfer metaproperties from one logic to another?, can we transfer completeness?, how can we create a new logic with such-and-such features?* and finally, *where does logicality lie?*

The answers are closely related to different methodologies that focus on the way certain metatheorems apply to some logics and the result enables us to formulate the equation:

Logicality = translatability into classical logic

Completeness through translation The "target logic" into which other logics are translated use to be either first-order or higher-order classical logic. By now let us restrict ourselves to the following question. Why should we translate a logic XL into first-order logic rather than into higher-order logic?

General arguments for the convenience of first-order logic are:

1. Once a logic XL is translated into first-order logic we don't need but a unique deductive calculus, which is known to be strongly complete, and a unique theorem prover.

2. Further, it is possible to borrow some metaproperties of first-order logic in order to characterize XL.

These arguments hold for unsorted first-order logic and for many-sorted first-order logic (MSL) too. Both logics have a strongly complete calculus, Compactness, Löwenheim-Skolem theorem, Enumerability, Łos theorem, etc. In section 3 we wiil argue in favor of MSL as a unifier framework.

The semanticaly inspired translation we'll countenace is performed at three levels, all of then will drive us to completeness issues. The point of departure will be the very abstract question on the mere possibility of defining a calculus for a given logic XL. We want to know first if the set of validities is recursively enumerable, what we called the question on *completeness of a logic*. The second stage will be the question on compactness of the logic under investigation and its ability to distinguish between infinite cardinalities. We know they are necessary conditions, because for a strong complete calculus to be defined as finiteness of deduction from a set of hypothesis is a must. The last stage will be completeness in all its extent. Translation into MSL gives us the clue.

Completeness and logicality Accepting that logicality has to do with translatability of a logical system in classical logic, one wonders where the logicality of classical logic could reside? Moreover it is possible now to talk about *logic* simpliciter, since we have reached a unifying framework where several different logical systems can be translated. This framework is also dealt with from different points of view. Thus in the area of *Foundations*, along with the universal characterization of the notion of deductive consequence, such as Tarski's celebrated definition in terms of a consequence operator, there are also the extant branches of *Set Theory*, *Model Theory* and *Proof Theory*.

A problematic question that arises now is whether characterizations of logicality reached from each of these perspectives, are equivalent in a certain sense. In case they were, the argument in favor de the adequeacy of the mathematical definition of consequence with the intuitive notion is reinforced. The situation contrasts with that of Recursion Theory, where several alternative mathematical notions coping with the intuitive concept of algorithm have been offered, but they are proved to be equivalent; and it served to justify —along with substantive empirical evidence— the so-called Church-Turing thesis, which affirms the equivalence between the intuitive notion and the mathematical definitions. Completeness theorem shows a sort of agreement between syntactic and semantical notions of consequence. Nonetheless, the fact that there were successful deductive calculuses designed to cope with the semantical notion of consequence, could not imply that the semantical definition accounts for the corresponding intuitive notion. There is still some persistent doubt about it [1].

[1] As Etchemendy [7] has extensively argued for.

2 Completeness in first order many-sorted logic

Many-sorted logic is not only a natural language for reasoning about more than one type of objects with an efficient proof theory, but also a unifier for many other logics. Therefore, the study of many-sorted logic and its model theory gives us a clue to the behavior of some other logics which can be translated into many-sorted logic and also furnish us with an easy way to compare different logics. As we shall show in section 3, it is indeed the perfect candidate to act as a unifying framework in which we may situate and compare most of the many logics which abound in the literature.

In section 3.5 we will show that MSL reduces to unsorted-FOL (USL).

We may wonder, why MSL is used instead of USL as unifier? Here we offer some arguments:

1. In speaking of several sorts of things, USL lacks of the naturalness we have in MSL. Moreover, formulas in USL becomes more cumbersome.

2. Deductions in a MSL calculus use to be shorter than the correspondent deductions in the USL calculus obtained by reduction.

3. It is proved in Hook [19] that a MSL theory could be interpretable in another MSL theory without their corresponding USL translations being interpretable in one another.

4. Feferman [8] showed that MSL exhibits versions of Craig's interpolation theorem that are stronger (and not reductible to) the USL version.

2.1 Structures and languages of MSL

By many-sorted logics, MSL-logics, we mean a family of logics with more than one sort of variables (at the syntactic level), each of them ranging over a different domain (at the semantic level).

In many-sorted first order logic a structure is defined as a tuple having a family of non empty sets as domains and a family of operations (functions and relations) over the domains. We classify mathematical structures by signature in such a way that two structures have the same signature if and only if the same language can be used for both. The alphabet of the language contains operation symbols, variables of different sorts and the usual logical constants. The set of expressions of the logic are obtained by a recursive definition.

Given a language and a structure, both of them sharing the same signature, each closed term of the language will denote an element in the structure and each sentence is true or false in the structure. Nevertheless, we want to widen the scope of our definition so that each term and formula gets a denotation and so we define assignments from the set of variables into the domains of the structure.

We begin with many-sorted structures, and then we build languages for describing them.

By $S_\omega(X)$ we mean all finite sequences of elements of a given set X.

Definition 1 (Signature) *A* MSL*-signature is an ordered pair* $\Sigma = \langle Sort, Func \rangle$ *where*

- *Sort is an index set with* $0 \in Sort$ *(we will write* $Sort^+$ *for* $Sort - \{0\}$*).*
- $Func : Oper.Sym \longrightarrow S_\omega(Sort) \cup \omega^+$ *is a function that assigns to every operation symbol either an element of* $S_\omega(Sort)$ *or an arity from* ω^+

Every MSL-structure is of a given signature, and we want two structures to have the same signature iff they can be described by the same MSL-language.

