

Aristotle's Logic and Metaphysics

Roger Bishop Jones

Abstract

Formalisation in higher order logic of parts of Aristotle's logic and metaphysics.

Created 2009/05/21

Last Change Date: 2012/01/23 21:40:02

Id: t028.doc,v 1.33 2012/01/23 21:40:02 rbj Exp

<http://www.rbjones.com/rbjpub/pp/doc/t028.pdf>

© Roger Bishop Jones; Licenced under Gnu LGPL

Contents

1	Prelude	3
2	Introduction	4
2.1	Preliminary Formalities	5
3	Metaphysics (I)	5
3.1	The Grice/Code/Speranza Formulae	6
3.2	References to Plato	6
3.3	Aristotelian References	6
3.4	Formal Principles	7
3.4.1	Categories	7
3.4.2	Predication	9
3.4.3	The Principles in HOL	10
3.5	Total Definitions	11
3.6	Partial Definitions	13
3.7	Ontological Theorems	14
3.8	Platonic Principles and Theorems	15
3.9	Some Comments on The Conjectures	16
4	The Organon	18
4.1	Models and Their Significance	18
4.2	Preliminaries	21
4.2.1	Generating the Syllogisms	21
4.2.2	The Square of Opposition	25
4.2.3	Are The Syllogisms Tautologous?	26
4.3	Interpretation in Set Theory	28
4.3.1	Generating The Propositions	29
4.3.2	Proving the Syllogisms	30
4.4	Propositional Interpretation	31
4.5	Naive Interpretation in Predicate Calculus	33
4.5.1	Semantics	33
4.5.2	Predication	33
4.5.3	The Laws of Immediate Inference	34
4.5.4	The Square of Opposition	35
4.5.5	The Syllogisms	35
4.5.6	Generating Syllogisms	36
4.6	Predicate Calculus Without Empty Terms	36
4.6.1	Semantics	37
4.6.2	Predication	37
4.6.3	Laws of Immediate Inference	38
4.6.4	The Valid Syllogisms	40
4.6.5	Proving the Syllogisms	40
4.7	Existential Import in Universals	40
4.7.1	Semantics	40
4.7.2	Predication	41
4.7.3	The Laws of Immediate Inference	42
4.7.4	The Square of Opposition	43
4.7.5	The Syllogisms	44
4.8	Existential Import in Affirmations	44
4.8.1	Semantics	44

4.8.2	Predication	45
4.8.3	The Laws of Immediate Inference	46
4.8.4	The Square of Opposition	47
4.8.5	The Syllogisms	47
4.9	Modal Syllogisms	48
4.9.1	Semantics	48
4.9.2	Predication	49
4.9.3	Laws of Immediate Inference	51
4.9.4	The Valid Modal Syllogisms	52
4.9.5	General Results	53
4.9.6	Proving the Syllogisms	54
4.10	Demonstrative Truth	55
5	Metaphysics (II)	56
5.1	Semantics	56
5.2	Predication	59
5.3	Propositional Operators	61
5.4	Quantification	63
5.5	Judgements	63
5.6	Conversions	64
5.7	Modal Conversions	65
5.8	Other Conversions	66
5.9	Syllogisms for Essential Predication	67
5.10	Some Accidental Syllogisms	67
5.11	Grice and Code	67
5.11.1	Common Material	67
6	Conclusions	69
7	Postscript	69
A	Theory Listings	71
A.1	The Theory ariscat	72
A.2	The Theory syllog1	75
A.3	The Theory syllog2	76
A.4	The Theory syllog3	77
A.5	The Theory syllog4	79
A.6	The Theory syllog5	81
A.7	The Theory syllog6	83
A.8	The Theory modsyllog	85
A.9	The Theory syllmetap	87
A.10	The Theory gcon	91
	Bibliography	93
	Index	95

1 Prelude

This document is intended ultimately to form a chapter of Analyses of Analysis [6].

Some of the material is not expected to be in that history including:

- the material up to and including the Prelude
- the Postscript and any material following it.
- possibly some other parts which have been marked for exclusion

My original purpose in preparing this document was to analyse certain semi-formal statements, relating to the philosophy of Aristotle, which were posted to the *hist-analytic* mailing list (see [message in archive](#)) originating primarily in joint work by Grice [4] and Code [3].

This has now been overtaken by various other philosophical motivations.

Of these the most important for me at present lie in the perceived relevance of Aristotle's metaphysics to what I am trying elsewhere to write about *Metaphysical Positivism*. One tentative idea in this exposition involves three comparisons intended to illuminate the tension between essentialism and nominalism and inform the search for a middle ground. These three are between Plato and Aristotle, between Hume and Kant, and between Carnap and one or more twentieth century metaphysicians.

For this purpose I seek some kind of understanding of Aristotle's essentialism, and it is for me natural to use formal modelling as one way of realising that understanding.

Since my own background in formal modelling comes from Computer Science and Information Systems Engineering, my own preferred languages, methods and tools, which I believe can be effectively applied to some kinds of philosophical problems, are probably alien to most if not all philosophers, and it is therefore a secondary purpose of this material to try to make this kind of modelling intelligible to some philosophers. This is not a presentation of established methods with proven philosophical benefits. It is an exploration and adaptation of methods from other domains to philosophy, and the benefits, are to be discovered, not merely displayed.

The present state of the document is rather rough and ready. Formal modelling takes time, but presenting such material takes longer, and while I am hot on the trail of better, more illuminating models, the presentation will not be polished and transparent.

Further discussion of what might become of this document in the future may be found in my postscript (Section 7).

In this document, phrases in coloured text are hyperlinks, like on a web page, which will usually get you to another part of this document (the blue parts, the contents list, page numbers in the Index) but sometimes take you (the red bits) somewhere altogether different (if you happen to be online) like [the hist-analytic archives](#).

For description of the formal languages, methods and tools used in or in producing this document see: [5].

2 Introduction

My purpose here is to use formal models to aid in understanding the philosophies of Plato and Aristotle, both in relation to their contribution generally to the areas of interest, philosophical logic, semantics and metaphysics, and also more specifically in relation to the extent to which these philosophers laid the ground for the distinction which was later expressed in Hume's fork.

In doing this I began with some enquiries into Aristotle's metaphysics published by Code [3] and produced from this a preliminary model (Section 3). In these an important defect is that the model does not support the u-p syllogisms on which Code's analysis depends more heavily than one might have expected, and also does not allow for modal operators, which not only enter into Code's material

but are also important for the kinds of comparison with later philosophers which I had hoped to undertake. I have also failed at this stage to bring out the distinction between Plato and Aristotle. Perhaps more important is that I did not arrive at a good understanding of Code and the model is therefore unlikely to fully reflect his intentions. It is also the case that the method I adopted for the analysis of Code is one which he would have been likely to question. His paper does briefly discuss unfavourably the interpretation of Aristotle in terms of modern idiom such as that of set theory. It is an important part of my objectives in this document to discuss this kind of formal exegesis, and hopefully to explain contra Code why the use of modern set theoretic language is appropriate and helpful in the analysis of materials which could not have been originally conceived in such terms.

I then went back from the *Metaphysics* to the *Organon* and used formal models to come to a better understanding of Aristotle's formal syllogistic logic (Section 4). In this seven models of increasing sophistication were produced and formed the basis for undertaking a further model of the *metaphysics* which incorporated the u-p syllogisms and modal operators (Section 5). All of this is preliminary to addressing the real issues, which has not yet seriously begun.

2.1 Preliminary Formalities

In the document several different formal models are presented. By and large they are independent, but a some features are common and are therefore presented here for use in all the models.

SML

```
|open_theory "misc2";
|force_new_theory "aristotle";
```

We define inequality:

SML

```
|declare_infix (300, "≠");
```

HOL Constant

```
|$≠ : 'a → 'a → BOOL
```

```
|∀x y • x ≠ y ⇔ ¬ x = y
```

3 Metaphysics (I)

In this section we consider some material on Aristotle's *Metaphysics* [2] which originated in work of Grice and Code [3] and came to me from a [posting of J.L. Speranza](#) on the hist-analytic mailing list. Code's paper is also partially available at [Google Books](#).

What Speranza posted was the list of formulae which are named below as c01 through c31 (though not exactly as given, I have massaged them to be acceptable to HOL and also have quantified over all free variables).

The analysis in this section is independent of the preceding analysis of Aristotle's syllogism, and considers predication from a rather different point of view, which hangs around the distinction between essential and accidental predication. In the next section I will produce another model in which essence and accident are combined with a full treatment of modal syllogism so that some conclusions might be drawn about the relationship between essence and necessity in Aristotelean philosophy.

A new theory is needed which I will call “ariscat” which is created here:

SML

```
|open_theory "aristotle";  
|force_new_theory "ariscat";
```

3.1 The Grice/Code/Speranza Formulae

This work began with an attempt to analyse using ProofPower a set of formulae posted by J.L. Speranza to the [hist-analytic mailing list](#). These were a Speranzan transcript of formulae published in a paper by Code [3], the work presented in that paper having begun with some joint work with H.P. Grice [4].

