Rigid Logic as a Framework
for the Empirical Sciences

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

December 10, 2011

Abstract

A ‘rigid logic’ is a logic that is restricted to fixed domains of objects. This paper presents a formal account of rigid logic and illustrates this account in comparison with classical predicate logic. Moreover, it shows how to use rigid logic as a framework for modal logic and illustrates its merits in the context of philosophy of the empirical sciences, formal epistemology, and formal ontology. The overall argument is that rigid logic may provide a proper framework for all kinds of theories beyond the field of pure mathematics.

1 Introduction

A rigid logic is based on a fixed (therefore rigid) ontology.1 This kind of formalism shall appear useful in every context where we have to consider concrete (empirical) objects with some concrete properties, without being able to provide a purely structural specification. Thus, it may turn out that rigid logic provides a proper framework for almost every context outside of pure mathematics. As rigid logic assumes set theory and mathematics and tries to provide a formal framework for the empirical sciences only, it may be seen as a ‘logification’ of Patrick Suppes’ famous motto that “philosophy of science should use mathematics, and not meta-mathematics”.2

Rigid logic, in principle, is not at all a new idea. The logics that were considered by Frege, Russell, Carnap, and other classical philosophers of logic,
until the end of the 1920s, were rigid. The present paper, indeed, is a plea to go back to the rigid roots of pure logic, wherever we are concerned with theories of the empirical sciences and not with theories of pure mathematics. Rudolf Carnap, who developed in his seminal study [8] an account of theories of the empirical sciences, which at first glance may seem to be purely structural (cf. §16, in particular) came to the conclusion, at the end of his book, that a purely structural characterization fails and has to be replaced by an account that is based on ‘founded relations’ (§ 154) and, therefore, is rigid in fact (see also p. 11 below). Moreover, the whole philosophical discussion in the context of the so-called ‘new theory of reference’ (Kripke, Putnam) is nothing other than a broad establishment of the claim that a purely structural identification of empirical objects is generally impossible. Concrete examples of logical frameworks for the empirical sciences that are rigidly construed in fact are Bas van Fraassen’s semi-interpreted languages (as inspired by Evert Willem Beth) and David Lewis’ conception of theoretical terms in [21]. At least some aspects of a rigid framework are implemented in ‘structuralism’ as developed by Joseph Sneed, Wolfgang Stegmüller, and others. More recently, Jeff Ketland has shown (on the basis of a classical result by M. H. A. Newman and suggestions of William Demopoulos and Michael Friedman) that a proper conception of ‘ramsification’ of theoretical entities may also (and have to) include rigid elements, by means of cardinality claims. More precisely, a theory of the empirical sciences is true, if and only if its Ramsey sentence has a model that is isomorphic to a proper model $M$ of the empirical and theoretical world, with respect to both the empirical and the theoretical objects that $M$ refers to. In other words, in a theory of the empirical sciences both the domain of empirical objects and the domain of theoretical objects have to be fixed. Thus, Ketland’s theorem clearly shows that rigid logic is an adequate tool for the formalization of empirical theories. Hannes Leitgeb’s recent approach towards a re-evaluation of Carnap’s Aufbau also demonstrates that a ‘constitutional system’ for empirical concepts inevitably has to be rigid or interpreted in some way.

In spite of the fact that the very idea of empirical sciences as based on some sort of rigid ontologies is rather well-established in the philosophy of science community, there is surprisingly no well-established account of a logic that implements this restriction. (Rather, the above mentioned authors mostly used ad hoc restrictions of classical logic that simulated a rigid setting.) The aim of this paper is to develop such an account, in order to have a simpler toolkit at hand, for the purpose of formalizations of empirical theories. After a short presentation of the technical details of rigid logic (section 2) we compare this framework with classical (higher-order) predicate logic (section 3) and show how

---

3For an account of the development of the modern semantic picture of logic, from that rigid starting point, see [16].
4See [18, 23].
5See [31, 32, 33] and [4].
6See [25, 26, 2] and [11].
7See [17, 22] and [12].
8See [19], in particular, section 8.
to implement modal notions, in the context of a hybrid rigid logic that allows us to quantify over the structures and formulas of a basic rigid logic (section 4). We finally illustrate the merits of rigid logic, in the context of philosophy of the empirical sciences and formal epistemology (section 5) and formal ontology (section 6). In the latter two sections we also will formulate some proposals for future work in the field of formal philosophy that may be developed in a setting of rigid logic.

