GI Meeting Deduction and Logic (26.3.2021)

Paraconsistent and paracomplete logics

(in Isabelle/HOL)

David Fuenmayor
University of Luxembourg & Freie Universität Berlin

A “teaser talk” for:

D. Fuenmayor (2020) “Topological semantics for paraconsistent and paracomplete logics”
Isabelle’s Archive of Formal Proofs. https://www.isa-afp.org/entries/Topological_Semantics.html

Paraconsistent and Paracomplete Logics

Motivation in CS: Useful for knowledge representation and reasoning in presence of partial/incomplete and excessive/contradictory information.

A logic is called

- **paraconsistent** if it ‘tolerates contradictions’, i.e.

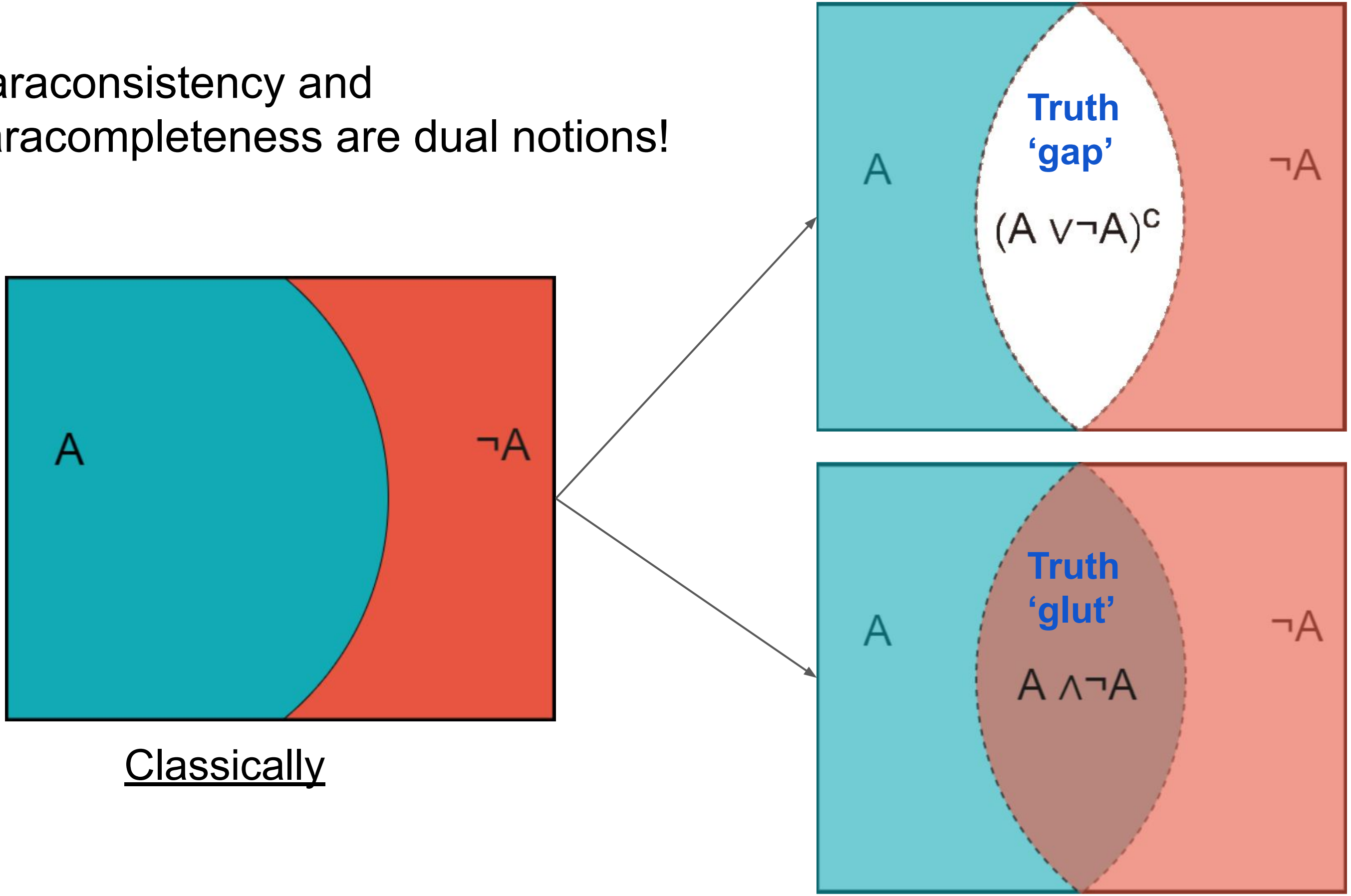
  ‘Principle of explosion’ or *ex contradictione (sequitur) quodlibet* (ECQ) is not valid, i.e. from a contradiction (A & ¬A) not everything follows!