The following material labelled “Code/Speranza” began as a transcription from Speranza’s email, and was later updated when Speranza pointed me to the partial availability of the Code paper on Google Books. I then put back some of the detail missed in the Speranza version, enclosed in square brackets, and enclosed in curly braces some of the material which Speranza had added to Code’s.

The terminology used is Grice’s. Code uses “Is” and “Has” instead of “izz” and “hazz” (which were coined by Grice and used by Speranza). Aristotle’s originals have been translated as “SAID OF” and “IN”, according to Cohen ([Grice and Code on IZZing and HAZZing](#)).

The material is interspersed with a formalisation in ProofPower-HOL. I have adopted some of Code’s headings for sections.

3.2 References to Plato

By footnote in Code.

30 *Phaedo*. 102B8C – 1 with C10ff (cf. B 4-6)

3.3 Aristotelian References

These are gathered together temporarily and will later be distributed as footnotes.

Cat. 1^a, 12–15 [Equivocally, Univocally and Derivatively](#).

Cat. 1^b5, **3b12**

Cat. 2^a, 19–34 [Presence and Predication](#).

Cat. 3^a, 15–20 [Subclasses and Predicability](#).

Cat. 10^a, **27ff.** [Opposite, Contrary, Privative, Positive](#).

De. Int. **7**, 17^a9–40 [Universal and Individual Subjects](#).

Post. An. A4, 73^a**34**–^b5 [What are the premisses of demonstration](#). Distinguishing essential and accidental predication.

Metap. I1, 1059^a10 – 14 [The One](#).

Metap. $\Delta 9$, 1018^a1 – 4 **Same and Different**. ‘Socrates’ and ‘musical Socrates’ are thought to be the same; but ‘Socrates’ is not predicable of more than one subject, and therefore we do not say ‘every Socrates’ as we say ‘every man’.

Metap. $\Delta 18$, 1022^a25 – 27 **In Virtue Of**. Admissibility of self-predication of particulars.

Metap. $\Delta 23$, 1023^a11 – 13 **Having and Being**. Two kinds of predication.

Metap. *Z5* **Only substance is definable**.

Metap. *Z6*, 1032^a4 – 6 **Each thing and its essence are one and the same**.

Metap. *Z8*, 1034^a7 **different in virtue of their matter**

Metap. *Z11*, 1037^a33 –^b 4 **Formulae, Parts, Substance and Essence**.

Topics. *H(VII)1*, 152^b25 – 29 **Numerical identity**.

3.4 Formal Principles

We begin with “Formal Principles” which we take as implicit definitions of two kinds of Aristotelian predication.

Code/Speranza

|[(A) *Formal Principles*]

|[FP1] 1. $A \text{ izz } A$.

|[FP2] 2. $(A \text{ izz } B \ \& \ B \text{ izz } C) \text{ --> } A \text{ izz } C$.

|[FP3] 3. $A \text{ hazz } B \text{ --> } \neg(A \text{ izz } B)$.

|[FP4] 4. $A \text{ hazz } B \text{ iff } A \text{ hazz } \textit{Some-Thing}$ [something] *that izz B*.

The modelling in ProofPower-HOL will be entirely conservative, so we provide explicit definitions for “izz” and “hazz” and prove that they satisfy these principles and suffice also for the definitions and theorems which follow.

If the formal development is complete the definitions will have been shown to be sufficient. In order to test whether the principles suffice I will attempt to proceed on these alone, though I suspect that will not be possible. Failing that I will offer informal arguments to the contrary (a formal argument would require ascent to a metatheory which would involve too much work).

In order to define these concepts we have to decide what they are about, and this is not straightforward.

3.4.1 Categories

Aristotle has a system of categories, and these seem central to the topic. Much hangs on what these are, and to get a nice structure to our theory it seems advisable to do a bit of “category theory” first. Of course this is not at all the same thing as the branch of mathematics which now goes by that name, but the choice of name for the mathematics was not entirely quixotic and at some point it might be interesting to think about the relationship between the two kinds of category theory.

Among these categories that of substances plays a special role. Substances can be particular in which case they correspond to some individual, or not, in which case they are sets of individuals. The particulars of the other categories are attributes, and the non-individuals are sets of attributes. I

don't think you can have singleton sets, so we can model all these categories as sets of sets in which the singleton sets are the individuals. Attributes can also be considered as sets of individual substances and so there is a type difference between the category of substances and the other categories.

The following introduce new types and type abbreviations for modelling Aristotle's categories.

ACAT is a type of attribute categories

ISUB is a type of individual substances

CATM is the type of the things which are in categories.

This is a 'disjoint union', which means that there are two kinds of thing which one finds in categories, either a set of individual substances (using singleton sets to represent individual substances), or a set of properties of individual substances tagged with an attribute category.

CAT is a type abbreviation for a notion of category which is either an attribute category or some other category (which will stand for the category of substances).

There are some oversimplifications here which I am hoping will not be too serious for a useful first cut.

Modal Operators The main one is that this general approach will not permit the definition of modal operators, which are used by Code. Whether there is real need for them seems to me doubtful, but if necessary a modal model could be provided.

Empty Sets The second is that I have not excluded empty sets, and hence that there can be predicates with null extensions. This is in fact consistent with Code's "principles", but we find that his definitions are written as if there were no empty predicates, even though this is not entailed by the principles. Where Code's definitions presume non-emptiness of predicates I have chosen another definition which does not. Some of the theorems are then unprovable.

Extensionality Intracategorical equality will be extensional. That appears to be what is required, so it probably isn't a problem!

Predicability Aristotle defines particulars in terms of Predicability, they are the impredicables. Its not clear how to deal with this, and it appears to be in terminological conflict with Code, who appears to use "predicable" to mean "truly predicable". Code has the general principle "A izz A" which implicitly asserts that everything is truly predicable of itself.

SML

```
new_type("ACAT", 0);
new_type("ISUB", 0);
declare_type_abbrev ("CATM", [],  $\lceil$ :ISUB  $\mathbb{P}$  + (ACAT  $\times$  (ISUB  $\rightarrow$  BOOL) $\mathbb{P}$ ) $\lceil$ );
declare_type_abbrev ("CAT", [],  $\lceil$ :ONE + ACAT $\lceil$ );
```

We name the category of substances.

HOL Constant

```
CatSubs : CAT
-----
CatSubs = InL One
```


Now we define various operators over categories and their constituents which suffice for the development of an appropriate theory, in the context of which rest of the Aristotelian terminology will we hope prove definable.

First “projection” functions which yield the constituents of *MCATs*.

HOL Constant

$$\mathbf{Cat} : \mathit{CATM} \rightarrow \mathit{CAT}$$

$$\forall x \bullet \mathit{Cat} \ x = \mathit{if} \ \mathit{IsL} \ x \ \mathit{then} \ \mathit{CatSubs} \ \mathit{else} \ \mathit{InR}(\mathit{Fst} \ (\mathit{OutR} \ x))$$

HOL Constant

$$\mathbf{IndvSet} : \mathit{CATM} \rightarrow \mathit{ISUB} \ \mathbb{P}$$

$$\forall x \bullet \mathit{IndvSet} \ x = \mathit{OutL} \ x$$

HOL Constant

$$\mathbf{AttrSet} : \mathit{CATM} \rightarrow (\mathit{ISUB} \rightarrow \mathit{BOOL})\mathbb{P}$$

$$\forall x \bullet \mathit{AttrSet} \ x = \mathit{Snd}(\mathit{OutR} \ x)$$

This one turns out handy.

HOL Constant

$$\mathbf{CatSet} : \mathit{CATM} \rightarrow (\mathit{ISUB} + (\mathit{ISUB} \rightarrow \mathit{BOOL}))\mathbb{P}$$

$$\begin{aligned} \forall x \bullet \mathit{CatSet} \ x = \\ & \mathit{if} \ \mathit{Cat} \ x = \mathit{CatSubs} \\ & \mathit{then} \ \{y \mid \exists z \bullet z \in \mathit{IndvSet} \ x \wedge y = \mathit{InL} \ z\} \\ & \mathit{else} \ \{y \mid \exists z \bullet z \in \mathit{AttrSet} \ x \wedge y = \mathit{InR} \ z\} \end{aligned}$$

With these definitions in place we get a useful characterisation of identity for elements of *CATM*.

$$\mathit{catm_eq_lemma} =$$

$$\vdash \forall A \ B \bullet A = B \Leftrightarrow \mathit{Cat} \ A = \mathit{Cat} \ B \wedge \mathit{CatSet} \ A = \mathit{CatSet} \ B$$

3.4.2 Predication

Now we can define predication. We do this in terms of Grice’s *izz* and *hazz*.

