institute *for* logic, language *and* computation

FRANÇOIS LEPAGE, SERGE LAPIERRE

The Functional Completeness of 4-value Monotonic Protothetics

LP-96-06, *received*: June 1996

ILLC Research Report and Technical Notes Series
Series editor: Dick de Jongh

Logic, Philosophy and Linguistics (LP) Series, ISSN: 0928-3307

Institute for Logic, Language and Computation (ILLC)
University of Amsterdam
Plantage Muidergracht 24
NL-1018 TV Amsterdam
The Netherlands
e-mail: illc@fwi.uva.nl

The Functional Completeness of 4-value Monotonic Protothetics

François Lepage
Département de philosophie
Université de Montréal
C.P. 6128 Succursale Centre ville
Montréal, H3C 3J7
e-mail: lepagef@ere.umontreal.ca

Serge Lapierre
Collège de Bois-de-Boulogne
e-mail: lapierrs@ere.umontreal.ca

ABSTRACT

By *protothetics* we mean the language of type theory having t (the type of truth values) as only basic type. This language can be interpreted by various non classical semantics. In particular, it can be interpreted by a semantics based on four truth values, forming an approximation lattice, and in which all function spaces are *restricted* to monotonic functions. We show that interpreted by this non classical semantics, the language of protothetics is *functionaly complete* in the sense that for every monotonic function there is a name in the language which denotes it. More precisely, using only functional abstraction, application and Boolean operators, we provide a recursive formula which gives a name of every object of any type. We conclude on remarks concerning the definability of the quantifiers.

0. Protothetics

The term *protothetics* refers to the theory of propositional types. By propositional types we understand the following.

Definition 1 The set of propositional types is the smallest set T such that

(i) $t \in T$ (t is the type of propositions);
(ii) if α, $\beta \in T$, then $<\alpha\beta> \in T$.

From now on we will abbreviate $<\alpha\beta>$ by $\alpha\beta$ whenever convenient.
The domains of each type are defined as follows.

Definition 2 For each $\alpha \in T$, the set of entities of type α is the set D_α such that

(i) $D_t = \{0, 1\}$ (the set of truth values);
(ii) $D_{\alpha\beta} = D_\beta{}^{D_\alpha}$ (the set of functions of D_α in D_β).

The syntax of protothetics is the following.

Definition 3 First, for every type α, there is a denumerable set $Var_\alpha = \{X_{\alpha i}\}_{i \in \omega}$ of *variables* of that type. The set of *terms* of type α is the smallest set Trm_α such that :

(i) $Var_\alpha \subseteq Trm_\alpha$;
(ii) if $A, B \in Trm_t$, then $\neg A$, $[A \wedge B] \in Trm_t$;
(iii) $A \in Trm_\alpha$ and $X \in Var_\beta$, then $\lambda XA \in Trm_{\beta\alpha}$;
(iv) if $A \in Trm_{\alpha\beta}$ and $B \in Trm_\alpha$, then $[AB] \in Trm_\beta$;
(v) $\mathrm{F} \in Trm_t$ and $\mathrm{T} \in Trm_t$.

Let us call terms of type t *sentences*. F and T are two special sentences which will denote respectively the false and the true. If A and B are sentences, then $[A \vee B]$ is an abbreviation for $\neg[\neg A \wedge \neg B]$. The *scope* of λX_α in $\lambda X_\alpha A$ is A, and a term is *closed* if every occurrence of every variable X_α is in the scope of λX_α. Henceforth A_α, B_α, C_α, ... will refer to terms of type α.

We then define an *assignment of value.*

Definition 5 An assignment of value j is a function

$$j : \bigcup_{\alpha \in T} Var_\alpha \rightarrow \bigcup_{\alpha \in T} D_\alpha$$

such that $j(X_\alpha) \in D_\alpha$. We write $j(a/X)$ for the assignment that differs from j at most by the fact that it assigns the value a to X.

Finally, we define a *valuation based on j.*

Definition 6 A valuation based on j is a function

$$\mathrm{V}_j : \bigcup_{\alpha \in T} Trm_\alpha \rightarrow \bigcup_{\alpha \in T} D_\alpha$$

such that

(i) $\mathrm{V}_j(X_\alpha) = j(X_\alpha)$;
(ii) $\mathrm{V}_j(\neg A_t) = 1$ if $\mathrm{V}_j(A_t) = 0$
$= 0$ if $\mathrm{V}_j(A_t) = 1$;
(iii) $\mathrm{V}_j([A_t \wedge B_t]) = 1$ if $\mathrm{V}_j(A) = \mathrm{V}_j(B) = 1$

$= 0$ if $V_j(A) = 0$ or $_j(B) = 0$;

(iv) $V_j(\lambda X_\alpha A_\beta)$ is the function which associates $V_{j(a/X)}(A)$ with every $a \in D_\alpha$;

(v) $V_j([A_{\alpha\beta} B_\alpha]) = V_j(A)(V_j(B))$;

(vi) $V_j(F) = 0$ and $V_j(T) = 1$.