Definition 2 (Structure) *A* MSL*-structure of signature* $\Sigma = \langle Sort, Func \rangle$ *is a tuple*

$$\mathcal{A} = \langle \{A_i\}_{i \in Sort}, \{f^{\mathcal{A}}\}_{f \in Oper.Sym} \rangle$$

given by the following conditions:

- $\{A_i\}_{i \in Sort}$ *is a family of non-empty sets, and* $A_0 = \{\mathrm{t}, \mathrm{f}\}$
- $\{f^{\mathcal{A}}\}_{f \in Oper.Sym}$ *is a family of functions defined over the domains, which can adopt two different forms: either they are* ***typed functions*** *or* ***untyped functions****. Equality* $E^{\mathcal{A}}$ *is an untyped function whose interpretation is identity.*

Given a structure $\mathcal{A}$ of signature Σ, we need a language L adequate for speaking about $\mathcal{A}$. The alphabet of L includes all the operation symbols in $Oper.Sym$ plus the quantifier $\exists$, and for each sort $i \in Sort^+$, we have a countable set of variables, $\mathcal{V}_i = \{v_0^i, v_1^i, v_2^i \ldots\}$. $\mathcal{V}$ being the union of all those sets of variables.

Definition 3 (Expressions) *Given a* MSL*-language* L *of signature* Σ*, the corresponding set of* Exp(MSL) *is the smallest set such that:*

- *Each variable* v^i *of sort* $i \in Sort^+$ *is an* Exp(MSL) *of type* i*.*
- *If* $f \in Oper.Sym$ *with* $Func(f) = \langle i_0, i_1, \ldots, i_m \rangle$*, and* $\varepsilon_1, \ldots, \varepsilon_m$ *are* Exp(MSL) *of types* $i_1, \ldots, i_m$ *then* $f(\varepsilon_1, \ldots, \varepsilon_m)$ *is an* Exp(MSL) *of type* i_0*. If* $R \in Oper.Sym$ *with* $Func(f) = m$*, and* $\varepsilon_1, \ldots, \varepsilon_m$ *are* Exp(MSL) *of arbitrary nonzero types, then* $R\varepsilon_1, \ldots, \varepsilon_m$ *is an* Exp(MSL) *of type* 0*.*
- *For every* Exp(MSL) *of type* 0 *and* x^i *of sort* $i \in Sort$*,* $\exists x^i \varepsilon$ *is an* Exp(MSL) *of type* 0*.*

Satisfiability, consequence and validity: Let $\mathcal{A}$ be a structure and L a language, both sharing the same signature Σ. To define the adequate satisfiability relation we need assignments from the set of variables to the domains and such that $M[\mathcal{V}_i] \subseteq A_i$. An interpretation over $\mathcal{A}$ is then a tuple $\Im = \langle \mathcal{A}, M \rangle$. Definitions of satisfaction, consequence and validity are straightforward.

2.2 Deductive calculus

Now we introduce a calculus to generate validities as logical theorems and to allow mechanizing the reasoning process, namely to derive conclusions from a set of hypothesis using an effective procedure. The notion of proof is effective in the sense that there is a method by which, whenever a finite sequence of formulas is given, it can always be determined effectively whether or not it is a proof.

A deduction[2] is a finite non-empty sequence of lines, each of which is a finite non-empty sequence of formulas: $\langle \varphi_1 ... \varphi_n \psi \rangle$ is called a *sequent* with *antecedent* $\varphi_1, ..., \varphi_n$ and *consequent* ψ. The rules are the following: *Hypothesis introduction, Monotony, Proof by cases, Non contradiction, Introducing disjunction in the antecedent, Introducing disjunction in the consequent, Introducing particularization in the antecedent*, *Introducing particularization in the consequent*, *Reflexivity of equality and* and *Equals substitution.*

2.3 Completeness

The completeness proof for this calculus follows the well known Henkin's strategy. The important issue is to be able to show that every consistent set of formulas have a model, therefore syntactical consistency and semantical satisfiabibily are equivalent. The proof is performed in two steps:

1. Every consistent set of formulas can be extended to a maximal consistent set with witnesses.

2. Once we have the maximal consistent set with witnesses, we use it as a guide to build the precise model the formulas of this set are describing. This is possible because a maximally consistent set is a very detailed description of a structure.

Completeness theorem is proved in its strong sense, $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$ —for any Γ, φ such that $\Gamma \cup \{\varphi\} \subseteq \mathsf{Form}(L)$—. We prove completeness and its corollaries following the path:

$$\left.\begin{array}{l}\text{Lindenbaum's}\\ +\\ \text{Henkin's}\end{array}\right\} \rightarrow \left.\begin{array}{l}\text{Corollary}\\ +\\ \text{Lemma 1}\end{array}\right\} \rightarrow \begin{array}{l}\text{Henkin's theorem}\\ \downarrow\\ \text{Strong completeness}\\ \downarrow\\ \text{Weak completeness}\end{array} \begin{array}{l}\nearrow \text{Compactness}\\ \searrow \text{Löwenheim-Skolem}\end{array}$$

These theorems should be understood as follows:

[2] See Manzano [25] for a detailed presentation of this calculus which is proved to be *sound* and *strongly complete.*

- *Lindenbaum lemma:* If $\Gamma \subseteq \mathsf{Form}(L)$ is consistent and the set of free variables in Γ is finite, there exists Γ^* such that $\Gamma \subseteq \Gamma^* \subseteq \mathsf{Form}(L)$, Γ^* is maximally consistent and contains witnesses.

- *Henkin's lemma:* If Γ^* is maximally consistent and contains witnesses, then Γ^* has a countable model.

- *Corollary:* If $\Gamma \subseteq \mathsf{Form}(L)$ is consistent and the set of free variables in Γ is finite, then Γ has a countable model.

- *Lemma 1:* If $\Gamma \subseteq \mathsf{Form}(L)$ is consistent and $\Delta \subseteq \mathsf{Sent}(L')$ is the class of sentences obtained from formulas in Γ by replacing every free variable by a new constant in L' and if it happens that Δ has a countable model, then so has Γ.

- *Henkin's theorem:* If $\Gamma \subseteq \mathsf{Form}(L)$ is consistent, then Γ has a model whose domain is countable.

- *Strong completeness:* If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$

- *Weak completeness:* If $\models \varphi$ then $\vdash \varphi$.

- *Compactness theorem:* Γ has a model iff every finite subset of it has a model.

- *Löwenheim-Skolem:* If Γ has a model, then it has a countable model.