SML

$$\begin{aligned} & \mathit{declare_infix} \ (300, \ \mathit{"izz"}); \\ & \mathit{declare_infix} \ (300, \ \mathit{"hazz"}); \end{aligned}$$

HOL Constant

$$\mathbf{\$izz} : \mathit{CATM} \rightarrow \mathit{CATM} \rightarrow \mathit{BOOL}$$

$$\forall A \ B \bullet A \ \mathit{izz} \ B \Leftrightarrow \mathit{Cat} \ A = \mathit{Cat} \ B \wedge \mathit{CatSet} \ A \subseteq \mathit{CatSet} \ B$$

HOL Constant

```
$hazz : CATM → CATM → BOOL
```

```
∀A B● A hazz B ⇔ Cat A = CatSubs ∧ ¬ Cat B = CatSubs  
  ∧ ∃a● a ∈ AttrSet B ∧ ∀s● s ∈ IndvSet A ⇒ a s
```

That was reasonably neat, but the definition of *izz* isn't terribly convenient for proving things. Lets have some *izz* lemmas:

```
izz_lemma1 =  
  ⊢ ∀ A B● Cat A = CatSubs ⇒ (A izz B ⇔ Cat B = CatSubs ∧ IndvSet A ⊆ IndvSet B)  
  
izz_lemma2 =  
  ⊢ ∀ A B● Cat B = CatSubs ⇒ (A izz B ⇔ Cat A = CatSubs ∧ IndvSet A ⊆ IndvSet B)  
  
izz_lemma3 =  
  ⊢ ∀ A B● ¬ Cat A = CatSubs ⇒ (A izz B ⇔ Cat B = Cat A ∧ AttrSet A ⊆ AttrSet B)  
  
izz_lemma4 =  
  ⊢ ∀ A B● ¬ Cat B = CatSubs ⇒ (A izz B ⇔ Cat B = Cat A ∧ AttrSet A ⊆ AttrSet B)
```

3.4.3 The Principles in HOL

Here are the HOL versions of the Code “Principles”.

The following is a bit of program in a programming language called SML, which stands for “Standard Meta Language”! It names various terms in HOL, the name on the left ‘c01’ (short for ‘conjecture 1’), the term on the right quoted in “Quine corners”.¹

SML

```
val c01 = ⌈∀A● A izz A⌋;  
val c02 = ⌈∀A B C● A izz B ∧ B izz C ⇒ A izz C⌋;  
val c03 = ⌈∀A B● A hazz B ⇒ ¬ A izz B⌋;  
val c04 = ⌈∀A B● A hazz B ⇔ ∃C● A hazz C ∧ C izz B⌋;
```

One would expect this set of principles to be sufficient to characterise *izz* and *hazz* (i.e. sufficient to derive any other true facts about them) but this seems doubtful.

Here are some supplementary conjectures.

SML

```
val c01b = ⌈∀A● ¬ A hazz A⌋;  
val c03b = ⌈∀A B● A izz B ⇒ ¬ A hazz B⌋;  
val c03c = ⌈∀A B● A hazz B ⇒ ¬ A = B⌋;  
val c04a = ⌈∀A B C● A hazz B ∧ B izz C ⇒ A hazz C⌋;  
val c04b = ⌈∀A B C● A izz B ∧ B hazz C ⇒ A hazz C⌋;
```

¹“Quine corners” are a notation originally used by Quine for Godel numbers, i.e., in Quine’s use ‘⌈43⌋’ is a friendly way of writing down the Godel number of the numeral ‘43’. In ProofPower HOL these corners are used to refer to HOL terms in the metalanguage SML. In HOL, a formula is a term of type $\lceil : \text{BOOL} \rceil$ (the opening \lceil : is used when quoting a type rather than a term).

Of the supplementaries:

- c01b is derivable from c03 and c01.
- c03b is the contrapositive of c03.
- c03c would be derivable for Code once he has defined equality, it is provable for us now because we have a primitive equality.
- c04a is a preferable formulation of the right-left implication in c04, and we have used it to prove c04.
- c04b is an obvious further transitivity-like property, which does not look like it's provable from the stipulated principles.

Proven Theorems

	$l01$	$= \vdash \forall A \bullet A \text{ izz } A$
	$l01b$	$= \vdash \forall A \bullet \neg A \text{ hazz } A$
	$l02$	$= \vdash \forall A B C \bullet A \text{ izz } B \wedge B \text{ izz } C \Rightarrow A \text{ izz } C$
	$l03$	$= \vdash \forall A B \bullet A \text{ hazz } B \Rightarrow \neg A \text{ izz } B$
	$l03b$	$= \vdash \forall A B \bullet A \text{ izz } B \Rightarrow \neg A \text{ hazz } B$
	$l03c$	$= \vdash \forall A B \bullet A \text{ hazz } B \Rightarrow \neg A = B$
	$l04a$	$= \vdash \forall A B C \bullet A \text{ hazz } B \wedge B \text{ izz } C \Rightarrow A \text{ hazz } C$
	$l04$	$= \vdash \forall A B \bullet A \text{ hazz } B \Leftrightarrow (\exists C \bullet A \text{ hazz } C \wedge C \text{ izz } B)$
	$l04b$	$= \vdash \forall A B C \bullet A \text{ izz } B \wedge B \text{ hazz } C \Rightarrow A \text{ hazz } C$

3.5 Total Definitions

Code/Speranza

|[(B) Total Definitions]

|{6. ($A \text{ hazz } B$ & A is a particular) \rightarrow there is a C such that ($C \neq A$) & ($A \text{ izz } B$).}

|[D1] 7. A is predicable of B iff ($(B \text{ izz } A) \vee (B \text{ hazz } \text{Something that izz } A)$).

|[D2] 8. A is essentially predicable [L -predicable] of B iff $B \text{ izz } A$.

|[D3] 9. A is accidentally predicable [H -predicable] of B iff $B \text{ hazz something that izz } A$.

|[D4] 10. $A = B$ iff $A \text{ izz } B$ & $B \text{ izz } A$.

|[D5] 11. A is an individual iff (Nec)(For all B) $B \text{ izz } A \rightarrow A \text{ izz } B$

|[D6] 12. A is a particular iff (Nec)(For all B) A is predicable of $B \rightarrow (A \text{ izz } B \wedge B \text{ izz } A)$

|[D7] 13. A is a universal iff

(Poss) (There is a B) A is predicable of $A[B]$ & $\neg(A \text{ izz } B \wedge B \text{ izz } A)$

There is a certain amount of duplication of terminology here, since essential and accidental predication seem to be just *izz* and *hazz* backwards. I'm not so happy with the "ables" here, for what is clearly meant is "truly predicable", which is not quite the same thing. Better names would be simply "is_essentially" and "is_accidentally", lacking the ambiguity of "able" (but then they would have to be the other way round, exactly the same as *izz* and *hazz*).

Anyway here are the definitions (keeping the names (more or less) as they were for the present):

SML

```
declare_infix (300, "predicable_of");  
declare_infix (300, "essentially_predicable_of");  
declare_infix (300, "accidentally_predicable_of");
```

HOL Constant

```
$essentially_predicable_of : CATM → CATM → BOOL  
-----  
∀A B • A essentially_predicable_of B ⇔ B izz A
```

HOL Constant

```
$accidentally_predicable_of : CATM → CATM → BOOL  
-----  
∀A B • A accidentally_predicable_of B ⇔ B hazz A
```

Aristotelian predication is then:

HOL Constant

```
$predicable_of : CATM → CATM → BOOL  
-----  
∀A B • A predicable_of B ⇔ A essentially_predicable_of B ∨ A accidentally_predicable_of B
```

Because we have not precluded empty predicates Code's definition will not do, and we have to make "individual" primitive, insisting on an individual being a singleton.

HOL Constant

```
individual : CATM → BOOL  
-----  
∀A • individual A ⇔ ∃a • CatSet A = {a}
```

According to Code's definition a particular is a substantial individual, we also have to use a more direct statement of that principle.

HOL Constant

```
particular : CATM → BOOL  
-----  
∀A • particular A ⇔ individual A ∧ Cat A = CatSubs
```

Again we have a problem with Code's definition and therefore define a universal as a non-particular.

