# A FRAMEWORK FOR LOGICS. RIGIDITY, FINITISM AND AN ENCYCLOPEDIA OF LOGICS*

Christian Damböck
Institute Vienna Circle
University of Vienna
christian.damboeck@univie.ac.at

*This work was supported by the Austrian Science Fund (FWF), research projects P18596 and P18066.

## Universal logic and my proposal

Universal logic is based on the idea of presenting logics as algebraic structures.

I will follow this approach here in some sense. In my understanding

- a *deductive system* $(L, \vdash)$ is a set $L$ plus a relation $\vdash$ over $\wp(L)$,

- a *semantic system* $(W, L, \models)$ is a relation $\models$ between the sets $W$ and $L$.

☞ $W$ is thought to be a set here, i. e. $\models$ is a 'set-theoretical predicate' in Patrick Suppes' sense.

☞ However, I do not follow the universal logic approach insofar as I take neither a deductive nor a semantic system as an expression for such things that we call 'logics'.

## But what is a logic?

In my view there are two fundamentally different ways to understand the term ‘logic’.

(1) the mathematical way: a logic is a formal language with a particular *expressive power*.

(2) the philosophical way: a logic is a formal language that allows us to express some particular *philosophical notions*.

☞ Very roughly, a logic in the philosophical sense is a collection of philosophical *devices* like quantifiers, existence predicates, first or higher order predicates, functions, modal operators, etc.

**Main question:**
**How to present such a collection of philosophical devices in a set-theoretic environment?**

In a classical framework of mathematical logic (like first-order logic) we can characterize a deductive calculus as an algebraic structure ('set theoretical predicate') but not a semantic system, simply because the class of all semantic interpretations is not a set.

☞ In order to be able to express the most important philosophical features of languages in a set-theoretic framework we need a completely different layout for our languages:

**A logic in the philosophical sense must be based on an *interpreted language*, i. e. a language where the names have fixed denotations (direct reference).**

## Semantic systems for interpreted languages

Let $L_a$ be the language (set of formulas) of propositional logic and $L_p$ the language of first-order logic (without free variables).

- In a semantic system for $L_a$ there is no difference between the interpreted and an the uninterpreted case (because for the specification of the semantic system the difference between propositional constants and propositional variables is insignificant).

- A semantic system for $L_p$ (interpreted case) must be based on the stock of individuals $D$ that is fixed by the individual constants (direct reference!). Thus a structure consists of a subset of $D$ plus relations and functions over this subset.

☞ The class of all structures of an interpreted language is always constructed as a set of combinations, in an obvious way.

## Philosophical logics

Given the semantic system $S = (W, L, \models)$ I propose to define a *philosophical logic* in the following way:

$L'(S)$ I call the class of all formulas of the form $w \Vdash \phi$ with $w \in W$ and $\phi \in L$.

Then we have a truth value for every $L'$-formula $w \Vdash \phi$, defined in an obvious way:

$$w \Vdash \phi = \begin{cases} T & \text{iff it holds that } w \models \phi \\ F & \text{otherwise.} \end{cases}$$

This $L'(S)$ I call *the philosophical logic* over $S$.

☞ A philosophical logic is an interpreted language, insofar as every sentence of the language has a fixed truth value.

## Rigid and finitistic logics (propositional logic as a framework for logics)

Let $S_a$ be a usual semantic system for the propositional language $L_a$ with the logical connectives $\neg$ and $\bigwedge$ (generalized conjunction) and the relation of satisfaction $\models_a$.

Then I call a philosophical logic $L'(W, L, \models)$ *rigid*, if there exists a set $F \subseteq L$ that defines the set $\hat{F}$ of formulas

$$\phi ::= \ p \mid \neg\phi \mid \ \bigwedge \Gamma,$$

where $p$ ranges over $F$ and $\Gamma$ over sets of finite formulas; then there exists a function $\Theta$ that (1) maps $W$ injective onto $\wp(F)$ and (2) maps $L$ onto $\hat{F}$ so that for every $w \in W$ and every $\phi \in L$ it holds:

$$w \models \phi \quad \text{iff} \quad \Theta(w) \models_a \Theta(\phi).$$

If the set $F$ of a rigid logic is finite and the function $\Theta$ is recursive, then we call this logic *finitistic*.