2.4 What have we achieved?

From the mere fact that we have a calculus, we know that there is an effective procedure for theoremhood; that is, the set Theo of logical theorems is (at least) recursively enumerable. By soundness and weak completeness we know that this property also applies to validities, as $\mathsf{Theo} = \mathsf{Val}$. We obtained this result from the strong version of completeness, which was a consequence of Henkin's theorem. Thus, the most important result was that consistency and satisfiability match in MSL. From this theorem we derived the three main result: strong completeness, compactness and Löwenheim-Skolem. Strong completeness acts as a bridge between the effective syntactical side of logic and its semantical appearance. Compactness and Löwenheim-Skolem are semantical theorems by nature, but in our proof we use the aforementioned bridge. Both theorems are tied to the fact that proofs in the calculus are finite and that to characterize cardinality a logic stronger than first order is required. We would like to stress that in our proof the more powerful result is Henkin's theorem from where the other results are easily obtained.

Once a particular logical system has been introduced, a corpus of mathematical definitions, theorems and proofs is developed. In this section the more basic ones have been anwsered. There remains nevertheless a residue of inquiry beyond the borders of mathematics which is a fertile area for philosophical scrutiny. Typical questions are: *What is the distinctive character of this logic?,* *what counts as a logical inference?, what is a connective?, are the expressive and computability power of this particular logic well balanced?, is the logic decidable?, is it complete?* and finally, *where does logicality lie?*

Several of this questions have been answered already for many-sorted logic, we want to stress that MSL is not only a natural and well balanced logic with a strong completeness calculus, but an excellent framework to place other logics.

In section 3 we will in fact reverse the investigation. The point of departure will be the very abstract question on the mere possibility of defining a calculus for a given logic. The logic under investigation might not have a deductive calculus. The second stage will be the question of finiteness of deduction from a set of hypothesis and the last one, completeness in all its extent. Translation into MSL gives us the clue.

To sum up, in section 3 we give a detailed account of completeness at several levels by means of translation, in section 4 we try to answer the question on logicality.

3 Completeness through translations

In this section we will in fact reverse the challenge done in the previous one as the logic under scrutiny might not have a deductive calculus. It is argued in [25] for MSL as the target logic in translation issues, due to its efficient proof theory, flexibility, naturalness and versatility to adapt to reasoning about more than one type of objects. As we shall show in what follows, it is indeed the perfect candidate to act as a unifying framework in which we may situate and compare most of the many logics. In what follows we are presenting translations as the path to completeness, in three stages.

3.1 Translating into MSL (three levels)

Suppose we are given a logic XL which is to be translated into MSL. What follows is a description of our methodology, which consists of a number of desiderata –rather than concrete results applicable straightforward for each specific case.

3.2 Level one: representation theorems

Let XL be the logic to be translated. By Exp(XL) and Str(XL) we mean respectively its class of expressions and its class of structures; and the same holds for MSL. If Σ is the signature of language L of logic XL, we denote with Σ^*, L^* and MSL*, respectively, its corresponding many-sorted signature, language and logic. Of course this vague notion of "correspondence" must be clarified for each translation.

We need to define two things: a recursive function Trans from Exp(XL) to Exp(MSL*), and a direct conversion Conv_1 from Str(XL) to Str(MSL*). This is depicted in the following diagram, whereby vertical lines stand for the satisfaction relationship between structures and sentences.

$$\begin{tikzcd}
\mathsf{Exp(XL)} \arrow[r, "\textsc{Trans}"] \arrow[d, dash] & \mathsf{Exp(MSL^*)} \arrow[d, dash] \\
\mathsf{Str(XL)} \arrow[r, "\textsc{Conv}_1"] & \mathsf{Str(MSL^*)}
\end{tikzcd}$$

In addition, $\textsc{Trans}$ must be defined recursively in such a way that the translation of closed XL-formulas contains at most a finite number of free variables. The conversion $\textsc{Conv}_1(\mathcal{A})$ of a XL-structure $\mathcal{A}$, will probably contain as many universes as categories of mathematical objects referred to in the syntax of XL. A possibility is to add to the many-sorted structures universes containing all categories of mathematical objects of which we may want —and are able— to refer to in the logic XL. Therefore, in $\textsc{Conv}_1(\mathcal{A})$ we shall add universes containing those sets and relations which are definable in the original structure $\mathcal{A}$ using XL.

Our first goal is to state and prove the following theorem.

Theorem 4 *For every sentence φ of the logic XL,*

$$\vDash_{\mathsf{Str(XL)}} \varphi \textit{ in } \mathsf{XL} \textit{ iff } \vDash_{\mathcal{S}*} \textsc{Trans}(\varphi) \textit{ in } \mathsf{MSL}$$

where $\mathcal{S}^*$ stands for $\textsc{Conv}_1$[$\mathsf{Str(XL)}$], and $\forall\textsc{Trans}(\varphi)$ stands for be the universal closure of $\textsc{Trans}(\varphi)$.

The next desired step is to replace the semantical restriction to S^* for a suitable set $\Delta \subseteq \mathsf{Sent}(L^*)$ such that the representation theorem holds; namely, a theorem in the following form:

Theorem 5 (Representation Theorem) *There is a recursive set of L^*-sentences Δ, with $\mathcal{S}^* \subseteq Mod(\Delta)$ and such that*

$$\vDash_{\mathsf{Str(XL)}} \varphi \textit{ in } \mathsf{XL} \textit{ iff } \Delta \vDash_{\mathsf{Str(MXL^*)}} \forall\textsc{Trans}(\varphi) \textit{ in } \mathsf{MSL}$$

for every sentence φ of the logic XL.

Remark 6 *From the previous result the completeness of this logic is straightforward; namely, $Val(\mathsf{XL})$ is recursively enumerable. Therefore, we know that XL is, in principle, a (weak) complete logic. In case the definition of logic XL were only semantically given, a complete calculi for XL is a natural demand. We also know that validity in this logic can be simulated in many-sorted logic, due to the strong completeness of MSL. Thus, the first step of our investigation on the path to completeness is performed.*

A first step towards a strong completeness result for XL may rest upon the following comparison between consequences in logics XL and MSL.