HOL Constant

```
universal : CATM → BOOL  
-----  
∀A • universal A ⇔ ¬ particular A
```

SML

```
val c06 =  $\ulcorner \forall A B \bullet A \text{ hazz } B \wedge \text{particular } A \Rightarrow \exists C \bullet C \neq A \wedge A \text{ izz } B \urcorner$ ;
val c06n =  $\ulcorner \neg \forall A B \bullet A \text{ hazz } B \wedge \text{particular } A \Rightarrow \exists C \bullet C \neq A \wedge A \text{ izz } B \urcorner$ ;
val c07 =  $\ulcorner \forall A B \bullet A \text{ predicable\_of } B \Leftrightarrow (B \text{ izz } A) \vee \exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A \urcorner$ ;
val c08 =  $\ulcorner \forall A \bullet A \text{ essentially\_predicable\_of } B \Leftrightarrow B \text{ izz } A \urcorner$ ;
val c09 =  $\ulcorner \forall A \bullet A \text{ accidentally\_predicable\_of } B \Leftrightarrow \exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A \urcorner$ ;
val c10 =  $\ulcorner \forall A B \bullet A = B \Leftrightarrow A \text{ izz } B \wedge B \text{ izz } A \urcorner$ ;
val c11 =  $\ulcorner \forall A B \bullet \text{individual } A \Leftrightarrow \Box(\forall B \bullet B \text{ izz } A \Rightarrow A \text{ izz } B) \urcorner$ ;
val c12 =  $\ulcorner \forall A \bullet \text{particular } A \Leftrightarrow \Box(\forall B \bullet A \text{ predicable\_of } B \Rightarrow A \text{ izz } B \wedge B \text{ izz } A) \urcorner$ ;
val c13 =  $\ulcorner \forall A \bullet \text{universal } A \Leftrightarrow \Diamond(\exists B \bullet (A \text{ predicable\_of } B \wedge \neg(A \text{ izz } B \wedge B \text{ izz } A))) \urcorner$ ;
```

```
l05 =  $\vdash \forall x \bullet \text{form } x \Rightarrow \text{universal } x$ 
l06n =  $\vdash \neg (\forall A B \bullet A \text{ hazz } B \wedge \text{particular } A \Rightarrow (\exists C \bullet C \neq A \wedge A \text{ izz } B))$ 
l07 =  $\vdash \forall A B \bullet A \text{ predicable\_of } B \Leftrightarrow B \text{ izz } A \vee (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$ 
l08 =  $\vdash \forall A \bullet A \text{ essentially\_predicable\_of } B \Leftrightarrow B \text{ izz } A$ 
l09 =  $\vdash \forall A \bullet A \text{ accidentally\_predicable\_of } B \Leftrightarrow (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$ 
l10 =  $\vdash \forall A B \bullet A = B \Leftrightarrow A \text{ izz } B \wedge B \text{ izz } A$ 
```

c06 is false (see l06n), probably a typo. However, I couldn't work out what was intended.

c11-13 are not provable in our model because of the existence of empty predicates (and the lack of modal operators).

3.6 Partial Definitions

Code/Speranza

```
[(C) Partial Definitions]
```

```
[D8] 14. If A is Some Thing [a this somewhat], A is an individual.
```

```
[D9] 15. If A is a [(seperable) Platonic] Form,
```

```
    A is Some Thing [a this somewhat] and Universal.
```

This is D8/c14.

HOL Constant

```
SomeThing : CATM → BOOL
```

```
 $\forall x \bullet \text{SomeThing } x \Rightarrow \text{individual } x$ 
```

A form is a non-substantial individual.

This is D9/c15

HOL Constant

```
form : CATM → BOOL
```

```
 $\forall x \bullet \text{form } x \Rightarrow \text{SomeThing } x \wedge \text{universal } x$ 
```

3.7 Ontological Theorems

Code/Speranza

[[*(D) Ontological Theorems*]]

- [T1] 16. *A is predicable of B iff (B izz A) v (B hazz Some Thing that Izz A).*
- [T2] 17. *A is essentially predicable [L-predicable] of A.*
- [T3] 18. *A is accidentally predicable [H-predicable] of B \rightarrow A \neq B*
- [T4] *A is not accidentally predicable [H-predicable] of A*
- {19. *– (A is accidentally predicable of B) \rightarrow A \neq B.*}
- [T5] 20. *A is a particular \rightarrow A is an individual.*
 [Note that the converse of T5 is not a theorem]
- [T6] 21. *A is a particular \rightarrow No Thing [nothing] that is Not Identical with A izz A.*
- [T7] 22. *No Thing is both particular & a [(seperable) Platonic] Form.*
- [T8] 23. *A is a (seperable Platonic) Form \rightarrow nothing that is not identical with A izz A.*
- [T9] 24. *A is a particular \rightarrow there is no (seperable Platonic) form B such that A izz B.*
- [T10] 25. *A is a (seperable Platonic) form*
 $\rightarrow ((A \text{ is predicable of } B \ \& \ A \neq B) \rightarrow B \text{ hazz } A)$
- [T11] 26. *(A is a (seperable Platonic) form & B is a particular)*
 $\rightarrow (A \text{ is predicable of } B \text{ iff } B \text{ hazz } A).$

SML

```
val c05 =  $\ulcorner \forall x \bullet \text{universal } x \Rightarrow \text{form } x \urcorner$ ;
val c05b =  $\ulcorner \forall x \bullet \text{form } x \Rightarrow \text{universal } x \urcorner$ ;
val c16 =  $\ulcorner \forall A B \bullet A \text{ predicable\_of } B \Leftrightarrow (B \text{ izz } A) \vee \exists C \bullet (B \text{ hazz } C \wedge C \text{ izz } A) \urcorner$ ;
val c17 =  $\ulcorner \forall A \bullet A \text{ essentially\_predicable\_of } A \urcorner$ ;
val c18 =  $\ulcorner \forall A \bullet A \text{ accidentally\_predicable\_of } B \Rightarrow A \neq B \urcorner$ ;
val c19 =  $\ulcorner \forall A \bullet \neg A \text{ accidentally\_predicable\_of } A \urcorner$ ;
val c20 =  $\ulcorner \forall A \bullet \text{particular } A \Rightarrow \text{individual } A \urcorner$ ;
val c21 =  $\ulcorner \forall A \bullet \text{particular } A \Rightarrow \neg \exists C \bullet C \neq A \wedge C \text{ izz } A \urcorner$ ;
val c22 =  $\ulcorner \neg \exists A \bullet \text{particular } A \wedge \text{form } A \urcorner$ ;
val c23 =  $\ulcorner \forall A \bullet \text{form } A \Rightarrow \neg \exists C \bullet C \neq A \wedge C \text{ izz } A \urcorner$ ;
val c23b =  $\ulcorner \forall A \bullet \text{form } A \Rightarrow \text{individual } A \urcorner$ ;
val c24a =  $\ulcorner \forall A B \bullet \text{particular } A \wedge \text{individual } B \wedge A \text{ izz } B \Rightarrow \text{particular } B \urcorner$ ;
val c24 =  $\ulcorner \forall A \bullet \text{particular } A \Rightarrow \neg \exists B \bullet \text{form } B \wedge A \text{ izz } B \urcorner$ ;
val c24b =  $\ulcorner \forall A \bullet \text{particular } A \Rightarrow \neg \text{form } A \urcorner$ ;
val c25 =  $\ulcorner \forall A B \bullet \text{form } A \Rightarrow A \text{ predicable\_of } B \wedge A \neq B \Rightarrow B \text{ hazz } A \urcorner$ ;
val c26 =  $\ulcorner \forall A B \bullet \text{form } A \wedge \text{particular } B \Rightarrow (A \text{ predicable\_of } B \Leftrightarrow B \text{ hazz } A) \urcorner$ ;
```

These are the ones I have proved.

```
l16 =  $\vdash \forall A B \bullet A \text{ predicable\_of } B \Leftrightarrow B \text{ izz } A \vee (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$ 
l17 =  $\vdash \forall A \bullet A \text{ essentially\_predicable\_of } A$ 
l19 =  $\vdash \forall A \bullet \neg A \text{ accidentally\_predicable\_of } A$ 
l20 =  $\vdash \forall A \bullet \text{individual } A \Rightarrow \text{particular } A$ 
l22 =  $\vdash \neg (\exists A \bullet \text{particular } A \wedge \text{form } A)$ 
```

$|l23b = \vdash \forall A \bullet \text{form } A \Rightarrow \text{individual } A$
 $|l24a = \vdash \forall A B \bullet \text{particular } A \wedge \text{individual } B \wedge A \text{ izz } B \Rightarrow \text{particular } B$
 $|l24 = \vdash \forall A \bullet \text{particular } A \Rightarrow \neg (\exists B \bullet \text{form } B \wedge A \text{ izz } B)$
 $|l26 = \vdash \forall A B \bullet \text{form } A \wedge \text{particular } B \Rightarrow (A \text{ predicable_of } B \Leftrightarrow B \text{ hazz } A)$

T6/c21, T8/c23, T10/c25 are all unprovable because of the existence of empty predicates.

3.8 Platonic Principles and Theorems

This section is a bit of a mess. I now see that the reason for this is that Code is now presenting a different theory here, which is Aristotle's conception of Plato's metaphysics. This explains why these principles at least augment (and possibly contradict) concepts which have already been defined. In our method, which involves, for the sake of ensuring consistency, the use of only conservative extensions, this cannot be done simply by adding new principles. We have to develop two systems in separate theories in which the differences of conception between Aristotle and Aristotle's conception of Plato are investigated in distinct contexts (though we could place in a single parent theory the elements which are common to both).

This will be considered later.