When a term A is closed, $V_j(A)$ is independent of j and $V_j(A)$ is called the *denotation* of A and is written A^d.

Henkin (1963) showed how to provide a canonical name of each object in D_α for every type α, using λ, fonctional application, $\wedge$, $\vee$, $\neg$ and an identity operator $\equiv$. The identity operator has the classical meaning : $V_j([A_\alpha \equiv B_\alpha]) = 1$ iff $V_j(A) = V_j(B)$. One interesting question concerns the possibility of providing names using only λ, application, $\wedge$, $\vee$, and $\neg$, i.e., using only functional abstraction, application and the Boolean operators. van Benthem (1995) gave a positive answer to this question. One exact formulation of his suggestion is the following.

Definition 7 Let $\alpha = <\alpha_1 \ldots <\alpha_n t> \ldots>$. Let us call any sequence (empty if $\alpha = t$) $<p_1,\ldots,p_n> \in D_{\alpha_1} \times \ldots \times D_{\alpha_n}$ an *α-projector*. Clearly, for every $f \in D_\alpha$, $f(p_1)\ldots(p_n) \in D_t$ for every α-projector $<p_1,\ldots,p_n>$. Let $Pr(\alpha)$ be the set of all α-projectors and let $f \in D_\alpha$. A *projector* of f any sequence in $Pr(\alpha)$. $\mathbf{1}(f)$ will denote the set of projectors $<p_1,\ldots,p_n>$ of f such that $f(p_1)\ldots(p_n) = 1$.

Clearly, for every f, $g \in D_\alpha$, $f = g$ iff $f(p_1)\ldots(p_n) = g(p_1)\ldots(p_n)$ for every α-projector $<p_1,\ldots,p_n>$. We will write $\vec{p}$ for $<p_1,\ldots,p_n>$ and $f(\vec{p})$ for $f(p_1)\ldots(p_n)$.

A similar syntactic notion will be used. For any term A_α, with $\alpha = <\alpha_1 \ldots <\alpha_n t> \ldots>$, we will call a sequence $\vec{B} = <B_1,\ldots,B_n> \in Trm_{\alpha_1} \times \ldots \times Trm_{\alpha_n}$ a projector of A. Clearly, $[\ldots[AB_1]\ldots B_n] \in Trm_t$ for every projector $<B_1,\ldots,B_n>$ of A. We will write $[A\vec{B}]$ for $[\ldots[AB_1]\ldots B_n]$.

Let $f \in D_\alpha$ with $\alpha = <\alpha_1 \ldots <\alpha_n t> \ldots>$ and suppose, by induction, that we have a canonical name $(g)^c$ for every object g of lower types. For any variable X_{α_i} $(1 \leq i \leq n)$, $[X_{\alpha_i}(\vec{g})^c]$ is $[\ldots[X_{\alpha_i}(g_1)^c]\ldots(g_k)^c]$, i.e., the term of type t made of the variable X_{α_i} applied to the projector $(\vec{g})^c = <(g_1)^c,\ldots,(g_k)^c>$. Now let $\vec{p} = <p_1,\ldots,p_n>$ be a projector of f and let us define δ_{p_i} as follows :

$$\delta_{p_i}([X_{\alpha_i}(\vec{g})^c]) = [X_{\alpha_i}(\vec{g})^c)] \quad \text{if } p_i(\vec{g}) = 1$$

$$= \neg[X_{\alpha_i}(\vec{g})^c)] \ \text{if } p_i(\vec{g}) = 0.$$

Then

Proposition

$$\lambda X_{\alpha_1} \ldots \lambda X_{\alpha_n} \Big[\bigvee_{\vec{p} \in \mathbf{1}(f)} \Big(\bigwedge_{\vec{g} \in Pr(\alpha_1)} \delta_{p_1}([X_{\alpha_1}(\vec{g})^c]) \wedge \ldots \wedge \bigwedge_{\vec{g} \in Pr(\alpha_n)} \delta_{p_n}([X_{\alpha_m}(\vec{g})^c]) \Big) \Big]$$

is a rigid designator or a canonical name of f, i.e., $((f)^c)^d = f$. Let's call this formula **F1**.

With **F1**, we have a name for every classical propositional function. van Benthem suggested that this result can be easily extended to the 'many-valued' case. A special case is the three-valued logic where the third value is the *undefined* value. Can we generalize the above result in order to have a name for very *partial* propositional function? The problem is of some interest because, as we will see, it is not possible to use an identity operator in partial logic, so we cannot use Henkin's strategy.