- **paracomplete** if it does not ‘enforce completeness/exhaustiveness’, i.e.

  Law of excluded middle or *tertium non datur* (TND) is not valid, it can be the case that neither A nor its negation is the case (cf. intuitionistic logics).

Paraconsistent and Paracomplete Logics

Paraconsistency and paracompleteness are dual notions!



Classically

Paracomplete

$W \neq |A| \cup |\neg A|$

$\top \nvdash A \vee \neg A$

dual

$A \wedge \neg A \nvdash \bot$

$|A| \cap |\neg A| \neq \emptyset$

Paraconsistent

Paraconsistent and Paracomplete Logics

- Paraconsistent (resp. paracomplete) logics validate TND (resp. ECQ) only.
- Some logics do not validate either, they are termed “paranormal” or “paradefinite”. Classical logic validates both.
- By giving up TND/ECQ other properties of negation become ‘negotiable’, e.g.
    - DS1-4: disjunctive syllogism
    - DM1-4: De Morgan laws (finite & infinite)
    - DNI/DNE: double negation intro/elim (rule & axiom)
    - CoP1-4: contraposition (rules & axioms)
    - MT1-4: modus tollens (rules & axioms)
    - … and many others
- Automated theorem proving can be employed to explore (minimal) semantic conditions under which they obtain.

Logics with “Recovery Operators”: LFIs & LFUs

Giving up TND or ECQ does not necessarily ‘weaken’ our logic!

We can have special operators (‘○’,‘☆’) allowing us to recover classical properties in a ‘sentence-wise’ fashion. (Think of a sort of “quality seal” for formulas.)

For (paraconsistent) LFI’s:

read ‘○A’ as “A is consistent”

For (paracomplete) LFU’s:

read ‘☆A’ as “A is determined”

Logics with “Recovery Operators”: LFIs & LFUs

For (paraconsistent) LFIs we have that:

The Principle of “Explosion” (ECQ)

$$A , \neg A \vdash B$$

is NOT valid. Instead we have:

The Principle of “Gentle Explosion”
(cf. W. Carnielli & J. Marcos (2001). “A Taxonomy of C-Systems”)

$$\circ A , \; A , \neg A \vdash B$$

Logics with “Recovery Operators”: LFIs & LFUs

Dually, for (paracomplete) LFUs we have that:

The Law of Excluded Middle (TND)

$$\vdash A \vee \neg A \quad (\text{i.e. } \Gamma \vdash A, \neg A)$$

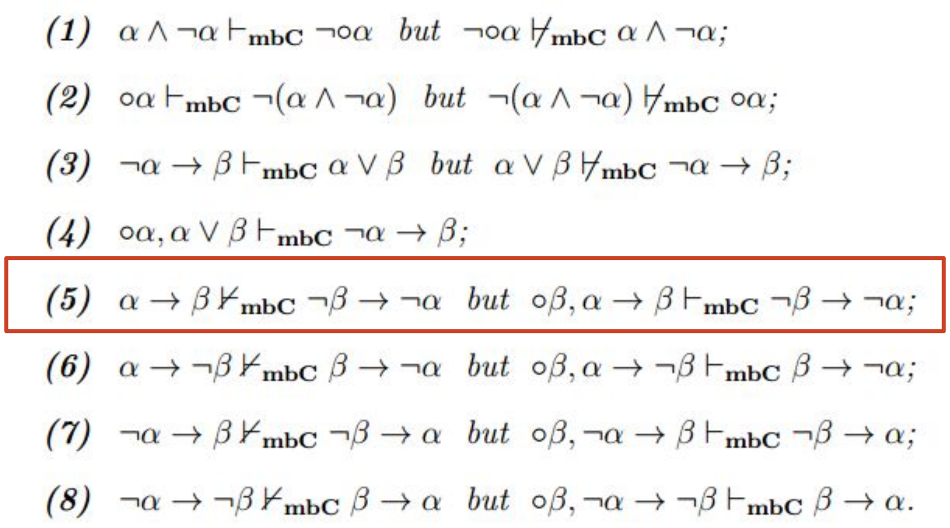
is NOT valid. Instead we have:

$$\vdash \text{☆}A \rightarrow A \vee \neg A \quad (\text{i.e. } \Gamma \vdash \bigstar A,\ A, \neg A \ ;\text{ where } \bigstar A = {\sim}\text{☆}A)$$

