Logica Trianguli, 3, 1999, 85-109

COMPLETENESS AND REPRESENTATION THEOREM FOR EPISTEMIC STATES IN FIRST-ORDER PREDICATE CALCULUS

Serge LAPIERRE, François LEPAGE

Abstract

The aim of this paper is to present a strongly complete first order functional predicate calculus generalized to models containing not only ordinary classical total functions but also arbitrary partial functions. The completeness proof follows Henkin's approach, but instead of using maximally consistent sets, we define saturated deductively closed consistent sets (SDCCS). This provides not only a completeness theorem but a representation theorem: any SDCCS defines a canonical model which determine a *unique partial value* for every predicate symbol and any function symbol. Any SDCCS can thus be interpreted as an epistemic state.

1. Introduction

The aim of this paper is to present a proof *à la* Henkin [1] of strong completeness for functional predicate calculus with identity generalized to models containing not only total functions but arbitrary partial functions as well.

These partial functions are arbitrary because the domains contain functions of an arbitrary degree of definition, that is, totally undefined *n*-ary functions and predicates and totally defined *n*-ary functions and predicates, as well as all the functions having an intermediate degree of definition. The resultant logic could have turned out to be trivial, with an empty class of sentences valid for partial interpretations — i.e., with an empty class of sentences true for the interpretation where every function and every predicate is totally undefined.

This, as we shall see, is not the case, as we adopt what may be called a Quinean point of view: functions and predicates can be arbitrarily undefined, but quantification (the ontological commitment) applies only to the defined objects. One of the characteristic of our logic is that sentences such as “(x)($x = x$)” are valid, whereas, contrary to free logics, those of the form “($a = a$)” or more generally “($f^n t_1 \dots t_n = f^n t_1 \dots t_n$)” are not (even though they are never false).

Our approach is classical; we present the syntax, that of the familiar first-order functional calculus, followed by the semantics. We then introduce a system of sequents and show that it is complete.

The development of the proof is therefore relatively straightforward, except where certain traits peculiar to the partial aspect of the interpretation are concerned.

One of these traits concerns the interpretation of the partial character itself. As we will show, a partial function will be understood here to be a function that is not totally defined. Therefore, the level of definition of every function could be made total. We will see that this property, once formalized, is expressed in the form of a general constraint of monotonicity relative to an order whose immediate interpretation is that of “level of definition”. Indeed, when a function takes a value, partial or total, for an argument that is not totally defined, the value must be compatible with all the values the function takes for better-defined arguments. The underlying idea in defining partial models is to represent epistemic states: a non-omniscient agent can be represented by a partial model which is an approximation of the intended model.

2. Syntax

Primitive symbols:

1. Individual variables $x, y, z, x_1, y_1, z_1, \dots, x_2, y_2, z_2, \dots$
2. For every $n \geq 1$, n-place predicate symbols $P^n_0, P^n_k, \dots$
3. For every $n \geq 1$, n-argument function symbols $f^n_0, \dots, f^n_k, \dots$
4. Logical constants ‘=’, ‘¬’, ‘ ’, ‘ ’.
5. Parentheses ‘(’, ‘)’.

Definition of terms:

(i) Every individual variable is a term;
(ii) if f^n is an n-arguments function symbol and $t_1, \dots, t_n$ are terms, then $f^n t_1 \dots t_n$ is a term;

(iii) nothing else is a term.

Definition of formulae:

(i) if t and t' are terms, then $(t = t')$ is a formula;

(ii) if P^n is an n-places predicate symbol and $t_1,\ldots, t_n$ are terms, then $P^n t_1\ldots t_n$ is a formula;

(iii) if A is a formula, $\neg A$ is a formula;

(iv) if A and B are formulae, then $(A \quad B)$ is a formula;

(v) if A is a formula and x is a variable, then $\ xA$ is a formula;

(vi) nothing else is a formula.

Abbreviations:

$(A \quad B) := (\neg A \quad B)$

$(A \quad B) := \neg(\neg A \quad \neg B)$

$(A \equiv B) := ((A \quad B) \quad (B \quad A))$

$\ xA := \neg \ x\neg A$

$T := \ x(x = x)$

$F := \neg T$

$\ (A) := ((A \equiv T) \quad (A \equiv F))$

$\ (t) := \ x(t = x)$, where x does not appear in t.

$\ (f^n) := \ x_1\ldots \ x_n \ (f^n x_1 \ldots x_n)$, where $x_1,\ldots, x_n$ are distinct variables.

$\ (P^n) := \ x_1\ldots \ x_n \ (P^n x_1 \ldots x_n)$, where $x_1,\ldots, x_n$ are distinct variables.

The last four abbreviations serve to introduce a functor whose interpretation will, in its positive component, be “is totally defined”. We will return to this question after having introduced the semantics, to which we now turn.

3. Semantics[1]

A *model* is an ordered pair $< E \quad \{ \ _e\}, g>$ satisfying the following properties.

1. E is a non-empty set (a set of individuals) and $\ _e$ is the undefined object. The undefined object is an artefact that proves to be of great utility since it allows us to treat expressions that are defined and those

[1] For a general presentation of the notion of partial functions in terms of monotonic functions, see [3].

that are not in a uniform way: for a term, to be undefined will simply be to have $_e$ for value. The introduction of an undefined object makes it necessary, moreover, to introduce an order which we will designate as "⊩" and whose interpretation will be "$a \Vdash b$ if and only if a is less defined or equal to b". We have, formally, that for all a, b E $\{ {}_e\}$, $a \Vdash b$ if and only if either $a = {}_e$ or $a = b$. E $\{ {}_e\}$ is therefore a flat meet semi lattice such that all the elements of E strictly dominate $_e$ and are incomparable to each other.

2. g is a function such that:
(i) if P^n is an n-place predicate symbol, then $g(P^n)$ is a monotonic function from $(E \quad \{ {}_e\})^n$ into $\{0, 1, \quad \}$ (the false, the true and the undefined);
(ii) if f^n is an n-argument function symbol, then $g(f^n)$ is a monotonic function from $(E \quad \{ {}_e\})^n$ into $E \quad \{ {}_e\}$.

is the undefined truth-value and plays a role similar to $_e$, that is to say, one has $\Vdash$, $\Vdash 1$, $\Vdash 0$, $1 \Vdash 1$ and $0 \Vdash 0$. Relative to "⊩", monotonicity is defined in the following way:

$g(f^n)$ is monotonic for its ith place if and only if for all $a_1,\ldots, a_n$ E $\{ {}_e\}$, if $a_i \Vdash a'_i$ then $g(f^n)(<a_1,\ldots, a_i ,\ldots, a_n >) \Vdash g(f^n)(<a_1,\ldots, a'_i ,\ldots, a_n >)$
and
$g(P^n)$ is monotonic for its ith place if and only if for all $a_1,\ldots, a_n$ E $\{ {}_e\}$, if $a_i \Vdash a'_i$ then $g(P^n)(<a_1,\ldots, a_i ,\ldots, a_n >) \Vdash g(P^n)(<a_1,\ldots, a'_i ,\ldots, a_n >)$.

A function is monotonic if it is monotonic for all its places. The constraint of monotonicity is a consequence of the interpretation of and $_e$ as undefined elements: a function that takes a certain value for some argument must take values at least as defined for more defined arguments, that is to say that the definedness of the values of the function increases with the definedness of the arguments.