Proposition 7 *There is a recursive set of sentences, $\Delta \subseteq \mathsf{Sent}(L^*)$ with $\mathcal{S}^* \subseteq$* $\mathrm{Mod}(\Delta)$ *and such that*

$$\textsc{Trans}(\Pi) \cup \Delta \vDash_{\mathsf{Str(MXL^*)}} \textsc{Trans}(\phi) \textit{ implies } \Pi \vDash_{\mathsf{Str(XL)}} \phi \textit{ in } \mathsf{XL}$$

for all $\Pi \cup \{\phi\} \subseteq$$\mathsf{Sent(XL)}$

3.3 Level two: the main theorem

When the logic XL under scrutiny has a concept of logical consequence, we may try to prove *the main theorem*; that is, that a consequence in XL is equivalent to the consequence of its translation, *modulo* the theory Δ. So, the purpose now is to prove a stronger version of proposition 7, one that reverses the implication. To achieve this, first we define a reverse conversion Conv$_2$ of structures from Str(MSL*) into Str(XL).

$$\begin{tikzcd}
\mathsf{Exp(XL)} \arrow[r, "\textsc{Trans}"] \arrow[d, dash] & \mathsf{Exp(MSL^*)} \arrow[d, dash] \\
\mathsf{Str(XL)} & \mathsf{Str(MSL^*)} \arrow[l, "\textsc{Conv}_2"']
\end{tikzcd}$$

Once Conv$_2$ is defined and we have all the results that steam from the first level, the goal is to clearly state and prove the following:

Theorem 8 (Main Theorem) *There is a recursive set* $\Delta \subseteq \mathsf{Sent}(L^*)$ *with* $\mathcal{S}^* \subseteq \mathrm{Mod}(\Delta)$ *and such that*

$$\Pi \vDash_{\mathsf{Str(XL)}} \phi \quad \textit{iff} \quad \textsc{Trans}(\Pi) \cup \Delta \vDash_{\mathsf{Str(MXL^*)}} \textsc{Trans}(\phi)$$

for all $\Pi \cup \{\phi\} \subseteq$*Sent(XL).*

Remark 9 *From theorem 8 it is possible to prove **Compactness** and **Löwenheim-Skolem** for XL. Thus the second stage of our path to completeness is finished. The logic under investigation could have a strong complete calculus.*

3.4 Level three: deductive correspondence

When the logic XL also have a deductive calculus, Cal(XL), we can try to use the machinery of correspondence to prove, if possible, soundness and completeness for XL.

Before applying previous results to this search, we have to ensure that: (i) Trans respects connectives, (ii) Cal(XL) is finitary, (iii) Cal(XL) contains the classical propositional calculus, and (iv) deduction theorem holds in Cal(XL).

After a series of previous lemmas, the main goal of this level is to clearly state and prove the following result.

Theorem 10 (Deductive correspondence) *Let* Δ *be defined as in the Main Theorem. Then*

$$\Pi \vdash_{\mathsf{Cal(XL)}} \phi \quad \textit{iff} \quad \textsc{Trans}(\Pi) \cup \Gamma \vdash_{\mathsf{Cal(MSL)}} \textsc{Trans}(\phi)$$

for all $\Pi \cup \{\phi\} \subseteq$ *Sen(XL).*

Now we get the last of our intended results, namely soundness and completeness for the logic XL.

Proposition 11 (Soundness and Completeness of XL)

$$\Pi \models_{\mathit{Str(XL)}} \phi \quad \textit{iff} \quad \Pi \vdash_{\mathrm{Cal(XL)}} \phi$$

Proof. Consider the following equivalences:

$$\begin{array}{ccccr} \textsc{Trans}(\Pi) \cup \Delta \models_{\mathsf{Str(MSL)}} \textsc{Trans}(\phi) & \Longleftrightarrow & \Pi \models_{\mathsf{Str(XL)}} \phi & & (*) \\ \Updownarrow & & & & (**) \\ \textsc{Trans}(\Pi) \cup \Delta \vdash_{\mathrm{Cal(MSL)}} \textsc{Trans}(\phi) & \Longleftrightarrow & \Pi \vdash_{\mathrm{Cal(XL)}} \phi & & (***) \end{array}$$

Notice that equivalence $(*)$ is the main theorem, equivalence $(**)$ is soundness and completeness of MSL, and equivalence $(***)$ is theorem 10. ■

Remark 12 *We have reached the end of the road to completeness, it is important to stress that we are using the already proven completeness results of MSL to prove strong completeness for XL.*

3.5 Reduction of MSL to unsorted-FOL

Here we point out that MSL itself reduces to unsorted-FOL in the sense of theorems 13 (below), which *states that* given a many-sorted structure $\mathcal{A}$, every MSL sentence true at $\mathcal{A}$ has a translation into unsorted first order logic that is true at $\mathcal{A}^*$, where $\mathcal{A}^*$ is the result of unifying all the sorts in $\mathcal{A}$. From this result we obtain the main theorem 14 of equivalence between consequences in both logics, modulo a theory Π.

Actually, this *theorem* could be interpreted as an argument for the convenience of *using* unsorted-FOL instead of MSL within the "translation paradigm." Are we arguing against ourselves? Well, our preference for MSL is well grounded *in spite of* theorems 13 and 14. We mention this translation in order to stress the point that we ended up in classical logic, even unsorted-FOL if you wished.

For the syntactical translation (known as "relativization of quantifiers"), let L be a many-sorted language of signature Σ, and let $\mathit{Oper.Sym}$ be its set of operation symbols. Then we need an unsorted language L^* with $\mathit{Oper.Sym}^* = \mathit{Oper.Sym} \cup \{Q_i : i \in \mathit{Sort}\}$. Variables of L will be treated as variables of L^*. Let us call $\textsc{Trans}$ the translation from L-expressions to L^*-expressions:

$$\begin{array}{rcl} \textsc{Trans}(x^i) & := & x^i \text{ for each sort } i \in \mathit{Sort} \\ \textsc{Trans}(f(\tau_1, \ldots, \tau_n)) & := & f(\textsc{Trans}(\tau_1), \ldots, \textsc{Trans}(\tau_n)) \\ \textsc{Trans}(R\tau_1, \ldots, \tau_n) & := & R\textsc{Trans}(\tau_1), \ldots, \textsc{Trans}(\tau_n) \\ \textsc{Trans}(\exists x^i \phi) & := & \exists x^i (Q^i x^i \wedge \textsc{Trans}(\phi)) \end{array}$$

