To appear in Studia Logica, vol. 62 no. 2 (1999)

Monotone majorizable functionals

Helmut Schwichtenberg

Abstract

Several properties of monotone functionals (MF) and monotone majorizable functionals (MMF) used in the earlier work by the author and van de Pol are proved. It turns out that the terms of the simply typed lambda-calculus define MF, but adding primitive recursion, and even monotonic primitive recursion changes the situation: already $\lambda Z.Z(1-\mathrm{sg})$ is not MMF. It is proved that extensionality is not Dialectica-realizable by MMF, and a simple example of a MF which is not hereditarily majorizable is given.

Keywords: Monotone functionals, monotone majorizable functionals, hereditarily majorizable functionals, simply typed lambda-calculus, extensionality, Dialectica interpretation

1 Introduction

For higher order functionals there is a rather natural notion of monotonicity. In the present paper we adress a question asked by Mints (at the International Congress for Logic, Methodology and the Philosophy of Science in Florence 1995): How do monotone functionals relate to Howard's hereditarily majorizable functionals [2, p.454]?

Clearly the two notions to be compared should be the monotone *majorizable* and the hereditarily majorizable functionals. The first and easy observation is that not every monotone majorizable functional is hereditarily majorizable; this will be shown by means of a counterexample below. Next we go back to Howard's original motivation for introducing the hereditarily majorizable functionals: he used them as a tool to prove that "the Dialectica interpretation of the simplest nontrivial case $\forall y^2 E_2(y)$ of the axiom of extensionality cannot be carried out by use of a primitive recursive functional". We will show that the same holds if we replace 'primitive recursive' by 'monotone majorizable'. Since there are monotone majorizable functionals that are not hereditarily majorizable and hence not primitive recursive, we obtain a proposition not contained in Howard's. However, not every primitive recursive functional is monotone majorizable (this was conjectured in a first draft of this paper and recently proved by van de Pol); hence our result is independent of Howard's.

We also give a characterization of the monotone majorizable functionals by means of a variant of another notion introduced by Howard: that of a *compact* set of functionals; see [2, p.164]. Call a set $\mathcal{A}$ of functionals of type $\rho = \rho_1 \to \cdots \to \rho_n \to \iota$ *$\mathcal{M}$-compact* ($\mathcal{M}$ for monotone) if for all $\mathcal{M}$-compact sets $\mathcal{B}_1, \dots, \mathcal{B}_n$ of monotone functionals of types $\rho_1, \dots, \rho_n$ the set $\mathcal{A}\mathcal{B}_1 \dots \mathcal{B}_n$ is finite. Then a functional F is monotone majorizable if and only if the singleton set $\{F\}$ is $\mathcal{M}$-compact. A similar characterization of the hereditarily majorizable functionals has been given by Bezem in [1, p.276].

It seems to be an interesting problem to study what happens if everything is based of the total continuous functionals (cf. the book of Stoltenberg-Hansen et al. [3] for the relevant theory), instead of the full type structure $(\mathcal{T}_\rho)$. Some of the counterexamples below make heavy use of non-continuous functionals.

2 Monotone functionals

As semantic domains we take the full type structure over the natural numbers.

$$\mathcal{T}_\iota := \mathbb{N},$$
$$\mathcal{T}_{\rho\to\sigma} := \{\, \mathbf{f} \mid \mathbf{f}\colon \mathcal{T}_\rho \to \mathcal{T}_\sigma \,\}.$$

We also use the natural order $>_\iota$ on $\mathbb{N}$. The interpretation of a term r in the type structure $(\mathcal{T}_\rho)_\rho$ under an environment U is denoted by $[\![r]\!]_U$.

We begin with the notion of a monotone functional, following [4, 5].

Definition. *For every type ρ we define the set $\mathcal{M}_\rho \subseteq \mathcal{T}_\rho$ of* monotone functionals *of type ρ and simultaneously a relation $\geq_\rho$ on $\mathcal{T}_\rho$.*

(i) $\mathbf{f} \in \mathcal{M}$ iff $\mathbf{f} \in \mathcal{T}$ and in case $\mathbf{f}$ is of arrow type $\rho \to \sigma$:

$$\forall \mathbf{x}{\in}\mathcal{M}_\rho(\mathbf{f}(\mathbf{x}) \in \mathcal{M}_\sigma) \quad \textit{and} \quad \forall \mathbf{x}, \mathbf{y}{\in}\mathcal{M}_\rho(\mathbf{x} \geq_\rho \mathbf{y} \implies \mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{f}(\mathbf{y})).$$