Given a model M = $<E \quad \{ {}_e\}, g>$, an *assignment* on M is a function μ that assigns an element of E $\{ {}_e\}$ to each variable. Given an element a E $\{ {}_e\}$ and an individual variable x, μ(a/x) is that assignment on M that differs at most from μ by assigning the value a to x.

Let $M = \langle E \quad \{ {}_e\}, g\rangle$ be a model and let μ be an assignment on M. For each term *t*, the *value* $||t||_{M,\mu}$ in $E \quad \{ {}_e\}$ according to M and μ is recursively defined as follows:

(i) $||x||_{M,\mu} = \mu(x)$ for every variable *x*;
(ii) $||f^n(t_1, \ldots, t_n)||_{M,\mu} = g(f^n)(<||t_1||_{M,\mu}, \ldots, ||t_n||_{M,\mu}>)$.

For each formula *A*, the *value* $||A||_{M,\mu}$ in $\{0, 1, \quad\}$ according to M and μ is defined recursively as follow:

$$\text{(i) } ||(t = t')||_{M,\mu} = \begin{cases} 1 \text{ iff } ||t||_{M,\mu} = ||t'||_{M,\mu} \neq {}_e \\ 0 \text{ iff } ||t||_{M,\mu} \neq {}_e, ||t'||_{M,\mu} \neq {}_e \text{ and } ||t||_{M,\mu} \neq ||t'||_{M,\mu} \\ \quad \text{otherwise} \end{cases}$$

$$\text{(ii)} \qquad ||P^n t_1, \ldots, t_n||_{M,\mu} = g(P^n)(<||t_1||_{M,\mu}, \ldots, ||t_n||_{M,\mu}>)$$

$$\text{(iii)} \qquad ||\neg A||_{M,\mu} = \begin{cases} 1 \text{ iff} ||A||_{M,\mu} = 0 \\ 0 \text{ iff} ||A||_{M,\mu} = 1 \\ \quad \text{otherwise} \end{cases}$$

$$\text{(iv)} \qquad ||(A \quad B)||_{M,\mu} = \begin{cases} 1 \text{ iff} ||A||_{M,\mu} = 1 \text{ or } 1||B||_{M,\mu} = 1 \\ 0 \text{ iff } ||A||_{M,\mu} = 0 \text{ and } ||B||_{M,\mu} = 0 \\ \quad \text{otherwise} \end{cases}$$

$$\text{(v)} \qquad || \; xA||_{M,\mu} = \begin{cases} 1 \text{ iff there is an} a \quad E \text{ such that } ||A||_{M,\mu(a/x)} = 1 \\ 0 \text{ iff for every} a \quad E, \; ||A||_{M,\mu(a/x)} = 0 \\ \quad \text{otherwise} \end{cases}$$

As it has been previously mentioned, the non-trivial character of partial logic follows from the interpretation of the quantifier, which covers only defined values.

For instance, we see that the expression (*t*) (an abbreviation of $x(t = x)$, where *x* does not appear in *t*) will be true if and only if *t* is defined and undefined if *t* is undefined, but will never be false. So we can express, in the object language, the fact that a term is defined, but we cannot express the fact that a term is undefined: the expression (*t*) is either true or undefined. In fact, as we said earlier, it is impossible to introduce a functor *H* whose interpretation would be $||H(t)||_{M,\mu} = 1$ if and only if $||t||_{M,\mu} = \ _e$, for such a functor is not monotonic.

We leave to the reader the exercise of verifying that other expressions such as (*A*), (f^n) and (P^n) fulfill the required role, namely, of being true if and only if the value of the expression is totally defined and undefined otherwise.

One can easily verify that definitions (i)-(v) are *maximal* in the following sense: any stronger definition of $|| \; ||_{M,\mu}$ (i.e., that provides more 0 or more 1) is non-monotonic.

Definition of the notion of validity[2]

Given a set Γ of formulae and a formula *A*, we define the notions of *logical consequence* and of *valid formula* in the following manner:

- $\Gamma \models A$ if and only if for every model M and every assignment μ on M, if $||B||_{M,\mu} = 1$ for every $B \quad \Gamma$, then $||A||_{M,\mu} = 1$.

- $\models A$ if and only if $\emptyset \models A$.

Notational conventions and definitions

- External parentheses of formulae may be omitted.
- An *occurrence* of a term *t* in a formula is *free* if and only if all occurrences of all the variables in *t* are free.
- $A(t'/t)$ designates the formula obtained by replacing every free occurrence of *t* in *A* by an occurrence of *t'*.
- $t(t'/x)$ designates the term obtained by replacing every occurrence of *x* in *t* with an occurrence of *t'*.

[2] For a more general study of the notion of validity for the domains of partial functions, see [2].

4. System

The rules of the system can be grouped according to their "meaning". The first group includes the two general rules of the deductive system.

Group 1

R: If $A \quad \Gamma$, then $\Gamma \vdash A$

T: $$\frac{\Gamma \vdash A}{\Gamma \quad \Delta \vdash A}$$

The second group concerns the behaviour of identity.

Group 2

ID1: $$\frac{\Gamma \vdash \quad (t)}{\Gamma \vdash t = t}$$

ID2: $$\frac{\Gamma \vdash t = t' \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

ID3: $$\frac{\Gamma \vdash \quad (t = t')}{\Gamma \vdash \quad (t) \qquad (t')}$$

ID4: $$\frac{\Gamma \vdash \quad (t) \qquad (t')}{\Gamma \vdash \quad (t = t')}$$

ID5: $\vdash \quad x(x = x)$

where in ID2, B is the result of replacing some free occurrences of t in A by occurrences of t' and these occurrences are free in B.

The fifth rule ensures that there is at least one object which is the value of a variable and thereby excludes the trivial model in which the "undefined" would be the only possible value for variables.

The third group imposes the constraints of monotonicity.

Group 3

M1: $$\frac{\Gamma \vdash \quad (P^n t_1, \ldots, t_i, \ldots, t_n)}{\Gamma \vdash \quad (t_i) \quad (P^n t_1, \ldots, t_i, \ldots, t_n \equiv P^n t_1, \ldots, t'_i, \ldots, t_n)}$$

M2: $$\frac{\Gamma \vdash \quad (f^n t_1, \ldots, t_i, \ldots, t_n)}{\Gamma \vdash \quad (t_i) \quad (f^n t_1, \ldots, t_i, \ldots, t_n = f^n t_1, \ldots, t'_i, \ldots, t_n)}$$

where t'_i is any term.

The fourth group concerns Boolean terms.

Group 4

BOOL1: $$\frac{\Gamma \vdash \quad (A)}{\Gamma \vdash \quad (\neg A)}$$

BOOL2: $$\frac{\Gamma \vdash \quad (\neg A)}{\Gamma \vdash \quad (A)}$$

BOOL3a: $$\frac{\Gamma \vdash \quad (A) \quad \Gamma \vdash \quad (B)}{\Gamma \vdash \quad (A \quad B)}$$

BOOL3b: $$\frac{\Gamma \vdash \neg (A \quad B)}{\Gamma \vdash \quad (A) \qquad (B)}$$

BOOL4a: $$\frac{\Gamma \vdash \quad x \ (A)}{\Gamma \vdash \quad (\ xA)}$$

BOOL4b: $$\frac{\Gamma \vdash \neg \ xA}{\Gamma \vdash \quad x \ (A)}$$

BOOL5: $$\frac{\Gamma \vdash \quad x(A \quad B)}{\Gamma \vdash A \qquad xB}$$

where in BOOL5, x has no free occurrence in A.