1. Partial Functions

In the following, we identify a partial function with a special kind of monotonic function[1]. Formally, we have

Definition 8 For any $\alpha \in T$, the set PM_α of partial functions of type α is recursively defined as follows :

(i) $PM_t = \{0, 1, \bot\}$
(ii) $PM_{\alpha\beta} = (PM_\alpha \rightarrow PM_\beta)$

where $(PM_\alpha \rightarrow PM_\beta)$ is the set of monotonic functions of PM_α in PM_β, the monotonicity being relative to the following order :

(i) for any $x \in PM_t$, $x \sqsubseteq x$ and $\bot \sqsubseteq x$;

[1] For a more extensive presentation of the following, see F. Lepage (1992) or (1995) and S. Lapierre (1992).

(ii) for any $f,\ g \in PM_{\alpha\beta}$, $f \sqsubseteq g$ if and only if for any $x \in PM_{\alpha}$, $f(x) \sqsubseteq g(x)$.

Proposition For any α, PM_{α} is a meet-semi-lattice, where the meet $\wedge$ and (when it exists) the sup $\vee$ are defined respectively by the recursive clause :

(i) for $x, y \in PM_t$, $x \wedge y = x$ if $x = y$ and $\bot$ otherwise;

(ii) for $f,\ g \in PM_{\alpha\beta}$, $f \wedge g$ is the function h such that for any $x \in PM_{\alpha}$, $h(x) = f(x) \wedge g(x)$;

and

(iii) for $x, y \in PM_t$, $x \vee y = x$ if $y = \bot$ or $x = y$
$x \vee y = y$ if $x = \bot$ and does not exist otherwise;

(iv) for $f,\ g \in PM_{\alpha\beta}$, $f \vee g$ is the function h such that for any $x \in PM_{\alpha}$, $h(x) = f(x) \vee g(x)$ if $f(x) \vee g(x)$ exists.

The notion of projector is defined as usual. We just need the following additional notions.

Definition 9 Let $f \in PM_{\alpha}$. **1**(f) (resp. **0**(f) and $\bot(f)$) is the set of projectors $<p_1,\ldots,p_n>$ of f such that $f(p_1)\ldots(p_n) = 1$ (resp. 0 and $\bot$).

Again we have this

Proposition Let $f,\ g \in PM_{\alpha}$; $f = g$ iff $f(p_1)\ldots(p_n) = g(p_1)\ldots(p_n)$ for every α-projector $<p_1,\ldots,p_n>$.

Now we can define a partial interpretation for the terms of the propositional types theory.

Definition 10 : A *partial value assignment* j is a function

$$j : \bigcup_{\alpha \in T} Var_{\alpha} \rightarrow \bigcup_{\alpha \in T} PM_{\alpha}$$

such that $j(X_\alpha) \in PM_\alpha$. As before, we write $j(a/X)$ for the assignment that differs from j at the most by the fact that it assigns the value a to X

Finally, we define a *partial valuation based on j.*

Definition 11 A partial valuation based on j is a function

$$V_j : \bigcup_{\alpha\in T} Trm_\alpha \rightarrow \bigcup_{\alpha\in T} PM_\alpha$$

such that

(i) $V_j(X_\alpha) = j(X_\alpha)$;

(ii) $V_j(\neg A_t) = 1$ if $V_j(A_t) = 0$
$= 0$ if $V_j(A_t) = 1$
$= \perp$ otherwise;

(iii) $V_j([A_t \wedge B_t]) = 1$ if $V_j(A) = V_j(B) = 1$
$= 0$ if $V_j(A) = 0$ or $_j(B) = 0$
$= \perp$ otherwise;

(iv) $V_j(\lambda X_\alpha A_\beta)$ is the function which associates $V_{j(a/X)}(A)$ with every $a \in D_\alpha$;

(v) $V_j([A_{\alpha\beta} B_\alpha]) = V_j(A)(V_j(B))$;

(vi) $V_j(F) = 0$ and $V_j(T) = 1$.

It is easy to verify that negation and conjunction are both monotonic, and that functional application and abstraction preserve monotonicity. It follows that for every type α, every term A_α and every j, $V_j(A) \in PM_\alpha$. Now, if we want to keep our semantics sound, it is not possible - as suggested at the end of the previous section - to introduce into the language an identity operator "≡" which behaves classically, because $V_j([A_\alpha \equiv B_\alpha])$ would not be monotonic with regard to $V_j(A)$ and $V_j(B)$. (For example, let A_t and B_t such that $V_j(A) = \perp$ and $V_j(B) = \perp$; then we would have $V_j([A \equiv B]) = 1$. Furthermore let C_t and D_t such that $V_j(C) = 1$ and $V_j(D) = 0$; then we would have $V_j([C \equiv D]) = 0$. But since $V_j(A) \leq V_j(C)$ and $V_j(B) \leq V_j(D)$, monotonicity would be broken.) We have here a very particular property of partial protothetics : the strongest equivalence relation definable in the object language of partial protothetics is *definitively* weaker than identity.

Again, one easily verifies that, for any closed term A and any j, j', $V_j(A) = V_{j'}(A)$. In that case, we will write $V_j(A) = V_{j'}(A) = A^d$.

