Henkin on Completeness

María Manzano*
Universidad de Salamanca
Enrique Alonso
Universidad Autónoma de Madrid

June 3, 2012

Abstract

The Completeness of Formal Systems is the title of the thesis that Henkin presented at Princenton in 1947, under the guiadance of Alonzo Church. His renowned results on completeness for both *type theory* and *first order logic* are part of his thesis. It is interesting to note that he obtained the proof of completeness of first order logic readapting the argument found for the theory of types.

It is surprising that the first-order proof of completeness that Henkin explained in class was not his own but was developed by using Herbrand's theorem and the completeness of propositional logic.

"Since we use the completeness of sentential logic in our proof, we effectively reduce the completeness problem for first order logic to that of sentential logic."

We conclude this paper by pointing two of the many influences of his completeness proofs, one is the completeness of *basic hybrid type theory* and the other is in correspondence theory, as developed in [9].

1 The completeness of FOL in Henkin's course

The story behind this is that of María Manzano, who during the academic year of 1977-1978 attended his class of *metamathematics* for doctorate students at Berkeley. Before each class Henkin would give us a text of some 4-5 pages that summarized what was to be addressed in the class. The texts were printed in purple ink, with the old multicopiers that we called *"Vietnamese copiers"* and that were so often used to (illegally) print pamphlets in our past revolutionary days in Spain against Franco's regime.

It is surprising that the first-order completeness proof that Henkin explained in class was not his own but was developed by using Herbrand's theorem and

*This research has been possible thanks to the research project sustained by **Ministerio de Ciencia e Innovación** of Spain with reference FFI _2009 _09345MICINN.

the completeness of propositional logic. In what follows I will summarize the proof, but remaining true to the spirit of Henkin's purple notes.

Theorem 1 (Herbrand's Theorem) *For each first-order sentence A there exist an (infinite) set of sentences of propositional logic Ψ such that: $\vdash A$ in FOL iff there is some $H \in \Psi$ such that $\vdash H$ in PL ($\vdash_{PL}$means that we just use sentential axioms and detachment).*

The above result can be regarded as a special case of the following

Theorem 2 *Let L be a first order language: We can extend L to L' by adjoining a set $\mathcal{C}$ of individual constants, and we can effectively give a set Δ of sentences of L' with the following property: For any set of sentences $\Gamma \cup \{A\} \subseteq Sent(L)$*

$$\Gamma \vdash A \textit{ iff } \Gamma \cup \Delta \vdash_{PL} A$$

Proof. In the first place, we build a set Δ where

$$\Delta = \Delta_1 \cup \Delta_2 \cup \Delta_3$$

Δ_1 consists of the sentences $\exists x_i B \to B(c_{i,B})$ (for each $\exists x_i B \in Sent(L')$). Δ_2 consists of various formal axioms for quantifiers (from first order logic), and Δ_3 consists of various formal axioms for the equality symbol (if there is one in the language L, otherwise is $\varnothing$).

In the spirit of Herbrand's theorem, an effective method of transforming any given derivation of A from $\Gamma \cup \Delta$ in PL into a formal derivation of A from Γ in FOL was given, which solves half of the theorem.

$$\Gamma \cup \Delta \vdash_{PL} A \text{ implies } \Gamma \vdash A$$

As for the other half,

$$\Gamma \vdash A \text{ implies } \Gamma \cup \Delta \vdash_{PL} A$$

suppose now that we do not have $\Gamma \cup \Delta \nvdash_{LP} A$. Then if we use completeness of propositional logic, $\Gamma \cup \Delta \not\models_{PL} A$ and we conclude that there is an assignment g for atoms of L'_0 that extends to an interpretation $\Im$ such that $\Im(A) = F$ but $\Im(\Gamma \cup \Delta) = T$.
In order to prove the theorem, from this interpretation $\Im$ we obtain a first order structure $\mathcal{A}$ such that $\mathcal{A} \models \Gamma$ but $\mathcal{A} \not\models A$ and so, $\Gamma \not\models A$.
By soundness of first order logic, $\Gamma \nvdash A$. ■

Predicate logic: Reduction to sentential logic Using the previous theorem we effectively reduce the completeness problem for first order logic to that of sentential logic. To this effect the following proposition was proved.