(W. Carnielli, M. Coniglio & A. Rodrigues (2020). Recovery operators, paraconsistency and duality. LJIGPL)

Logics with “Recovery Operators”: LFIs & LFUs

Recovery extends to other properties too. For example, we have for LFI system **mbC:**



(1) $\alpha \wedge \neg\alpha \vdash_{\mathbf{mbC}} \neg\circ\alpha$ *but* $\neg\circ\alpha \nvdash_{\mathbf{mbC}} \alpha \wedge \neg\alpha$;

(2) $\circ\alpha \vdash_{\mathbf{mbC}} \neg(\alpha \wedge \neg\alpha)$ *but* $\neg(\alpha \wedge \neg\alpha) \nvdash_{\mathbf{mbC}} \circ\alpha$;

(3) $\neg\alpha \to \beta \vdash_{\mathbf{mbC}} \alpha \vee \beta$ *but* $\alpha \vee \beta \nvdash_{\mathbf{mbC}} \neg\alpha \to \beta$;

(4) $\circ\alpha, \alpha \vee \beta \vdash_{\mathbf{mbC}} \neg\alpha \to \beta$;

(5) $\alpha \to \beta \nvdash_{\mathbf{mbC}} \neg\beta \to \neg\alpha$ *but* $\circ\beta, \alpha \to \beta \vdash_{\mathbf{mbC}} \neg\beta \to \neg\alpha$;

(6) $\alpha \to \neg\beta \nvdash_{\mathbf{mbC}} \beta \to \neg\alpha$ *but* $\circ\beta, \alpha \to \neg\beta \vdash_{\mathbf{mbC}} \beta \to \neg\alpha$;

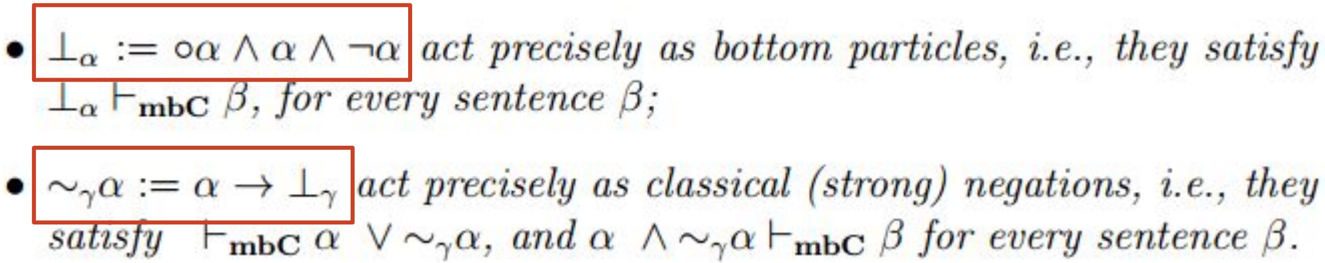
(7) $\neg\alpha \to \beta \nvdash_{\mathbf{mbC}} \neg\beta \to \alpha$ *but* $\circ\beta, \neg\alpha \to \beta \vdash_{\mathbf{mbC}} \neg\beta \to \alpha$;

(8) $\neg\alpha \to \neg\beta \nvdash_{\mathbf{mbC}} \beta \to \alpha$ *but* $\circ\beta, \neg\alpha \to \neg\beta \vdash_{\mathbf{mbC}} \beta \to \alpha$.

mbC negation (¬) is indeed very ‘weak’. (e.g. contraposition, DNI/DNE, etc. are not valid)
However, we can recover classical properties by employing the consistency operator: ‘○’

Logics with “Recovery Operators”: LFIs & LFUs

In a sense, **mbC** extends classical logic (i.e. it is more ‘expressive’). We can indeed define a ‘bottom particle’ (**⊥**), and with it a classical negation (**~**), inside **mbC**:

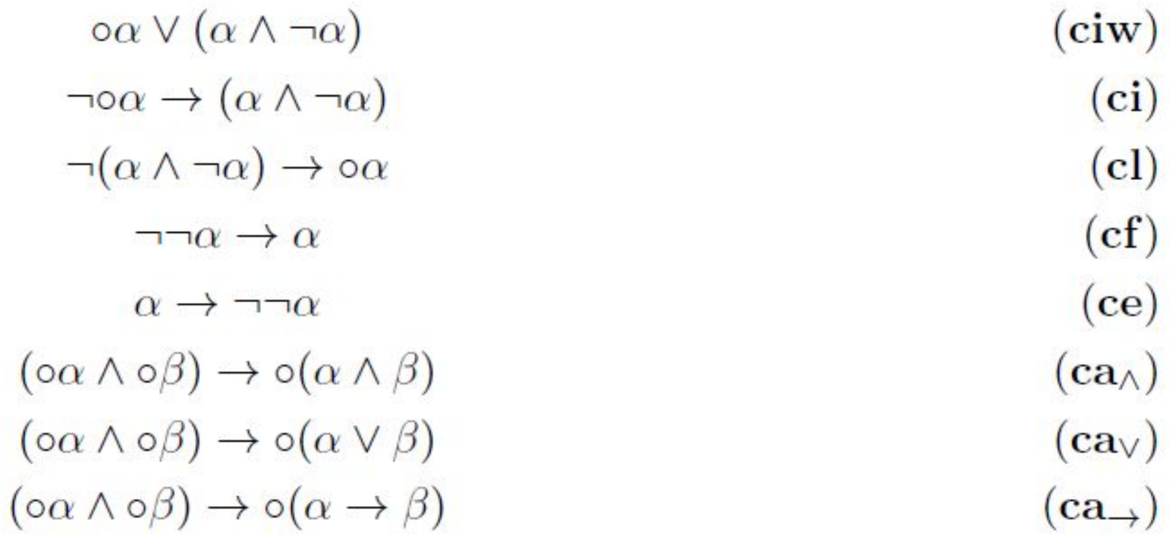


- $\boxed{\bot_\alpha := \circ\alpha \wedge \alpha \wedge \neg\alpha}$ *act precisely as bottom particles, i.e., they satisfy* $\bot_\alpha \vdash_{\mathbf{mbC}} \beta$, *for every sentence* β;
- $\boxed{\sim_\gamma\alpha := \alpha \to \bot_\gamma}$ *act precisely as classical (strong) negations, i.e., they satisfy* $\vdash_{\mathbf{mbC}} \alpha \vee \sim_\gamma\alpha$, *and* $\alpha \wedge \sim_\gamma\alpha \vdash_{\mathbf{mbC}} \beta$ *for every sentence* β.

Employing the above interpretation for **⊥** and **~** classical logic becomes a ‘subsystem’ of **mbC**

Logics with “Recovery Operators”: LFIs & LFUs

Several well-studied axiomatic extensions of the minimal (LFI) logic **mbC** employ:



$$\circ\alpha \vee (\alpha \wedge \neg\alpha) \tag{\textbf{ciw}}$$

$$\neg\circ\alpha \to (\alpha \wedge \neg\alpha) \tag{\textbf{ci}}$$

$$\neg(\alpha \wedge \neg\alpha) \to \circ\alpha \tag{\textbf{cl}}$$

$$\neg\neg\alpha \to \alpha \tag{\textbf{cf}}$$

$$\alpha \to \neg\neg\alpha \tag{\textbf{ce}}$$

$$(\circ\alpha \wedge \circ\beta) \to \circ(\alpha \wedge \beta) \tag{$\textbf{ca}_\wedge$}$$

$$(\circ\alpha \wedge \circ\beta) \to \circ(\alpha \vee \beta) \tag{$\textbf{ca}_\vee$}$$

$$(\circ\alpha \wedge \circ\beta) \to \circ(\alpha \to \beta) \tag{$\textbf{ca}_\to$}$$

W. Carnielli and M. Coniglio (2016). *Paraconsistent Logic: Consistency, Contradiction and Negation*. Springer

Logics with “Recovery Operators”: LFIs & LFUs

On (paraconsistent) LFIs:

W. Carnielli and M. Coniglio (2016). *Paraconsistent Logic: Consistency, Contradiction and Negation*. Springer

W. Carnielli, M. Coniglio, J. Marcos (2007). “Logics of Formal Inconsistency” *Handbook of Phil. Logic*. Springer

On (paracomplete) LFUs:

J. Marcos (2005). *Nearly every normal modal logic is paranormal*. Logique et Analyse

Recent developments:

W. Carnielli, M. Coniglio, D. Fuenmayor (2020). “Logics of Formal Inconsistency enriched with replacement: an algebraic and modal account”. Preprint at arXiv: https://arxiv.org/abs/2003.09522