Can we provide a name for any partial object? Yes, but only if we add in the language a new special symbol ϕ referring to $\bot$. Then following Blamey (1986), we may define

$$[A_t \downarrow B_t] =_{\text{def}} [A \wedge \phi] \vee [A \wedge B] \vee [\phi \wedge B].$$[2]

One can easily verify that for any assignment V_j, $V_j (A_t \downarrow B_t) = V_j (A) \wedge V_j (B)$, i.e., the *infimum* of $V_j (A)$ and $V_j (B)$. So, $V_j (A_t \downarrow B_t)$ is the total Boolean value of A_t and B_t if they have both the same total value, and is $\bot$ otherwise. If we add that

$$\delta_{p_i}([X_{\alpha_i}(\vec{g})^c]) = \text{V if } p_i(\vec{g}) = \bot,$$

then the following formula **F2** is the partial version of **F1**[3].

Proposition

$$\lambda X_{\alpha_1}\ldots\lambda X_{\alpha_n}\Big[\Big[\bigvee_{\vec{p}\,\in\,\mathbf{1}(f)}\Big(\bigwedge_{\vec{g}\in Pr(\alpha_1)}\delta_{p_1}([X_{\alpha_1}(\vec{g})^c])\wedge\ldots\wedge\bigwedge_{\vec{g}\in Pr(\alpha_n)}\delta_{p_n}([X_{\alpha_n}(\vec{g})^c])\Big)\Big]$$
$$\downarrow\Big[\neg\bigvee_{\vec{p}\,\in\,\mathbf{0}(f)}\Big(\bigwedge_{\vec{g}\in Pr(\alpha_1)}\delta_{p_1}([X_{\alpha_1}(\vec{g})^c])\wedge\ldots\wedge\bigwedge_{\vec{g}\in Pr(\alpha_n)}\delta_{p_n}([X_{\alpha_n}(\vec{g})^c])\Big)\Big]\Big]$$

is a canonical name of any partial monotonic f (with the convention that when $\mathbf{1}(f)$ or $\mathbf{0}(f)$ is empty, the default value is F).

3. Four-value Monotonic Protothetics

Let's go a step further and let's introduce $\top$, the "top".

Definition 12 For any $\alpha \in T$, the set FM_α of partial functions of type α is recusively dedined as follows :

(i) $FM_t = \{\bot, 0, 1, \top\}$;
(ii) $FM_{\alpha\beta} = (FM_\alpha \rightarrow FM_\beta)$.

T is a fourth Boolean value which strictly dominates 0 and 1. FM_t is then the lattice BOOL of Dana Scott (1973). By a classical proof we know that for every type α, the set

---

[2] Of course, we could introduce $\downarrow$ as primitive instead, and then define ϕ as [T $\downarrow$ F]. But neither ϕ nor $\downarrow$ is definable by means of classical resources.
[3] **F2** is a generalisation of Thijsse's formula (in E. Thijsse (1992)), which is a simplification of Blamey's formula (in S. Blamey (1986)).

FM_α is a complete lattice. Most of the definitions concerning partial monotonic functions are easily extended to monotonic functions of FM_α.

We can define values for conjunction and disjunction which are quite intuitive, and are extensions of Kleene strong connectives in the following sense.

(1) When the arguments are taken from $\{0, 1, \bot\}$, then the value is the strong Kleene value;
(2) All the other values are classical inasmuch as monotonicity is preserved.

The truth tables of $\neg$, $\wedge$ and $\vee$ are then

$\neg$	
$\bot$	$\bot$
0	1
1	0
$\top$	$\top$

$\wedge$	$\bot$	0	1	$\top$
$\bot$	$\bot$	0	$\bot$	0
0	0	0	0	0
1	$\bot$	0	1	$\top$
$\top$	0	0	$\top$	$\top$

$\vee$	$\bot$	0	1	$\top$
$\bot$	$\bot$	$\bot$	1	$\bot$
0	$\bot$	0	1	$\top$
1	1	1	1	1
$\top$	1	$\top$	1	$\top$

Notice that De Morgan laws hold according to those definitions, so that $A \vee B$ may be considered as an abbreviation of $\neg[\neg A \wedge \neg B]$.

One more time, it is possible to generalize van Benthem's formula. However, if we try the formula **F2**, the result is not in general the name of the intended function but of another function. One reason why that does not work is that **F2** is the infimum of two formulas. The left one describes the lines where the formula is true and the right one describes those where the formula is false. The default values being respectively 0 and 1, when a line is not described the default values appear and the infimum is $\bot$. Obviously, with four values, we need a much more sophisticated device.