(ii) $\mathbf{f} \geq \mathbf{g}$ iff $\mathbf{f}, \mathbf{g} \in \mathcal{T}$ are of the same type and in case $\mathbf{f}$ is of

- *ground type ι: $\mathbf{f} \geq_\iota \mathbf{g}$;*
- *arrow type $\rho \to \sigma$: $\forall \mathbf{x}{\in}\mathcal{M}_\rho(\mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{g}(\mathbf{x}))$.*

Note that $\geq_\rho$ only is a quasi order on $\mathcal{T}_\rho$: from $\mathbf{x} \geq \mathbf{y} \geq \mathbf{x}$ we cannot conclude $\mathbf{x} = \mathbf{y}$.

The previous definition can also be written more compactly as follows. Let $\mathcal{M} := \bigcup_\rho \mathcal{M}_\rho$.

Lemma. *For all $\mathbf{f}, \mathbf{g} \in \mathcal{T}_\rho$,*

(i) $\mathbf{f} \in \mathcal{M}$ iff $\forall \vec{\mathbf{x}}, \vec{\mathbf{y}}{\in}\mathcal{M}(\vec{\mathbf{x}} \geq \vec{\mathbf{y}} \implies \mathbf{f}(\vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\vec{\mathbf{y}}))$.

(ii) $\mathbf{f} \geq \mathbf{g}$ iff $\forall \vec{\mathbf{x}}{\in}\mathcal{M}(\mathbf{f}(\vec{\mathbf{x}}) \geq_\iota \mathbf{g}(\vec{\mathbf{x}}))$.

Here $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ only range over vectors for which $\mathbf{f}(\vec{\mathbf{x}})$ and $\mathbf{f}(\vec{\mathbf{y}})$ are of base type. $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$ means: For all i we have $\mathbf{x}_i \geq \mathbf{y}_i$

Proof. (ii). By induction on the type. $\Longrightarrow$. Assume $\mathbf{f} \geq_{\rho\to\sigma} \mathbf{g}$, and let $\mathbf{x}, \vec{\mathbf{x}} \in \mathcal{M}$. Then $\mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{g}(\mathbf{x})$ by definition, hence $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\sigma \mathbf{g}(\mathbf{x}, \vec{\mathbf{x}})$ by induction hypothesis. $\Longleftarrow$. Assume $\forall \mathbf{x}, \vec{\mathbf{x}}{\in}\mathcal{M}(\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{g}(\mathbf{x}, \vec{\mathbf{x}}))$. For $\mathbf{x} \in \mathcal{M}_\rho$ we show $\mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{g}(\mathbf{x})$ using the induction hypothesis. So let $\vec{\mathbf{x}} \in \mathcal{M}$. Then $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\sigma \mathbf{g}(\mathbf{x}, \vec{\mathbf{x}})$ by assumption.