Finally, the fifth and last group concerns the logical connectives. Note that these are classical, except I, ¬ I, I and E.

Group 5

I: $\dfrac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \quad B}$ E: $\dfrac{\Gamma \vdash A \quad B}{\Gamma \vdash A}$ $\dfrac{\Gamma \vdash A \quad B}{\Gamma \vdash B}$

I: $\dfrac{\Gamma \vdash A}{\Gamma \vdash A \quad B}$ $\dfrac{\Gamma \vdash B}{\Gamma \vdash A \quad B}$

E: $\dfrac{\Gamma \quad \{A\} \vdash C \quad \Gamma \quad \{B\} \vdash C \quad \Gamma \vdash A \quad B}{\Gamma \vdash C}$

I: $\dfrac{\Gamma \quad \{A\} \vdash B}{\Gamma \quad \{\ (A)\} \vdash A \quad B}$ E: $\dfrac{\Gamma \vdash A \quad \Gamma \vdash A \quad B}{\Gamma \vdash B}$

¬ I: $\dfrac{\Gamma \quad \{A\} \vdash B \quad \Gamma \quad \{A\} \vdash \neg B \qquad \Gamma \vdash \ (A)}{\Gamma \vdash \neg A}$

¬ E: $\dfrac{\Gamma \vdash A \quad \Gamma \vdash \neg A}{\Gamma \vdash B}$

I: $\dfrac{\Gamma \quad \{\ (t)\} \vdash A}{\Gamma \vdash \quad xA(x/t)}$ E: $\dfrac{\Gamma \vdash \quad xA}{\Gamma \quad \{\ (t)\} \vdash \mathrm{A}(t/x)}$

where in I, neither *x*, nor any variable of *t* occurs freely in any member of Γ and *x* is free for every variable of *t* in *A*.

I: $\dfrac{\Gamma \vdash A(t/x)}{\Gamma \vdash \quad xA}$ E: $\dfrac{\Gamma \vdash \quad xA \qquad \Gamma \qquad \{A(t/x)\} \vdash B}{\Gamma \vdash B}$

where in E, no variable of t occurs freely either in B, or in A, or in any member of Γ and t is free for x in A.

The following provable derived rules will be useful for the completeness proof.

R1: $\dfrac{\Gamma \vdash A \qquad \Gamma \qquad \{A\} \vdash B}{\Gamma \vdash B}$ V: $\dfrac{\Gamma \vdash \quad xA}{\Gamma \vdash \quad yA(y/x)}$

where y does not occur freely in A and is free for x in A.

T1: $\vdash T$ R2: $\dfrac{\Gamma \vdash A}{\Gamma \vdash \quad (A)}$

R3: $\dfrac{\Gamma \vdash A \equiv B \quad \Gamma \vdash A}{\Gamma \vdash B}$ $\dfrac{\Gamma \vdash A \equiv B \quad \Gamma \vdash B}{\Gamma \vdash A}$

R4: $\dfrac{\Gamma \vdash t = t'}{\Gamma \vdash \quad (t\,)}$ R5: $\dfrac{\Gamma \vdash t = t'}{\Gamma \vdash t' = t}$ R6: $\dfrac{\Gamma \vdash t = t'}{\Gamma \vdash \quad (t')}$

R7: $\dfrac{\Gamma \vdash \quad A}{\Gamma \vdash \quad \neg\neg\, A}$ T2: $\vdash F \quad A$ R8: $\dfrac{\Gamma \vdash \neg A}{\Gamma \vdash A \equiv F}$

R9: $\dfrac{\Gamma \vdash A \quad B}{\Gamma \quad \{A\} \vdash B}$ R10: $\dfrac{\Gamma \vdash A \equiv F}{\Gamma \vdash \neg A}$ R11: $\Gamma \quad \{A \equiv T\} \vdash A$

R12: $\Gamma \quad \{A \equiv F\} \vdash \neg A$ R13: $\dfrac{\Gamma \vdash \quad (A)}{\Gamma \vdash A \quad \neg A}$ R14: $\dfrac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A \quad \neg A}$

R15: $\dfrac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}$ R16: $\dfrac{\Gamma \vdash \neg A \quad \neg B}{\Gamma \vdash \quad (A \quad B)}$ R17: $\dfrac{\Gamma \vdash A \quad B}{\Gamma \vdash \neg\neg A \quad \neg\neg B}$

R18: $\dfrac{\Gamma \vdash \neg A \quad \neg B}{\Gamma \vdash \neg(A \quad B)}$ R19: $\dfrac{\Gamma \vdash \quad x\mathrm{A}}{\Gamma \vdash \quad x\neg\neg A}$ R20: $\dfrac{\Gamma \vdash \quad x\neg \mathrm{A}}{\Gamma \vdash \neg \quad xA}$

R21: $\dfrac{\Gamma \vdash \neg\neg A \quad \neg\neg B}{\Gamma \vdash A \quad B}$ R22: $\dfrac{\Gamma \vdash \neg(A \quad B)}{\Gamma \vdash \quad (\neg\neg A \quad \neg\neg B)}$

R23: $\dfrac{\Gamma \vdash \neg(A \quad B)}{\Gamma \vdash \quad \neg A \quad \neg B}$ R24: $\dfrac{\Gamma \vdash \quad x\neg\neg \mathrm{A}}{\Gamma \vdash \quad xA}$ R25: $\dfrac{\Gamma \vdash \neg \quad xA}{\Gamma \vdash \quad x\neg \mathrm{A}}$

5. Completeness

From this point forward, our demonstration of completeness follows the classical method introduced by Henkin. It is nevertheless made more complicated than Henkin's proof for one thing by the absence of the excluded-middle: a sentence that is not true can be false or undefined. The strategy, therefore, will not be to construct, from some consistent set, a maximally consistent set in the sense that the addition of any foreign formula to the set breaks the consistency. We clearly need a weaker notion of maximal extension. It is necessary to construct, from any consistent set, a consistent set which contains all its valid consequences, and thus all the valid consequences of the initial set. But since the intended interpretation is partial, there will possibly be some formula A such that neither A nor $\neg A$ belong to the extension. The stages of the proof will be the following:

1) first, we will show that every consistent set of formulae can be extended to a saturated (as in [5] and [4]) for , , , deductively closed and consistent set (SDCCS);
2) we will then show that this set allows us to define a partial model and a partial valuation;
3) finally, we will show that for this partial model and valuation, and for every formula A, A receives the value 1 if and only if A is in the extension, A receives the value 0 if and only if $\neg A$ is in the extension, and otherwise A receives the undefined value.