W. Carnielli, M. Coniglio & A. Rodrigues (2020). Recovery operators, paraconsistency and duality. LJIGPL

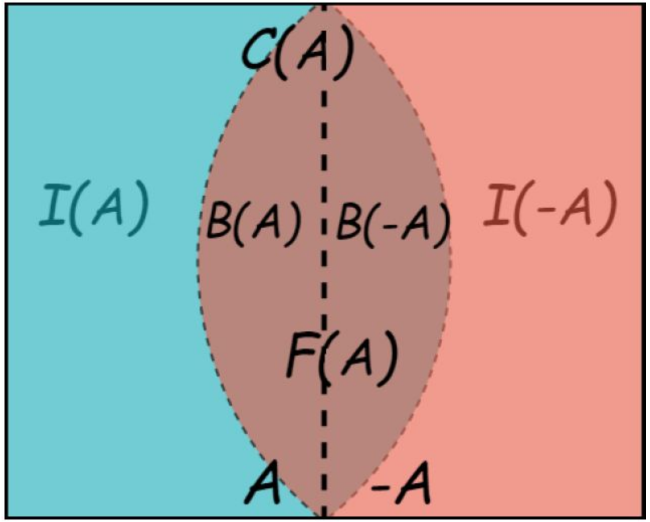
Topological semantics for LFIs & LFUs **(with replacement)**

Consider Boolean algebras extended with an additional unary operator (e.g. **C**losure, **I**nterior, **B**order, or **F**rontier algebras)

Ex. for LFIs we can use *frontier algebras* (with F(.) primitive)

- $C(A) = A \cup F(A)$
- $I(A) = A \setminus F(A)$
- $\boxed{\neg A := C(-A) = -A \cup F(-A)}$
- $\boxed{\circ A} := (I)^{\mathbf{fp}}(A) = -B(A) = -A \cup I(A) = \boxed{A \rightarrow I(A)}$

where $(\Phi)^{\mathbf{fp}}(A) := \Phi(A) \leftrightarrow A$



D. Fuenmayor (2020) “Topological semantics for paraconsistent and paracomplete logics” (Isabelle AFP)
www.isa-afp.org/entries/Topological_Semantics.html

or extended abstract in ResearchGate: “Paraconsistent and paracomplete logics in Isabelle/HOL”
https://www.researchgate.net/publication/349043183_Paraconsistent_and_paracomplete_logics_in_IsabelleHOL

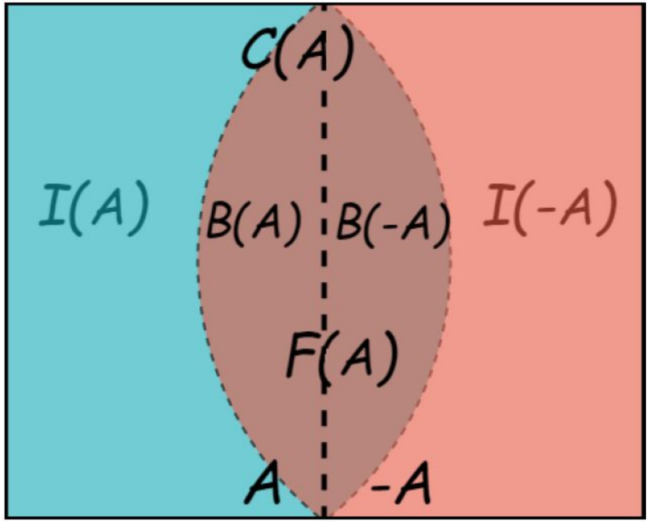
Topological semantics for LFIs & LFUs **(with replacement)**

Consider Boolean algebras extended with an additional unary operator (e.g. **C**losure, **I**nterior, **B**order, or **F**rontier algebras)

Ex. for LFUs we can use *frontier algebras* (with F(.) primitive)

- $C(A) = A \cup F(A)$
- $I(A) \ = A \setminus F(A)$
- $\boxed{\neg A \ := \ I(-A) \ = \ -A \setminus F(-A)}$
- $\boxed{\star A} \ := (C)^{\mathbf{fp}}(A) = -B(-A) = A \cup -C(A) = \boxed{C(A) \rightarrow A}$