(i). By induction on the type, using (ii). $\Longrightarrow$. Assume $\mathbf{f} \in \mathcal{M}_{\rho\to\sigma}$, and let $\mathbf{x}, \vec{\mathbf{x}}, \mathbf{y}, \vec{\mathbf{y}} \in \mathcal{M}$ such that $\mathbf{x}, \vec{\mathbf{x}} \geq \mathbf{y}, \vec{\mathbf{y}}$. We must show $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\mathbf{y}, \vec{\mathbf{y}})$. From $\mathbf{f} \in \mathcal{M}_{\rho\to\sigma}$ we obtain $\mathbf{f}(\mathbf{x}) \in \mathcal{M}_\sigma$ and $\mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{f}(\mathbf{y})$ (since $\mathbf{x} \geq_\rho \mathbf{y}$). Hence by induction hypothesis $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\mathbf{x}, \vec{\mathbf{y}})$, and by (ii) $\mathbf{f}(\mathbf{x}, \vec{\mathbf{y}}) \geq_\iota \mathbf{f}(\mathbf{y}, \vec{\mathbf{y}})$. $\Longleftarrow$. Assume $\forall \mathbf{x}, \vec{\mathbf{x}}, \mathbf{y}, \vec{\mathbf{y}}{\in}\mathcal{M}(\mathbf{x}, \vec{\mathbf{x}} \geq \mathbf{y}, \vec{\mathbf{y}} \implies \mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\mathbf{y}, \vec{\mathbf{y}}))$. We first show $\forall \mathbf{x}{\in}\mathcal{M}_\rho(\mathbf{f}(\mathbf{x}) \in \mathcal{M}_\sigma)$. So let $\mathbf{x} \in \mathcal{M}_\rho$. We show $\mathbf{f}(\mathbf{x}) \in \mathcal{M}_\sigma$ using the induction hypothesis. So let $\vec{\mathbf{x}}, \vec{\mathbf{y}} \in \mathcal{M}$ with $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$ be given. We must show $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\mathbf{x}, \vec{\mathbf{y}})$. But this follows from our assumption, since $\mathbf{x}, \vec{\mathbf{x}} \geq \mathbf{x}, \vec{\mathbf{y}}$. We now show $\forall \mathbf{x}, \mathbf{y}{\in}\mathcal{M}_\rho(\mathbf{x} \geq_\rho \mathbf{y} \implies \mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{f}(\mathbf{y}))$. So let $\mathbf{x}, \mathbf{y} \in \mathcal{M}_\rho$ with $\mathbf{x} \geq_\rho \mathbf{y}$ be given. We show $\mathbf{f}(\mathbf{x}) \geq_\sigma \mathbf{f}(\mathbf{y})$ using (ii). So let $\vec{\mathbf{x}} \in \mathcal{M}$. Then $\mathbf{x}, \vec{\mathbf{x}} \geq \mathbf{y}, \vec{\mathbf{x}}$, hence $\mathbf{f}(\mathbf{x}, \vec{\mathbf{x}}) \geq_\iota \mathbf{f}(\mathbf{y}, \vec{\mathbf{x}})$ by our assumption. $\square$

Lemma. *For every term r of the simply typed λ-calculus we have*

(i) $[\![r]\!]_U \in \mathcal{M}$ for every monotone environment U (i.e. mapping variables to monotone functionals).

(ii) $U \geq V \implies [\![r]\!]_U \geq [\![r]\!]_V$ *for monotone environments* U, V*.*

Proof. By induction on r we show $[\![r]\!]_U \vec{\mathbf{x}} \geq [\![r]\!]_V \vec{\mathbf{y}}$ for monotone environments $U \geq V$ and monotone arguments $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$; this clearly implies (i) and (ii). We may assume that r is in long normal form, i.e. $r = \lambda\vec{x}.z\vec{r}$. Let $U' := U[\vec{x} \mapsto \vec{\mathbf{x}}]$ and $V' := V[\vec{x} \mapsto \vec{\mathbf{y}}]$.

$$
\begin{aligned}
[\![\lambda\vec{x}.z\vec{r}]\!]_U(\vec{\mathbf{x}}) &= [\![z\vec{r}]\!]_{U'} \\
&= U'(z)[\![\vec{r}]\!]_{U'} \\
&\geq V'(z)[\![\vec{r}]\!]_{U'} && \text{since } U'(z) \geq V'(z) \text{ and } [\![\vec{r}]\!]_{U'} \in \mathcal{M} \text{ by IH(i)} \\
&\geq V'(z)[\![\vec{r}]\!]_{V'} && \text{since } [\![\vec{r}]\!]_{U'} \geq [\![\vec{r}]\!]_{V'} \text{ by IH(ii)} \\
&= [\![z\vec{r}]\!]_{V'} \\
&= [\![\lambda\vec{x}.z\vec{r}]\!]_V(\vec{\mathbf{y}}) \quad \square
\end{aligned}
$$

Definition. $\mathbf{f}$ *is monotone majorizable iff* $\mathbf{g} \geq \mathbf{f}$ *for some monotone* $\mathbf{g}$*.*

3 $\mathcal{M}$-compact functionals

We now give a characterization of the monotone majorizable functionals as those that are '$\mathcal{M}$-compact'. A similar characterization of the hereditarily majorizable functionals has been given by Bezem in [1].

