Metaphysical Positivism

Roger Bishop Jones

Abstract

Formal models of aspects of Metaphysical Positivism

Created 2004/07/15

Last Change Date: 2010/08/08 15:50:44

http://www.rbjones.com/rbjpub/pp/doc/t001.pdf

Id: t001a.tex,v 1.6 2010/08/08 15:50:44 rbj Exp

Contents

1 Prelude

This document is intended as the penultimate chapter of the second part of "An Analytic History of Philosophical Analysis" [2]. It is made available as a separate document prior to publication of the whole and serves also as a guide to the other parts of the work in progress.

For an overview of the work as a whole see the first chapter of the first part [1].

The present content of this document was not written for the purpose of inclusion in the book, but is some material broadly in the scope of the intended chapter which is expected to be replaced in due course by models more carefully engineered for the intended purpose.

2 Introduction

Metaphysical Positivism is a philosophical system among whose principle objectives is that of articulating a method which suffices to render rigorous, deductive reasoning in philosophy and in all other areas of knowledge in which deductive methods may be applicable.

This chapter is intended to provide formal models which underpin the proposed methods and their applications, and constitutes an application of the method to its own exposition.

A central fulcrum, around which the exposition is expected to turn, is the concept of "logical truth" in a broad sense, which we also call "analyticity" and "logical Necessity". These concepts take central place in an account of the analytic method, because the results of a sound deduction can be expressed as a logical truth. (Some prefer to take entailment as the fundamental concept, which idea I will neither dispute nor adopt.)

At present I envisage that this chapter will be concerned primarily with giving precise meanings to certain funcamental concepts through models of certain kinds of language.

3 Logical Truth

To model the fundamental notion of "logical truth", we consider various ways in which the semantics of languages can be formally defined. There are many ways of doing this. The choice of how to render the semantics may be part of the process of defining a language or class of languages. Definitions of logical truth are then specific to classes of languages which have the same kind of formal semantics.

3.1 Bare-Boned Truth-Conditions

3.2 Types

The following "primitive" types are introduced:

SML

```
new_type("S",0); (* sentences *)
new_type("C",0); (* contexts *)
new_type("W",0); (* possible worlds *)
new_type("P",0); (* propositions *)
```

3.3 The Semantics

The semantics of our language comes in two parts. A semantic map which delivers propositions, the meanings of sentences in context, and a propositional evaluation map, which extracts truth conditions from a proposition.

The purpose of this document is not to consider the semantics of any particular language but to reason about semantics and concepts defined in terms of semantics. We could have used variables for the semantics, but this time I decided to use loosely defined constants. So the following definitions only tell you the type of the semantic map and the evaluation map, they don't tell you anything more than that, so any results we subsequently obtain using these definitions will hold good for any language with a semantics of the type stipluated here.

HOL Constant

$\boldsymbol{sm} : S \times C \to P$

T

HOL Constant

$\boldsymbol{pem} : P \times W \to TTV$

T

Note that TTV is a type consisting of three 'truth' values, whose names are: $pTrue$, $pFalse$ and pU.

3.4 Necessity

A proposition is 'necessarily t' if it takes truth value 't' in every possible world.

HOL Constant

$\boldsymbol{necessarily} : TTV \to P \to BOOL$

$\forall t{:}TTV;\ p{:}P\bullet\ necessarily\ t\ p \Leftrightarrow \forall w{:}W\bullet\ pem(p,\ w) = t$

A proposition is *necessary* (simpliciter) if it is *necessarily t* for some truth value t.

HOL Constant

$\boldsymbol{necessary} : P \to BOOL$

$\forall p{:}P\bullet\ necessary\ p \Leftrightarrow \exists t\bullet\ necessarily\ t\ p$

HOL Constant

$\boldsymbol{contingent} : P \to BOOL$

$\forall p{:}P\bullet\ contingent\ p \Leftrightarrow \exists w1\ \ w2\bullet\ \neg\ pem(p,\ w1) = pem(p,\ w2)$

3.5 True in Virtue of Meaning

A common definition of "analytic" is as 'true in virtue of meaning', so we will now try to formalise that idea. If the truth value of a sentence can be ascertained from its meaning only, i.e. without taking into account any 'extra-linguistic fact' (in Quine's words), i.e. without knowing anything about what possible world is actual. This can only be known if it takes the same truth value in every possible world.

This can be generalised to an arbitrary truth value.

Therefore we define:

SML

```
declare_infix (300, "by_meaning");
```

HOL Constant

```
$by_meaning : TTV → (S × C) → BOOL
────────────────────────────────────
∀t:TTV; s:S; c:C• t by_meaning (s, c) ⇔ ∀p• pem (sm(s, c), p) = t
```

3.6 Expresses a Necessary Proposition

My preferred definition of analyticity is that a sentence is analytic if the proposition it expresses is necessarily true. Again we generalise to an arbitrary truth value. I'll make this infix as well.