For the conversion of MSL-structures into unsorted structures, let $\mathcal{A}$ be a many-sorted structure of signature Σ. Then we construct its corresponding one-sorted structure $\mathcal{A}^*$ by something called "unification of domains." Formally, we define the structure

$$\mathcal{A}^* = \langle \bigcup_{i \in \mathit{Sort}^+} A_i, \{f^{\mathcal{A}^*}\}_{f \in \mathit{Oper.Sym} - \{\neg, \vee\}}, \{Q_i^{\mathcal{A}^*}\}_{i \in \mathit{Sort}^+} \rangle$$

by means of the following clauses:

- The domain of $\mathcal{A}^*$ is the union of the domains of $\mathcal{A}$
- For each $f \in \mathit{Oper.Sym}$ with $\mathit{Func}(f) = \langle i_0, i_1, \ldots, i_n\rangle$ and $i_0 \neq 0$, $f^{\mathcal{A}^*}$ is any extension of $f^{\mathcal{A}}$ such that
$$\mathrm{Dom}(f^{\mathcal{A}^*}) = \Big(\bigcup_{i \in \mathit{Sort}^+} A_i\Big)^n \quad \text{and} \quad f^{\mathcal{A}^*} \restriction (A_{i_1} \times \ldots \times A_{i_n}) = f^{\mathcal{A}}$$
where new values in $f^{\mathcal{A}^*}$ are arbitrarily chosen.
- For each $R \in \mathit{Oper.Sym}$ with either $\mathit{Func}(R) = \langle i_0, i_1, \ldots, i_n\rangle$ or $\mathit{Func}(R) = n$,
$$R^{\mathcal{A}^*} = \{\langle x_1, \ldots, x_n\rangle \in (A^*)^n : R^{\mathcal{A}}(x_1, \ldots, x_n) = \mathrm{t}\}$$
- $Q_i^{\mathcal{A}^*} = A_i$ for each $i \in \mathit{Sort}^+$

It is worth noting that for every $f^{\mathcal{A}}$ there are many different operations extending it, which means that there are different USL-structures generated by the above conversion. Still, the following theorem holds for all of them.

Theorem 13 *Let $\mathcal{A}$ be a normal MSL-structure, $\mathcal{A}^*$ one of its USL counterparts, L a language for $\mathcal{A}$, and ϕ some sentence in L. Then*

$$\mathcal{A} \vDash \phi \iff \mathcal{A}^* \vDash \textsc{trans}(\phi)$$

From this theorem we obtain the following result.

Theorem 14 *There is a recursive set $\Pi \subseteq \mathsf{Sent}(L^*)$ with $\mathcal{S}^* \subseteq \mathrm{Mod}(\Pi)$ and such that*

$$\Gamma \vDash_{\mathit{Str(MSL)}} \phi \;\; \textit{iff} \;\; \textsc{Trans}(\Gamma) \cup \Pi \vDash_{\mathit{Str(FOL^*)}} \textsc{Trans}(\phi)$$

for all $\Gamma \cup \{\phi\} \subseteq$Sent(XL).

For all this, see Manzano [25, pp. 257–262].

3.6 What else?

The questions we pose now are more abstract: *What is a logical system?, how can we compare logics?, how can we transfer metaproperties from one logic to another?, how can we create a new logic with such-and-such features?* and finally, *where does logicality lie?*

All of them were answered in this section by means of translations into a common framework, with the only exception of the last one in case the logic considered is the one used as framework, MSL.

Allowing that we can make equivalent logicality of several logical systems with their translatability into classical logic, it still remains to account for the logicality of classical logic. It is worthwhile to say that we are now just considering classical logic as a special unifying logical framework where many logics could find an adequate counterpart. With respect to this neutral logical framework is that we ask now where its logicality may reside. Notably it is hardly likely that the roots of the logicality of our assumed logical framework will be placed outside this formal scheme. Appealing to ordinary logical intuitions sounds inappropriate in the present case. Thus the answer has to be sought resorting to more abstract and purely theoretical considerations.

4 Completeness logicality and representation

Since the concept of consequence is presented as the core of our discipline, it seems natural to study its properties and try to grasp *logicality* via an abstract consequence relation. There are several ways to capture the intuitive notion of consequence, most notably Tarski's semantic definition and the syntactical approaches provided by a variety of deductive calculuses.

It can also be seen as an operator that acts between sets of formulas of the formal language, the notion of consequence being defined by the properties of the operator.

Conjecture 15 ***Logicality could be identified with the properties of the consequence operator****: reflexivity, monotonicity and cut.*
1.- $\Delta \mid\!\sim \varphi$, *if* $\varphi \in \Delta$ (reflexivity)
2.- If $\Delta \mid\!\sim \varphi$ *then* $\Delta \cup \Delta\prime \mid\!\sim \varphi$ (monotonicity)
3.- If $\Delta \mid\!\sim \varphi$ *and* $\Delta \cup \{\varphi\} \mid\!\sim \psi$ *then* $\Delta \mid\!\sim \psi$ (cut, or transitivity)

To achieve a better understanding of this abstract relation let us go on to investigate the mathematical universe from where we extract the models of our formulas, the mathematical structures we study in logic.

4.1 Set Theory

The idea intuitively more fruitful and the most widespread is that our universe is a hierarchy of mathematical sets, the so-called Zermelo hierarchy. The fundamental assumption is that the sets are built by levels. We must answer the following questions:

1. *What collection do we take as our initial collection?*

2. *Which collections of objects from lower levels of the hierarchy do we take as elements of new levels of the hierarchy?*

3. *How far does the hierarchy extend?*

To answer the first question we should consider whether we want to have objects that are not sets or if you just want sets. In the last case, we begin with the empty set, $\mathcal{V}_0 = \varnothing$. There are two possible answers to the second question:

one is to allow just those collections that are describable in the language for set theory; another possible answer is to have all possible subsets. In the first case, we are adopting a fundamental axiom of set theory, *Axiom of Constructibility.* Zermelo solution is the second one. Given $\mathcal{V}_\alpha$ we set

$$\mathcal{V}_{\alpha+1} = \wp(\mathcal{V}_\alpha)$$

Now, the above definition tells us how to pass from $\mathcal{V}_\alpha$ to $\mathcal{V}_{\alpha+1}$. But what do we take when λ is a limit ordinal? We define:

$$\mathcal{V}_\lambda = \bigcup_{\alpha<\lambda} \mathcal{V}_\alpha$$

Finally, the answer to the third question is that there is no end, you can always build new levels. Thus, for each ordinal α there should be a corresponding level $\mathcal{V}_\alpha$. Zermelo Fraenkel hierarchy is the union of all levels, the *universe of sets*:

$$\mathcal{V} = \bigcup_{\alpha\in Ord} \mathcal{V}_a$$

$\mathcal{V}$ contains all sets, but it is not a set.