Definition. *A set* $\mathcal{A}$ *of functionals is* $\mathcal{M}$*-compact iff*

$$\forall \vec{\mathcal{B}} \subseteq \mathcal{M} (\vec{\mathcal{B}}\ \mathcal{M}\text{-compact} \to \mathcal{A}(\vec{\mathcal{B}})\ \text{finite}).$$

Lemma. $\mathcal{A}$ *is* $\mathcal{M}$*-compact iff there is a monotone functional* $\mathbf{f}$ *such that* $\mathbf{f} \geq \mathcal{A}$

Proof. By induction on the type. $\Longleftarrow$. Let $\vec{\mathcal{B}} \subseteq \mathcal{M}$ be $\mathcal{M}$-compact. By induction hypothesis $\mathbf{x}_i \geq \mathcal{B}_i$ for some monotone $\mathbf{x}_i$. Then for every $\mathbf{g} \in \mathcal{A}$, $\mathbf{y}_i \in \mathcal{B}_i$: $\mathbf{f}(\vec{\mathbf{x}}) \geq \mathbf{f}(\vec{\mathbf{y}}) \geq \mathbf{g}(\vec{\mathbf{y}})$. Hence $\mathcal{A}(\vec{\mathcal{B}})$ is bounded by $\mathbf{f}(\vec{\mathbf{x}})$, i.e. finite.

$\Longrightarrow$. For monotone $\mathbf{x}$ define

$$\mathbf{x}{\downarrow} := \{\, \mathbf{y} \text{ monotone} \mid \mathbf{x} \geq \mathbf{y} \,\};$$

by induction hypothesis $\mathbf{x}{\downarrow}$ is $\mathcal{M}$-compact. Hence $\mathcal{A}(\vec{\mathbf{x}}{\downarrow})$ is finite for monotone $\vec{\mathbf{x}}$. Let $\mathbf{f}(\vec{\mathbf{x}}) := \max \mathcal{A}(\vec{\mathbf{x}}{\downarrow})$ for monotone $\vec{\mathbf{x}}$ and arbitrary otherwise.

$\mathbf{f}$ is monotone: Let $\vec{\mathbf{x}}, \vec{\mathbf{y}}$ be monotone, $\vec{\mathbf{x}} \geq \vec{\mathbf{y}}$. Then $\mathbf{x}_i{\downarrow} \supseteq \mathbf{y}_i{\downarrow}$, so $\mathcal{A}(\vec{\mathbf{x}}{\downarrow}) \supseteq \mathcal{A}(\vec{\mathbf{y}}{\downarrow})$, hence $\mathbf{f}(\vec{\mathbf{x}}) \geq \mathbf{f}(\vec{\mathbf{y}})$.

$\mathbf{f} \geq \mathcal{A}$: Let $\mathbf{g} \in \mathcal{A}$ and $\vec{\mathbf{x}}$ be monotone. Then $\mathbf{f}(\vec{\mathbf{x}}) = \max \mathcal{A}(\vec{\mathbf{x}}{\downarrow}) \geq \mathbf{g}(\vec{\mathbf{x}})$. $\square$

4 Dialectica interpretation of the extensionality axiom

Theorem. *Assume* $\mathcal{F}$ *satisfies the Dialectica interpretation of the extensionality axiom*

$$\forall n(\alpha(n) = \beta(n)) \to Y(\alpha) = Y(\beta). \qquad \textit{(Ext)}$$

Then $\mathcal{F}$ *is not monotone majorizable*

Proof. By assumption

$$Y(\alpha) \neq Y(\beta) \rightarrow \alpha(\mathcal{F}(Y, \alpha, \beta)) \neq \beta(\mathcal{F}(Y, \alpha, \beta)). \tag{$*$}$$

Assume for a contradiction that $\mathcal{G} \geq \mathcal{F}$ for some monotone $\mathcal{G}$. By the characterization of monotone majorizable functionals in the previous section we know that $\{\mathcal{F}\}$ is $\mathcal{M}$-compact, hence $\mathcal{F}(1^2{\downarrow}, 1^1{\downarrow}, 1^1{\downarrow})$ is finite, say $< k$. Define

$$X_k(\alpha) := \mathrm{sg}(\alpha(k)) \qquad \alpha_k(n) := \begin{cases} 0 & \text{if } n < k \\ 1 & \text{otherwise.} \end{cases}$$

Clearly $X_k \in 1^2{\downarrow}$, $\alpha_k \in 1^1{\downarrow}$ and $X_k(\alpha_k) = 1 \neq 0 = X_k(0^1)$, so by $(*)$ $\alpha_k(\mathcal{F}(X_k, \alpha_k, 0^1)) \neq 0$, hence $\mathcal{F}(X_k, \alpha_k, 0^1) \geq k$, a contradiction. □

5 Comparison with hereditarily majorizable functionals

Howard has introduced the hereditarily majorizable functionals in [2, p.455], as follows.