Code/Speranza

$|[(E) \text{ Platonic Principle}]$
 $|$
 $|[PP1] \text{ 5. Each universal is a (seperable Platonic) form.}$
 $|[PP2] \text{ 27. (A is particular \& B is a universal \& predicable of A)}$
 $| \quad \rightarrow \text{there is a C such that (A \neq C \& C is essentially predicable of A)}$

SML

$|val \text{ c05} = \ulcorner \forall x \bullet \text{universal } x \Rightarrow \text{form } x \urcorner;$
 $|val \text{ c05b} = \ulcorner \forall x \bullet \text{form } x \Rightarrow \text{universal } x \urcorner;$
 $|val \text{ c27} = \ulcorner \forall A B \bullet \text{particular } A \wedge \text{universal } B \wedge B \text{ predicable_of } A$
 $| \quad \Rightarrow \exists C \bullet (A \neq C \wedge C \text{ essentially_predicable_of } A) \urcorner;$

c05 is not provable, its converse c05b is. c27 is not provable, since it would require that there be more than one particular and we have no reason to believe that to be the case.

$|l05b = \vdash \forall x \bullet \text{form } x \Rightarrow \text{universal } x$

Code/Speranza

$|[(F) \text{ Platonic Theorem}]$
 $|$
 $| \{28. \text{ If there are particulars, of which universals are predicable,}$
 $| \quad \text{not every universal is Some Thing.}\}$
 $|[PT1] \text{ 29. Each universal is Some Thing [a this somewhat].}$
 $|[PT2] \text{ 30. If A is a particular, there is no B such that}$
 $| \quad (A \neq B \& B \text{ is essentially predicable of A}).$
 $|[PT3] \text{ 31. (A is predicable of B \& A \neq B) \rightarrow A is accidentally predicable of B.}$

SML

```
val c28 =  $\ulcorner (\exists P \bullet \text{particular } P \wedge \exists U \bullet \text{universal } U \wedge U \text{ predicable\_of } P) \Rightarrow \neg (\forall U \bullet \text{universal } U \Rightarrow \text{thing } U) \urcorner$ ;
val c29 =  $\ulcorner \forall U \bullet \text{universal } U \Rightarrow \text{thing } U \urcorner$ ;
val c30 =  $\ulcorner \forall A \bullet \text{particular } A \Rightarrow \neg \exists B \bullet (A \neq B \wedge B \text{ essentially\_predicable\_of } A) \urcorner$ ;
val c31 =  $\ulcorner \forall A B \bullet A \text{ predicable\_of } B \wedge A \neq B \Rightarrow A \text{ accidentally\_predicable\_of } B \urcorner$ ;
```

These are the ones I have proved.

```
l06n =  $\vdash \neg (\forall A B \bullet A \text{ hazz } B \wedge \text{particular } A \Rightarrow (\exists C \bullet C \neq A \wedge A \text{ izz } B))$ 
```

3.9 Some Comments on The Conjectures

The main problem with the conjectures is that as a group they are inconsistent. Consequently, one cannot find definitions which are consistent with all the conjectures.

So first I will expose some of the most obvious contradictions which flow from the conjectures.

1. From 5 and 15 we conclude that form and universal are coextensive. I think it may be that this is part of the Platonic view but not of the Aristotelian one.

Here are some observations on specific conjectures (now out of date).

c01 Note here that this is quantified over everything, and hence over individuals, whereas Aristotle describes individuals as thing of which one may predicate, but which are not themselves predicable. Perhaps this inderdiction applies to *hazz* but not to *izz*, to accidental but not essential predication. (We should add the rule $\ulcorner \neg A \text{ hazz } A \urcorner$ which is easy to prove.) I don't know any more detail about Aristotle's attitude towards predication by individuals. If one cannot, where do we stand when we do, as in "Socrates is Socrates" and "Socrates is Aristotle". Anyway, if these were to have a truth value (which surely they do) then the truth value will be as in this rule.

c02 Behind the scenes this is transitivity of set inclusion.

c03 This is because *izz* is intracategorical and *hazz* is intercategorical. An obvious but useful corrolary is that they are not equal (c03b).

c04 This, and most of the other theorems involving existential quantification, is rather odd. Its proof depends on the conjecture *c04a*, which we have proven, and which involves no existential quantification, but its content is not significantly greater than that rule. From right to left *l04* is *l04b* (you just pull out the existential and it turns into a universal). From left to right *l04* is trivial, since *B* serves as a witness for the existential.

Ideally we would be working with claims which are expressible syllogistically, i.e. without benefit of quantifiers. We can make an exception for universals on the left, since these are interconvertible with the conjecture with free variables instead which we can think of as schemata. Where an existential quantifier appears in a negative context it will turn universal if pulled out to the top level and can therefore be dispensed with. Elsewhere its worth asking whether the content is significant (including, as here, one half of the content implicit in putting an existential under an equivalence).

co4b is an obvious similar result to *c04a*.

- c05** According to my definition this is the wrong way round.
- c06** As it stands this is provably false since we have $\lceil AhazzB \rceil$ on the left, which entails that A and B are not of the same category, and $\lceil AizzB \rceil$ in the right, which entails that they are of the same category.
- c07** This, at the expense of using an existential, nevertheless tells us nothing that is not immediate from the definitions.
- c08** Is just our definition.
- c09** This turns into $c04$ (apart from the variable names) once you expand the definition of *accidentally_predicable_of*.
- c10** Behind the scenes, *izz* is set inclusion, so this is obvious.
- c11-13** These contain modal operators which cannot be defined using this model.
- c14-c15** I've not worked out what a "thing" is. Not even sure that I should have rendered Speranza's version using that term. However, I do think I know that *particular* and *universal* are opposites (contradictories), and hence I could conclude from these two conjectures that there can be no *forms* since they entail that a form is both individual and universal. Sounds like I have the wrong end of some stick or other.
- c16** I don't know enough about "thing"s to prove this one.
- c17** This is $c01$ in other words.
- c18** This is $c03b$ in other words.
- c19** This is the contrapositive of the claim that *accidentally_predicable_of* is reflexive, which is false. Would be true for *essentially_predicable_of*, but we already have that stated directly as $c17$.
- c20** Immediate from my definitions.
- c21** This turns out to be false under my definitions, because I have not excluded the possibility of an empty predicate. However, one wonders why this should be excluded.
If I go over to a model adequate for modal operators then it will be easy to exclude this possibility.
- c21** Fails for same reason as $c20$, though I could fix this by making the definition of *universal* insist on more than one member.
- c23** I don't know why this should be true. Any particular which partakes of a form contradicts it.
- c24** This is not true in the present model, because we might have only one particular, and hence no non-trivial forms.
- c25-c26** These two tell me that forms are not substance, but attributes, which contradicts $c24$ which tells us that there are substantial forms (if particulars are substances).
- c27** I don't see why this should be true.
- c28-c29** These two together entail that no universal is predicable of any particular.
- c30** This says that nothing is essentially predicable of a particular except itself.
- c31** This is a stronger version of $c30$ which says that, even if something isn't particular, nothing but itself is essentially predicable of it.

4 The Organon

The **Organon** is a collection of 6 books by Aristotle which form the main part of his work on Logic. The first of these is the **Categories** [1], on which Aristotle’s *Metaphysics* depends. The *Metaphysics*, at least the parts involved in the Grice/Code analysis [4, 3], is concerned with predication, which is also central to the formal core of Aristotle’s logic, the theory of the syllogism, presented in the **Prior Analytic, Book 1** [1].

Aristotle’s account of syllogistic logic covers modal reasoning. In attempting to understand Aristotelian essentialism, one of the key problems is to establish the relationship between the two distinctions between necessary and contingent proposition, and between essential and accidental predication.

Though Code [3] does not conceive of himself as engaged in formalising Aristotle in a modern predicate logic, his presentation seems much closer to predicate logic than to syllogistic logic. I would like to understand the *metaphysics* if possible in terms of the kind of logic which Aristotle had at his disposal. To explore the extent to which this might be possible, some models of syllogistic logic might be helpful.

When we look at the syllogism with particular concern for the notion of predication involved, we find that the Grice’s “izz/hazz” distinction (in Aristotle “said of” and “in”) is not relevant. In this respect predication is simpler in the syllogism, but instead we have an orthogonal distinction into four kinds of predication according to whether the subject is universal or particular, and whether the predication is affirmative or negative, over which are later added the modal operators.

The semantics of the syllogism remains a matter of controversy in some respects. The majority of the syllogisms held to be valid by Aristotle would be valid if universal and particular propositions were translated as universal and existential quantification in a modern predicate logic. Four of the syllogisms held to be valid by Aristotle would not be sound under such an interpretation. These four are distinguished by having universal premises but a particular conclusion. I propose to call these the *universal-particular* syllogisms, which I may abbreviate *u-p*.

There are three most common approaches to the u-p syllogisms:

1. Consider them to be fallacious (possibly admitting the implicature).
2. Consider Aristotle’s formal logic to be concerned exclusively with non-empty terms.
3. Consider universal propositions to be, in effect, the conjunction of a universal and an existential quantification.