Proposition 3 *Theorem 2 and completeness of PL implies completeness of FOL.*

Note that a proof of the kind described above, provides a completeness proof for first order logic. For the theorem shows

$$\Gamma \nvdash A \text{ implies } \Gamma \cup \Delta \nvdash_{PL} A$$

On the other hand, using the structure $\mathcal{A}$ we show that

$$\Gamma \cup \Delta \nvdash_{PL} A \text{ implies } \Gamma \not\models A$$

Therefore, $\Gamma \models A$ implies $\Gamma \vdash A$, which is completeness for first order logic.

Another completeness proof he also developed in class was his result based on Craig's interpolation theorem [7].

2 His renowned proofs of completeness

The theorem of completeness establishes the correspondence between deductive calculus and semantics. Gödel had solved it positively for first-order logic and negatively for any logical system able to contain arithmetic. The lambda calculus for the theory of types [3], with the usual semantics over a standard hierarchy of types, was able to express arithmetic and hence could only be incomplete. Henkin showed that if the formulae were interpreted in a less rigid way, accepting other hierarchies of types that did not necessarily have to contain all the functions but at least the definable ones, it is easily seen that all consequences of a set of hypotheses are provable in the calculus. The valid formulae with this new semantics, called general semantics, are reduced to coincide with those generated by the rules of calculus.

As is well known, Henkin's completeness theorem rests on the proof that every consistent set of formulae has a model. The proof is essentially given in two parts:

1. The consistent set is extended to one that is maximally consistent and contains witnesses.

2. The model that the formulae of this maximally consistent set describe is constructed, since a maximally consistent set is a detailed description of a structure.

Surprisingly, the model uses the expressions themselves as objects; in particular their elements are equivalence classes of expressions, the equivalence relationship being that of formal derivability of equality.

It is interesting to note how he discovered it[1].

Completeness in the theory of types [5] Church created the lambda calculus using functions as the basic concept, and he made a clear distinction between the value of a function for a given argument and the function itself. Functions and function values have proper names in lambda calculus. This language attracted Henkin very much as well as the hierarchy of types.

[1] We strongly recomend reading [8] and [1].

Hierarchy of types The types are structured in a hierarchy that has the following as basic types:

- $\mathcal{D}_1$ is a non-empty set; that of individuals of the hierarchy
- $\mathcal{D}_0$ is the domain of truth values (since we are in binary logic, these values are reduced to T and F)
- The other domains are constructed from the basic types as follows: if $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ have already been constructed, we define $\mathcal{D}_{(\alpha\beta)}$ as the domain formed by all the functions from $\mathcal{D}_\beta$ to $\mathcal{D}_\alpha$.

To talk about this hierarchy, a formal language is introduced.
Henkin says:

> I decided to try to see just which objects of the hierarchy of types did have names in T.
>
> (Henkin 1996, p. 146)
>
> As I struggled to see the action of functions more clearly in this way, I was struck by the realization that I have used λ-conversion, one of the formal rules of inference in Church's deductive system for the language of the Theory T. All my efforts had been directed toward interpretations of the formal language, and now my attention was suddenly drawn to the fact that these were related to the formal deductive system for that language.
>
> (Ibid, p. 150)

The hierarchy described above is standard, and the completeness result depended on accepting other hierarchies of types that did not necessarily have to contain all the functions but at least the definable ones.

He developed the hierarchy of the objects that have a name in the hierarchy of types and realised that in order to remove repetitions he was using the rules of lambda-conversion and that the syntax and calculus were therefore involved in the description of the semantic objects. In particular, to identify the objects named by means of M^α and N^α he was using calculus theorems as a criterion: in particular, that $\vdash M^\alpha = N^\alpha$.

> In particular, I saw that using the symbol $\vdash$ for formal provability (or derivability) as usual, we can define for each type symbol α a domain $\mathcal{D}'_\alpha$

in which each sentence of the formal language acquires denotation, each functional expression denotes a function of the hierarchy and the deductively equivalent expressions denote the same object.

Finally, the proof was completed when he realized that for the universe of the objects named by propositions (that of truth values) to be reduced to only two it would be necessary to expand the axioms until they formed a maximally consistent set, so that the number of equivalence classes induced by the relation $\vdash M^\alpha = N^\alpha$ would be reduced to two.

In particular, his main theorem reads:

Theorem 4 *If Λ is any consistent set of cffs (sentences), there is a general model (in which each domain $\mathcal{D}_\alpha$ of $\mathcal{M}$ is denumerable) with respect to which Λ is satisfiable.*

To prove this theorem the set Λ is extended to a maximal consistent set which serves both as an oracle and as building bricks for the model. Specifically, to identify objects named by using M^α and N^α he made use of a criterion based on the calculus, in particular the fact that $\vdash M^\alpha = N^\alpha$.