Following Muskens (1989), we introduce ψ as a name of $\top$ and a new operator @ defined as

@	$\bot$	1	0	$\top$
$\bot$	$\bot$	$\bot$	0	0
1	1	1	$\top$	$\top$
0	$\bot$	$\bot$	0	0
$\top$	1	1	$\top$	$\top$

We define δ_{p_i}:

If $f(\vec{p}) \in \{0, 1, \top\}$, then

$$\delta_{p_i}([X_{\alpha_i}(\vec{g})^c]) = [X_{\alpha_i}(\vec{g})^c)] \qquad \text{if } p_i(\vec{g}) = 1;$$

$$= \neg[X_{\alpha_i}(\vec{g})^c)] \qquad \text{if } p_i(\vec{g}) = 0;$$

$$= [X_{\alpha_i}(\vec{g})^c)] \wedge \neg[X_{\alpha_i}(\vec{g})^c)] \qquad \text{if } p_i(\vec{g}) = \top$$

$$= V \qquad \text{if } p_i(\vec{g}) = \bot.$$

If $f(\vec{p}) = \bot$, then $\delta_{p_i}([X_{\alpha_i}(\vec{g})^c]) = F$.

The following term denotes f:

$$\lambda X_{\alpha 1} \ldots \lambda X_{\alpha n}[[\bigvee_{\vec{p} \in \bot(f) \cup \mathbf{1}(f) \cup \top(f)} (\bigwedge_{\vec{g} \in Pr(\alpha 1)} \delta_{p1}([X_{\alpha 1}(\vec{g})^c]) \wedge \ldots \wedge \bigwedge_{\vec{g} \in Pr(\alpha n)} \delta_{pn}([X_{\alpha n}(\vec{g})^c]))]$$

$$@\neg[\bigvee_{\vec{p} \in \bot(f) \cup \mathbf{0}(f) \cup \top(f)} (\bigwedge_{\vec{g} \in Pr(\alpha 1)} \delta_{p1}([X_{\alpha 1}(\vec{g})^c]) \wedge \ldots \wedge \bigwedge_{\vec{g} \in Pr(\alpha n)} \delta_{pn}([X_{\alpha n}(\vec{g})^c]))]]$$

4. About the definability of the 'usual' logical operators

One interesting question now concerns the definability of operators like identity and the quantifiers. Let's consider first the 3-value case.

We have shown elsewhere (F. Lepage (1995)) that the strongest 3-value monotonic identity is the following. First, we need the notion of *total object.*

Definiton 13 For any $\alpha \in T$, the set PT_α of total objects of type α is the following

(i) $PT_t = \{0,1\}$;
(ii) $PT_{\alpha\beta}$ is the set of all the $f \in PM_{\alpha\beta}$ such that, for any $a \in PT_\alpha$, $f(a) \in PT_\beta$.

Two relations are then introduced.

Definition 14 Two object a and b are *weakly equivalent* (we write $a =^* b$) iff

(i) for $a, b \in PM_t$, $a =^* b$ iff $a = b$;
(ii) for $a, b \in PM_{\alpha\beta}$, $a =^* b$ iff for any $c \in PT_\alpha$, $a(c) =^* b(c)$.

It can be shown that, for $a, b \in PT_\alpha$, $a =^* b$ iff $a \vee b$ exists. $a =^* b$ can thus be seen as a kind of compatibility.

Definition 15 Two object a and b are *strongly different* (we write $a \neq^* b$) iff

(i) for $a, b \in PM_t$, $a \neq^* b$ iff $a \neq b$, $a \neq \bot$ and $b \neq \bot$
(ii) for $a, b \in PM_{\alpha\beta}$, $a \neq^* b$ iff there is a $c \in PM_\alpha$ such that $a(c) \neq^* b(c)$.

Using these two relations we can define identity.

Definition 16 The relation of monotonic identity I_α between objects of type α (i.e., I_α is of type $<\alpha<\alpha t>>$) is

$$I_\alpha(a,b) = 1 \text{ iff } a =^* b \text{ and } a, b \in PT_\alpha$$
$$= 0 \text{ iff } a \neq^* b$$
$$= \varphi \text{ otherwise.}$$

Since we have a name for every object, we have a name for every I_α. Once we have a name for such a function, let's say $\equiv_\alpha$, we can define the universal quantifier as

$$\forall X_\alpha A_t =_{\text{def}} [\lambda X_\alpha A \equiv_{\alpha t} \lambda X_\alpha \mathrm{T}].$$

It is worth noting that with $\equiv_\alpha$ having the truth conditions of I_α defined above, we have

$$\mathrm{V}j(\forall X_\alpha A_t) = 1 \text{ iff for every } a \in PT_\alpha, \mathrm{V}j_{(a/X)}(A) = 1$$
$$= 0 \text{ iff there is an } a \in PT_\alpha \text{ such that } \mathrm{V}j_{(a/X)}(A) = 0$$
$$= \bot \text{ otherwise.}$$

These truth conditions are the *strongest possible* for the universal quantifier, i.e., there is no monotonic functor that stricly dominates this one that behaves as the universal quantifier when the arguments are total. Moreover, with $\exists X_\alpha A_t =_{\text{def}} \neg\forall X_\alpha \neg A$, the same is true for $\exists$.