Construction of an $\vee$-*,* $\exists$- *and* $\forall$*-saturated set from any consistent set*

Let Γ be a consistent set of formulae and let C be a formula such that $\Gamma \nvdash C$ (C is called the test formula). We will construct a set Γ' such that
(i) Γ Γ'
(ii) $\Gamma' \nvdash C$
(iii) for every formula A, $\Gamma' \vdash A$ if and only if A Γ'.
(iv) Γ' is -*saturated,* i.e., if $\Gamma' \vdash A$ B, then $\Gamma' \vdash A$ or $\Gamma' \vdash B$
(v) Γ' is -*saturated,* i.e., if $\Gamma' \vdash$ xA, then $\Gamma' \vdash A(y/x)$ for some variable y.
(vi) Γ' is -*saturated,* i.e., if $\Gamma' \nvdash$ xA, then $\Gamma' \nvdash A(y/x)$ for some variable y such that $\Gamma' \vdash$ (y).

We will then show the truth-lemma itself: there is a valuation such that A Γ' if and only if A is satisfied by that valuation, which gives us strong completeness.

We assume that we have an enumeration Δ of all formulae of the language, such that each formula appears countably many times in it, and we designate by D_n the n-th formula according to this enumeration. We construct a sequence $<\Gamma_i>$ of nested sets in the following way.

1) $\Gamma_0 = \Gamma$;
2) if $\Gamma_{2n} \nvdash D_n$, then $\Gamma_{2n+2} = \Gamma_{2n+1} = \Gamma_{2n}$;
3) if $\Gamma_{2n} \vdash D_n$, then $\Gamma_{2n+1} = \Gamma_{2n}$ $\{D_n\}$ and
a) if $D_n = P^m t_1 \ldots t_m$, then $\Gamma_{2n+2} = \Gamma_{2n+1}$;
b) if $D_n = (t = t')$, then $\Gamma_{2n+2} = \Gamma_{2n+1}$;
c) if $D_n = \neg A$, then $\Gamma_{2n+2} = \Gamma_{2n+1}$;
d) (-*saturation*) if $D_n = (A$ $B)$, then
(i) if Γ_{2n+1} $\{A\} \nvdash C$, then $\Gamma_{2n+2} = \Gamma_{2n+1}$ $\{A\}$;

(ii) if Γ_{2n+1} $\{A\} \vdash C$, then $\Gamma_{2n+2} = \Gamma_{2n+1}$ $\{B\}$;
e) (-*saturation*) if $D_n =$ xA, then $\Gamma_{2n+2} = \Gamma_{2n+1}$ $\{A(y/x)\}$, where y is a new variable that does not appear in Γ_{2n+1}.

Let $\Gamma^* = \bigcup_n \Gamma_n$. Then:

1) Γ^* is closed for $\vdash$, that is to say $\Gamma^* \vdash A$ if and only if A Γ^*.
The proof is standard.

2) C Γ^*.
The proof is standard.

3) Γ^* is -saturated. This clearly follows from the condition d) in the construction of Γ^*.

4) Γ^* is -saturated. This clearly follows from the condition e) in the construction of Γ^*.

We will say that Γ^* is the maximal - and -saturated extension of Γ for the enumeration Δ and the test formula C.

Nothing in this construction guarantees that Γ^* is -saturated. In the classical case, this saturation is ensured by syntactic negacompletion: if $\Gamma^* \nvdash$ xA, then $\Gamma^* \vdash \neg$ xA. Therefore $\Gamma^* \vdash$ $x\neg A$ and $\Gamma^* \vdash \neg A(y/x)$ for some y and finally, Γ^* being consistent, $\Gamma^* \nvdash A(y/x)$. Such a derivation is not valid for partial interpretations. It could be the case that Γ^* $\nvdash$ xA and $\Gamma^* \nvdash \neg A(y/x)$: for every variable y, $A(y/x)$ will be true or undefined. To obtain the -saturation, we should ensure that the construction of the maximal extension is such that if xA is not derivable, then the extension admits at least one instanciation $A(y/x)$ which is not derivable either, without bringing in the derivability of $\neg A(y/x)$.

We will construct an -saturated set Γ' such that Γ^* Γ'. First, we are given:
a) an enumeration $Q_0,\ldots, Q_n,\ldots,$ of all the formulae of the form xA,
b) a denumerable sequence of new variables $x_0,\ldots, x_n, \ldots,$

c) a denumerable sequence of enumerations $\Delta^0, \ldots, \Delta^n, \ldots$, of formulae of the language obtained through the successive enrichment of the initial vocabulary with the variables $x_0, \ldots, x_n, \ldots$.

Let $<\Gamma'_n>$ be the sequence of -saturated and -saturated supersets of Γ^* defined as follows.

(i) $\Gamma'_0 = \Gamma^*$ and $C_0 = C$ (where C is the test formula that served in the construction of Γ^*).

(ii) If $\Gamma'_n \vdash Q_n$, then $\Gamma'_{n+1} = \Gamma'_n$ and $C_{n+1} = C_n$.

(iii) If $\Gamma'_n \nvdash Q_n$ (with $Q_n = \quad xA$), then let $\Gamma''_n = \Gamma'_n \quad \{ \quad (x_n)\}$.

We then have $\Gamma''_n \nvdash (C_n \quad A(x_n/x))$ since, by I we would have $\Gamma'_n \vdash \quad x_n\ (C_n \quad A(x_n/x))$, and, by BOOL5, $\Gamma'_n \vdash C_n \quad x_n A$, x_n not being free in C_n. By the induction hypothesis, $\Gamma'_n \nvdash C_n$, Γ''_n is -saturated and therefore $\Gamma'_n \vdash C_n$ or $\Gamma'_n \vdash \quad x_n A(x_n/x)$. Therefore $\Gamma'_n \vdash \quad x_n A(x_n/x)$. By rule V, $\Gamma'_n \vdash \quad xA$, which is contrary to the hypothesis.

So we define Γ'_{n+1} as the maximal -and -saturated extension of Γ''_n with $C_{n+1} = (C_n \quad A(x_n/x))$ as the test formula of consistence for the enumeration Δ^n.

Let $\Gamma' = \bigcup_n \Gamma'_n$.

Γ' is clearly consistent, -saturated, -saturated and -saturated.

Definition of the interpretation

We will now define a pre-order relation between the expressions of the language, the natural interpretation of which will be “is at most as defined as” relative to Γ'. This stage of the construction has no equivalent in the classical case. Indeed, in order to define an interpretation on the basis of Γ', we should define a relation of equivalence on the set of expressions of the language the interpretation of which is “to have the same level of definition”. Although this is relatively easy for the formulae, the fact that we cannot for reasons of monotonicity express that an expression is undefined in the object language somewhat complicates things. Fortunately, we can speak of the undefined in the metalanguage.

Definition 1

Let “ ” be the following relation:

1) if A and B are formulae, $A \quad B$ if and only if
if $A \quad \Gamma'$, then $B \quad \Gamma'$ and
if $\neg A \quad \Gamma'$, then $\neg B \quad \Gamma'$;

2) if t and t' are terms, $t \quad t'$ if and only if
if $(t = t) \quad \Gamma'$, then $(t = t') \quad \Gamma'$
that is to say that either $(t = t) \quad \Gamma'$ or $(t = t') \quad \Gamma'$;

3) if P^n and Q^n are two n-places predicate symbols, then $P^n \quad Q^n$ if and only if for all terms $t_1,\ldots,t_n$, $P^n t_1 \ldots t_n \quad Q^n t_1 \ldots t_n$.

4) if f^n and g^n are two n-arguments function symbols, then $f^n \quad g^n$ if and only if for all terms $t_1,\ldots,t_n$, $f^n t_1 \ldots t_n \quad g^n t_1 \ldots t_n$.