How does Henkin construct type hierarchies? On page 86 of *Completeness in the Theory of Types* he says this:

> We now define by induction on a a frame of domains $\{D_a\}$ and simultaneously a one-to-one mapping Φ of equivalence classes onto the domains D_a such that $\Phi([\alpha_a])$ is in D_a.

The completeness of the first order functional calculus [4] Surprisingly, he obtained the proof of completeness of first-order logic later , readapting the argument found for the theory of types. Another interesting aspect that Henkin himself pointed out is the non-constructive nature of the proof, despite coming from a tradition as tightly bound to proofs with a constructive nature as those developed by Church.

An Extension of the Craig-Lyndon Interpolation Theorem [7] In 1963 Henkin published the paper *An Extension of the Craig-Lyndon Interpolation Theorem* where we can find a different proof of completeness for first order logic. Craig had shown the following theorem:

Theorem 5 *If A and C are any formulas of predicate logic such that $A \vdash C$, then there is a formula B such that (i) $A \vdash B$ and $B \vdash C$, and (ii) each predicate symbol occurring in B occurs both in A and in C.*

Henkin recalls that due to the fact that the relations $\vdash$ and $\models$ coincide in extension (by the strong completeness theorem), the above theorem is also valid if we replace the syntactic notion of derivability by the semantical notion of consequence. However, his idea was to obtain completeness from a slightly modified version of Craig's theorem.

> Notice, however, that if we alter Craig's theorem by replacing the symbol "$\vdash$" with "$\models$" in the hypothesis, but leaving "$\vdash$" unchanged in condition (i) of the conclusion, then the resulting proposition yields the completeness theorem as an immediate corollary.

The main theorem to be proves is:

Theorem 6 *Let Γ and Δ any sets of nnf's (negation normal formula) such that $\Gamma \models \Delta$. There is a nnf B such that (i) $\Gamma \vdash B$ and $B \vdash \Delta$, and (ii) any predicate symbol with a positive or negative occurrence in B has an occurrence of the same sign in some formula of Γ and in some formula of Δ.*

The strong completeness theorem is implied by the previous one.

The proof of the theorem is done by contraposition and to arrive to the conclusion that $\Gamma \not\models \Delta$ Henkin inductively builds two sets of sentences and define a model based on them using the technique he himself developed in his classical completeness proof [5].

3 Two results based on Henkin's ideas

Let us highlight how Henkin's *general models* are related to the *theory of representation*, or in other words: the *correspondence theory* and *non-standard models*. A more detailed examination of this can be consulted in the article by Manzano entitled *"Divergencia y rivalidad entre Lógicas"* [10] or in her book *Extensions of First Order Logic* [9]. Currently, the proliferation of logics used in Philosophy, Informatics, Linguistics and Mathematics make it crucial to achieve an operative reduction for all of them. We attribute most of the ideas handled in the reduction to many-sorted logic *[9]* to two articles by Henkin: *"Completeness in the theory of types"* from 1950, and the one from 1953, *"Banishing the rule of substitution for functional variables"*. Nevertheless, with all the foregoing we do not wish to deceive possible readers. In the article from 1950, there are no translations of formulae, and the language and many-sorted calculus do not even appear explicitly. Regarding higher-order logic, as far as is known many-sorted calculus appears for the first time in the 1953 article. In it, Henkin proposes the axiom of comprehension as an alternative to the substitution rule used in the calculuses previously proposed for higher-order logic. If the axiom of comprehension is removed from this calculus, one obtains the MSL calculus. There is also another idea —this time from the 1953 article— that is also interesting and goes as follows: If we weaken the axiom of comprehension (for example, we restrict it to first-order formulae or to translations of dynamic or modal formulae or to any other recursive set), we obtain calculi in between MSL and SOL. And it is easy to find their corresponding semantics. Naturally, the class of structures corresponding to them will be situated in between $\mathcal{F}$ and $\mathcal{GS}$. The new logic, let us call it XL, will also be complete. The reason is because this class of models is axiomatizable.

Completeness, translations, logicality and representation theorems It is argued in [9] 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. In the following we are presenting translations as the path to completeness, in three stages[2].

[2] See [11] for a detailed presentation.

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, the corresponding many-sorted signature, language and logic.