2 The logic

A structure frame $\mathcal{S} = (S, T, S)$ is based on a non-empty set of non-empty and pairwise disjoint sets $S$ of sorts and a set $T$ of types that is defined as a subset of the set of all finite sequences of elements of $S$. The elements of sorts are called the objects of a structure frame. We define the set $S$ of all objects as

$$S := \bigcup_{s \in S} s.$$ 

If $T$ is the set of all possible assignments of types with suitable objects, i.e.,

$$T := \bigcup_{(s_1, \ldots, s_i) \in T} s_1 \times \ldots \times s_i,$$

then the set of structures $S$ of a structure frame is defined as a set of subsets of $T$, i.e., we have

$$S \subseteq \mathcal{P}(T).$$

On this basis, we define a rigid logic $L_r = (\mathcal{S}, \mathcal{F}, \models)$ that consists of a structure frame $\mathcal{S} = (S, T, S)$, a set of formulas $\mathcal{F}$, and a satisfaction relation $\models \subseteq S \times \mathcal{F}$. More precisely, $\mathcal{F}$ and $\models$ are defined in the following way:

We have a countable set of variables $x, x', \ldots$. Every object in $S$ is defined as a constant, and there is an additional constant NULL with NULL $\notin S$. Constants and variables are terms. If $\phi$ is a formula (in the sense defined below) and $x$ is a variable, then $\forall x. \phi$ is a term. Additionally, every sort in $S$ is represented by a unique $S$-constant $s$.

Given a structure $\mathcal{S} \in S$, we define interpretations for constants $c$ and formulas $\phi$ with the only free variable $x$:

$$\mathcal{S}(c) := \begin{cases} c & \text{if } c \in \mathcal{S}, \\
null & \text{otherwise.} \end{cases}$$

$$\mathcal{S}(\forall x. \phi) := \begin{cases} o & \text{if } o \in S \text{ is the definite object} \\
null & \text{such that } \phi[o/x] \text{ is true,} \\
null & \text{if there is no such definite object.} \end{cases}$$

Every finite sequence of $S$-terms is defined as an atomic formula. If $t$ and $t'$ are $S$-terms, then $t = t'$ is a formula. If $x$ is a variable, $s$ is an $S$-constant,
and $\phi$ is a formula, then $\forall x : \phi$ and $\forall x \in s : \phi$ are formulas, i.e., we introduce the latter as a device for quantification over the objects of a particular sort $s$. We further introduce the usual logical connectives, and the $\lambda$-abstractor: if $x$ is an $S$-variable, $t$ is a term and $\phi$ is a formula, then $(\lambda x.\phi)(t)$ is also a formula.

Among others, we have the following rules for satisfaction of a formula in a structure $\mathcal{S}$:

$$
\begin{align*}
\mathcal{S} \models (t_1, \ldots, t_n) & \iff (\mathcal{S}(t_1), \ldots, \mathcal{S}(t_n)) \in \mathcal{S} \\
\mathcal{S} \models t = t' & \iff \mathcal{S}(t) = \mathcal{S}(t') \\
\mathcal{S} \models \forall x. \phi & \iff \text{for every } c \in \mathcal{S} \text{ it holds } \mathcal{S} \models \phi[c/x] \\
\mathcal{S} \models \forall x \in s. \phi & \iff \text{for every } c \in s \text{ it holds } \mathcal{S} \models \phi[c/x] \\
\mathcal{S} \models (\lambda x. \phi)(t) & \iff \mathcal{S} \models \phi[\mathcal{S}(t)/x]
\end{align*}
$$

Here, the $t_1, \ldots, t_n, t, t'$ are terms that do not contain free variables, $s$ is an $S$-constant, and $x$ is a variable. (We do not define satisfaction for formulas that contain free variables.) The remaining connectives and quantifiers ($\land$, $\lor$, $\exists$ etc.) are specified in a straightforward way.

Rigid logic is based on a limited set $\mathcal{S}$ of objects, whereas in classical predicate logic, the domain is unrestricted. The rigid layout implies in particular that the whole logic is trivially decidable (regarding both satisfaction and validity), if both $\mathcal{S}$ and $T$ are finite, because in this case we obtain a finite set of finite structures. The rigid logic over a structure frame $\mathcal{F}$ can be seen as an instance of propositional logic (with possibly uncountably many propositional constants), as we can reduce the above formal construct to propositional logic with the set $T$ of propositional constants.\textsuperscript{10} Another obvious reduction strategy would be many-sorted first-order logic, where the sorts are restricted to the respective sorts of a structure frame (in the sense of a restricted form of Henkin- semantics). However, we do not discuss these strategies any further here.

