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

Christian Damböck

Institute Vienna Circle, University of Vienna
Spitalgasse 2-4, Hof 1
1090 Vienna, Austria
`christian.damboeck@univie.ac.at`

A *logic* is understood here as an algebraic structure $\mathcal{L} = (F_{\mathcal{L}}, \mathbb{S}_{\mathcal{L}}, \vDash_{\mathcal{L}})$ that consists of a set $F_{\mathcal{L}}$ of *formulas* (sentences) plus a class $\mathbb{S}_{\mathcal{L}}$ of *structures* and a *relation of satisfaction* $\vDash_{\mathcal{L}}$ between them. We have the usual metalogical notions like logical consequence and logical equivalence, defined in the usual way. A logic is *rigid*, if there exists a set $F_{\mathbf{at}} \subseteq F_{\mathcal{L}}$ so that the logic can be reduced to the propositional logic that is defined over the set $F_{\mathbf{at}}$ of propositional constants, in an obvious way. If this reduction is recursive and the basic set $F_{\mathbf{at}}$ is finite, we call the rigid logic *finitistic*. – Finitistic logics are an important subclass of the class of all rigid logics, because of its metalogical merits: every rigid logic is decidable, regarding both satisfaction and logical consequence. I however discuss mainly the general rigid case here.

The main advantage of rigid languages is that we can describe them in terms of set theory. In other words: rigid languages are not a construction of pure logics or meta-mathematics, respectively, but they are a proper construction of mathematics: we can take every rigid logic as a mathematical structure and we can describe every property of this logic in terms of this mathematical structure. – A construction, similar to this, is the well-known Henkin-trick that reduces richer languages to first-order logic. But, whereas in the case of the Henkin-reduction we have a language that is not a part of the mathematical universe in itself and thus has the well-known properties of expressive power on the one hand and incompleteness (in the sense of: not being able to express everything in the realm of set theory) on the other, in the case of reduction to propositional logic we have a less powerful language which however has the serious advantage that it *is* part of the mathematical universe and thus it is complete in a pretty obvious sense.

Of course, rigid logics are useless for the purpose of mathematical foundation (because we have to assume mathematics to be able to define them). But they are extremely useful for practically every application that is not intended for the definition of mathematical languages. Thus rigid languages should be a good choice for *philosophical* logics of any kind, because we can define them in a pretty straightforward and unifying way.

My first example is the rigid first-order logic $\mathsf{RIG_p}(D, \mathbf{P}, \alpha)$ which is built over a (possibly infinite) set D of *individuals*, a finite or countable set $\mathbf{P}$ of *predicates* and a function $\alpha : \mathbf{P} \mapsto \mathbb{N}$ that assigns to each predicate its 'arity'. A *structure* $\mathfrak{S}$

is defined by a pair $(D_\exists, \pi)$, where $D_\exists \subseteq D$ provides the set of 'existing entities' of $\mathfrak{S}$ and π is a function that assigns to every predicate $P \in \mathbf{P}$ with $\alpha(P) = i$ a set $\pi(P) \subseteq D_\exists^i$. The syntax and semantics of the language is defined in an obvious way. – This language is defined as a *free logic*, because the set D should contain *every possible basic object* of the underlying universe; therefore a structure or possible world must contain only those basic objects that actually exist. Apart from that, the difference between rigid and non-restricted first-order logic lies only in the fact that the rigid language is restricted to a particular set of objects, whereas the universe of first-order logic (with set theoretical axioms) is the class of all sets.

In a similar way we can define more powerful relational or functional languages with arbitrarily complex hierarchies of types as rigid logics. I will briefly describe some examples in my talk.

The second example, which I will describe in more detail, is the construction of a *modal logic* over an arbitrary basic logic (which is usually but not necessarily thought to be rigid). Let $\mathcal{L} = (F_\mathcal{L}, \mathbb{S}_\mathcal{L}, \vDash_\mathcal{L})$ be any logic. Then we define $\mathsf{FLP}_\mathcal{L}(\mathbf{P}_w, \alpha_m, \mathfrak{M})$ as a language over $\mathcal{L}$, where $\mathbf{P}_w$ is a set of *modal predicates* (i. e. relations of comparability of possible worlds), α_m is a function that assigns to each modal predicate its arity and $\mathfrak{M} = (\mathfrak{W}, \Pi)$ is a *modal structure*. Here, $\mathfrak{W}$ is a set of structures out of $\mathbb{S}_\mathcal{L}$, called the set of all *possible worlds* of the modal structure. (If the logic is rigid, it possibly holds that $\mathfrak{W} = \mathbb{S}_\mathcal{L}$.) Π assigns to every modal predicate $P \in \mathbf{P}_w$ with $\alpha_m(P) = n$ a set $\Pi(P) \subseteq \mathfrak{W}^n$. We have the formulas of the basic language in $\mathsf{FLP}_\mathcal{L}$ plus some obvious syntactic elements (with an obvious semantic interpretation) for quantification over possible worlds. Additionally, we have the syntactic device $\Vdash$, defined as a relation between possible worlds and formulas. (The purpose of this device is that it allows us to define modal operators in the sense of Kripke-semantics.) The semantics for $\Vdash$ is defined as

$$\mathfrak{S} \vDash a \Vdash \phi \quad \text{iff} \quad \mathfrak{S}(a) \vDash \phi.$$

Here $\mathfrak{S}$ is a structure and a is a $\mathfrak{W}$-constant, $\mathfrak{S}(a)$ assigns to the constant a structure (the structure $\mathfrak{S}'$ that is assigned to the constant a by the structure $\mathfrak{S}$). This language provides a generalized framework for modal logics of arbitrary complexity. We also can enrich the framework by allowing some more complex operations with possible worlds (by introducing functions and higher-order terms).

Finally, I will discuss briefly some examples for non-classical languages like many-valued logic in such a rigid framework.

The concluding remark of my talk will be this. The rigid framework is intended as a unifying account that should allow us to develop an *encyclopedia of philosophical logics*, in a more straightforward way than we will be able to in rather syntactically oriented frameworks (frameworks that are based on the notion of a deductive system).