The structures we study in mathematical logic, the models of our formulas, the models of our theories, are sets located in one of the levels of the hierarchy. They are like small galaxies in the mathematical universe.

Remark 16 *In first order logic a structure is defined as a tuple having a family of non empty sets as domains and a family of operations (functions and relations) over the domains. It seems natural in this case to look at what all structures share, to investigate what remains after abstracting from the specific nature of the objects that constitute the universes.*

Invariance Tarski does not ask *what is logic?* or even *what is a logical inference?* but, *what are logical notions?* In fact, he is searching into the hierarchy of sets trying to figure out what remains after abstracting from it the specific nature of the constituent objects. He uses his own definition of semantic consequence and tries to implement a program that have had great results in mathematics: he uses invariance under certain groups of transformations. So a notion is a logical one if we can define relevant transformations over the universe of sets and we can prove the invariance under transformations of the notion under consideration. Finite type hierarchy ([5], [16]) is the most frequent idealization of the mathematical universe. The transformations used to define invariance goes from permutations to bijections and homomorphisms [24], [33], [31], [14]. Tarski analyzes the logical notions of *Principia Mathematicae* and shows that they are invariant under permutations of the universe of individuals in the hierarchy, in particular, no individual is invariant while the empty set and the universal class are invariant. As examples of binary relations that are logical notions he includes again the empty set and the universal binary relations, identity and its complements.

Conjecture 17 *Logicality is invariance under transformations.*

To speak of mathematical structures to characterize them, to fix them or compare them, we can use the tools of *Universal Algebra*, but it is also possible to introduce a formal language and exhaust the resources of logic. In doing so we enter the sanctuary of *Model Theory.*

4.2 Model Theory

Classical model theory, is a branch of mathematical logic that deals with the relationships between descriptions in first-order languages and the structures that satisfy these descriptions. We are considering now the first order logic. The bridge between these two types of reality is the concept of truth, namely, the notion

$$\textit{formula} \quad \varphi \quad \textit{is true under interpretation } \Im$$

A *theory* is a set of sentences closed under deducibility. Being first order logic a complete logic, we can use in the definition the semantical relation of consequence.

Given a theory Γ, we obtain the class of its models,

$$Mod(\Gamma) = \{\mathcal{A} \mid \mathcal{A} \vDash \gamma, \textit{ for every } \gamma \in \Gamma\}$$

or, given a class of structures $\mathfrak{K}$ sharing a signature, we define its theory, namely, all the sentences true in each structure in the class

$$Th(\mathfrak{K}) = \{\varphi \in \mathsf{Sent}(L) \mid \mathcal{A} \vDash \varphi, \textit{ for each } \mathcal{A} \in \mathfrak{K}\}$$

Each model $\mathcal{A}$ selects from the set of all sentences of the language, the set $Th(\mathcal{A})$ describing $\mathcal{A}$. We see that all theories share a core set of valid formulas Val, which, being true in all models necessarily fail to describe any particular one. The relevant question is

Do these formulas characterize anything?

In there we find what is common to all theories, therefore we can say that these formulas describe the logic itself.

Conjecture 18 ***Logicality is what all theories share.*** *So that we identify it with the set* Val*, namely, the smallest possible theory.*

$$\mathit{\mathsf{Val}} = \bigcap \{Th(\mathcal{A}) \mid \mathcal{A} \in \mathfrak{U}\}$$

Therefore, to capture *logicality*, the essence of logic, is the same as catching the set of valid formulas.

How do we do?

We now think that *logicality* is encrypted in Val. But analyzing this theory more closely we realize that it is far too abstract and so, to characterize Val using the tools of model theory is provably not very useful. We turn now to *Proof Theory* and ask

Can we generate Val*?*

4.3 Proof Theory

Conjecture 19 *Although the set* Val *of valid formulas do not describe any particular structure, we try to generate this set using logical rules. We then define the logic by using the rules of the calculus. Finally, we identify* ***logicality*** *with logical rules, at least as an operational criteria.*

Remark 20 *This identification is possible due to the fact that the logic is weak complete and thus this theorem and logicality fit tightly .*

Conjecture 21 ***Logicality can be identified with logical theorems.*** *Let us introduce* Theo *to refer to the set of logical theorems, which can be derived from the empty set of hypothesis,* $\mathsf{Theo} = \{\varphi \mid \vdash \varphi\}$*. By soundness and weak completeness, we obtain* $\mathsf{Theo} = \mathsf{Val}$*.*

Have we achieved anything?

Remark 22 *Although these sets are extensionally the same, and so* ***logicality*** *still rests in the set of validities, we have just made an important move. We now have a syntactical procedure to generate this set, the proposal 19 is reinforced.*

FOL is sound and complete in the strong sense

$$\Gamma \models \varphi \;\; \textit{if and only if} \;\; \Gamma \vdash \varphi$$

Two of the most significant results of model theory are derived from the preceding theorem, namely, *compactness* theorem and *Löwenheim-Skolem* theorem.

1. **Compactness:** Proofs in a logical calculus have a finite character and so the completeness theorem guarantees that consequence from any Γ is always reductible to a finite set of hypothesis, the ones used in the proof. This very simple observation allows the proof of compactness as an easy corollary of completeness.

2. **Löwenheim-Skolem:** There are several theorems under this name, all of them about the size of models. The oldest one by Löwenheim (1915) establishes that a sentence with an infinite model have a countable infinite model as well.

What is the most important characteristic of a logical calculus? Most logicians will agree that effectiveness of deductions and the property that theoremhood is either decidable or at least recursively enumerable. In MSL, as well as in FOL, the notion of a theorem is recursively enumerable.

4.4 Recursion Theory

What does it mean to be recursive?

The intuitive idea is that of an algorithm, that is, a mechanical procedure that can be applied in a finite number of steps, not including hazardous actions (like throwing a coin and acting in accordance to the result).