Definition 2

For all expressions α, β, $\alpha \approx \beta$ if and only if $\alpha \quad \beta$ and $\beta \quad \alpha$.

Lemma 1

“ ” is a pre-order relation (that is, it is reflexive and transitive).

Proof: trivial.

Lemma 2

“≈” is an equivalence relation.

Proof: trivial.

We designate by $C_{\approx}(\alpha)$ the equivalence class of the expression α.

For the canonical interpretation that we are building from Γ', all the expressions of one equivalence-class will be seen to take the same value. Now we will define domains M_i and a function which assigns an element of one of the domains to each equivalence-class $C_{\approx}(\alpha)$.

Construction of the domains M_i

We recursively define

a) a function K which assigns exactly one value to each equivalence class;

b) sets of values M_i such that M_1 will be the set of values of formulae, M_2 will be the set of values of terms, $M_{3n} \quad (M_2^n \to M_1)$ will be the set of value of n-places predicate symbols, i.e. M_{3n} will be a subset of the monotonic functions from M_2^n into M_1 and finally M_{4n} $(M_2^n \to M_2)$ will be the set of values of n-arguments function symbols, i.e. M_{4n} will be a subset of the monotonic functions from M_2^n into M_2;

and we show that:

c) for all expressions of the language α, β (formulae, terms, predicate symbols and function symbols), $K(C_{\approx}(\alpha)) \Vdash K(C_{\approx}(\beta))$ if and only if α β.

(i) For formulae

a) $K(C_{\approx}(A)) = 1$ iff A Γ'
$K(C_{\approx}(A)) = 0$ iff $\neg A$ Γ', and
$K(C_{\approx}(A)) =$ otherwise

From definitions 1 and 2 and lemma 2 it is obvious that $K(C_{\approx}(A)) = K(C_{\approx}(B))$ if and only if $A \approx B$, if and only if [A Γ' and B Γ'] or [$\neg A$ Γ' and $\neg B$ Γ'] or [A Γ' and B Γ' and $\neg A$ Γ' and $\neg B$ Γ']. So, $K(C_{\approx}(A))$ does not in any way depend on the particular formula A which is chosen;

b) $M_1 = \{0, 1, \ \}$ is the set of values of formulae[3].

c) One easily verifies that $K(C_{\approx}(A)) \Vdash K(C_{\approx}(B))$ if and only if A B.

(ii) For terms

a) $K(C_{\approx}(t)) = C_{\approx}(t)$. Clearly $K(C_{\approx}(t))$ does not in any way depend on the element t which is chosen.

b) $M_2 = \{ C_{\approx}(t) : (t = t)$ Γ'$\}$ $\{ {}_e\}$, where ${}_e = \{t : (t = t)$ Γ'$\}$.

Rule ID5 ensures that $M_2 \neq \{ {}_e\}$. Indeed, $x(x = x)$ Γ' and so by -saturation, $(y = y)$ Γ' for some y. The inclusion of empty models involves the usual inconveniences.

c) Let us verify that $K(C_{\approx}(t)) \Vdash K(C_{\approx}(t'))$ if and only if t t'

Suppose that $K(C_{\approx}(t)) \Vdash K(C_{\approx}(t'))$.

If $K(C_{\approx}(t)) = {}_e$, then $(t = t)$ Γ'. Thus by definition of , t t'.

If $K(C_{\approx}(t)) \neq {}_e$, then $K(C_{\approx}(t)) = K(C_{\approx}(t'))$ by the definition of $\Vdash$ and so $C_{\approx}(t) = C_{\approx}(t')$, $t \approx t'$ and t t'.

Suppose that t t'.

[3] We could define $0 =_{\text{def}} \{A : \neg A$ Γ'$\}$, $1 =_{\text{def}} \{A : A$ Γ'$\}$ and $=_{\text{def}} \{A : A$ Γ' and $\neg A$ Γ'$\}$ which ensures the existence of .

If ($t = t$) Γ', then $K(C_{\approx}(t)) =$ $_e$ and so, by the definition of ⊩ , $_e$ ⊩ $K(C_{\approx}(t'))$.
If ($t = t$) Γ', then by the definition of , ($t = t'$) Γ'. But by rule ID2, this implies that ($t' = t'$) Γ', ($t' = t$) Γ' and so t' t . Thus by the definition of ≈, $t \approx t'$. So $K(C_{\approx}(t)) = K(C_{\approx}(t'))$ and $K(C_{\approx}(t))$ ⊩ $K(C_{\approx}(t'))$.

(iii) For predicate symbols

a) $K(C_{\approx}(P^n))$ is the function h of M_2^n in M_1 such that
$h(<a_1,\dots,a_n>) = K(C_{\approx}(P^n t_1\dots t_n))$, where $C_{\approx}(t_i) = a_i$.

In order to show that this definition is adequate, it is necessary to verify that 1) for every a M_2, there is a term t such that $C_{\approx}(t) = a$ and that 2) if $C_{\approx}(t_{i1}) = C_{\approx}(t_{i2})$, then $K(C_{\approx}(P^n t_1\dots t_{i1}\dots t_n)) = K(C_{\approx}(P^n t_1\dots t_{i2}\dots t_n))$.
1) follows directly from (ii) b).
For 2) one must examine two cases.
The first is the one in which ($t_{i1} = t_{i2}$) Γ'. Rules ID2 and R5 guarantee that Γ' is closed for the intersubstitution of t_{i1} and t_{i2}. Hence we will have either $P^n t_1\dots t_{i1}\dots t_n$ and $P^n t_1\dots t_{i2}\dots t_n$ both in Γ', or their negations both in Γ' or, finally, neither the one nor the other. By definition of K for formulae, $P^n t_1\dots t_{i1}\dots t_n$ and $P^n t_1\dots t_{i2}\dots t_n$ both belong to the same equivalence class.

The second case is the one in which ($t_{i1} = t_{i2}$) Γ'. From $C_{\approx}(t_{i1}) = C_{\approx}(t_{i2})$ and the definition of ≈ and one easily shows that ($t_{i1} = t_{i1}$) Γ' and ($t_{i2} = t_{i2}$) Γ'. It follows, by ID1, that (t_{i1}) Γ' and (t_{i2}) Γ'. If neither $P^n t_1\dots t_{i1}\dots t_n$, nor $P^n t_1\dots t_{i2}\dots t_n$, nor $\neg P^n t_1\dots t_{i1}\dots t_n$, nor $\neg P^n t_1\dots t_{i2}\dots t_n$ belongs to Γ', then $P^n t_1\dots t_{i1}\dots t_n$ and $P^n t_1\dots t_{i2}\dots t_n$ belong to the same class. If not, we take the one (one of the ones) that belongs to Γ'. Suppose, for example, that it is $P^n t_1\dots t_{i1}\dots t_n$. By M1 and the -saturation of Γ' we show that ($P^n t_1\dots t_{i1}\dots t_n \equiv P^n t_1\dots t_{i2}\dots t_n$) Γ'. From R3 it follows that $P^n t_1\dots t_{i2}\dots t_n$ Γ'. The other cases are similar.

b) Now we must show that $K(C_{\approx}(P^n))$ is monotonic. Since an n-ary predicate is monotonic if and only if it is monotonic for each of its n arguments, it will suffice to verify for any given argument. Moreover, because the proof is the same for all n, it will suffice to verify the monotonicity of unary predicates.