However, it now seems that none of these are correct and that the approach closest to Aristotle’s text may be to have existential import in assertions (whether universal or particular) and absent from denials.

4.1 Models and Their Significance

Three different methods of analysis are illustrated in a very simple way in application to the laws of direct inference and the assertoric (i.e. non-modal) syllogistic.

These consisting in interpreting or modelling *aspects* of Aristotelian logic using modern languages and tools. It is important to understand that the construction of an interpretation or a model says by itself nothing about Aristotelian logic until the author makes a claim about the relationship between the model and the thing modelled. It is never the intention that a model should be in all respects similar to the original.

In producing these models I prepare the model first, with only a vague sense of what it might be good for, and then make certain comparisons, and describe the results. Thus the first two interpretations, which may be called translations or reductions, serve only to reduce the question of truth or provability in Aristotelian logic to that of truth or provability in some other notation or deductive system.

This they do incompletely. The first interpretation fails to demonstrate a subset of the syllogisms (those which I here refer to as u-p syllogisms). Its purpose is therefore primarily to show that the most naive interpretations of Aristotle's language fail to validate his reasoning, and thereby to cast doubt on such interpretations.

The next set of models may also be understood in similar ways, but they can also be considered as attempts to capture the truth conditions of categorical assertions. The evaluation of these models is based again in the first instance on the correspondence between truth and provability between the model and the original. Only one of these models fails to show significant differences in these respects from Aristotle. It is therefore only in respect of the last of the models that we are motivated to enquire closer into the fidelity of the truth-conditional semantics which they exhibit.

The claims made for the models are very limited, and the purposes of the models are:

- to exhibit certain methods and consider their merits
- to provide a basis for formal reasoning using a modern proof tool in a manner consistent with Aristotle's logic
- to provide a basis for an exploration of Aristotle's metaphysics using the same tools and methods

Three methods of modelling are used. These methods may be described as:

- translation
- shallow embedding
- deep embedding

In a translation syntax is not preserved, the aristotelian assertions are translated into some other form altogether.

In an embedding of either depth, an attempt is made to preserve and work with the original syntax, though in practice there may be alterations to the syntax to enable the analysis to be undertaken using some preferred software for formal analysis. In this document the software is **ProofPower**.

A series of mappings into HOL have been used to evaluate these possibilities:

1. a set theoretic interpretation (Section 4.3)
2. an interpretation in propositional logic (Section 4.4)
3. a naive predicate calculus interpretation (Section 4.5)
4. predicate calculus with no empty terms (Section 4.6)
5. existential import in universals (Section 4.7)
6. existential import in affirmations (Section 4.8)

7. reasoning about deep embeddings
8. modal syllogisms (by non-emptiness) (Section 4.9)

Of these the first two are simplistic translations. The categorical propositions are mapped to sentences in HOL which resemble set theory, or the predicate calculus and by this means is obtained a partial reduction of the decision problem for truth of syllogisms.

The next three are ‘shallow embeddings’. This means that the established representation of categorical assertions in the AEIO form is retained as syntax, so no translation of syntax is involved. These are given meaning by defining constants in HOL named ‘a’, ‘e’, ‘i’, and ‘o’ to capture the truth conditions of the four forms. This involves in the first instance a choice of type for the term variables and then the definition of these forms as functions from two such values to the type `BOOL`.

The interpretation is then evaluated by proving the soundness of various laws and syllogisms represented syntactically much as they are in the relevant literature.

Of these three shallow embeddings, the first two are based upon Strawson [11]. The third benefits from discussion on the *phil-logic* mailing list, and in particular on advice on the interpretation of Aristotle from Drake O’Brien supported by textual references which convinced me that neither of the Strawsonian interpretations could be correct.² These two points are firstly that Aristotle cannot be construed as presuming non-emptiness of terms (or as committing an existential fallacy), and that existential import is associated with the quality of the proposition, attaching to affirmative but not to negative propositions.

To facilitate comparison between the different interpretations the following list of ‘features’ will be checked. These are the various rules which might or might not be valid. My aim is to check which of these hold in each interpretation and then to summarise the results in a table.

- Laws of immediate inference.

These fall into the following categories:

- Simple conversion.
- Conversions per accidens.
- Obversion.
- Contraposition and inversion.
- The square of opposition.

- The syllogisms.

The present treatment was based firstly on the wikipedia account of [Aristotle’s Logic](#), secondly on Strawson [11] from whom I first obtained the syllogisms not in Aristotle and began incorporation of direct inference, and then from Spade[10] I obtained the names for the extra syllogisms and a fuller account of how the syllogisms can be derived which is not yet reflected in the following. Using these sources I didn’t get very close to the real Aristotle. Drake O’Brien, in discussion on *phil-logic* helped get be closer.

The main interest at present is the sequence of models which contributes to the formulation of an integrated model for the syllogism and the metaphysics in Section 5.

²This seems to be the accepted interpretation among contemporary scholars, and is described by Terence Parsons in his “Overview of Aristotle’s Logic” [8] available from his website.

4.2 Preliminaries

4.2.1 Generating the Syllogisms

This following presentation of the forms of syllogism is based primarily on [the one at wikipedia](#).

In the following several different interpretations of the syllogism are formalised as shallow semantic embeddings in HOL. Part of the evaluation of these interpretations consists in demonstrating that certain syllogisms are sound relative to the particular versions of the semantics. This is done by constructing a conjecture in HOL which will be true if the syllogism is sound, and then proving that the conjecture is true (i.e. proving the corresponding theorem). This procedure does not amount to a proof of the soundness of the syllogisms relative to the semantics, but gives almost the same level of confidence. A standard soundness proof would require a ‘deep embedding’, which would involve formalisation of the syntax of the syllogism and of the semantics as a relation between the syntax and the models. This would be considerably more arduous, and would not deliver comparably greater insights.

Before looking at alternative semantics for the syllogistic some machinery for generating syllogisms which will be used throughout is provided in this section.

Structurally a syllogism consists of two premises and a conclusion. The premises and conclusion are propositions, each of which is built from two terms in one of four ways. The different interpretations of the syllogism are obtained by choosing different types for the terms and different ways of constructing the propositions from the terms. For some purposes distinct ways of combining the propositions to form the completed syllogism will be necessary.

The information determining a particular interpretation will therefore be certain values in the met-language:

mkAT a function which converts a name to a term of that name (having type: $\overline{\text{ML}}\text{string} \rightarrow \text{TERM}^\top$)

mkAP a function which takes a propositional form and delivers a proposition constructor taking two terms and delivering a proposition (having type: $\overline{\text{ML}}\text{string} \rightarrow (\text{TERM} * \text{TERM}) \rightarrow \text{TERM}^\top$)

mkAS a function which takes a list of propositions (the premises) and a single proposition (the conclusion) and delivers a syllogism as a sequent goal (having type: $\overline{\text{ML}}(\text{TERM list} * \text{TERM}) \rightarrow \text{GOAL}^\top$)

A choice about the semantics will be encapsulated as a record having these three components.

SML

```
| type mapkit =  
|   {mkAT:string -> TERM,  
|     mkAP:string -> (TERM * TERM) -> TERM,  
|     mkAS:(TERM list * TERM) -> GOAL};
```

There are four forms of predication which are normally presented as infix operators over terms using the vowels “a”, “e”, “i”, “o”.

These are to be construed as follows:

HOL term	Meaning
$\lceil A a B \rceil$	All A are B
$\lceil A e B \rceil$	All A are not B
$\lceil A i B \rceil$	An A is B
$\lceil A o B \rceil$	An A is not B

Though exactly what these mean remains controversial and is a principle point of variation in the interpretations presented here.

Syllogisms come in four figures, according to the configuration of variables in the premises:

Figure 1 M-P, S-M \vdash S-P

Figure 2 P-M, S-M \vdash S-P

Figure 3 M-P, M-S \vdash S-P

Figure 4 P-M, M-S \vdash S-P

Where S, P and M are the subject, predicate and middle term respectively.

The following function generates a list of four quadruples of HOL variables which correspond to the four figures of syllogisms. It is parameterised by the HOL type used for predicates to that it can be re-used when we change the representation type.

SML

```
|fun figureS mkt =
|  let val M = mkt "M"
|     and P = mkt "P"
|     and S = mkt "S"
|  in [(M,P,S,M), (P,M,S,M), (M,P,M,S), (P,M,M,S)]
|  end;
```

In most cases terms are mapped to variables of appropriate types for the chosen semantics. The following function when given a type will return a term constructor mapping term names to variables of that type.

SML

```
|fun mkATvar tt s = mk_var(s,tt);
```

For these cases the list of figures with the relevant terms is obtained by this function:

SML

```
|fun figurest tt = figureS (mkATvar tt);
```

and the following function when instantiated to some type will give the nth figure with terms as variables of that type.

SML

```
|fun nthfig tt n = nth n (figurest tt);
```

These four figures are then repeated for each combination of the four types of premise in each of the premises and the conclusion. This gives $4 \times 4 \times 4 \times 4 = 256$ possibilities, of which 19 were held to be valid by Aristotle, four of them u-p syllogisms.