### 3 A comparison with classical predicate logic

Meta-logical questions aside, there are interesting parallels and differences between rigid logic and classical predicate logic. The actual formal conception of rigid logic deviates significantly from classical logic, even on the level of syntax. Individual constants have a fixed reference here, and because we refer to objects only via individual constants, every sort is restricted to a particular set of objects, namely, these objects that are named by the constants of the sort.

The types of rigid logic can be seen as a generalization of the type structures of classical higher-order logic with a ramified type structure. If $t = (s, s_1, \ldots, s_n)$ is defined as a type of a structure frame $\mathcal{F}$, then we call $s$ the labeling sort of this type, i.e., $s$ contains the names of the relations of type $t$. Thus, we mostly describe $t$ in the more illustrative symbolic form $s(s_1, \ldots, s_n)$. The atomic formulas of type $t$, in turn, are described as $c(c_1, \ldots, c_n)$ etc. As a structure

\textsuperscript{10}See [10] for a discussion of this kind of reduction.
specifies a subset of $s \times s_1 \times \ldots \times s_n$, we just obtain a particular relation for each label in $s$, that is, we can also quantify only over this limited range of relations that are specified for the elements of $s$ and generally not over the whole set $\wp(s_1 \times \ldots \times s_n)$. In particular, we obtain a structure that contains all possible relations of this type, only if the cardinality of $s$ is greater than or equal to the cardinality of $\wp(s_1 \times \ldots \times s_n)$. There are also some obvious parallels with Henkin-semantics, as both in the latter and in rigid logic, a structure specifies a limited range of relations for each type. Thus, we may specify a structure frame in such a way that it only contains structures that satisfy the comprehension axioms, etc.\footnote{For an account of (higher-order) Henkin-semantics, in comparison with standard semantics see [13, ch. 2].}

If $s(s_1, \ldots, s_n)$ is a type of a structure frame $\mathcal{S}$, then, unlike the case of classical logic, there may also be a number of further types with the sort $s$ as labeling sort. In principle, we may obtain a type $(s, x)$ for any finite sequence of sorts $x$. This is always the case if our type structure is total insofar as $T$ contains every finite sequence of sorts. In classical logic, by contrast, there is only one type $t(t_1, \ldots, t_n)$ with the labeling type $t$. We may react to this flexibility in two ways. Firstly, we simply may restrict a structure frame in such a way that a classical hierarchical type structure results. We call a structure frame regular, if it holds:

1. There are some sorts that do not function as labeling sorts of any type: the basic sorts.

2. Every sort is the labeling sort of one type at most.

Basic sorts represent the first-order types of classical logic here. The second rule ensures that our type structure is homomorphic. This strategy may be used in the context of philosophy of science, where clear and unambiguous type structures are mostly desirable (see section 5, below).\footnote{A regular type structure in this sense generally may build a finite hierarchy. If we want to ensure that every possible type of the classical ramified type hierarchy is contained in $T$, we have to require also that every finite sequence of sorts $s_1, \ldots, s_n$ is represented by a type $(s, s_1, \ldots, s_n)$.} The second strategy would be to pass on such restrictions and allow the type structure to be polymorphic in some sense. This strategy may be used in the context of formal ontologies for everyday language where open and flexible type structures are desired (see section 6, below).

To describe relations, we need types with two or more places, as the first place always represents the labeling sort of a relation and the second and the following places represent its arguments. Unary types, therefore, clearly must have a different function here. In principle, there are two ways to interpret unary types. Firstly, we may interpret them by analogy with zero-level types or propositional types of classical logic, i.e., every object $c$ of a sort $s$ that is defined as a unary type represents a particular proposition, which is true in a structure $\mathcal{S}$, iff $c \in \mathcal{S}$. Secondly, we may interpret them as existence predicates, i.e., $c \in \mathcal{S}$
means that the object \( c \) exists in the context of structure \( \mathcal{G} \). Whether we prefer the first or the second interpretation clearly is more a matter of convention. The second interpretation, in particular, allows us to interpret a rigid logic as a free logic.\(^{13}\) If we do not adopt any further restriction, then our logic turns out to be a free logic in the unrestricted sense, i.e., we have objects \( c \) that do not exist in a particular structure and, nevertheless, have some positive properties insofar as there are relations \( P \), etc., such that \( P(c) \), etc., is true. To avoid this mostly undesired feature of free logic, we have to adopt the following obvious rule, for every structure \( \mathcal{G} \) of a structure frame and every sequence of constants \((c_1, \ldots, c_n)\):

\[
(c_1, \ldots, c_n) \in \mathcal{G} \rightarrow c_i \in \mathcal{G}, \text{ for every } i.
\]

In this case, non-existing entities cannot have positive properties at all. We call a structure frame that is restricted to the structures that fulfil this rule as existentially consistent.