There are several mathematical definitions of this concept, all of then trying to characterize what an algorithm is, and all of then were proven to be equivalent, *i.e.* they define the same class of functions. Several of these definitions appeared in the 1930s to characterize notions that seemed, in principle, to be different: the first one was Gödel's characterization of the functions defined by means of recursion; the second was the concept of functions defined by the $\lambda-$operator, which Church and Kleene introduced; and the third was the notion of a function computable by an abstract machine —Turing machines.

Around the same time Church formulated the thesis[3] that the precise mathematical concept of a recursive function, as defined by Turing, corresponds exactly to the intuitive concept of effective computability. Being a thesis involving intuitive notions, there is no mathematical proof for it. There are, however, strong reasons for accepting it and in fact is globally accepted.

Invariance In this context it is natural and expected Feferman's result [9] associating logical operations that are invariant under homomorphisms with the $\lambda-$definable.

4.5 Algebraic Logic

Around 1850, Boole observed that certain classical laws of logic could be expressed by means of algebraic equations. The fact that one could use familiar algebraic methods to operate with these laws, served to show an underlying identity between logic and the algebra of numbers.

Further his theory was enriched by providing an interpretation for his algebraic equations, in which variables represented sets instead of propositions. Stone followed through with his interpretations, by considering the class of all structures which satisfy them, calling these Boolean algebras, and studying their relations to one another in terms of such concepts as homomorphisms, subalgebras, and direct products.

The use of Boole's equations to define the class of Boolean algebras, and the central role played in their theory by Stone's representation theorem, had a profound effect on the further development of algebraic logic. One of the first was the formulation of cylindric algebras by Tarski in the fifties, with the purpose of providing a class of algebraic structures that bear the same relation to first-order first-order logic as the class of Boolean algebras bears to propositional logic.

Needless to say, the concept of cylindric algebra is only one among several classes of algebraic structures that have been introduced in connection with

[3] Church's thesis is treated in [1] and [26].

logical systems. Certain classes are obtained by modifying the Boolean laws in order to deal with "nonclassical logics". Among types of algebras employed for studying classical logic, polyadic algebras are those which, along with cylindric algebras, have been studied most intensively[4].

4.6 Completeness as representation

Eli Dresner [6] also paid attention to the interpretation of first order completeness theorem as a mapping "between two structured domains: the set of sentences in a given first-order language structured through the deductive consequence relation and the same set structured through model-theoretic consequence". This second domain usually construed as consisting of sets of models, related to one another set-theoretically. Moreover there is no a ready-made interpretation for these mappings. On the contrary, possible alternatives should be looked over.

Dresner refers other cases where the association of semantic values with syntactic entities is made on similar grounds, the Hilbert and Bernay's 1918 proof of the consistency of the propositional calculus —referred also by Zach [34]— using assignments of 0 and 1 to formulas and showing that

- all axioms get 0 under every assignment,
- all inference rules preserve this property, and
- a formula and its negation cannot be both assigned 0 and therefore cannot be both provable.

As Dresner points out, this mechanism is not viewed by Hilbert and Bernays as grounding syntactic consistency in a more basic truth-theoretic apparatus, but rather as a means for capturing and proving syntactic relations that are in no need of further grounding in any way. Similarly, in the approach to completeness for first-order logic following Henkin' proof, syntactic and semantic validity stand on their own.

It is relevant to point out that this construal of the completeness theorem is in tune with actual mathematical and logical practice. Considering the way in which the theorem is formulated when it is proved by Henkin [21], we can see that it is stated under the claim that every deductively consistent set of sentences of a first-order language has a model[5]. The fact that the completeness theorem was formulated in that way, is indicative of its significance, since we often start with a theory that is known (or hypothesized) to be deductively consistent and use the theorem as indicating that there is indeed a model that attests to this fact.

[4] For further reference, [22], [23], [32], [15].

[5] In section 2.3 we used the term Henkin's theorem for this formulation from where both strong and weak completeness follows easily.

Conjecture 23 *Completeness theorem proves a useful representation between deductive consistency and model-theoretic validity for our basic logical framework. Consequently it also shows a balance among these different components which conform our approach to the notion of logicality.*

4.7 Recapitulation and further goals

If we analyze the various proposals in this section we see that both in Set Theory and Recursion Theory *invariance* is used to characterize logical notions while in Model Theory and Proof Theory the set of logical theorems (or its rules) and the set of validities are identified with *logicality.* Moreover, in Algebraic Logic, representation theorems play an important role concerning *logical properties* of deductive systems.

What is needed to identify all these proposals?

No doubt, *Completeness theorem* is the key to conjectures number 18, 19 and 21. We will take an algebraic perspective to be able to analyze the other two.

Roughly speaking, logicality seems to have many faces. We can recognize perhaps the most conspicuous ones. The most important point is whether there exists equivalence among these characterizations. Whether they boil down to the same concept or different notions are involved instead. The concept of logical consequence bears on this issue. Unfortunately we don't have a clear equivalence between the intuitive notion of logical consequence and its formal counterparts, tailored by using proof and model-theoretical notions. Admissibly completeness theorem acts like a bridge between semantic and syntactic representations of the notion of validity or logical consequence. It guarantees that both semantical and syntactic notion match up. It must be noted that we do not mean that the theorem is carrying over some intuitive content in order to improve a weaker notion of deducibility by means of the stronger model-theoretic underpinnings of deduction. Even though historically model theory consolidates in a more general (and possibly extended) way these formal underpinnings. In other words, what the theorem tunes or calibrates are the formal counterparts of these notions of syntactic and semantic consequence, obtained into the formal framework of the logical theory.

5 Conclusion

We found ourselves within the forest dark of logic, and managed to pinpoint the significance of logicality and completeness theorem. Starting from the unifying framework of translation in Many Sorted Logic, introduced to cope with the extant proliferation of logics, we have seen how different methodologies converge to the framed of classical logic. It is understandable then that one wonders where logicality of classical logic could reside. An answer to this contest can be looked for by searching through the branches of logical foundation. The upshot of this search revolves around the equivalence of several characterizations of logicality obtained from each of these perspectives. Completeness theorem seems to be relevant for solving some conjectures regarding these proposals. This is the reason why our section 2 is devoted to the completeness theoremof MSL. In section 3 we have given special consideration to the strategy of translation into MSL. Different levels of translations were distinguished and general outcomes concerning completeness of the logical calculuses were introduced in subsection 3.1. In subsection 3.5 we proved the reduction of MSL to unsorted FOL. In this way we ended up in classical logic, even unsorted-FOL, although our preference for keeping MSL within the translation paradigm is still well grounded.