The use of vowels for the predicators allows the valid cases to be named using names in which the vowels tell you the form of the syllogism (if you also know the figure). The first vowel tells you the kind of syllogism in the first premise, the second vowel that in the second premise, and the third vowel that in the conclusion.

In the following table the names in square brackets are for u-p syllogisms. The names followed by exclamation marks are “subalternate mood”, they do not appear in aristotle but are valid in the models here for which the u-p syllogisms hold ³.

Figure 1	Figure 2	Figure 3	Figure 4
Barbara	Cesare	[Darapti]	[Bramantip]
Celarent	Camestres	Disamis	Camenes
Darii	Festino	Datisi	Dimaris
Ferio	Baroco	[Felapton]	[Fesapo]
Barbari!	Cesaro!	Bocardo	Fresison
Celaront!	Camestrop!	Ferison	Camenop!

A syllogism may be identified either by the name in the above table (for the valid syllogisms), or by the three vowels, and the figure.

The above table is captured by the following definitions in our metalanguage:

We now define a data structure from which the valid syllogisms can be generated. With the model we are using only the 15 non-u-p syllogisms. They are shown in the following data structure.

SML

```
val syllogism_data1 =
  [ ("Barbara", 1),
    ("Celarent", 1),
    ("Darii", 1),
    ("Ferio", 1),
    ("Cesare", 2),
    ("Camestres", 2),
    ("Festino", 2),
    ("Baroco", 2),
    ("Disamis", 3),
    ("Datisi", 3),
    ("Bocardo", 3),
    ("Ferison", 3),
    ("Camenes", 4),
    ("Dimaris", 4),
    ("Fresison", 4)];
```

The rest of the table consists of “universal-particular” syllogisms, i.e. syllogisms with universal premises but a particular conclusion.

The following four u-p syllogisms were held to be valid by Aristotle (the ones in square brackets in the table).

³This I got from Spade [10].

SML

```
val syllogism_data2 =  
  [("Darapti", 3),  
   ("Felapton", 3),  
   ("Bramantip", 4),  
   ("Fesapo", 4)];
```

The following five further syllogisms are also sound, though not noted as such by Aristotle (the ones followed by exclamation marks in the table).

SML

```
val syllogism_data3 =  
  [("Barbari", 1),  
   ("Celaront", 1),  
   ("Cesaro", 2),  
   ("Camestrop", 2),  
   ("Camenop", 4)];
```

We now assume available the triple of constructors mentioned above and define a function which converts syllogism identification in the form of a string of three vowels and a number in the range 1-4 into a syllogism. This is parameterised by a *mapkit* so that we can use it to deliver several different interpretations of the syllogism.

SML

```
fun vowels_from_string s = filter (fn x => x mem (explode "aeiou")) (explode s);  
  
fun mkSyll (sk:mapkit) (s, n) =  
  let val [pa, pb, co] = vowels_from_string s  
      val (a,b,c,d) = nth (n-1) (figureS (#mkAT sk))  
      val pl = [#mkAP sk pa (a,b), #mkAP sk pb (c,d)]  
      val c = #mkAP sk co (#mkAT sk "S", #mkAT sk "P")  
  in #mkAS sk (pl, c)  
  end;
```

We can then map this operation over the a list of syllogism data as follows:

SML

```
fun mkGoals sk = map (fn (s,n) => (s, mkSyll sk (s, n)));
```

If we further supply a proof tactic and a labelling suffix then we can prove a whole list of goals and store the results in the current theory.

SML

```
fun proveGoals tac suff = map  
  (fn (s,g) =>  
   (s,  
    save_thm (s^suff, tac_proof (g, tac))  
             handle _ => t_thm));
```


SML

```
|fun proveSylls sk tac suff sd = proveGoals tac suff (mkGoals sk sd);
```

The following functions take a string which is the name of a syllogism extract the vowels which occur in it and convert them into the corresponding predication operator to give a triple of operators.

SML

```
|fun op_from_char ot c = mk_const (if c = "o" then "u" else c, ot);

|fun optrip_from_text ot s =
  let val [a, b, c] = (map (op_from_char ot) o vowels_from_string) s;
  in (a, b, c)
  end;
```

A Common Special Case The most common pattern is that:

- Terms are mapped to variables of some type.
- Propositions are formed using infix operators *a*, *e*, *i*, *o*.
- Syllogisms are sequents, the premises in the assumptions.

The following constructs a mapkit for such cases when supplied with the type of the term variables, and the four proposition constructors.

SML

```
|fun mkSimpMapkit ty opl =
  let val mkcop = fn s => snd (find (combine (explode "aeio") opl) (fn (n,v) => n = s))
  in
    {mkAT = fn s => mk_var(s, ty),
     mkAP = fn s => fn (l,r) => mk_app(mk_app (mkcop s, l), r),
     mkAS = fn x => x}
  end;
```

4.2.2 The Square of Opposition

A	contraries	E
subalternates	contradictories	subalternates
I	subcontraries	O

Table 1: The square of Opposition 1

The relationships between corners of the square of opposition have the following names:

The full set of relationships exhibited by the square is:

The goal conclusions for theorems expressing compliance are (details of syntax will vary between interpretations):

SML

```

| val ao_contrad =  $\lceil A a B \Leftrightarrow \neg A o B \rceil$ ;
| val ei_contrad =  $\lceil A e B \Leftrightarrow \neg A i B \rceil$ ;
| val ae_contrar =  $\lceil \neg (A a B \wedge A e B) \rceil$ ;
| val io_subcont =  $\lceil A i B \vee A o B \rceil$ ;
| val ai_subalt =  $\lceil A a B \Rightarrow A i B \rceil$ ;
| val eo_subalt =  $\lceil A e B \Rightarrow A o B \rceil$ ;

```

This provides a compact way of comparing different interpretations of the assertoric forms.

The principal point of difference is in subalternation which fails when the quantifiers are interpreted in the modern sense and empty terms are admitted. One remedy for this defect might be to add existential import to the universal quantifier, but this by itself would cause the contradictories to fail. Another remedy is to add existential import to the universal and define the particular modes through the contradictories.

Peter Strawson enumerates 14 laws of immediate inference, not all of which seem to have been embraced by Aristotle. The list nevertheless provides a good collection against which an interpretation of the propositions can be evaluated, along with the 24 syllogisms.

The following table shows the correspondence between the numbers used by Strawson, the names used here, and a fuller name.

4.2.3 Are The Syllogisms Tautologous?

This section is a kind of preliminary skirmish provoked by some discussions on [phil-logic](#) in which one or two specific questions are addressed.

The question which provoked this section was:

- Are the valid syllogisms tautologous?

This is complex because it depends on a view about the semantics of the syllogism and also upon what particular notion of tautology is in play.

Two related questions come from remarks by P.V.Spade. P.V. Spade, in a thumbnail history of logic [10] remarks that some later peripatetics attempted to show that stoic propositional logic was simply the syllogism in other clothes, and others that the two are in some sense equivalent.

- Is there any sense in which the syllogism can be said to encompass Stoic propositional logic?
- Is there any sense in which the syllogism might be seen to be equivalent to Stoic propositional logic?

contradictories	A and B are contradictories if $\lceil A \Leftrightarrow \neg B \rceil$
contraries	A and B are contraries if they are not both true.
subcontraries	A and B are sub-contraries if they are not both false.
subalternates	B is subalternate to A if A implies B

Table 2: The square of Opposition 2

I have no knowledge of Stoic propositional logic, and this section is therefore concerned with the relationship between the syllogism and modern propositional logic, which I presume is a superset (probably but not necessarily proper) of Stoic propositional logic.

I distinguish three notions of tautology (though this is not intended to be an exhaustive list) on which something can be said in this connection.

propositional A valid sentence of the propositional calculus.

first order A valid sentence of first order logic.

analytic A sentence whose truth conditions assign truth to it in every situation or interpretation consistent with the meanings of the language in which it is expressed.

The connection between these is in the notions of truth function and truth condition. The first two are notions of truth functional tautology. In both languages the truth value of a sentence is a function of the truth values of the atomic sentences it contains, and a sentence is tautologous if that truth function is the one which always delivers the value “true”. In the second of these two the notion of truth function is slightly more complex and some explanation may be necessary of how the quantifiers can be so construed. It may be noted however that it is a thesis of Wittgenstein’s Tractatus[12] that logical truth in general is tautologous. In the last case the restriction to truth functions (i.e. functions which take truth values as arguments and deliver truth values as results) is dropped, allowing that the essential element is that the relevant part of the semantics can be rendered as *truth conditions* in which the result is a truth value but the argument need not be, and that it is of the essence of the concept tautology to single out from truth functions or conditions in general those which always deliver the truth value ‘true’.