We have that $K(C_{\approx}(P))$ is monotonic
if and only if
for every $C_{\approx}(t)$ and $C_{\approx}(t')$, if $C_{\approx}(t)$ ⊩ $C_{\approx}(t')$, then $K(C_{\approx}(P))(C_{\approx}(t))$ ⊩ $K(C_{\approx}(P))(C_{\approx}(t'))$
if and only if

(by (ii) c), (ii) a) and (iii) a)) for every t, t', if $t \quad t'$, then $K(C_{\approx}(Pt)) \Vdash K(C_{\approx}(Pt'))$

if and only if

(by (i) c)) for every t, t', if $t \quad t'$ then, $Pt \quad Pt'$.

By *reductio ad absurdum* let us suppose that for some t, t', we have $t \quad t'$ and not $Pt \quad Pt'$. Then by the definition of this means that (the "[,]" serving to clarify)

$[(t = t) \quad \Gamma'$ or $(t = t') \quad \Gamma']$ and $[[Pt \quad \Gamma'$ and $Pt' \quad \Gamma']$ or $[\neg Pt \quad \Gamma'$ and $\neg Pt' \quad \Gamma']]$.

We have to verify two cases:

Case 1

$[(t = t) \quad \Gamma'$ or $(t = t') \quad \Gamma']$ and $[Pt \quad \Gamma'$ and $Pt' \quad \Gamma']$

Case 1.1

$(t = t') \quad \Gamma'$ and $Pt \quad \Gamma'$ and $Pt' \quad \Gamma'$, which is forbidden by ID2.

Case 1.2

$(t = t) \quad \Gamma'$ and $Pt \quad \Gamma'$ and $Pt' \quad \Gamma'$. From $Pt \quad \Gamma'$ and R2, we have that (Pt). But by M1, $(t) \quad (P(t) \equiv P(t')) \quad \Gamma'$. By the *-saturation* of Γ', we have either $(t) \quad \Gamma'$ or $(P(t) \equiv P(t')) \quad \Gamma'$. $(t) \quad \Gamma'$ is not possible since by ID1, we would then have $(t = t) \quad \Gamma'$ which is contrary to the hypothesis. Therefore $(P(t) \equiv P(t')) \quad \Gamma'$. It follows by R3 that $P(t') \quad \Gamma'$, which is absurd.

Case 2

$[(t = t) \quad \Gamma'$ or $(t = t') \quad \Gamma']$ and $[\neg Pt \quad \Gamma'$ and $\neg Pt' \quad \Gamma']$

The proof is similar to that of case 1.

(iv) For n-places function symbols

$K(C_{\approx}(f^n))$ is the function h of M_2^n in M_2 such that $h(<a_1,\ldots,a_n>) = K(C_{\approx}(f^n t_1\ldots t_n))$

where $C_{\approx}(t_i) = a_i$.

In order to show that this definition is adequate, we must verify that 1) for every $a \quad M_2$, there is a term t such that $C_{\approx}(t) = a$ and 2) if $C_{\approx}(t_{i1}) = C_{\approx}(t_{i2})$ then

$K(C_{\approx}(f^n t_1\ldots t_{i1}\ldots t_n)) = K(C_{\approx}(f^n t_1\ldots t_{i2}\ldots t_n))$.

1) has already been proven.

For 2), let us suppose by *reductio ad absurdum* that $C_{\approx}(t_{i1}) = C_{\approx}(t_{i2})$ and

$K(C_{\approx}(f^n t_1\ldots t_{i1}\ldots t_n)) \neq K(C_{\approx}(f^n t_1\ldots t_{i2}\ldots t_n))$,

that is to say that $C_{\approx}(f^n t_1\ldots t_{i1}\ldots t_n) \neq C_{\approx}(f^n t_1\ldots t_{i2}\ldots t_n)$.

There are two cases

Case 1
$(t_{i1} = t_{i1})$ Γ' and by definitions 1 and 2, we have $(t_{i1} = t_{i2})$ Γ'.
Then by using R2, ID3 and E we have that (t_{i1}) Γ'.
Case 1.1
Suppose that $(f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ'. By ID1 we have $(f^n t_1 \ldots t_{i1} \ldots t_n = f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ, and by ID2, $(f^n t_1 \ldots t_{i1} \ldots t_n = f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ'. By definitions 1 and 2, we have that $f^n t_1 \ldots t_{i1} \ldots t_n \approx f^n t_1 \ldots t_{i2} \ldots t_n$, which is contrary to the hypothesis.
Case 1.2
Suppose that $(f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ'. In that case, the same argument as in 1.1 will bring us to conclude that $(f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ'. By R4 it follows that
$(f^n t_1 \ldots t_{i1} \ldots t_n = f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ' and $(f^n t_1 \ldots t_{i2} \ldots t_n = f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ' and from there, in virtue of definitions 1 and 2, that $f^n t_1 \ldots t_{i1} \ldots t_n \approx f^n t_1 \ldots t_{i2} \ldots t_n$, which is contrary to the hypothesis.
Case 2
$(t_{i1} = t_{i1})$ Γ'. It can easily be proved, from definitions 1 and 2 that $(t_{i2} = t_{i2})$ Γ'.
By ID1, it is clear that (t_{i1}) Γ' and (t_{i2}) Γ'.
Suppose that $(f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ'. By M2 and the -saturation of Γ', $(f^n t_1 \ldots t_{i1} \ldots t_n = f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ', which is absurd. Therefore $(f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ'. A similar argument leads us to $(f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ'. By R4 we then have that $(f^n t_1 \ldots t_{i1} \ldots t_n = f^n t_1 \ldots t_{i1} \ldots t_n)$ Γ' and $(f^n t_1 \ldots t_{i2} \ldots t_n = f^n t_1 \ldots t_{i2} \ldots t_n)$ Γ', which by definitions 1 and 2 entails that $f^n t_1 \ldots t_{i1} \ldots t_n \approx f^n t_1 \ldots t_{i2} \ldots t_n = {}_e$, which is absurd, and which completes the proof.
Now we can define the interpretation.

A canonical model for partial predicate calculus
Definition 3
Let $\mathrm{M}_K = <M_2, g_K>$ such that M_2 is as above, $g_K(P^n) = K(C_{\approx}(P^n))$ and $g_K(f^n) = K(C_{\approx}(f^n))$ (we will sometimes simply write M).

Definition 4
Let μ_K : Var → M_2 be the assignment such that $\mu_K(x) = K(C_{\approx}(x))$.

To prove the main lemma, we will need some secondary lemmas. The simplest proofs are left to the reader.

Lemma 3

For every term t, $\|t\|_{\mathrm{M},\mu K} = C_{\approx}(t)$

The next two lemmas are very important: they show that our ID rules adequately characterize the behaviour of identity in partial domains.

Lemma 4

$C_{\approx}(t) = C_{\approx}(t') \neq {}_{e}$ if and only if $(t = t')$ Γ'

Lemma 5

$C_{\approx}(t) \neq C_{\approx}(t')$, $C_{\approx}(t) \neq {}_{e}$ and $C_{\approx}(t') \neq {}_{e}$ if and only if $\neg(t = t')$ Γ'.