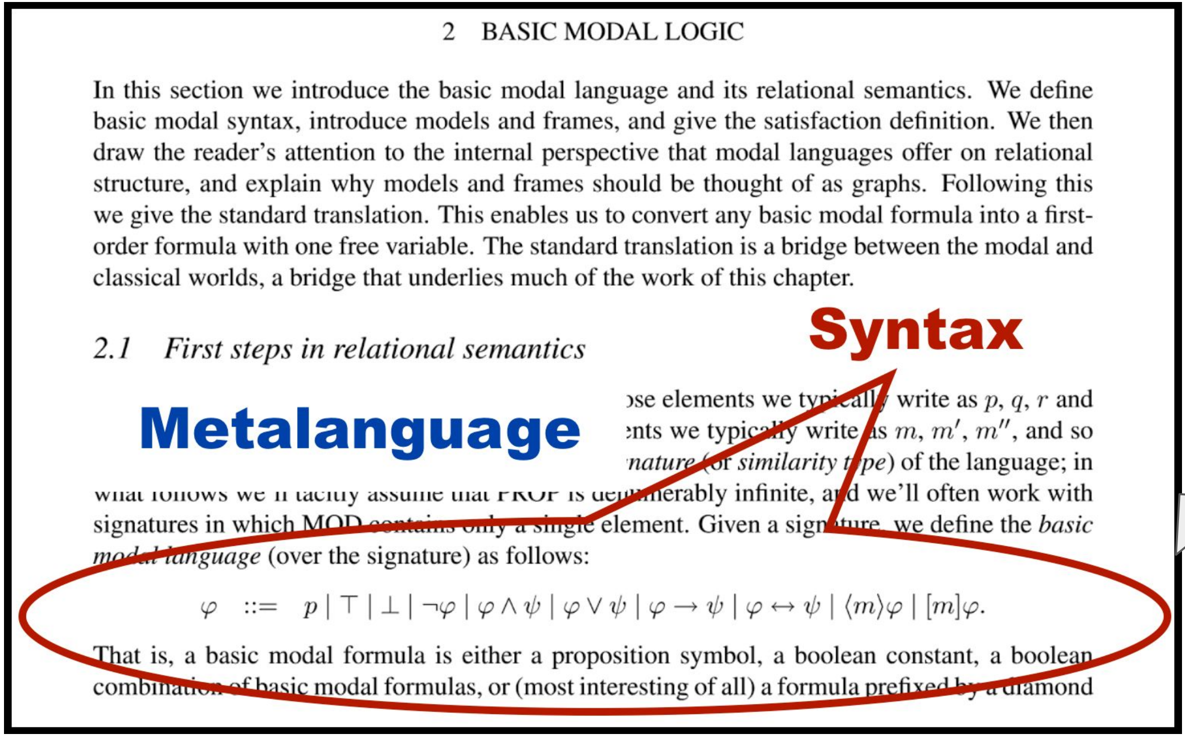
where $(\Phi)^{\mathbf{fp}}(A) \ := \ \Phi(A) \leftrightarrow A$



D. Fuenmayor (2020) “Topological semantics for paraconsistent and paracomplete logics” (Isabelle AFP)
www.isa-afp.org/entries/Topological_Semantics.html

or extended abstract in ResearchGate: “Paraconsistent and paracomplete logics in Isabelle/HOL”
https://www.researchgate.net/publication/349043183_Paraconsistent_and_paracomplete_logics_in_IsabelleHOL

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)



2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 *First steps in relational semantics*