## The connection between interpreted languages, rigidity and finitism

1. Philosophical logics (in the technical sense just described) are always rigid.
2. A rigid logic is finitistic, iff the set of structures $W$ is finite.
3. Every finitistic logic is decidable (regarding both satisfaction and logical consequence).
4. Uninterpreted languages generally are not rigid (because the class of structures generally is not a set). (An important counter-example is propositional logic.)

☞ Although finitistic languages are decidable via truth table method, we will also need *deductive systems* in a rigid framework, because of questions of speed.

## Toward an encyclopedia of philosophical logics

☞ The notion of a rigid/finitistic language allows a reduction of interpreted languages to propositional logic (cf. Henkin-semantics).

☞ We can discuss the philosophical features of logics in the realm of set theory here, what clearly is impossible in the case of mathematical logics.

☞ The project of an *encyclopedia of philosophical logics* is the project of the development of a catalogue of definitions of 'set-theoretical predicates' for philosophical features of logics.

**Some examples:**

## Example I: names for propositions, predicates and functions

- If a rigid logic contains a set $A$ of *propositional constants* then we have the power set of $A$ that provides the structures (possible worlds) over $A$.

- If a rigid logic contains a set $D$ of *individual constants* and a set $P$ of *first-order predicates* then we have a set of possible worlds, provided by $(D, P)$ in an obvious combinatorial way (subsets of $D$ and relations over those subsets).

- If a rigid logic contains a set $T$ of *type-theoretical objects*, together with a function $\omega$ that assigns to every element of $T$ its place in the ramified hierarchy of types $\tau$ then we have a set $W$ of possible worlds, provided by $(T, \omega)$ that is also constructed in an obvious combinatorial way. ($W$ is finite, iff $T$ is finite.)

- In a similar sense we can introduce functions, many-sorted relations and functions, etc.

## Example II: names for relations between worlds

Let $L'$ be any rigid language. Then we introduce

1. the set of possible worlds $W$ as a set of individual constants.

2. a set of $W$-variables and a set of $W$-predicates.

3. a constant $\aleph$ that designates on every place of a formula the world which is actual on this place.

4. $\Vdash$ as an additional syntactic element: if $y$ is a $W$-term and $\phi$ is a formula then $y \Vdash \phi$ is also a formula.

If $r$ is a binary modal predicate we can define

$$\Box\phi := \forall y : r(\aleph, y) \rightarrow y \Vdash \phi$$

and have a perfect expression for modality in the Kripkean sense.

## Example III:
## modal interpretations

☞ Semantic interpretation for the $W$-predicates can be provided either via a fixed interpretation or on a second level of semantic interpretation (in an increased version of the language with an additional factor of first-order complexity over $W$).

☞ Possibly there can be introduced a ramified hierarchy of semantic interpretations and relations over semantic interpretations.

☞ Relations can be defined between sets of semantic interpretations and arbitrary types of other sets.

☞ Of course, we also can introduce functions from arbitrary sets (of semantic interpretations) to arbitrary sets (of semantic interpretations).

☞ It seems likely that *every* aspect of reasoning about possible worlds (modal logic, relevance logic, dynamic logic, etc.) can be formalized in such a framework.

## Example IV:
## the many-valued case

Because rigid languages are languages in a propositional environment, it is very easy to implement many-valued versions of them. We simply have to modify the truth-functional interpretations of the basic language $L_a$ in a many-valued sense and get for every such interpretation $F_m$ a class of many-valued rigid logics.

## Conclusion

The aim of my approach is to develop a concise encyclopedia of philosophical logics. The main advantages of this approach are:

1. Logics can be described in a set-theoretical framework (as ‘set-theoretical predicates’ in Suppes’ sense).
2. It is easy to develop a unifying language for the specification of ‘features’ of logics here.
3. Questions of speed aside, there is no substantial need for deductive calculi in this framework, because there is a truth-table interpretation for every formula.
4. Therefore, the philosophical properties of logics can be discussed here without endless discussions of purely technical questions (completeness proofs, etc.).