SML

```
declare_infix (300, "analytic");
```

HOL Constant

```
$analytic : TTV → (S × C) → BOOL
────────────────────────────────────
∀t:TTV; s:S; c:C• t analytic (s, c) ⇔ necessarily t (sm(s, c))
```

Now we prove that these two conception of analyticity are the same.

The proof is trivial, exanding the relevant definitions yields a universally quantified identity equation (apart from the names of the bound variables). In the following proof script, the necessary rewriting is broken into two stages to show the identity.

SML

```
set_goal([], ⌜∀t s c• t analytic (s,c) ⇔ t by_meaning (s, c)⌝);
a (pure_rewrite_tac (map get_spec [⌜$analytic⌝, ⌜$by_meaning⌝, ⌜necessarily⌝]));
```

```
(* *** Goal "" *** *)

(* ?⊢ *)  ⌜∀ t s c• (∀ w• pem (sm (s, c), w) = t) = (∀ p• pem (sm (s, c), p) = t)⌝
```

SML

```
a (rewrite_tac[]);
val analyticity_lemma1 = save_pop_thm "analyticity_lemma1";
```

4 KANT'S DEFINITION OF ANALYTICITY

4.1 First Shot

Kant defined analyticity only for "subject predicate" sentences, and some have therefore supposed this to be less general than more recent formulations. However, asssuming only that the notion of analyticity is to be preserved by logical equivalence we can show that Kant's definition is equivalent to the preceding ones.

There is an awkwardness in generalising this notion to three truth values, so I will do it only for the one.

We will first define predicate inclusion.

SML

```
declare_infix (300, "contains");
```

HOL Constant

```
$contains : ('a → BOOL) → ('a → BOOL) → BOOL
──────────────────────────────────────────
∀P Q• P contains Q ⇔ ∀x• Q x ⇒ P x
```

P contains Q is a way of writing a subject predicate assertion in which the subject is Q and the predicate is P (this is Aristotelian terminology, we don't use predication in this way in modern logic).

Now we show that every judgement is equivalent to one in "subject predicate" form:

```
kantian_lemma =
      ⊢ ∀ SS• ∃ P Q• SS ⇔ P contains Q
```

SML

```
set_goal([], ⌜∀SS• ∃P Q• SS ⇔ P contains Q⌝);
a (strip_tac THEN ∃_tac ⌜λx• SS⌝ THEN ∃_tac ⌜λx• T⌝);
a (rewrite_tac [get_spec ⌜$contains⌝]);
val kantian_lemma = save_pop_thm "kantian_lemma";
```

This lemma may be applied generally, thus:

SML

```
val ℕ_gt_trans = ∀_elim ⌜∀x y z:ℕ• x > y ∧ y > z ⇒ x > z⌝ kantian_lemma;
```

yields:

```
val ℕ_gt_trans =
   ⊢ ∃ P Q• (∀ x y z• x > y ∧ y > z ⇒ x > z) ⇔ P contains Q
```

However, this is smoke and mirrors, because though it appears to be saying something about propositions, it is really about truth values. i.e. we have proven that every sentence has the same truth value as some sentence in subject predicate form, i.e. that there is a false sentence and a true sentence in subject predicate form.

To have a relevance to the scope of Kant's definition of analyticity we need some real metatheoretic reasoning in which we talk about meanings of judgements.

4.2 Fuller Treatment

The treatment in the previous section is not wholly convincing.

A sufficient reason for this is that the central thesis is not itself formalised. Insofar as there is any doubt about the thesis it therefore fails to improve the situation, and the question arises whether the theorems proven really establish the intended result.

The thesis is that, under certain provisos, Kant's definition of analyticity is equivalent to a definition along the lines of "true in virtue of meaning". However, it seems probable that the conditions under which Kant's definition holds good differ significantly from those in which the other definition is applicable. To formulate the thesis it is therefore necessary to establish some sufficient (and preferably necessary) conditions for both definitions to be applicable.

The required result is a general result covering a class of descriptive languages, and I think this will be better expressed if we abandon the previous methods of formalising in relation to some loosely specified but fixed (constant) language, and talk about languages using variables.

I will therefore start from scratch but replicate the same basic idea of what a descriptive language is.

Kant talks about "judgements", I will treat these as sentences in context.

4.3 Types

In this version type variables will be used where constants were previously used, as follows:

'S Sentences

'C Contexts

'P Propositions

'W Possible Worlds

4.4 The Semantics

The semantics of a 'descriptive' language comes in two parts. A semantic map which delivers propositions, the meanings of sentences in context, and a propositional evaluation map, which extracts truth conditions from a proposition.

The purpose of this document is not to consider the semantics of any particular language but to reason about semantics and concepts defined in terms of semantics.

The following type abbreviations give the type of the semantics of a descriptive language.

SML

```
(* Semantic Map *)
declare_type_abbrev ("SM", ["'C", "'P", "'S"], ⌜:'S × 'C → 'P⌝);

(* Propositional Evaluation Map *)
declare_type_abbrev ("PEM", ["'P", "'W"], ⌜:'P × 'W → BOOL⌝);

(* Language *)
declare_type_abbrev ("LAN", ["'C", "'P", "'S", "'W"], ⌜:('C, 'P, 'W)SM × ('P, 'W)PEM⌝);
```

4.5 Necessity

A proposition is 'necessary' if it takes truth value 'T' in every possible world, the definition is parameterised by a propositional evaluation map.