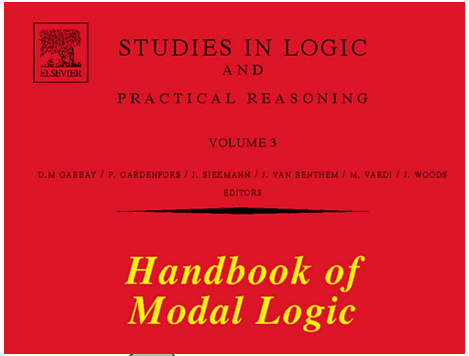
Syntax

Metalanguage

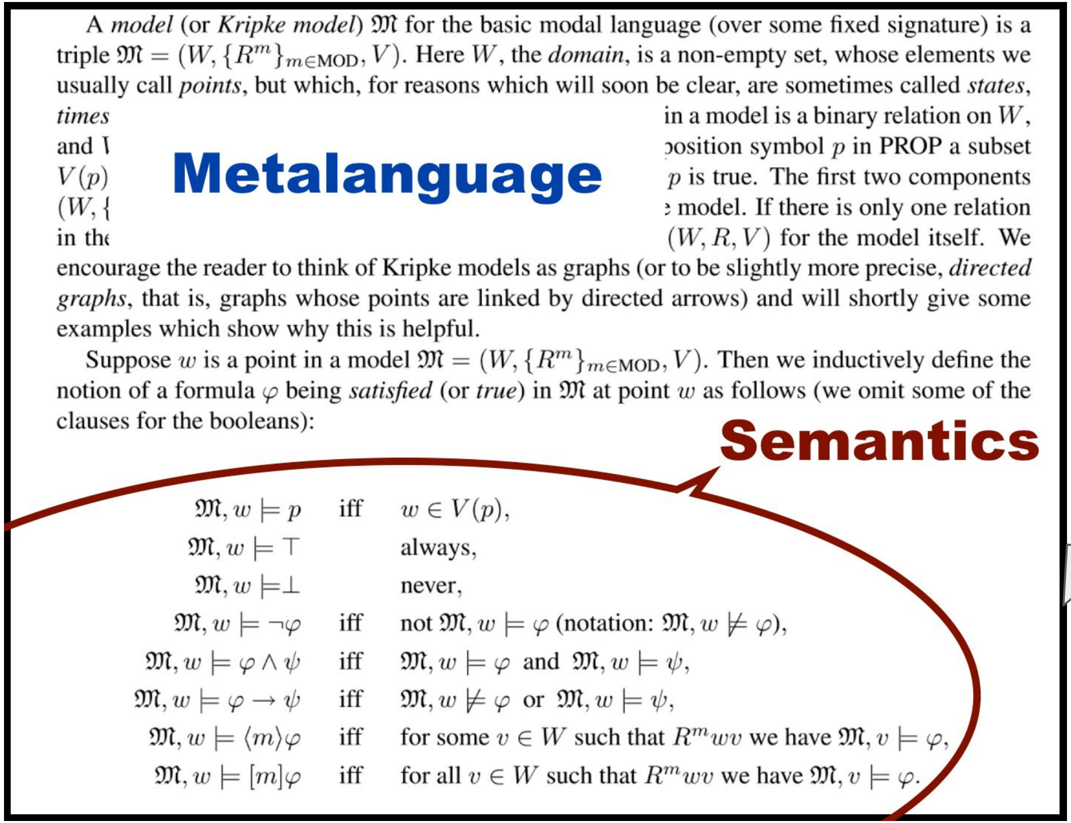
ose elements we typically write as p, q, r and nts we typically write as m, m', m'', and so *nature* (*or similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond


ELSEVIER
STUDIES IN LOGIC AND PRACTICAL REASONING
VOLUME 3
D.M GABBAY / P. GARDENFORS / J. SIEKMANN / J. VAN BENTHEM / M. VARDI / J. WOODS
EDITORS
Handbook of Modal Logic

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)



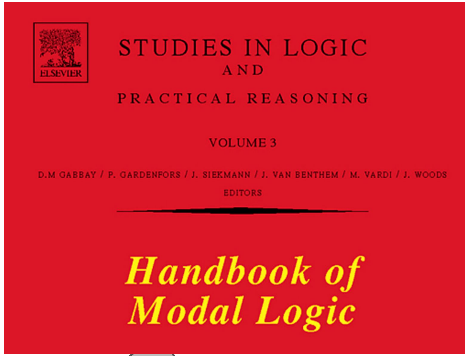
A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times* … in a model is a binary relation on W, and I … position symbol p in PROP a subset $V(p)$ … p is true. The first two components $(W, \{$ … model. If there is only one relation in the … (W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Metalanguage

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

Semantics

$$\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m] \varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}$$



ELSEVIER

STUDIES IN LOGIC AND PRACTICAL REASONING

VOLUME 3

D.M GABBAY / P. GARDENFORS / J. SIEKMANN / J. VAN BENTHEM / M. VARDI / J. WOODS

EDITORS

Handbook of Modal Logic

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)

HOL (meta-logic) φ ::= 🟪

L (object-logic) ψ ::= 🟥

Embedding of 🟥 in 🟪

🟥 = 🟪

🟥 = 🟪

🟥 = 🟪

🟥 = 🟪

🟥 = 🟪

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)

HOL $\quad s, t \quad ::= \quad c_\alpha \mid x_\alpha \mid (\lambda x_\alpha s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \mid \neg s_o \mid s_o \vee t_o \mid \forall x_\alpha\, t_o$

HOML $\quad \varphi, \psi \quad ::= \quad \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \to \psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x_\gamma\, \varphi \mid \exists x_\gamma\, \varphi$

HOML in HOL: HOML formulas φ are mapped to HOL predicates $\varphi_{\mu\to o}$
(explicit representation of labelled formulas)

$$\boxed{\begin{array}{lcl}
\neg & = & \lambda\varphi_{\mu\to o}\lambda w_\mu \neg\varphi w \\
\wedge & = & \lambda\varphi_{\mu\to o}\lambda\psi_{\mu\to o}\lambda w_\mu(\varphi w \wedge \psi w) \\
\to & = & \lambda\varphi_{\mu\to o}\lambda\psi_{\mu\to o}\lambda w_\mu(\neg\varphi w \vee \psi w) \\
\\
\forall & = & \lambda h_{\gamma\to(\mu\to o)}\lambda w_\mu \forall d_\gamma\, hdw \\
\exists & = & \lambda h_{\gamma\to(\mu\to o)}\lambda w_\mu \exists d_\gamma\, hdw \\
\\
\Box & = & \lambda\varphi_{\mu\to o}\lambda w_\mu \forall u_\mu\, (\neg rwu \vee \varphi u) \\
\Diamond & = & \lambda\varphi_{\mu\to o}\lambda w_\mu \exists u_\mu\, (rwu \wedge \varphi u) \\
\\
\text{valid} & = & \lambda\varphi_{\mu\to o}\forall w_\mu \varphi w
\end{array}}$$

Ax (polymorphic over γ)

The equations in Ax are given as axioms to the HOL provers!