Proof

By lemma 4 and $C_{\approx}(t) \neq C_{\approx}(t')$, we have that $(t = t')$ Γ'. If $C_{\approx}(t) \neq {}_{e}$ and $C_{\approx}(t') \neq {}_{e}$, then by the definition of M_2, $(t = t)$ Γ' and $(t' = t')$ Γ'.
From $(t = t)$ Γ' and R4, we have (t) Γ' and by a similar argument, (t') Γ'. By I and ID4, we have $(t = t')$ Γ'.
By the definition of , we therefore have
$((t = t') \equiv T)$ $((t = t') \equiv F)$ Γ'.
By the -saturation of Γ', we have either $((t = t') \equiv T)$ Γ', or $((t = t') \equiv F)$ Γ'.
Therefore by T1 and R3, if $((t = t') \equiv T)$ Γ', then $(t = t')$ Γ', which is absurd.
Therefore $((t = t') \equiv F)$ Γ', which by R10 implies that $\neg(t = t')$ Γ'.

Suppose that $\neg(t = t')$ Γ' and let us show that $C_{\approx}(t) \neq C_{\approx}(t')$, $C_{\approx}(t) \neq {}_{e}$ and $C_{\approx}(t') \neq {}_{e}$.
If $\neg(t = t')$ Γ', then by R2, $(\neg(t = t'))$ Γ'.
We then show, by BOOL2, that
$(t = t')$ Γ'.
By ID3 and E, we have (t) Γ' and (t') Γ'. So by ID1, $(t = t)$ Γ' and
$(t' = t')$ Γ' and so $C_{\approx}(t) \neq {}_{e}$ and $C_{\approx}(t') \neq {}_{e}$.

Let us show that if $\neg(t = t')$ Γ', then $C_{\approx}(t) \neq C_{\approx}(t')$. By *reductio ad absurdum* let us suppose that $C_{\approx}(t) = C_{\approx}(t')$. By the definition of $\approx$, the following two sentences are true:
if $(t = t)$ Γ', then $(t = t')$ Γ', and if $(t' = t')$ Γ', then $(t = t')$ Γ'. The consistency of Γ' precludes that $(t = t')$ Γ'. So $(t = t)$ Γ' and $(t' = t')$ Γ' and by the definition of M_2, $C_{\approx}(t) = C_{\approx}(t') = {}_{e}$, which is absurd since we have shown that if $\neg(t = t')$ Γ', then $C_{\approx}(t) \neq {}_{e}$ and $C_{\approx}(t') \neq {}_{e}$.

Lemma 6
For every a M_2, $a \neq {}_{e}$ and every finite set X of variables, there is a variable x X such that $a = C_{\approx}(x)$.

Lemma 7
For every a M_2, every term t and every t' such that $a = C_{\approx}(t')$, $||t||_{\mathrm{M},\mu K(a/x)} = ||t(t'/x)||_{\mathrm{M},\mu K}$.

Lemma 8
For every a M_2 and every term t such that $a = C_{\approx}(t)$, $||A||_{\mathrm{M},\mu K(a/x)} = ||A(t/x)||_{\mathrm{M},\mu K}$

Proof
We proceed by induction.
(i) A is $(t = t')$
This follows directly from lemma 7.
(ii) A is $P^n t_1, ..., t_n$. In that case,
$||P^n t_1, ..., t_n||_{\mathrm{M},\mu K(a/x)} =$ definition of || || and g_K
$K(C_{\approx}(P^n))(<||t_1||_{\mathrm{M},\mu K(a/x)} \ldots ||t_n||_{\mathrm{M},\mu K(a/x)}>) =$ lemma 7
$K(C_{\approx}(P^n))(<||t_1(t/x)||_{\mathrm{M},\mu K} \ldots ||t_n(t/x)||_{\mathrm{M},\mu K}>) =$ definition of K
$||P^n t_1(t/x) \ldots t_n(t/x)||_{\mathrm{M}\mu K} =$ definition of (t/x)
$||P^n t_1, ..., t_n(t/x)||_{\mathrm{M},\mu K}$
(iii) A is $\neg B$, the demonstration is trivial.
(iv) A is B C, the demonstration is trivial.
(v) A is yB. There are two sub-cases.

a) $x = y$
$|| xB||_{\mathrm{M}\mu K(a/x)} =$ by the definitions of $\mu_K(a/x)$ and || ||
$|| xB||_{\mathrm{M}\mu K} =$ x is not free in xB
$|| xB(t/x)||_{\mathrm{M},\mu K}$

b) $x \neq y$
Let us show the identity for any term t such that $C_{\approx}(t) = a$.

$$\|\ yB\|_{M,\mu K(a/x)} = \begin{cases} 1 \text{ iff there is a} b \quad E \text{ such that } \|\ B\ \|_{M,\mu_K (a/x)(b/y)} = 1 \\ 0 \text{ iff for every} b \quad E,\ \ \|\ B\|_{M,\mu_K (a/x)(b/y)} = 0 \\ \quad \text{otherwise} \end{cases}$$

if and only if, since $x \neq y$ entails that $\mu_K(b/x)(a/y) = \mu_K(a/y)(b/x)$

$$\|\ yB\|_{M,\mu K(a/x)} = \begin{cases} 1 \text{ iff there is a} b \quad E \text{ such that } \|\ B\ \|_{M,\mu_K (b/x)(a/y)} = 1 \\ 0 \text{ iff for every} b \quad E,\ \ \|\ B\|_{M,\mu_K (b/x)(a/y)} = 0 \\ \quad \text{otherwise} \end{cases}$$

if and only if, by the induction hypothesis,

$$\|\ yB\|_{M,\mu K(a/x)} = \begin{cases} 1 \text{ iff there is a} b \quad E \text{ such that } \|\ B(t/x)\|_{M,\mu_K (b/y)} = 1 \\ 0 \text{ iff for every} b \quad E,\ \ \|\ B\|_{M,\mu_K (b/y)} = 0 \\ \quad \text{otherwise} \end{cases}$$

if and only if

$$\|\ yB\|_{M,\mu K(a/x)} = \ \|\ yB(t/x)\|_{M,\mu K}$$

Lemma 9
$(t) \quad \Gamma'$ if and only if $(t = t) \quad \Gamma'$.

Lemma 10
$(t) \quad \Gamma'$ if and only if $C_{\approx}(t) \neq \ {}_{e}$.

Lemma 11
Let A be any formula . For every a, $a \neq \ {}_{e}$, $\|A\|_{M,\mu K(a/x)} = 1$ if and only if for every term t free for x in A such that $(t) \quad \Gamma'$, $\|A(t/x)\|_{M,\mu K} = 1$, and for every a, $a \neq \ {}_{e}$, $\|A\|_{M,\mu K(a/x)} = 0$ if and only if for every term t free for x in A such that $(t) \quad \Gamma'$, $\|A(t/x)\|_{M,\mu K} = 0$.

Main lemma

M is a model such that for every formula A

$||A||_{M,\mu K} = 1$ iff $A \quad \Gamma'$

$||A||_{M,\mu K} = 0$ iff $\neg A \quad \Gamma'$

$||A||_{M,\mu K} = \quad$ otherwise.

Proof

By induction, we verify that $||A||_{M,\mu K} = 1$ if and only if $A \quad \Gamma'$ and $||A||_{M,\mu K} = 0$ if and only if $\neg A \quad \Gamma'$ simultaneously. The third possibility is satisfied automatically if the first two are.