In particular in the case of existentially consistent structure frames it may be useful to restrict quantification to these objects that actually exist in a structure. For this purpose, we may introduce an additional quantifier \( \forall^E \):

\[
\forall^E x : \phi := \forall x : (x) \rightarrow \phi.
\]

Here \((x)\) represents the unary atomic formula with the argument \( x \), i.e., the existence predicate in the above-mentioned form.

Another feature that identifies our logic as an instance of free logic is the additional constant \( \text{NULL} \), which is a constant that never refers and, therefore, represents non-existence.\(^{14}\) We use this constant, in particular, to deal with non-existing objects and definite descriptions. The latter are defined in such a way that \( \iota x. \phi \) either refers to the definite object \( x \) such that \( \phi \) or, if such an object does not exist, to the constant \( \text{NULL} \). In other words, \( \iota \)-terms are generally specified as partial functions.

In the case of \( \lambda \)-abstractions, we also have to take care about existence matters. Our actual definition implies, for example, that a formula \( (\lambda x. \phi)(t) \) may be true, even if \( t \) does not represent an existing object. If \( k \), for example, represents the present king of France, and \( B \) the property of being bald, then \( (\lambda x. \neg B(x))(k) \) turns out to be a true formula, i.e., ‘the present king of France is not bald’ turns out to be true. To avoid this absurdity, we have to replace \( \lambda \) in the formula above with the operator \( \lambda^E \), which is explicitly defined in the following way:

\[
(\lambda^E x. \phi)(t) := (t) \land (\lambda x. \phi)(t).
\]

Consider also the following somewhat funny case. We may represent ‘the nothing’ (‘das Nichts’) by means of our non-referring constant \( \text{NULL} \) and the property

\(^{13}\)See [3], for an overview.

\(^{14}\)Such a constant was introduced, for example, in [7, p. 37].
of being nothing (‘to noth’, ‘zu nichten’) by means of equality with NULL. Then, we obtain the formulas \((\lambda x. x = \text{NULL})(\text{NULL})\) and \((\lambda^E x. x = \text{NULL})(\text{NULL})\) as more or less obvious expressions for ‘The nothing noths’ or ‘Das Nichts nichtet’.\(^{15}\) The former is a tautology; the latter is a contradiction.

In the case of identity-formulas, we also have to take care about non-existing objects. Formulas of the form \(c = c'\) do not avoid that identities between non-existing objects like ‘Zeus = Pegasus’ turn out to be true formulas (because \(\text{NULL} = \text{NULL}\) is true). Again, we have to introduce a clause here that restricts identity to existing objects, for example, by means of a stronger identity relation \(\equiv:\)

\[
t \equiv t' \; := \; (t) \land t = t'.
\]

Another somewhat unusual feature of rigid logic, in its present syntactic treatment, is that we introduced only one type of variables. This shall appear useful because of the possibility of polymorphic type structures. With the formula

\[
\exists x : x(c) \land \phi
\]

we search for an object \(x\) that describes a property of \(c\) and fulfils condition \(\phi\). It may be not clear, from the beginning, from which sort \(x\) should be, because, in the polymorphic case, there may be a number of types that characterize properties of \(c\). On the other hand, if we want to restrict quantification to a particular sort \(s\), we can realize this in the following form:

\[
\exists x \in s : \psi.
\]

We conclude this section with a brief discussion of an obvious objection against the rigid layout of a logic, namely, the claim that there are names (in everyday language, even in the sciences) that do not have a rigid reference but rather refer to different objects in different contexts. It is important to understand that the existence of non-rigid names is not at all an argument against rigid logic. A rigid logic is rigid \textit{only because} objects are introduced here by means of rigid designators. However, rigid designators of this rather technical nature and the names of a natural language are obviously quite different things. In particular, each proper name \(n\) of a natural language may be considered to be an object that has both an intension and an extension. Let \(n\) be an object of a sort of names, in the context of our rigid logic. Then, we may interpret every direct attribution of properties to \(n\) as the attribution of ‘intensional properties’ to \(n\). Furthermore, we may introduce a partial function \(\downarrow\) into our logic, such that the extension or reference of \(n\) is defined as the term

\[
\ell x. \downarrow(n, x).
\]

This extension may change from structure to structure (and it may be absent in some or all structures). For further considerations on this topic, see section 6 below.