But having a canonical name for every I_α is not enough at least for some purposes. The lack of explicit recursive definition of identity could be an empediment to the elaboration of a finite axiomatization of 3-value monotonic protothetics. This can be the case if we have to introduce axioms or rules for every identity of every type.

The question is : can we design a formula denoting I_α which relies recursively on (1) identity on lower types, (2) canonical names of objects of lower types and (3) Boolean operators as defined above? The simpler formula is *prima facie*

$$[A_{\alpha\beta} \equiv_{\alpha\beta} B_{\alpha\beta}] =_{\text{def}} \forall X_\alpha [A\ X \equiv_\beta BX].$$

But that does not work because the definition of $\forall X_\alpha C_t$:

$$\forall X_\alpha C_t =_{\text{def}} [\lambda X_\alpha C \equiv_{\alpha t} \lambda X_\alpha \mathrm{T}]$$

uses the type αt which is not lower than $\alpha\beta$.

Unfortunatly, there does not seem to be a natural and very simple definition of $\forall$. The following one works but is not very elegant.

Firstly, we define the class of canonical names of *total* objects.

Definition 17 Let C_α be the set of canonical names of objects of type α. For any $\alpha \in T$, the set TC_α of canonical names of *total* objects of type α is the following

(i) $TC_t = \{\mathrm{T}, \mathrm{F}\}$;
(ii) $TC_{\alpha\beta} = \{A \in C_{\alpha\beta} :$ for any $B \in TC_\alpha$, $([AB]^d)^c \in TC_\beta\}$.

We can then define :

$$\forall X_\alpha A =_{\text{def}} \bigwedge_{a^c \in TC_\alpha} A\langle a^c/X\rangle$$

$$[A_t \equiv B_t] =_{\text{def}} [[A \wedge B] \vee [\neg A \wedge \neg B]]$$

$$[A_{\alpha\beta} \equiv_{\alpha\beta} B_{\alpha\beta}] =_{\text{def}} \forall X_\alpha [[AX] \equiv_\beta [BX]]$$

Finally, one can then introduce in the language a functor whose interpretation is 'to be total' :

$$\mathfrak{J}(A_\alpha) =_{\text{def}} \exists X_\alpha [A \equiv_\alpha X].$$

For the definition of validity as 'true for any assigment', a complete system can be provided.

What about for the 4-value case? The situation is much more intricate. As in the three value case, we need to define objects that behave like classical objects.

Definition 18 The set of *pseudo-classical* objects of type α, is the smallest set PS_α such that

(i) $PS_t = \{0,1\}$

(ii) $PS_{\alpha\beta}$ is the set of $f \in PM_{\alpha\beta}$ such that for any $a \in PS_\alpha, f(a) \in PS_\beta$.

Pseudo-classical functions behave like classical objects when their arguments are themselves pseudo-classical.

As in the three values case, let NC_α be the set of canonical names of the four values objects.

Definition 19 For any type α, the set PC_α of canonical names of type of pseudo-classical objects of type α is

(i) $PC_t = \{\mathrm{T}, \mathrm{F}\}$;
(ii) $PC_{\alpha\beta} = \{A \in NC_{\alpha\beta}$: for any $B \in PC_\alpha$, $([AB]^d)^c \in Pc_\beta\}$.

We can define the universal quantifier.

Definition 20

$$\forall X_\alpha A =_{\mathrm{def}} \bigwedge_{a^c \in PC_\alpha} A<a^c/X>$$

This definition brings us an unpleasant surprise. According to it, the truth conditions of $\forall X_\alpha A_t$ are

$\mathrm{V}j(\forall X_\alpha A_t) = 1$ iff $a \in PS_\alpha$, $\mathrm{V}j_{(a/X)}(A) = 1$;
$= 0$ iff there is an $a \in PS_\alpha$ such that $\mathrm{V}j_{(a/X)}(A) = 0$
or if there is an a and a $b \in PS_\alpha$ such that $\mathrm{V}j_{(a/X)}(A) = \bot$ and $\mathrm{V}j_{(b/X)}(A) = \top$
$= \bot$ if there is an $a \in PS_\alpha$ such that $\mathrm{V}j_{(a/X)}(A) = \bot$ and for any $b \in PS_\alpha$, $\mathrm{V}j_{(b/X)}(A) = \bot$ or $\mathrm{V}j_{(b/X)}(A) = 1$
$= \top$ elsewhere.

The second clause is the bad news because we would like to have an universal quantifier which behave pseudo-classically, i.e., we would like the universal quantifier to obey following condition

C condition

$Vj(\forall X_\alpha A_t) = 1$ iff for any $a \in PS_\alpha$, $Vj_{(a/X)}(A) = 1$
$Vj(\forall X_\alpha A_t) = 0$ iff there is at least an $a \in PS_\alpha$ such that $Vj_{(a/X)}(A) = 0$.

