On some self-referential four-valued languages

José Martínez Fernández
Logos - Universitat de Barcelona

July 14, 2013

Motivation for the fixed-point problem

- First-order language L.
  Expanded language L^+, adding a monadic predicate T.

Motivation for the fixed-point problem

- First-order language L.
  Expanded language L^+, adding a monadic predicate T.
- Ground model M for L. Can we extend M with an interpretation for T such that for any sentence A of L^+, $\text{val}_{M^+}(T\ulcorner A \urcorner) = \text{val}_{M^+}(A)$?

Motivation for the fixed-point problem

- First-order language L.
  Expanded language L^+, adding a monadic predicate T.
- Ground model M for L. Can we extend M with an interpretation for T such that for any sentence A of L^+, $\mathrm{val}_{M^+}(T\ulcorner A\urcorner) = \mathrm{val}_{M^+}(A)$?
- Classical logic: No, due to the paradoxes.

Motivation for the fixed-point problem

- First-order language L.
  Expanded language L^+, adding a monadic predicate T.
- Ground model M for L. Can we extend M with an interpretation for T such that for any sentence A of L^+, $\text{val}_{M^+}(T\ulcorner A \urcorner) = \text{val}_{M^+}(A)$?
- Classical logic: No, due to the paradoxes.
- A scheme of interpretation (say, the classical or strong Kleene one) has the *fixed-point property* when, for every ground model M there is an interpretation for truth.

Motivation for the fixed-point problem

- First-order language L.
  Expanded language L^+, adding a monadic predicate T.
- Ground model M for L. Can we extend M with an interpretation for T such that for any sentence A of L^+, $\text{val}_{M^+}(T\ulcorner A \urcorner) = \text{val}_{M^+}(A)$?
- Classical logic: No, due to the paradoxes.
- A scheme of interpretation (say, the classical or strong Kleene one) has the *fixed-point property* when, for every ground model M there is an interpretation for truth.
- **1.Theorem (Visser)**: Let E be the set of truth values. If $(E, \leq)$ is a ccpo and the logical connectives of a scheme are monotonic on $(E, \leq)$, then the scheme has the f.p.p.

The Kleene languages

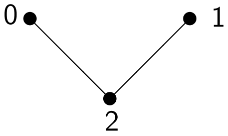
$E_3 = \{0, 1, 2\}$.

	$\neg_k$
0	1
1	0
2	2

$\wedge_s$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

$\wedge_w$	0	1	2
0	0	0	2
1	0	1	2
2	2	2	2

- The order of knowledge on E_3:



The Kleene languages

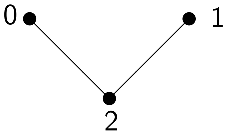
$E_3 = \{0, 1, 2\}$.

	$\neg_k$
0	1
1	0
2	2

$\wedge_s$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

$\wedge_w$	0	1	2
0	0	0	2
1	0	1	2
2	2	2	2

- The order of knowledge on E_3:



- **2.Corollary (Kripke, Martin, Woodruff)**: The Kleene interpreted languages have the fixed-point property.

The interpreted language of Gupta-Martin

- The operator of pathologicality:

	$\downarrow$
0	0
1	0
2	1

The interpreted language of Gupta-Martin

- The operator of pathologicality:

	$\downarrow$
0	0
1	0
2	1

- **3.Proposition (Gupta-Belnap)**: The weak Kleene scheme of interpretation with the operator $\downarrow$ has the f.p.p.

The interpreted language of Gupta-Martin

- The operator of pathologicality:

	$\downarrow$
0	0
1	0
2	1

- **3.Proposition (Gupta-Belnap)**: The weak Kleene scheme of interpretation with the operator $\downarrow$ has the f.p.p.
- **Problem (Gupta-Belnap)**: Characterize the schemes of interpretation which have the f.p.p.

Stipulation logic (Visser)

- Given an interpreted propositional language, a *stipulation* is a system of equations of the form $p_i = \varphi$, where p_i is an atomic proposition of the language and φ any formula of the language.