\(^{15}\)Cf. [6] for a famous critique of Heidegger’s (in)famous dictum.
4 Modal aspects

A main advantage of rigid logic is the possibility to define a ‘modal logic’ simply as a logic that quantifies over the sets of formulas and structures of a basic rigid logic. As both the formulas and structures of a rigid logic are defined as sets we can interpret them, without any restriction, as sorts of another (‘higher-order’) rigid logic.\footnote{In fact, modal logic of any reasonable kind is a logic that quantifies in some way over such things as ‘possible worlds’. First-order interpretations of modal logics in this sense are studied in so-called correspondence theory. See [30], for an overview. However, in most cases modal logics are specified in such a way that the quantification takes place rather indirectly, by means of some modal operators such as □ and ◻. However, there are exceptions, even in the realm of non-rigid logic. In hybrid logic there are some tools introduced that allow us to quantify directly over possible worlds. Cf. [5, ch. 14]. A famous example in the classical literature where quantification over possible worlds is explicitly realized is David Lewis’ ‘counterpart theory’ in [20]. In rigid logic, quantification over possible worlds becomes much simpler, indeed, because here the basic language contains a concrete set of structures, and not a whole class that is based on the universe of all sets. Thus, we may quantify over the range of all these structures, without having to leave the realm of set theory.}

The most elegant way to implement a rigid logic as a hybrid rigid logic in this sense would be the self-referential conception of a rigid logic \( L_r = (\mathfrak{F}, \mathfrak{F}, \models) \) with \( \mathfrak{F} = (\mathcal{S}, T, \mathcal{F}) \), such that the formulas, structures, and a set that contains \( \models \) are defined as sorts and there is a type that contains the relation \( \models \) between \( \mathcal{S} \) and \( \mathcal{F} \). We then simply had to restrict \( \mathcal{S} \) to the set of all structures in which the relation \( \models \) is interpreted correctly, following the above specifications. However, there can be no doubt that such a conception falls short of being ‘diagonalizable’ because we have terms that refer to formulas; therefore, we can define the satisfaction relation in the realm of \( L_r \)-formulas and inevitably must arrive at paradoxes, in the sense of Gödel and Tarski.\footnote{See [15, 27].}

To avoid this mostly undesirable constellation, we have to ensure that our logic is not self-referential, in the just mentioned sense. We can realize this by simply restricting ourselves to these rigid logics, where \( \mathcal{F} \cup \mathcal{S} \cup \{\models\} \) and \( \mathcal{S} \) are disjoint, i.e., where the ‘meta-lingual’ objects – the formulas, structures, and the relation \( \models \) – are not introduced as objects on the level of the ‘object language’. We call a rigid logic that fulfills this condition as non self-referential.

Let \( L_r = (\mathfrak{F}, \mathfrak{F}, \models) \) with \( \mathfrak{F} = (\mathcal{S}, T, \mathcal{F}) \) be a non self-referential rigid logic, then we can establish modal quantification over structures and formulas (in a non self-referential way), in the context of another non self-referential rigid logic \( L'_r = (\mathfrak{F}', \mathfrak{F}', \models') \) with \( \mathfrak{F}' = (\mathcal{S}', T', \mathcal{F}') \), such that

\[
\{\mathcal{F}, \mathcal{S}, \{\models, \ldots\}\} \subseteq \mathcal{S}' \quad \text{and} \quad (\{\models, \ldots\}, \mathcal{S}, \mathcal{F}) \in T'.
\]

Here, the set \( \mathcal{S}' \) of \( L'_r \)-structures has to be restricted to these structures that interpret \( \models \) correctly, in the sense of our above specifications. Self-referentiality is clearly ruled out here because of the strict distinction between the meta- and the object-levels.

Unary modal operators like \( \Box \) and \( K \) are defined here explicitly, by means
of a suitable accessibility-relation $R$:

$$\Box \phi := \forall x \in S : R(a, x) \rightarrow \models (x, \phi).$$

This is an $L'_r$-formula where $\phi$ refers to an object of the sort $\mathcal{F}$ and $a$ refers to the object of the sort $S$ that represents the actual world. $\models$ is the relation of satisfaction of $L_r$ as redefined as a relation of $L'_r$. More complex modal operators like the until-operator or the relevance implication are also introduced here explicitly, in an obvious approach.

Quantified modal logic is realized here in a pretty straightforward manner, as the basic rigid logic generally has to be considered a quantified one (except for the case that the basic structure frame contains unary types only). As the whole layout of the basic logic is rigid, we only have two possible cases here: firstly, a fixed domain account where we restrict ourselves to structures that all contain the same basic objects (i.e., where every basic object exists); secondly, a variable domain account where in different structures different basic objects are defined as existing objects.\(^\text{18}\)

We may also quantify over formulas and other objects of the basic logic $L_r$ as long as we have introduced them as sorts of $L'_r$. For example, the formula

$$K \neg \exists x \in \mathcal{F} : K x$$

provides an obvious formalization for Socrates’ dictum “I know that I know nothing”. Here, $K$ is a unary modal operator, defined as above and interpreted in the sense of “a particular subject knows $\phi$”. The formula is true if $K\phi$ is false, for every formula of the basic rigid logic, i.e., if the subject really knows nothing (at the level of the basic rigid logic).