Definition. *For* $\mathbf{f}, \mathbf{g} \in \mathcal{T}_\rho$

$$\mathbf{f} \geq_{\mathrm{maj}} \mathbf{g} \ :\equiv \ \forall \vec{\mathbf{x}}, \vec{\mathbf{y}}(\vec{\mathbf{x}} \geq_{\mathrm{maj}} \vec{\mathbf{y}} \rightarrow \mathbf{f}(\vec{\mathbf{x}}) \geq_\iota \mathbf{g}(\vec{\mathbf{y}})).$$

$\mathbf{f} \geq_{\mathrm{maj}} \mathbf{g}$ *reads '*$\mathbf{f}$ *hereditarily majorizes* $\mathbf{g}$*'.* $\mathbf{f}$ *is called hereditarily majorizable iff* $\mathbf{g} \geq_{\mathrm{maj}} \mathbf{f}$ *for some* $\mathbf{g}$*.*

Note that by definition the hereditarily majorizable functionals are closed under application.

We now show by means of an example that not every monotone majorizable functional is hereditarily majorizable. Define

$$Y_0(\alpha) := \begin{cases} \text{greatest 0 of } \alpha & \text{if it exists} \\ 0 & \text{otherwise} \end{cases} \qquad Z_0(\alpha) := \begin{cases} 0^2 & \text{if } \alpha \text{ mon.} \\ Y_0 & \text{otherwise.} \end{cases}$$

Then the following can be seen easily.

1. Y_0 is *not* monotone majorizable.
   (*Proof.* $\{Y_0\}$ is not $\mathcal{M}$-compact, since $Y_0(1^1{\downarrow}) \ni Y_0(\alpha_{k+1}) = k$).

2. Z_0 is monotone.

3. The monotone majorizable functionals are *not* closed under application.
   (*Proof.* $Z_0(\overline{\mathrm{sg}}) = Y_0$).

4. Y_0 is *not* hereditarily majorizable.
   (*Proof.* Let $Y \geq_{\mathrm{maj}} Y_0$, i.e. $\forall \alpha^*, \alpha(\alpha^* \geq_{\mathrm{maj}} \alpha \rightarrow Y(\alpha^*) \geq Y_0(\alpha))$. $1^1 \geq_{\mathrm{maj}} \alpha_{k+1}$, hence $Y(1^1) \geq Y_0(\alpha_{k+1}) = k$. Contradiction).

5. Z_0 is *not* hereditarily majorizable.
   (*Proof.* $Z_0(\overline{\mathrm{sg}}) = Y_0$ and $\overline{\mathrm{sg}}$ is hereditarily majorizable).

6 Primitive recursive functionals

The terms of GÖDEL's system T are built from typed variables $x^\rho, y^\rho, \dots$ and constants 0^ι for zero, $\mathrm{S}^{\iota\to\iota}$ for successor and $\mathrm{R}^{\rho\to(\iota\to\rho\to\rho)\to\iota\to\rho}$ for the primitive recursion operators by λ-abstraction $(\lambda x^\rho r^\sigma)^{\rho\to\sigma}$ and application $(r^{\rho\to\sigma}s^\rho)^\sigma$. It is well-known that any reduction sequence w.r.t. $\beta\eta$-conversion and the conversion rules

$$\mathrm{R}rs0 \mapsto r$$
$$\mathrm{R}rs(\mathrm{S}t) \mapsto st(\mathrm{R}rst)$$

terminates with a uniquely determined normal form. For every term r and every environment U mapping variables x^ρ to functionals $\mathbf{x} \in \mathcal{T}_\rho$ we clearly have a value $[\![r]\!]_U$ in the full type structure $\mathcal{T}$ such that the values are not changed under conversion. The values of closed terms are called *primitive recursive functionals.*

We have seen already that any closed term r of the typed λ-calculus defines a monotone functional $[\![r]\!]$. This clearly does not hold any more if we allow primitive recursion as in GÖDEL's T: for instance $r_0 := \mathrm{R}1(\lambda x\lambda y0)$ defines $[\![r_0]\!] = \overline{\mathrm{sg}}$. One might think that replacing R by a monotone primitive recursion operators $\overline{\mathrm{R}}$ with conversion rules

$$\overline{\mathrm{R}}rs0 \mapsto r$$
$$\overline{\mathrm{R}}rs(\mathrm{S}t) \mapsto \mathrm{R}rs(\mathrm{S}t) + \overline{\mathrm{R}}rst$$

will transform any term r into a term $\overline{r}$ whose monotone value satisfies $[\![\overline{r}]\!]_U \geq [\![r]\!]_U$, for every monotone environment U. However, somewhat surprisingly this does not hold.