Stipulation logic (Visser)

- Given an interpreted propositional language, a *stipulation* is a system of equations of the form $p_i = \varphi$, where p_i is an atomic proposition of the language and φ any formula of the language.
- Liar sentence: 'this sentence is not true'

Stipulation logic (Visser)

- Given an interpreted propositional language, a *stipulation* is a system of equations of the form $p_i = \varphi$, where p_i is an atomic proposition of the language and φ any formula of the language.
- Liar sentence: 'this sentence is not true'
- (ℓ) 'ℓ is not true'

Stipulation logic (Visser)

- Given an interpreted propositional language, a *stipulation* is a system of equations of the form $p_i = \varphi$, where p_i is an atomic proposition of the language and φ any formula of the language.
- Liar sentence: 'this sentence is not true'
- (ℓ) 'ℓ is not true'
- $\ell = \neg\ell$

Stipulation logic (Visser)

- Given an interpreted propositional language, a *stipulation* is a system of equations of the form $p_i = \varphi$, where p_i is an atomic proposition of the language and φ any formula of the language.
- Liar sentence: 'this sentence is not true'
- (ℓ) 'ℓ is not true'
- $\ell = \neg\ell$
- (1) If this sentence is true, then the following sentence is not true.
  (2) Either the previous sentence is not true or snow is white
  $p_1 = p_1 \to \neg p_2$
  $p_2 = \neg p_1 \vee p_3$
  $p_3 = \mathbf{1}$

Stipulation logic

- A stipulation is *consistent* when the system of equations has a solution, i.e, when there is an assignment v of truth values to the atomic propositions such that $v(p_i) = v(\varphi_i)$ for all i.

Stipulation logic

- A stipulation is *consistent* when the system of equations has a solution, i.e, when there is an assignment v of truth values to the atomic propositions such that $v(p_i) = v(\varphi_i)$ for all i.
- An interpreted propositional language has the *fixed-point property* (f.p.p.) when every stipulation is consistent.

Stipulation logic

- A stipulation is *consistent* when the system of equations has a solution, i.e, when there is an assignment v of truth values to the atomic propositions such that $v(p_i) = v(\varphi_i)$ for all i.
- An interpreted propositional language has the *fixed-point property* (f.p.p.) when every stipulation is consistent.
- **Fixed-point problem** (Gupta-Belnap): Given a set of truth values E, characterize the interpreted propositional languages on E that have the f.p.p.

Some results about the f.p.p.

- **4.Theorem** (Visser): Let E be the set of truth values. If $(E, \leq)$ is a ccpo and the logical operators of an interpreted language are monotone functions on that order, then the scheme has the f.p.p.

Some results about the f.p.p.

- **4.Theorem** (Visser): Let E be the set of truth values. If $(E, \leq)$ is a ccpo and the logical operators of an interpreted language are monotone functions on that order, then the scheme has the f.p.p.
- **5.Theorem**: Let F be an interpreted three-valued language. Then F has the f.p.p. iff every unary operator that can be defined in F has a fixed point. The same characterization is valid for two-valued interpreted languages.

Belnap's logic

- $\langle \neg_b, \wedge_b \rangle$

	$\neg_b$
0	1
1	0
2	2
3	3

$\wedge_b$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	2	0
3	0	3	0	3

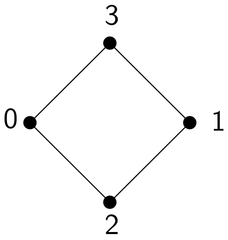
Belnap's logic

- $\langle \neg_b, \wedge_b \rangle$

	$\neg_b$
0	1
1	0
2	2
3	3

$\wedge_b$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	2	0
3	0	3	0	3

- The order of information on E_4:



Some unary operators

-

	$\neg_e$
0	1
1	0
2	1
3	1

	$\downarrow_1$
0	0
1	1
2	0
3	0

	$\downarrow_2$
0	0
1	0
2	1
3	0

	$\downarrow_3$
0	0
1	0
2	0
3	1

Some unary operators

	$\neg_e$
0	1
1	0
2	1
3	1

	$\downarrow_1$
0	0
1	1
2	0
3	0

	$\downarrow_2$
0	0
1	0
2	1
3	0

	$\downarrow_3$
0	0
1	0
2	0
3	1

	$\neg_\star$
0	1
1	0
2	1
3	3

	$\downarrow_1^\star$
0	0
1	1
2	0
3	3

	$\downarrow_2^\star$
0	0
1	0
2	1
3	3

Adding conditionals

-

$\to_l^\star$	0	1	2	3
0	1	1	1	3
1	0	1	2	3
2	2	1	1	3
3	3	3	3	3

$\leftrightarrow_l^\star$	0	1	2	3
0	1	0	2	3
1	0	1	2	3
2	2	2	1	3
3	3	3	3	3

$\leftrightarrow_{st}^\star$	0	1	2	3
0	1	0	0	3
1	0	1	0	3
2	0	0	1	3
3	3	3	3	3

Adding conditionals

-

$\to^\star_I$	0	1	2	3
0	1	1	1	3
1	0	1	2	3
2	2	1	1	3
3	3	3	3	3

$\leftrightarrow^\star_I$	0	1	2	3
0	1	0	2	3
1	0	1	2	3
2	2	2	1	3
3	3	3	3	3

$\leftrightarrow^\star_{st}$	0	1	2	3
0	1	0	0	3
1	0	1	0	3
2	0	0	1	3
3	3	3	3	3

- **6.Theorem**: Belnap's clone is maximal for the f.p.p.

Adding conditionals

-

$\to_l^\star$	0	1	2	3
0	1	1	1	3
1	0	1	2	3
2	2	1	1	3
3	3	3	3	3

$\leftrightarrow_l^\star$	0	1	2	3
0	1	0	2	3
1	0	1	2	3
2	2	2	1	3
3	3	3	3	3

$\leftrightarrow_{st}^\star$	0	1	2	3
0	1	0	0	3
1	0	1	0	3
2	0	0	1	3
3	3	3	3	3

- **6.Theorem**: Belnap's clone is maximal for the f.p.p.
- **7.Corollary**: Adding any of the operators $\neg_\star$, $\downarrow_1$, $\downarrow_1^\star$, $\downarrow_2$, $\downarrow_2^\star$, $\downarrow_3$, $\to_l^\star$, $\leftrightarrow_l^\star$ or $\leftrightarrow_{st}^\star$ to Belnap's language produces paradoxes.

Fitting's logic

	$\neg_b$
0	1
1	0
2	2
3	3

$\wedge_{sw}$	0	1	2	3
0	0	0	0	3
1	0	1	2	3
2	0	2	2	3
3	3	3	3	3

Fitting's logic

-

	$\neg_b$
0	1
1	0
2	2
3	3

$\wedge_{sw}$	0	1	2	3
0	0	0	0	3
1	0	1	2	3
2	0	2	2	3
3	3	3	3	3

- **8.Proposition**: $\langle \neg_b, \wedge_{sw}, \downarrow_2 \rangle$ does not have the fixed-point property.
  Proof: $x = \neg_b \downarrow_2 (\mathbf{2} \wedge_{sw} x)$.

Some definitions

- Given a k-valued language F, let us call $F^{(1)}$ the set of unary operators expressible in he language and $F^{(1-1)}$ the group of functions in $F^{(1)}$ which are permutations. For $a \in E_k$, the stabilizer of a (denoted as $\mathrm{St}(a)$) is the set of all permutations $f : E_k \to E_k$ such that $f(a) = a$.

Some definitions

- Given a k-valued language F, let us call $F^{(1)}$ the set of unary operators expressible in he language and $F^{(1-1)}$ the group of functions in $F^{(1)}$ which are permutations. For $a \in E_k$, the stabilizer of a (denoted as $\mathrm{St}(a)$) is the set of all permutations $f : E_k \to E_k$ such that $f(a) = a$.
- Given a partial order $(E, \leq)$, let us call $\mathrm{Mon}(\leq)$ the interpreted language generated by all functions monotonic on $\leq$. The *flat ccpo* on E_{k+1} is the partial order $\leq_k$ defined by $k \leq_k i$ for all $i \in E_{k+1}$.

Some definitions

- Given a k-valued language F, let us call $F^{(1)}$ the set of unary operators expressible in he language and $F^{(1-1)}$ the group of functions in $F^{(1)}$ which are permutations. For $a \in E_k$, the stabilizer of a (denoted as $\mathrm{St}(a)$) is the set of all permutations $f : E_k \to E_k$ such that $f(a) = a$.
- Given a partial order $(E, \leq)$, let us call $\mathrm{Mon}(\leq)$ the interpreted language generated by all functions monotonic on $\leq$. The *flat ccpo* on E_{k+1} is the partial order $\leq_k$ defined by $k \leq_k i$ for all $i \in E_{k+1}$.
- Let $f : E_{k+1} \to E_{k+1}$. The derived set of f, denoted $\operatorname{der} f$, is the set of all functions which can be obtained from f with some (all, none) of its variables replaced by constants. I_A, $A \subseteq E_k$, is the set of all functions on E_k that preserve the set A. The restriction of $f : E_{k+1} \to E_{k+1}$, denoted $\operatorname{re} f$, is the function $\operatorname{re} f : E_k \to E_{k+1}$ defined as $\operatorname{re} f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$, for all $x_1, \ldots, x_n \in E_k$.