In section 4 we turned to the problem of the logicality of classical logic, just considered as an universal logical framework. With this purpose we have explored different alternatives stemming from the area of logical foundations. We discovered that completeness theorem plays an important role in order to articulate these alternative approaches.

As for the latest of these foundational approaches to logical theory coming from algebraic logic, we focus our concerns on the purported relationship between completeness and representation theorems. In this case completeness theorem is considered as a mapping between two domains. It also entails to reject the construal of this theorem as founding a type of validity on the basis of the other, since a representation in this general sense consists in a structure-preserving mapping between two domains, while the former is certainly not viewed as being founded by the theorem on the latter.

It just remains to be said that by looking at completeness theorem this way we are following a novel suggestion on an old-standing matter. We are not merely considering its significance from the metalogical point of view, but reinterpreting the role it plays into the logical theory by articulating different perspectives on the unifying logical framework.

Acknowledments Both authors were supported by grant FFI2009-09345.

Second author was partially suported by grant Pid-2007-Bid-1606.

References

[1] Alonso, A. Manzano, M. "Diagonalisation and Church's thesis: Kleene'sxHomework". *History and Philosophy of Logic.* **26/2 .** 93-113. 2005.

[2] van Benthem, J. *Modal Logic and Classical Logic*, Bibliopolis, Naples, 1983.

[3] van Benthem, J. *Correspondence theory.* (in [10]), 1986

[4] Béziau, J.-Y. (ed.): *Logica Universalis*, Birkhäuser, Berlin, 2005

[5] Church, A. "A formulation of the simple theory of types". *Journal of Symbolic Logic,* 1940

[6] Dresner, E. ,"Logical Consequence and First-Order Soundness and Completeness: A Bottom Up Approach". *Notre Dame Journal of Formal Logic*, Vol. 52, Number 1, 2011.

[7] Etchemendy, J. *The Concept of Logical Consequence*, Harvard University Press, Cambridge, 1990.

[8] Feferman, S. "Lectures on proof theory," *Proceedings of the Summer School in Logic (Leeds 1967)*, LNM 70, Springer, Berlin, pp. 1–107, 1968.

[9] Feferman, S. "Logic, logics and logicism". *Notre Dame Journal of Formal Logic.* 1999

[10] Gabbay, D y Guenthner, F. *Handbook of Philosophical Logic 2^{nd}* edition. Kluwer Academic Publishers. Dordrecht. The Netherlands. 2001.

[11] Gabbay, D.M. (ed.). *What is a Logical System?*, Oxford University Press, Oxford, 1994

[12] Gabbay, D.M. *Labelled Deductive Systems*, Volume 1, Oxford University Press, Oxford, 1996

[13] Gödel, K. "Eine Interpretation des intuitionistischen Aussagenkalküls," Ergebnisse eines mathematischen Kolloquiums, 4 , pp. 34–40, 1933.

[14] Gómez-Torrente, M. "The problem of logical constants". *Bulletin of Symbolic Logic.* vol 8. 2002.

[15] Halmos, P. *Algebraic Logic.* New York: Chelsea. 1962.

[16] Henkin, L. "Completeness in the theory of types". *The Journal of Symbolic Logic.* vol. 15. pp. 81-91. 1950

[17] Henkin, L. "Banishing the rule of substitution for functional variables," *The Journal of Symbolic Logic*, 15, pp. 81–91. 1953.

[18] Henkin, L. "Mathematical foundations for mathematics". *American Mathematical Monthly*, vol. 97, N^o. 1, January 1990.

[19] Hook, J.L.: "A note on interpretations of many-sorted theories," *The Journal of Symbolic Logic*, 50, 1985, pp. 372–374

[20] Mc Gee, V. "Logical operations". *Journal of Philosophical Logic.* vol 25. 1996

[21] Henkin, L. "The completeness of the first-order functional calculus," *The Journal ofSymbolic Logic*, vol. 14, pp. 159–166. 1949.

[22] Henkin, L. "Mathematical foundations for mathematics". *American Mathematical Monthly*, vol. 97, N^o. 1, January 1990.

[23] Henkin, L. and Tarski, A. "Cylindric Algebras". Proceedings of Symposium in Pure Mathematics, Vol. II: Lattice Theory. pp. 83-113. 1959.

[24] Mc Gee, V. “Logical operations”. *Journal of Philosophical Logic.* vol 25. 1996

[25] Manzano, M. *Extensions of First Order Logic*, Cambridge University Press, Cambridge, 1996

[26] Manzano, M. “Alonzo Church: His Life, His Work and Some of His Miracles”, *History and Philosophy of Logic.* **18/4.** 211-232. 1997.

[27] Manzano, M. “Divergencia y rivalidad entre lógicas”. Orayen and Moretti eds: *Enciclopedia Iberoamericana de Filosofía.* Volumen 27 de Filosofía de la Lógica. Raúl . Editorial: Trotta. España. 2004..

[28] Manzano, M. “¿Qué es esa cosa llamada lógica?”. Atocha et al. (eds): *Representación y logicidad.* Fénix Editora. Sevilla. 2005.

[29] Manzano, M. and Alonso, E. “Completeness: from Gödel to Henkin”, to appear.

[30] Meseguer, J.: “General Logics,” in Ebbinghaus et al. (eds.): *Logic Colloquium ’87*, North-Holland, Amsterdam, pp. 275–239. 1989.

[31] Mostowski, A. “On a generalization of quantifiers”. *Fundamenta Mathematicae.* vol 44. 1957.

[32] Rasiowa, H. and Sikorski, R, *The Mathematics of Metamathematics.* Warsaw: Polska Akademia Nauk, Monografie Matematyczne, 1963.

[33] Sher, G. *The bounds of logic.* MIT. Cambridge. Massachusetts. 1991.

[34] Zach, R. “The practice of finitism: Epsilon calculus and consistency proofs in Hilbert’sprogram (History of Logic Helsinki, 2000),” *Synthese*, vol. 137 (2003), pp. 211–59.