Counterexample. *Let U be the environment mapping the variable z of type $(\iota \to \iota) \to (\iota \to \iota) \to \iota$ to the monotone functional Z_0, and as above $r_0 := \mathrm{R}1(\lambda x\lambda y0)$. Then $[\![z\overline{r_0}]\!]_U \not\geq [\![zr_0]\!]_U$.*

Proof. $[\![z\overline{r_0}]\!]_U = Z_0(1^1) = 0^2$ and $[\![zr_0]\!]_U = Z_0(\overline{\mathrm{sg}}) = Y_0$, but we cannot have $0^2 \geq Y_0$ since Y_0 is not monotone majorizable. □

In a first draft of this paper we conjectured that not every primitive recursive functional is monotone majorizable. This was recently proved by VAN DE POL.

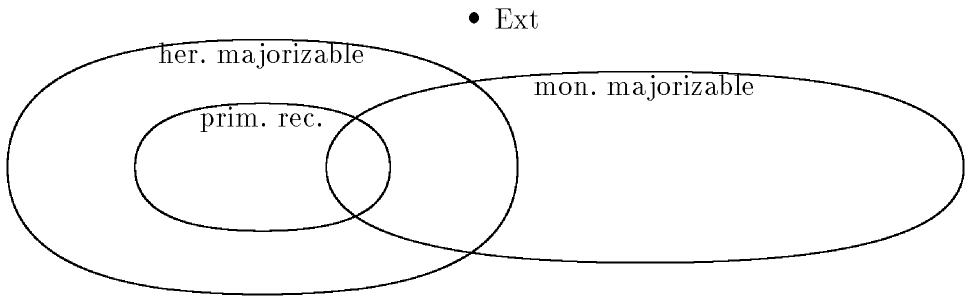
Theorem. *(*VAN DE POL*). The primitive recursive functional $\lambda Z.Z(\overline{\mathrm{sg}})$ is not monotone majorizable.*

Proof. Define $F_k\colon (\iota \to \iota) \to \iota$ by

$$F_k(\alpha) := \begin{cases} 0 & \text{if } \alpha(1) \geq \alpha(0) \\ k & \text{otherwise.} \end{cases}$$

Then for monotone α, β (with $\alpha \geq \beta$) we have $F_k(\alpha) = 0 = F_k(\beta)$, hence each F_k is monotone. Let $\mathcal{B} := \{\, F_k \mid k \in \mathbb{N} \,\}$. Then for every set $\mathcal{C}$ of monotone functions of type $\iota \to \iota$ we have $\mathcal{B}(\mathcal{C}) = \{0\}$, hence $\mathcal{B}$ is $\mathcal{M}$-compact. But $(\lambda Z.Z(\overline{\mathrm{sg}}))(\mathcal{B}) = \mathbb{N}$, hence $\{\, \lambda Z.Z(\overline{\mathrm{sg}}) \,\}$ is not $\mathcal{M}$-compact and therefore $\lambda Z.Z(\overline{\mathrm{sg}})$ is not monotone majorizable. □

We conclude with a picture showing how the monotone majorizable functionals relate to other well-known classes of functionals.



Here • denotes the Dialectica interpretation of the extensionality axiom.

References

[1] Marc Bezem. Compact and majorizable functionals of finite type. *The Journal of Symbolic Logic*, 54(1):271–280, March 1989.

[2] Anne S. Troelstra (editor). *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1973.

[3] Viggo Stoltenberg-Hansen, Edward Griffor, and Ingrid Lindström. *Mathematical Theory of Domains*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1994.

[4] Jaco van de Pol. Termination proofs for higher–order rewrite systems. In J. Heering, K. Meinke, B. Möller, and T. Nipkow, editors, *Higher–Order Algebra, Logic and Term Rewriting (HOA '93)*, volume 816 of *Lecture Notes in Computer Science*, pages 305–325. Springer Verlag, Berlin, Heidelberg, New York, 1994.

[5] Jaco van de Pol and Helmut Schwichtenberg. Strict functionals for termination proofs. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 350–364. Springer Verlag, Berlin, Heidelberg, New York, 1995.