Another example are formulas that contain references to objects of the basic logic, in the context of a meta-logic $L'_r$ where such things as the set $S$ of all basic objects are defined as sorts in $S'$. We may use this in the context of the formal specification of theories of the form

$$(\mathcal{O}, \mathcal{A}),$$

where $\mathcal{O}$ is a set of $L_r$-objects and $\mathcal{A}$ is a set of $L_r$-formulas that function as axioms, in the context of the ontology $\mathcal{O}$, such that the range of structures is restricted to these ones where only the elements of $\mathcal{O}$ are defined as existing objects (in the realm of an existentially consistent structure frame $\mathfrak{S}$).\(^\text{19}\)

Sometimes, we may even need a second or third level of a modal logic of the just-described form. For example, the axioms of a theory specification may also have to include modal formulas out of the context of $L'_r$. To realize this, we will need a second-order hybrid rigid logic $L''_r$, which is based on the basic rigid logic $L_r$ and the (first-order) hybrid rigid logic $L'_r$, in an obvious way.

\(^{18}\)For an overview of quantified modal logic, see [14].

\(^{19}\)Such rigid formalizations of theories are used, in particular, in [11].
5 Applications in philosophy of science and formal epistemology

The general philosophical point that we want to make about rigid logic here is that it may appear to be the most straightforward formal framework for applications in contexts of any kind where purely structural characterizations are not available, as the respective context depends in some sense from empirical circumstances. In a rigid logic, we characterize these empirical circumstances by means of rigid designators that point to empirical objects and properties of them. Given all the available empirical data, as characterized that way, we may also add some amount of characterizations of a (more or less) structural nature. For example, we may specify our structure frame in such a way that every structure must contain all possible relations of some or all types. This would imply that Ramsey sentences that quantify over the respective sorts of labels can be introduced here, more or less in the classical way. Let us assume, for example, that the structure frame $\mathfrak{F}$ of our basic rigid logic is regular, in the sense defined in section 3, and that there is a particular sort $s$ that represents the labels of a type $s(1,\ldots,n)$. Then we call the structure frame structurally complete with respect to $s$, iff (1) the cardinality of $s$ is greater or equal to the cardinality of $s(1 \times \ldots \times s_n)$ and (2) each structure in $\mathfrak{F}$ either contains all the formally possible relations of the type $s(1,\ldots,n)$ or fulfills the comprehension axioms w.r.t. $s$ (in the sense of Henkin-semantics). If the structure frame $\mathfrak{F}$ is structurally complete w.r.t. $s$, then formulas of the form

$$\exists x \in s : \phi$$

obviously must have essentially the same meaning (and extension) than Ramsey sentences in the realm of classical logic.

Note also that every reasonable case for ramsification, in the context of philosophy of the empirical sciences, obviously has to have the following form. There is a fixed empirical reality $E$ that is characterized by means of a set $e$ of empirical objects plus a number of relations that are based on $e$. In addition to this, we have a number of theoretical objects and relations between theoretical objects and between theoretical and empirical objects. The latter two categories of objects (i.e., theoretical terms and ‘bridge laws’) will have to be ‘structurally’ characterized here insofar as we have to replace each relation label for a theoretical term or a bridge law with a suitable variable that is bound to an existential quantifier. However, as mentioned in the introduction, above, Ketland’s theorem (as informed by Newman, Demopoulos, and Friedman) shows that even in the just described setting the domains of theoretical objects can be assumed to be fixed, insofar as a Ramsey sentence can be true here, only on the basis of a suitable cardinality claim, i.e. there has to be a model that fits isomorphically not only into the empirical but also the theoretical parts of our model for the actual world. Therefore, theories of the empirical sciences can never be ‘purely structural’ in any meaningful way that may rule out the setting of rigid logic.

In the rest of this section, we provide brief discussions of three more concrete
examples for possible applications of the rigid framework, in the context of philosophy of the empirical sciences and formal epistemology.