(i) a)

$||(t = t')||_{M,\mu K} = 1$ iff	definition of $|| \ ||$
$||t||_{M,\mu K} = ||t'||_{M,\mu K} \neq \quad_e$ iff	lemma 3
$C_{\approx}(t) = C_{\approx}(t') \neq \quad_e$ iff	lemma 4
$(t = t') \quad \Gamma'$	

b)

$||(t = t')||_{M,\mu K} = 0$ iff	definition of $|| \ ||$
$||t||_{M,\mu K} \neq ||t'||_{M,\mu K}$, $||t||_{M,\mu K} \neq \quad_e$ and $||t'||_{M,\mu K} \neq \quad_e$ iff	lemma 3
$C_{\approx}(t) \neq C_{\approx}(t')$, $C_{\approx}(t) \neq \quad_e$ and $C_{\approx}(t) \neq \quad_e$ iff	lemma 5
$\neg(t = t') \quad \Gamma'$	

(ii) a)

$||P^n t_1, \ldots, t_n||_{M,\mu K} = 1$ iff	definition of $|| \ ||$
$g_K(P^n)(<||t_1||_{M,\mu K} \ldots ||t_n||_{M,\mu K}>) = 1$ iff	lemma 3
$g_K(P^n)(< C_{\approx}(t_1), \ldots, C_{\approx}(t_n) >) = 1$ iff	definition of g_K
$K(C_{\approx}(P^n))(< C_{\approx}(t_1), \ldots, C_{\approx}(t_n) >) = 1$ iff	definition of $K(C_{\approx}(P^n))$
$K(C_{\approx}(P^n t_1 \ldots t_n)) = 1$ iff	definition of K
$P^n t_1 \ldots t_n \quad \Gamma'$	

b)

$||P^n t_1, \ldots, t_n||_{M,\mu K} = 0$ iff	definition of $|| \ ||$
$g_K(P^n)(<||t_1||_{M,\mu K} \ldots ||t_n||_{M,\mu K}>) = 0$ iff	lemma 3
$g_K(P^n)(< C_{\approx}(t_1), \ldots, C_{\approx}(t_n) >) = 0$ iff	definition of g_K
$K(C_{\approx}(P^n))(< C_{\approx}(t_1), \ldots, C_{\approx}(t_n) >) = 0$ iff	definition of $K(C_{\approx}(P^n))$
$K(C_{\approx}(P^n t_1 \ldots t_n)) = 0$ iff	definition of K
$\neg P^n t_1 \ldots t_n \quad \Gamma'$	

(iii) a)

$||\neg A||_{M,\mu K} = 1$ iff	definition of $|| \ ||$

$\|A\|_{\mathrm{M},\mu\kappa} = 0$ iff — induction hypothesis
$\neg A \quad \Gamma'$
b)
$\|\neg A\|_{\mathrm{M},\mu\kappa} = 0$ iff — definition of $\| \;\; \|$
$\|A\|_{\mathrm{M},\mu\kappa} = 1$ iff — induction hypothesis
$A \quad \Gamma'$

iv) a)
$\|A \quad B\|_{\mathrm{M},\mu\kappa} = 1$ iff — definition of $\| \;\; \|$
$\|A\|_{\mathrm{M},\mu\kappa} = 1$ or $\|B\|_{\mathrm{M},\mu\kappa} = 1$ iff — induction hypothesis
$A \quad \Gamma'$ or $B \quad \Gamma'$ iff — I and -saturation of Γ'
$(A \quad B) \quad \Gamma'$

b)
$\|A \quad B\|_{\mathrm{M},\mu\kappa} = 0$ iff — definition of $\| \;\; \|$
$\|A\|_{\mathrm{M},\mu\kappa} = 0$ and $\|B\|_{\mathrm{M},\mu\kappa} = 0$ iff — induction hypothesis
$\neg A \quad \Gamma'$ and $\neg B \quad \Gamma'$ iff — R18, I and R23
$\neg(A \quad B) \quad \Gamma'$

(v) a)
$\| \; xB\|_{\mathrm{M},\mu\kappa} = 1$ iff — definition of $\| \;\; \|$
there is an $a \quad E$ such that $\|B\|_{\mathrm{M},\mu\kappa(a/x)} = 1$ iff — lemma 8
there is an $a \quad E$ and a term t such that $a = C_{\approx}(\; t)$ and
$\|B(\mathrm{t}/x)\|_{\mathrm{M},\mu\kappa} = 1$ iff — induction hypothesis
$B(t/x) \quad \Gamma'$ iff — I and -saturation of Γ'
$xB \quad \Gamma'$

b) $\| \; xB\|_{\mathrm{M},\mu\kappa} = 0$ iff — definition of $\| \;\; \|$
for every $a \neq {}_{e}$, $\|B\|_{\mathrm{M},\mu\kappa(a/x)} = 0$ iff — lemma 10 and 11
for every t such that $(t) \quad \Gamma'$, $\|B(t/x)\|_{\mathrm{M},\mu\kappa} = 0$ iff — induction hypothesis
for every t such that $(t) \quad \Gamma'$, $\neg B(t/x) \quad \Gamma'$ iff — -saturation and E
$x\neg B \quad \Gamma'$ iff — R20 and R25
$\neg \; xB \quad \Gamma'$

Strong completeness theorem
If $\Gamma \models A$, then $\Gamma \vdash A$.

Proof
Let us suppose that it is not the case that $\Gamma \vdash A$. Let Γ' be a deductively closed , and -saturated set defined as above, taking A as the test

formula. Then A Γ'. By the main lemma, we have an interpretation for which A is either false or undefined and so it is not the case that $\Gamma \models A$.

This proof is, from a purely logical point of view, rather standard and intrinsicaly of no real interest. What is interesting is that it provides us with a representation theorem. Indeed, the classical Henkin’s proof relies on the following fundamental property: any maximally consistent set determines a canonical model i.e., any maximally consistent set determines a canonical classical value for each predicate and each functional term. Our result is an extension of this idea and what we have proved is that if we adopt the hypothesis that the interpretation of an epistemic state must use maximal monotonic partial functions as values for logical connectives and quantifiers, then the completeness theorem is a representation theorem: any SDCCS determines *one and only one* partial value for each predicate and each functional term. Thus under the maximality and the monotonicity hypotheses for epistemic states, our first-order logic is *the* logic of epistemic states.

Université de Montréal/Collège de Bois-de-Boulogne
lepagef@ere.umontreal.ca

REFERENCES

[1] HENKIN L., “Completeness in the Theory of Types”, *The Journal of Symbolic Logic*, 15, 1950, pp. 81-91.

[2] LAPIERRE S., “A Functional Partial Semantics for Intensional Logic”, *Notre Dame Journal of Formal Logic*, 33, 4, 1992, pp. 517-541.

[3] LEPAGE F., “Partial Functions in Type Theory”, *Notre Dame Journal of Formal Logic*, 33, 4, 1992, pp. 493-516.

[4] THIJSSE G. C. E., *Partial Logic and Knowledge Representation*, Eburon Publishers, Delft, The Netherlands, 1992.

[5] THOMASON R. H., “On the Strong Semantical Completeness of Intuitionistic Predicate Calculus”, *The Journal of Symbolic Logic*, 33, 1, 1968, pp. 1-7.