Unfortunatly, it is not possible to define such a quantifier because

(1) no function $f: PM_t^4 \to PM_t$ such that

$f(a_1, a_2, a_3, a_4) = 1$ iff $a_1 = a_2 = a_3 = a_4 = 1$
$f(a_1, a_2, a_3, a_4) = 0$ iff $a_1 = 0$ or $a_2 = 0$ or $a_3 = 0$ or $a_4 = 0$
is monotonic;

(2) with a quantifier $\forall$ satisfying the C condition, we can define an operator

$$W(A(\top), A(0), A(1), A(\bot)) =_{\text{déf}} \forall X_t A_t$$

and the value of W will have the property expressed in (1). The generalization to higher types is trivial.

For the same reasons, it is not possible to define in this logic an existential quantifier such that $Vj(\exists XA) = 1$ is true iff there is at least one pseudo-classical a such that $Vj_{(a/X)}(A) = 1$. These properties are related to the truth tables of $\vee$ and $\wedge$. The real problem is that it is not possible to fill up the following table

$\wedge$	$\bot$	0	1	T
$\bot$	?	0	?	?
0	0	0	0	0
1	?	0	1	?
T	?	0	?	?

without using more 0 (and obtaining a monotonic connector) nor to fill up the following table

$\vee$	$\bot$	0	1	T
$\bot$	?	?	1	?

0	?	0	1	?
1	1	1	1	1
T	?	?	1	?

without using more 1.

All this raises serious doubts about the very possibility of a four-value monotonic logic.

Bibliography

Andrews, P. B., 'A Reduction of the Axioms for the Theory of Propositional Types', *Fundamenta Mathematicæ* LII, 1963, 345-350.

van Benthem, J., *Language in Action*, Amsterdam, North-Holland/Cambridge, MIT Press, 1995.

Blamey, S., 'Partial Logic', in *Handbook of Philosophical Logic Vol. III*, Gabbay, D., *et al* (eds), Dordrecht, Reidel, 1986, 1-70.

Grzegorezyk, A. 'The Systems of Leś niewski in Relation to Contemporary Logical Research', *Studia Logica* 3, 1955, 77-95.

Henkin, L., 'A Theory of Propositionnal Types', *Fundamenta Mathematicæ*, 52, 1963, 321-344.

Lapierre, S., 'A Functional Partial Semantics for Intensional Logic', *Notre Dame Journal of Formal Logic*, Vol. 33, No 4, 1992, 517-541.

Lepage, F. 'Partial Functions in Type Theory', *Notre Dame Journal of Formal Logic*, Vol. 33, No 4, 1992, 493-516.

Lepage, F. 'Partial Propositional Logic', in M. Marion and R. S. Cohen (eds.), *Québec Studies in the Philosophy of Science I*, Kluwer Academic Publishers, 1995, 23-39.

Muskens, R. *Meaning and Partiality*, Ph.D. Dissertation, University of Amsterdam, 1989.

Scott, D., 'Models for Various Type-free Calculi', in P. Suppes et al. (eds) *Logic, Methodology and Philosophy of Science IV*, Amsterdam, North-Holland, 1973, 157-187.

Sł upecki, J., 'St. Leś niewski protothetics', *Studia Logica*, 1, 1953, 44-111.

Thijsse, G. C. E., *Partial Logic and Knowledge Representation*, Delft, Eburon Publishers, 1992.

institute *for* logic, language *and* computation

ILLC Research Reports and Technical Notes

Coding for Reports and Dissertations: *Series-Year-Number*, with LP = Logic, Philosophy and Linguistics; ML = Mathematical Logic and Foundations; CL = Computational Linguistics; CT = Computation and Complexity Theory; X = Technical Notes; DS = Dissertations.
All previous ILLC-publications are available from the ILLC bureau. For prepublications before 1994, contact the bureau.

CT-95-01 Marianne Kalsbeek, Yuejun Jiang *A Vademecum of Ambivalent Logic*

CT-95-02 Leen Torenvliet, Marten Trautwein *A Note on the Complexity of Restricted Attribute-Value Grammars*

CT-95-03 Krzysztof Apt, Ingrid Luitjes *Verification of Logic Programs with Delay Declarations*

CT-95-04 Paul Vitányi *Randomness*

CT-95-05 Joeri Engelfriet *Minimal Temporal Epistemic Logic*

CT-95-06 Krzysztof Apt, Rachel Ben-Eliyahu *Meta-variables in Logic Programming, or the Praise of Ambivalent Syntax*

CT-95-07 Frans Voorbraak *Combining unreliable pieces of evidence*

LP-95-01 Marten Trautwein *Assessing Complexity Results in Feature Theories*

LP-95-02 S.T. Baban, S. Husein *Programmable Grammar of the Kurdish Language*

LP-95-03 Kazimierz Świrydowicz *There exist exactly two Maximal Strictly Relevant Extensions of the Relevant Logic R**