The interpreted languages G_k

- G_k is the interpreted language generated by all functions $f : E_{k+1} \rightarrow E_{k+1}$ that satisfy the following conditions:
  1. For every $g \in \text{der}\, f$, if $g \neq c_k$, then $g \in I_{\{0\ldots k-1\}}$.
  2. If $f(a_0, \ldots, a_{n-1}) \neq k$, for some $a_i \in E_{k+1}$ and $a_{i_0} = \ldots = a_{i_j} = k$, for $0 \leq j \leq n-1$ and $0 \leq i_0 \leq \ldots \leq i_j \leq n-1$, then the function

$$\text{re}\, f(a_0, \ldots, a_{i_0-1}, x_1, a_{i_0+1}, \ldots, a_{i_j-1}, x_j, a_{i_j+1}, \ldots, a_{n-1})$$

     is constant.

The interpreted languages G_k

- G_k is the interpreted language generated by all functions $f : E_{k+1} \rightarrow E_{k+1}$ that satisfy the following conditions:
  1. For every $g \in \operatorname{der} f$, if $g \neq c_k$, then $g \in I_{\{0\ldots k-1\}}$.
  2. If $f(a_0, \ldots, a_{n-1}) \neq k$, for some $a_i \in E_{k+1}$ and $a_{i_0} = \ldots = a_{i_j} = k$, for $0 \leq j \leq n-1$ and $0 \leq i_0 \leq \ldots \leq i_j \leq n-1$, then the function

     $$\operatorname{re} f(a_0, \ldots, a_{i_0 -1}, x_1, a_{i_0+1}, \ldots, a_{i_j -1}, x_j, a_{i_j+1}, \ldots, a_{n-1})$$

     is constant.
- Examples:

f_1	0	1	2	3
0	0	2	1	3
1	0	2	1	3
2	0	2	0	3
3	0	1	2	3

f_2	0	1	2	3
0	1	1	1	2
1	0	0	0	2
2	2	2	2	0
3	3	3	3	3

The interpreted languages G_k

- G_k is the interpreted language generated by all functions $f : E_{k+1} \to E_{k+1}$ that satisfy the following conditions:
  1. For every $g \in \text{der} f$, if $g \neq c_k$, then $g \in I_{\{0...k-1\}}$.
  2. If $f(a_0, \ldots, a_{n-1}) \neq k$, for some $a_i \in E_{k+1}$ and $a_{i_0} = \ldots = a_{i_j} = k$, for $0 \leq j \leq n-1$ and $0 \leq i_0 \leq \ldots \leq i_j \leq n-1$, then the function

     $$\text{re}\, f(a_0, \ldots, a_{i_0-1}, x_1, a_{i_0+1}, \ldots, a_{i_j-1}, x_j, a_{i_j+1}, \ldots, a_{n-1})$$

     is constant.
- Examples:

f_1	0	1	2	3
0	0	2	1	3
1	0	2	1	3
2	0	2	0	3
3	0	1	2	3

f_2	0	1	2	3
0	1	1	1	2
1	0	0	0	2
2	2	2	2	0
3	3	3	3	3

- $f_1 \notin G_3$, $f_2 \in G_3$

A fixed point theorem

- **9.Theorem**: Let F be a k-valued interpreted language such that every function in $F^{(1)}$ has a fixed point and $F^{(1-1)} = \mathrm{St}(k)$. Then either $F \subseteq G_k$ or $F \subseteq \mathrm{Mon}(\leq_k)$.

A fixed point theorem

- **9.Theorem**: Let F be a k-valued interpreted language such that every function in $F^{(1)}$ has a fixed point and $F^{(1-1)} = \mathrm{St}(k)$. Then either $F \subseteq G_k$ or $F \subseteq \mathrm{Mon}(\leq_k)$.
- **10.Theorem**: Any (k+1)-valued interpreted language F such that every unary function defined in F has a fixed point and $F^{(1-1)} = \mathrm{St}(k)$ has the fixed-point property.