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

Theorem 7 (Representation Theorem) *There is a recursive set of L^*-sentences Δ, with $\mathcal{S}^* \subseteq Mod(\Delta)$ and such that*

$$\models_{Str(XL)} \varphi \text{ in } XL \text{ iff } \Delta \models_{Str(MXL^*)} \forall\text{TRANS}(\varphi) \text{ in } MSL$$

for every sentence φ of the logic XL.

From the previous result the enumerability theorem for this logic is straightforward; namely, Val(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.

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.

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

$$\Pi \models_{\mathcal{S}} \phi \quad \textit{iff} \quad \text{TRANS}(\Pi) \cup \Delta \models_{Str(MXL^*)} \text{TRANS}(\phi)$$

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

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.

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

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

Theorem 9 (Deductive correspondence) *Let Δ be defined as in the Main Theorem. Then*

$$\Pi \vdash_{Cal(XL)} \phi \quad \textit{iff} \quad \text{TRANS}(\Pi) \cup \Gamma \vdash_{Cal(MSL)} \text{TRANS}(\phi)$$

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 10 (Soundness and Completeness of XL)

$$\Pi \vDash_{\mathcal{S}} \phi \;\; \textit{iff} \;\; \Pi \vdash_{\mathrm{Cal(XL)}} \phi$$

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.

Completeness in Hybrid Type Theory In [2] a Basic Hybrid Type Theory is introduced. The goal of this paper is to investigate whether basic hybridization also leads to simple Henkin-style completeness proofs in the setting of (classical) higher-order modal logic (that is, modal logics built over Church's simple theory of types [3]), and as we shall show, the answer is "yes". The crucial idea is to use $@_i$ as a rigidifier for arbitrary types. We shall interpret $@_i\alpha_a$, where α_a is an expression of any type a, to be an expression of type a that rigidly returns the value that α_a receives at the i-world. As we show, this enables us to construct a description of the required model inside a single MCS and hence to prove (generalized) completeness for higher-order hybrid logic.

We now come back to Henkin's crucial idea for taming higher-order logic. The standard semantics (ignore for the moment the modal and hybrid components) is the usual semantics for higher-order logic and it is logically intractable: if we define validity as truth in all standard structures, we have a complex (indeed, provably unaxiomatizable) notion of validity. His notion of general interpretations simultaneously lowers the logical complexity of validity (as there are more general structures than standard ones, it is, so to speak, easier for a formula to be falsified, and indeed, higher-order validity becomes recursively enumerable) and makes clear just why those plausible looking axiomatizations were so plausible: they are complete with respect to Henkin's general semantics.

Our completeness theorem is essentially an adaptation of Henkin's hierarchy construction, using the rigidity and truth equivalence classes introduced at the end of the previous section.

References

[1] Alonso, E and Manzano, M. *"Completeness: from Gödel to Henkin".* To appear.

[2] Areces, C., Blackburn, P., Huertas, A. and Manzano, M. *"Completeness in Hybrid Type Theory".* To appear.

[3] Church, A. [1940]. *"A formulation of the simple theory of types"*. **The Journal of Symbolic Logic.** vol. 5, pp. 56-68.

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

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

[6] Henkin, L. [1953]. *"Banishing the Rule of Substitution for Functional Variables"*. **The Journal of Symbolic Logic.** 18(3): 201-208.

[7] Henkin, L. [1963]. *"An Extension of the Craig-Lyndon Interpolation Theorem"*. **The Journal of Symbolic Logic.** 28(3): 201-216.

[8] Henkin, L. [1996]. "*The discovery of my completeness proofs"*, Dedicated to my teacher, Alonzo Church, in his 91^{st} year, **Bulletin of Symbolic Logic***,* vol. 2, Number 2, June 1996. (presentado el 24 de Agosto de 1993 en el XIX International Congress of History of Science, Zaragoza, Spain).

[9] Manzano, M. [1996]. *Extensions of First Order Logic***.** Cambridge Tracts in Theoretical Computer Science. Cambridge: Cambridge University Press.

[10] Manzano, M. [2004]. *"Divergencia y rivalidad entre lógicas."* En **Enciclopedia Iberoamericana de Filosofía.** Volumen 27 de Filosofía de la Lógica. Raúl Orayen y Alberto Moretti eds. Editorial Trotta. España.

[11] Manzano, M. and Urtubey, L. *"Completeness, translations, logicality and representation theorems".* To appear.