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)

HOL $\quad s,t \;::=\; \boldsymbol{c}_\alpha \mid x_\alpha \mid (\lambda x_\alpha s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \mid \neg s_o \mid s_o \vee t_o \mid \forall x_\alpha\, t_o$

HOML $\quad \varphi,\psi \;::=\; \ldots \mid \neg\varphi \mid \varphi\wedge\psi \mid \varphi\to\psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x_\gamma\, \varphi \mid \exists x_\gamma\, \varphi$

HOML in HOL: HOML formulas φ are mapped to HOL predicates $\varphi_{\mu\to o}$
(explicit representation of labelled formulas)

$$
\begin{aligned}
\neg &= \lambda\varphi_{\mu\to o}\lambda w_\mu \neg\varphi w \\
\wedge &= \lambda\varphi_{\mu\to o}\lambda\psi_{\mu\to o}\lambda w_\mu(\varphi w \wedge \psi w) \\
\to &= \lambda\varphi_{\mu\to o}\lambda\psi_{\mu\to o}\lambda w_\mu(\neg\varphi w \vee \psi w) \\
\forall &= \lambda h_{\gamma\to(\mu\to o)}\lambda w_\mu \forall d_\gamma\, hdw \\
\exists &= \lambda h_{\gamma\to(\mu\to o)}\lambda w_\mu \exists d_\gamma\, hdw \\
\Box &= \lambda\varphi_{\mu\to o}\lambda w_\mu \forall u_\mu\, (\neg\mathbf{r}wu \vee \varphi u) \\
\Diamond &= \lambda\varphi_{\mu\to o}\lambda w_\mu \exists u_\mu\, (\mathbf{r}wu \wedge \varphi u) \\
\text{valid} &= \lambda\varphi_{\mu\to o}\forall w_\mu \varphi w
\end{aligned}
$$

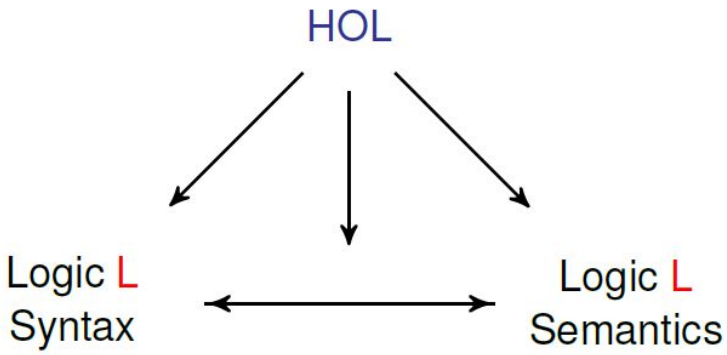
C. Benzmüller & L. Paulson (2013) "Quantified Multimodal Logics in Simple Type Theory" Logica Universalis

The equations in Ax are given as axioms to the HOL provers!

Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)

Logics **L** embedded using the semantic embeddings approach (all of them supporting quantification)

- Multi-modal & hybrid logics
- Deontic logics & conditional logics
- Many-valued logics
- Free logics (+ axiomatizing Category theory)
- 2D-semantics (Kaplan’s Logic of Indexicals)
- Dynamic logics (incl. preference logics, public announcement), and many others...
- **paraconsistent & paracomplete logics**



Non-classical logics in Isabelle/HOL via semantic embeddings: Simple Type Theory as universal meta-logic (jww. Christoph Benzmüller)

Theorem provers become universal logic reasoning engines:
interactive: Isabelle/HOL, PVS, HOL, Coq/HOL, Lean, …
automated: Leo-2/3, Satallax, Vampire, SMT solvers,
(counter)model finders (Nitpick, Nunchaku)

New approach towards combining logics:

- object logics correspond to different fragments of HOL (i.e. Church’s simple type theory).
- encoding semantic conditions and ‘bridge’ meta-axioms
- using theorem provers and model finders for verification

Coq

Isabelle/HOL encoding of LFIs & LFUs

Live 2min demo

+

Q&A