LP-95-04 Jaap van der Does, Henk Verkuyl *Quantification and Predication*

LP-95-05 Nataša Rakić *Past, Present, Future and Special Relativity*

LP-95-06 David Beaver *An Infinite Number of Monkeys*

LP-95-07 Paul Dekker *The Values of Variables in Dynamic Semantics*

LP-95-08 Jaap van der Does, Jan van Eijck *Basic Quantifier Theory*

LP-95-09 Jeroen Groenendijk, Marin Stokhof, Frank Veltman *Coreference and Modality*

LP-95-10 Jeroen Groenendijk, Martin Stokhof, Frank Veltman *Coreference and Contextually Restricted Quantification*

LP-96-01 Renate Bartsch *Understanding Understanding*

LP-96-02 David Beaver *Presupposition*

LP-96-03 Theo M.V. Janssen *Compositionality*

LP-96-04 Reinhard Muskens, Johan van Benthem, Albert Visser *Dynamics*

LP-96-05 Dick de Jongh, Makoto Kanazawa *Angluin's theorem for indexed families of r.e. sets and applications*

LP-96-06 François Lepage, Serge Lapierre *The Functional Completeness of 4-value Monotonic Protothetics*

ML-95-01 Michiel van Lambalgen *Randomness and Infinity*

ML-95-02 Johan van Benthem, Giovanna D'Agostino, Angelo Montanari, Alberto Policriti *Modal Deduction in Second-Order Logic and Set Theory*

ML-95-03 Vladimir Kanovei, Michiel van Lambalgen *On a Spector Ultrapower of the Solovay Model*

ML-95-04 Hajnal Andréka, Johan van Benthem, István Németi *Back and Forth between Modal Logic and Classical Logic*

ML-95-05 Natasha Alechina, Michiel van Lambalgen *Generalized Quantification as Substructural Logic*

ML-95-06 Dick de Jongh, Albert Visser *Embeddings of Heyting Algebras (revised version of ML-93-14)*

ML-95-07 Johan van Benthem *Modal Foundations of Predicate Logic*

ML-95-08 Eric Rosen *Modal Logic over Finite Structures*

ML-95-09 Hiroakira Ono *Decidability and finite model property of substructural logics*

ML-95-10 Alexei P. Kopylov *The undecidability of second order linear affine logic*

ML-96-01 Domenico Zambella *Algebraic Methods and Bounded Formulas*

ML-96-02 Domenico Zambella *On Forcing in Bounded Arithmetic*

ML-96-03 Hajnal Andréka, Johan van Benthem & István Németi *Modal Languages and Bounded Fragments of Predicate Logic*

ML-96-04 Kees Doets *Proper Classes*

ML-96-05 Søren Riis *Count(q) versus the Pigeon-Hole Principle*
X-95-01 Sophie Fischer, Leen Torenvliet *The Malleability of* TSP_{2Opt}
X-96-01 Ingmar Visser *Mind Rules: a philosophical essay on psychological rules and the rules of psychology*
DS-95-01 Jacob Brunekreef *On Modular Algebraic Protocol Specification*
DS-95-02 Andreja Prijatelj *Investigating Bounded Contraction*
DS-95-03 Maarten Marx *Algebraic Relativization and Arrow Logic*
DS-95-04 Dejuan Wang *Study on the Formal Semantics of Pictures*
DS-95-05 Frank Tip *Generation of Program Analysis Tools*
DS-95-06 Jos van Wamel *Verification Techniques for Elementary Data Types and Retransmission Protocols*
DS-95-07 Sandro Etalle *Transformation and Analysis of (Constraint) Logic Programs*
DS-95-08 Natasha Kurtonina *Frames and Labels. A Modal Analysis of Categorial Inference*
DS-95-09 G.J. Veltink *Tools for PSF*
DS-95-10 Giovanna Cepparello *Studies in Dynamic Logic*
DS-95-11 W.P.M. Meyer Viol *Instantial Logic. An Investigation into Reasoning with Instances*
DS-95-12 Szabolcs Mikulás *Taming Logics*
DS-95-13 Marianne Kalsbeek *Meta-Logics for Logic Programming*
DS-95-14 Rens Bod *Enriching Linguistics with Statistics: Performance Models of Natural Language*
DS-95-15 Marten Trautwein *Computational Pitfalls in Tractable Grammatical Formalisms*
DS-95-16 Sophie Fischer *The Solution Sets of Local Search Problems*
DS-95-17 Michiel Leezenberg *Contexts of Metaphor*
DS-95-18 Willem Groeneveld *Logical Investigations into Dynamic Semantics*
DS-95-19 Erik Aarts *Investigations in Logic, Language and Computation*
DS-95-20 Natasha Alechina *Modal Quantifiers*
DS-96-01 Lex Hendriks *Computations in Propositional Logic*
DS-96-02 Erik de Haas *Categories for Profit*
DS-96-03 Martin H. van den Berg *Some Aspects of the Internal Structure of Discourse: the Dynamics of Nominal Anaphora*