A fixed point theorem

- **9.Theorem**: Let F be a k-valued interpreted language such that every function in $F^{(1)}$ has a fixed point and $F^{(1-1)} = \mathrm{St}(k)$. Then either $F \subseteq G_k$ or $F \subseteq \mathrm{Mon}(\leq_k)$.
- **10.Theorem**: Any (k+1)-valued interpreted language F such that every unary function defined in F has a fixed point and $F^{(1-1)} = \mathrm{St}(k)$ has the fixed-point property.
- **11.Corollary**: The four-valued language that contains the constants and the operators $\neg_b, \neg_\star, \wedge_{sw}, \downarrow_1^\star, \downarrow_2^\star, \downarrow_3, \rightarrow_l^\star, \leftrightarrow_l^\star, \leftrightarrow_{st}^\star$ has the f.p.p.

Another four-valued generalization of weak Kleene logic

- $\langle \neg_b, \wedge_{ww} \rangle$.

$\wedge_{ww}$	0	1	2	3
0	0	0	2	3
1	0	1	2	3
2	2	2	2	3
3	3	3	3	3

$\wedge_{sw}$	0	1	2	3
0	0	0	0	3
1	0	1	2	3
2	0	2	2	3
3	3	3	3	3

Another four-valued generalization of weak Kleene logic

- $\langle \neg_b, \wedge_{ww} \rangle$.

$\wedge_{ww}$	0	1	2	3
0	0	0	2	3
1	0	1	2	3
2	2	2	2	3
3	3	3	3	3

$\wedge_{sw}$	0	1	2	3
0	0	0	0	3
1	0	1	2	3
2	0	2	2	3
3	3	3	3	3

- **12.Theorem**: The four-valued interpreted language that contains the constants and the operators $\neg_b, \wedge_{ww}, \downarrow_2, \downarrow_3$ has the fixed-point property.

Another four-valued generalization of weak Kleene logic

Sketch of the proof:

- Let us call $H(G_2)$ the set of all finitary functions f on E_4 that satisfy the following condition: for all $g \in \operatorname{der} f$, if $g \neq c_3$, then $g \in I_{\{0,1,2\}}$ and $g \in G_2$.

Another four-valued generalization of weak Kleene logic

Sketch of the proof:

- Let us call $H(G_2)$ the set of all finitary functions f on E_4 that satisfy the following condition: for all $g \in \operatorname{der} f$, if $g \neq c_3$, then $g \in I_{\{0,1,2\}}$ and $g \in G_2$.
- First step: Show that all the functions expressible in the language satisfy this condition. One can do this by checking the truth tables of the operators $\neg_b$, $\wedge_{ww}$, $\downarrow_2$ and $\downarrow_3$, and then proving that $H(G_2)$ is closed under composition of functions.

Another four-valued generalization of weak Kleene logic

Sketch of the proof:

- Let us call $H(G_2)$ the set of all finitary functions f on E_4 that satisfy the following condition: for all $g \in \operatorname{der} f$, if $g \neq c_3$, then $g \in I_{\{0,1,2\}}$ and $g \in G_2$.
- First step: Show that all the functions expressible in the language satisfy this condition. One can do this by checking the truth tables of the operators $\neg_b$, $\wedge_{ww}$, $\downarrow_2$ and $\downarrow_3$, and then proving that $H(G_2)$ is closed under composition of functions.
- Second step: Show that an interpreted language that expresses exactly the functions of $H(G_2)$ has the fixed-point property. Given a stipulation, the valuation that shows that it is consistent is found through a procedure based on the definition of the functions in $H(G_2)$ and the fact that G_2 has the fixed-point.

References

Melvin Fitting, Kleene's three-valued logics and their children, *Fundamenta Informaticae*, 20, pp. 113-131, 1994.

Anil Gupta and Nuel Belnap, *The Revision Theory of Truth*, MIT Press, 1993.

José Martínez Fernández, Maximal Three-Valued Clones with the Gupta-Belnap Fixed-Point Theory, *Notre Dame Journal of Formal Logic*, 48, 4, pp. 449-472, 2007.

Albert Visser, Semantics and the Liar Paradox, in D. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, vol. 4, Reidel, 1984, pp. 617-706.

Generalization of Visser's Theorem

- Let us say that a partial order $(E, \leq)$ is *stable* if all monotonic functions from E to E have a fixed point.

Generalization of Visser's Theorem

- Let us say that a partial order $(E, \leq)$ is *stable* if all monotonic functions from E to E have a fixed point.
- **13.Theorem**: Let $(E_k, \leq)$ $(k \geq 2)$ be a stable partial order and $F \subseteq \mathcal{O}_k$ a clone such that $F^{(1)} \subseteq (\text{Mon} \leq)^{(1)}$. Then F has the fixed-point property.