HOL Constant

```
necessary : ('P, 'W)PEM → 'P → BOOL
────────────────────────────
∀pem (p:'P)• necessary pem p ⇔ ∀w:'W• pem(p, w) = T
```

HOL Constant

```
contradictory : ('P, 'W)PEM → 'P → BOOL
────────────────────────────
∀pem (p:'P)• contradictory pem p ⇔ ∀w:'W• pem(p, w) = F
```

HOL Constant

```
contingent : ('P, 'W)PEM → 'P → BOOL
────────────────────────────
∀pem (p:'P)• contingent pem p ⇔ ∃w1 w2:'W• ¬ pem(p, w1) = pem(p, w2)
```

[to be completed!]

5 Postscript

A The Theory t001a

A.1 Parents

misc2

A.2 Constants

sm	$S \times C \to P$
pem	$(P \times W,\ BOOL)\ IX$
necessarily	$TTV \to P \to BOOL$
necessary	$P \to BOOL$
contingent	$P \to BOOL$
$***by_meaning***	$TTV \to S \times C \to BOOL$
$***analytic***	$TTV \to S \times C \to BOOL$
$***contains***	$('a \to BOOL,\ BOOL)\ BR$

A.3 Types

S
C
W
P

A.4 Fixity

Right Infix 300:

analytic ***by_meaning*** ***contains***

A.5 Definitions

pem

sm $\vdash T$

necessarily $\vdash \forall\ t\ p\bullet\ necessarily\ t\ p \Leftrightarrow (\forall\ w\bullet\ pem\ (p,\ w) = t)$

necessary $\vdash \forall\ p\bullet\ necessary\ p \Leftrightarrow (\exists\ t\bullet\ necessarily\ t\ p)$

contingent $\vdash \forall\ p$
$\bullet\ contingent\ p$
$\Leftrightarrow (\exists\ w1\ \ w2\bullet\ \neg\ pem\ (p,\ w1) = pem\ (p,\ w2))$

by_meaning $\vdash \forall\ t\ s\ c$
$\bullet\ t\ by\_meaning\ (s,\ c) \Leftrightarrow (\forall\ p\bullet\ pem\ (sm\ (s,\ c),\ p) = t)$

analytic $\vdash \forall\ t\ s\ c$
$\bullet\ t\ analytic\ (s,\ c) \Leftrightarrow necessarily\ t\ (sm\ (s,\ c))$

contains $\vdash \forall\ P\ Q\bullet\ P\ contains\ Q \Leftrightarrow (\forall\ x\bullet\ Q\ x \Rightarrow P\ x)$

A.6 Theorems

***analyticity_lemma*1**

$\vdash \forall\ t\ s\ c\bullet\ t\ analytic\ (s,\ c) \Leftrightarrow t\ by\_meaning\ (s,\ c)$

kantian_lemma

$\vdash \forall\ SS\bullet\ \exists\ P\ Q\bullet\ SS \Leftrightarrow P\ contains\ Q$

B The Theory t001b

B.1 Parents

misc2

B.2 Constants

necessary	$('P, 'W)\ PEM \to 'P \to BOOL$
contradictory	$('P, 'W)\ PEM \to 'P \to BOOL$
contingent	$('P, 'W)\ PEM \to 'P \to BOOL$

B.3 Type Abbreviations

$('C, 'P, 'S)$ ***SM***	$('C, 'P, 'S)\ SM$
$('P, 'W)$ ***PEM***	$('P, 'W)\ PEM$
$('C, 'P, 'S, 'W)$ ***LAN***	$('C, 'P, 'S, 'W)\ LAN$

B.4 Definitions

necessary

$\vdash \forall\ pem\ p\bullet\ necessary\ pem\ p \Leftrightarrow (\forall\ w\bullet\ pem\ (p,\ w) \Leftrightarrow T)$

contradictory

$\vdash \forall\ pem\ p\bullet\ contradictory\ pem\ p \Leftrightarrow (\forall\ w\bullet\ pem\ (p,\ w) \Leftrightarrow F)$

contingent

$\vdash \forall\ pem\ p$
$\bullet\ contingent\ pem\ p$
$\Leftrightarrow (\exists\ w1\ \ w2\bullet\ \neg\ (pem\ (p,\ w1) \Leftrightarrow pem\ (p,\ w2)))$

Bibliography

[1] Roger Bishop Jones. Analyses of Analysis: Part I - Introduction. *RBJones.com*, 2009-. http://www.rbjones.com/rbjpub/pp/doc/t029.pdf.

[2] Roger Bishop Jones. *Analyses of Analysis: Part II - Synthetic Analysis*. RBJones.com. 2009. http://www.rbjones.com/rbjpub/pp/doc/b003.pdf.

Index