First example: The framework of rigid logic can be used as a formal (meta-)framework for ‘structuralism’, as it was developed by Joseph Sneed, Wolfgang Stegmüller, and others. As demonstrated with some detail in [11] the structuralist framework, in its original form, suffers from being too structural in fact. This means that if we characterize a scientific theory in this framework, we first have to give a purely structural characterization, essentially, by means of a set of axioms \( \mathcal{A} \), and then we have to add an informal description \( I \) that shows how we may reduce the whole range of models of \( \mathcal{A} \) to these models that are really ‘intended applications’ of the theory. This practice is a little bit disappointing, as a typical structuralist specification begins with a highly sophisticated formal specification of axioms and ends up in the context of a purely informal description that completely lacks any well-defined formal criteria and rules. In the rigid setting, we can improve this framework insofar as we can replace the informal description \( I \) with the specification of an ontology \( \mathcal{O} \) that is an integral part of our formal framework. The basic claim of this rigid re-definition of structuralism is simply that the restriction of a theory to its intended models can be completely realized in restricting the theory to the range of objects it really wants to talk about, that is, to its concrete ontology \( \mathcal{O} \).

Second example: The framework of rigid logic is particularly suited as a framework for a formal description of the dynamic of scientific theories. As pointed out in [1] a specification of a scientific theory that is based on both an ontology \( \mathcal{O} \) and a set of axioms \( \mathcal{A} \) (in the aforementioned way) allows us to deal with such concepts as (in-)commensurability, (ir-)reducibility, and structural (dis-)continuity in quite an elegant manner. Essentially, we gain a second descriptive dimension here, in comparison with purely structural frameworks, because there are always two ways in which one instance of a theory may be compared with another. For example, there may be structural continuity insofar as the axioms remain unchanged, but discontinuity on the level of the ontology insofar as the meaning and/or the reference of some or all scientific terms has changed (in the sense of Kuhn’s notion of ‘incommensurability’). On the other hand, the ontological layout of a theory may remain unchanged, but the axioms may change (simply because we refuted some axioms of the old theory and replaced them with new ones).

Third example: The framework of rigid logic may be used as a framework for a new Aufbau-like system, along the lines that were drawn out by Hannes Leitgeb in [19].\(^2\) As pointed out in section 8 of the latter paper, a proper solution to ‘Quine’s problem’, namely, the so-called second dogma of empiricism that every theoretical concept can be reduced to immediate experience,\(^2\) has to make use of a reduction procedure that picks out a proper model from the range of all possible models. In order to pick out such a model, by means of a definite

\(^2\)Leitgeb’s work is not the only Aufbau-like project that is recently worked out. See also the Canberra plan [9] and David Chalmers’ recent book project (http://consc.net/constructing/).

\(^2\)See [24].
description, the fundamental relations of an Aufbau-like constitutional system have to be ‘founded’, in a sense that (as already mentioned) was pointed out by Carnap himself in §154 of the Aufbau. Roughly, a structure $S$ is ‘founded’, if and only if it points to a unique structural possibility insofar as there are no other structures $S'$ with $S \neq S'$, so that $S$ and $S'$ are isomorphic. The merit of founded structures, then, is simply that we may have axiomatic descriptions $\phi$ here that provide definite descriptions for structures, that is, the term $\iota x \in S. \phi$ may refer. The advantage of the rigid framework, in this context, is simply that it straightforwardly allows us to specify structures that are ‘founded’, in the aforementioned way. Therefore, rigid logic appears to be tailored as a framework for Aufbau-like systems (at least, if we do not want to adopt in such a system a logicist foundation of mathematics).

6 Formal ontology

Another application of rigid logic may be developed in the field of ‘formal ontologies’ for everyday language. The following sketch is an outline for future work of the author of this paper that partly will converge with the above mentioned Aufbau-like project.

A structure frame $\mathfrak{F}$ provides a particular ontology, in the context of the objects of $S$ that are distributed to a set $S$ of pairwise disjoint sets. These sorts of a structure frame may be more or less straightforwardly interpreted as different non-logical ‘categories’ of things (with the only formal restriction that these ‘categories’ must not overlap). We may introduce ‘categories’ such as the following:

- $S \ldots$ spatial objects
- $s \ldots$ subjects
- $n \ldots$ names (intensions)
- $p \ldots$ phenomena (elementary experiences)
- etc.

Just for convenience, we interpret names here in such a way that they represent a particular intensional structure, that is, the name ‘pegasus’ is considered as an object that has such properties as ‘is a winged horse’ (or ‘pegasises’), etc. Then, we may associate names with such things as symbols and extensions. A symbol may be simply defined as a set of spatial objects $sym$, and we may have a relation $\sigma$ that associates names with symbols, such that $(\sigma, n, sym)$ is the proposition that $sym$ is a symbol for the name $n$. For the association of extensions we use the above-introduced relation symbol $\downarrow$. Unlike in the case of classical (homomorphic) predicate logic, we do not have to explicitly introduce two parallel type-hierarchies, one that represents extensional objects, one that represents intensional objects. We, rather, may introduce intensions by means

---

22 We refer to a research tradition here that is represented, among others, by Rudolf Carnap, Richard Montague, Pavel Tichý, and Edward Zalta. See [7, 28, 29, 35, 36].

23 Such an account with two parallel type hierarchies can be found in [13, ch. 7].
of one single sort of intensional objects (i.e., our set $n$ of ‘names’). Then, we associate the respective objects of this sort with the respective type, in the context of a number of polymorphic types of the form $\langle \{\downarrow\}, n, x \rangle$, where $x$ is any finite sequence of sorts. Of course, we have to know the actual extension type of a particular intensional object $n$ (at least we have to know the cardinality of $x$), but the structure may also ensure that the correct extensions are associated with $n$. In the case of a name like pegasus, the atomic formula $\downarrow('pegasus', \bar{c})$ will be false, for every finite sequence of constants $\bar{c}$. In the case of a ‘proper name’ $p$ like ‘Caesar’, ‘the present king of France’, or ‘Barack Obama’, we either will have exactly one spatial object $c$ such that $\downarrow(p, c)$ is true or no such an object at all. In the case of ‘natural kind terms’ $k$ like ‘water’, ‘tiger’, or ‘human being’, we will obtain a possibly empty set $k'$ of spatial or other objects, such that $\downarrow(k, c)$ is true, for exactly every $c \in k'$. In the case of more-dimensional relational names $r$ like ‘is married to’, ‘are the child of’, ‘are the coordinates of’, etc., we will obtain a possibly empty set $r'$ of sequences of spatial or other objects such that $\downarrow(r, \bar{c})$ is true, for exactly every sequence of objects $\bar{c} \in r'$. In all these cases, we may have to use the possibilities of polymorphism in an extensive way, as we will need a large number of types of the form $\langle \{\downarrow\}, n, x \rangle$. Note also that just in the case of proper names $n$ that have a definite object as their reference, the term $\iota x. \downarrow(n, x)$ will refer.

Some names may refer in an indexical way. For example, we may have names like ‘I’ or ‘here’ whose reference may be determined in dependence from a subject $s$ by means of terms like $\iota x. \downarrow('I', s, x)$, $\iota x. \downarrow('here', s, x)$, etc. Moreover, subjects may be associated with epistemic operators ($s$ ‘knows’, $s$ ‘believes’) and the like, and we may also associate each subject (in each state of our universe) with a (possibly higher order) structure that represents the subject’s actual ‘world picture’. In an Aufbau-like formal ontology, we also may associate subjects with elementary experiences. We may associate each subject $s$ in each temporal state of a (possible) universe where $s$ exists with a particular elementary experience from $p$. Moreover, we may define a particular relation $R$ of ‘recollected similarities between elementary experiences’ (and evaluate it in the context of each particular subject $s$).

The temporal aspect will be introduced here on the level of a hybrid rigid logic, in the sense of section 4 above. We assume that the structures $\mathcal{S}$ of the basic rigid logic represent (possible) temporal states of our universe and establish a partial (or linear) order relation $<$ over $\mathcal{S}$. If $<$ is a partial order, then we obtain the ‘possible worlds’ of our account as the set of maximal linear sub-orders of $<$. On this basis, we can introduce modal operators like $\Box$ (true in every possible world), $\Box$ (always true in the actual world), $P$ (true in a future state of the actual world), etc., in a more or less obvious way.

This sketchy presentation does not represent a concrete ontology but only gives a cue on how we may be able to specify concrete formal ontologies in the realm of rigid logic. We do not claim that it will be impossible to obtain any of the formal features that we may obtain in rigid logic even in a more classical, non-rigid framework. However, the high polymorphic flexibility of rigid logic may be gained in a more classical framework only with considerably
more formal effort, and in most cases, it may ultimately turn out that we just
had simulated a rigid framework in the classical realm.

References


tonic for Science. The Structuralist Program. D. Reidel Publishing Com-


[4] Evert Willem Beth. Semantics of Physical Theories. Synthese 12, 172-175,
1960.

[5] Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors. Hand-


[8] Rudolf Carnap. Der logische Aufbau der Welt. Felix Meiner Verlag, Ham-
burg, 1998 [1928].


[12] William Demopoulos and Michael Friedman. Bertrand Russell’s The Anal-
ysis of Matter: Its Historical Context and Contemporary Interest. Philos-


ume 277.