In relation to these three notions of tautology, one thinks naturally of categorical propositions as involving quantification, and therefore, of syllogisms, if tautologous being so at best as first order validities, but in any case, if sound, as analytic. These intuitions do not support a connection between syllogisms and propositional tautologies.

The connection we illustrate here is obtained via elementary set theory. The intuition behind this is that Aristotelian predication is analogous to set inclusion, taking terms to denote their extensions (and singular terms as denoting singleton sets), together with complementation and negation. This works except for the problematic universal-particular syllogisms which infer particular conclusions from universal premises. A connection with propositional logic can be obtained from this, since the elementary set theory required for this interpretation is a boolean algebra, as is the propositional calculus.

Partial reductions to set theory and propositional logic are exhibited in sections 4.3 and 4.4. The reduction to set theory is semantically more plausible than the second step to propositional logic, and provides an account of those u-p syllogisms, which are not accounted for by the propositional reduction.

ao_contrad	contradictories
ei_contrad	contradictories
ae_contrar	contraries
io_subcont	subcontraries
ai_subalt	subalternates
eo_subalt	subalternates

Table 3: The square of Opposition 3

We obtain kind of reduction of the decision problem for the non-u-p syllogisms to the decision problem for truth functional tautologies, but only in a very informal sense, since there are only a small finite set of valid syllogisms (though schemata) and so the decision problem is trivial. For this reason syllogistic logic cannot be as expressive as a modern propositional logic, but conceivably might be closer to Stoic propositional logic.

The first step, to set theory is illustrated in the Wikipedia article on the syllogism, which gives Venn diagrams for all 24 “valid” syllogisms.

Though it is convenient to think of this as a two stage reduction, in the following implementation the two different reductions are generated independently.

4.3 Interpretation in Set Theory

SML

```
| open_theory "aristotle";
| force_new_theory "syllog1";
```

Under the proposed mapping terms of the syllogism are mapped to propositions. The term “Mortal” may be thought of as being mapped to the proposition “x is Mortal”.

Thus the Barbara syllogism:

- All As are Bs
- All Bs are Cs
- =====
- All As are Cs

might be rendered set theoretically as:

$$(A \subseteq B \wedge B \subseteq C) \Rightarrow (A \subseteq C)$$

where A, B and C are subsets of some universal set.

However, since we are working here in a sequent calculus we can render the theorems closely to the original as:

$$\left| \begin{array}{l} [A \subseteq B, \\ B \subseteq C] \\ \vdash A \subseteq C \end{array} \right.$$

A set theoretic presentation of the non-u-p syllogisms can be obtained by assuming that the sets are not empty, but no similarly plausible account of these syllogisms is available in propositional logic.

Here is a table describing the proposed mappings:

Aristotle	Meaning	Set Theory	Propositional Analogue
A a B	All A are B	$A \subseteq B$	$A \Rightarrow B$
A e B	All A are not B	$A \subseteq \sim B$	$A \Rightarrow \neg B$
A i B	An A is B	$\neg(A \subseteq \sim B)$	$\neg(A \Rightarrow \neg B)$
A o B	An A is not B	$\neg(A \subseteq B)$	$\neg(A \Rightarrow B)$

These mappings can be encapsulated as two ‘mapkits’.

First the set theory mapping:

```
SML
| val st_mkAP =
|   let   fun a (su, pr) =  $\ulcorner_{ML} su \urcorner \subseteq_{ML} pr \urcorner \urcorner$ 
|         fun e (su, pr) =  $\ulcorner_{ML} su \urcorner \subseteq \sim_{ML} pr \urcorner \urcorner$ 
|         fun i (su, pr) =  $\ulcorner \neg_{ML} su \urcorner \subseteq \sim_{ML} pr \urcorner \urcorner$ 
|         fun u (su, pr) =  $\ulcorner \neg_{ML} su \urcorner \subseteq_{ML} pr \urcorner \urcorner$ 
|   in fn s => case s of "a" => a | "e" => e | "i" => i | "o" => u
|   end;
|
| declare_type_abbrev("TermS", [],  $\ulcorner \text{!} a SET \urcorner$ );
|
| val st_mapkit:mapkit =
|   {
|     mkAT = fn s => mk_var(s,  $\ulcorner \text{TermS} \urcorner$ ),
|     mkAP = st_mkAP,
|     mkAS = fn x => x};
```

4.3.1 Generating The Propositions

This section is mainly given over to short programs in our metalanguage the end effect of which is to secure the proof of the 15 theorems of set theory and 15 propositional tautologies which are obtained from non-u-p syllogisms by this naive transformation.

The results are visible in the “theorems” section of the theory listing in Appendix A.2, and this section can be safely skipped by anyone whose interest is purely philosophical.

Mostly this uses functionality devised in the previous section to achieve two slightly different sets of theorems.

The following functions take a string which is the name of a syllogism, extract the vowels which occur in it and convert them into the corresponding propositional formula to give a triple of propositions.

giving the following list of “goals”:

```
| val it =
|   [ ( $\ulcorner M \subseteq P \urcorner, \ulcorner S \subseteq M \urcorner, \ulcorner S \subseteq P \urcorner$ ),
|     ( $\ulcorner M \subseteq \sim P \urcorner, \ulcorner S \subseteq M \urcorner, \ulcorner S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner M \subseteq P \urcorner, \ulcorner \neg S \subseteq \sim M \urcorner, \ulcorner \neg S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner M \subseteq \sim P \urcorner, \ulcorner \neg S \subseteq \sim M \urcorner, \ulcorner \neg S \subseteq P \urcorner$ ),
|     ( $\ulcorner P \subseteq \sim M \urcorner, \ulcorner S \subseteq M \urcorner, \ulcorner S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner P \subseteq M \urcorner, \ulcorner S \subseteq \sim M \urcorner, \ulcorner S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner P \subseteq \sim M \urcorner, \ulcorner \neg S \subseteq \sim M \urcorner, \ulcorner \neg S \subseteq P \urcorner$ ),
|     ( $\ulcorner P \subseteq M \urcorner, \ulcorner \neg S \subseteq M \urcorner, \ulcorner \neg S \subseteq P \urcorner$ ),
|     ( $\ulcorner \neg M \subseteq \sim P \urcorner, \ulcorner M \subseteq S \urcorner, \ulcorner \neg S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner M \subseteq P \urcorner, \ulcorner \neg M \subseteq \sim S \urcorner, \ulcorner \neg S \subseteq \sim P \urcorner$ ),
|     ( $\ulcorner \neg M \subseteq P \urcorner, \ulcorner M \subseteq S \urcorner, \ulcorner \neg S \subseteq P \urcorner$ ),
|     ( $\ulcorner M \subseteq \sim P \urcorner, \ulcorner \neg M \subseteq \sim S \urcorner, \ulcorner \neg S \subseteq P \urcorner$ ),
```

```

| ([ $\Gamma P \subseteq M^\top$ ,  $\Gamma M \subseteq \sim S^\top$ ],  $\Gamma S \subseteq \sim P^\top$ ),
| ([ $\Gamma \neg P \subseteq \sim M^\top$ ,  $\Gamma M \subseteq S^\top$ ],  $\Gamma \neg S \subseteq \sim P^\top$ ),

```

4.3.2 Proving the Syllogisms

Here is a proof tactic for the non-u-p syllogisms under the set theory interpretation.

```

SML
| val st_tac = REPEAT (POP_ASM_T ante_tac)
|   THEN PC_T1 "hol1" rewrite_tac []
|   THEN prove_tac[];

```

We now apply this to the non-u-p syllogisms, saving the results in the theory (Appendix A.2).

```

SML
| proveSylls st_mapkit st_tac "" syllogism_data1;

```

We now adjust the set theoretic reduction to deliver u-p syllogisms by including a non-emptiness assumption and obtain proofs of the nine syllogisms valid u-p syllogisms.

giving the following list of “goals”:

```

| val it =
|   [ ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma M \subseteq P^\top$ ,  $\Gamma M \subseteq S^\top$ ],  $\Gamma \neg S \subseteq \sim P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma M \subseteq \sim P^\top$ ,  $\Gamma M \subseteq S^\top$ ],  $\Gamma \neg S \subseteq P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma P \subseteq M^\top$ ,  $\Gamma M \subseteq S^\top$ ],  $\Gamma \neg S \subseteq \sim P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma P \subseteq \sim M^\top$ ,  $\Gamma M \subseteq S^\top$ ],  $\Gamma \neg S \subseteq P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma M \subseteq P^\top$ ,  $\Gamma S \subseteq M^\top$ ],  $\Gamma \neg S \subseteq \sim P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma M \subseteq \sim P^\top$ ,  $\Gamma S \subseteq M^\top$ ],  $\Gamma \neg S \subseteq P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma P \subseteq \sim M^\top$ ,  $\Gamma S \subseteq M^\top$ ],  $\Gamma \neg S \subseteq P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma P \subseteq M^\top$ ,  $\Gamma S \subseteq \sim M^\top$ ],  $\Gamma \neg S \subseteq P^\top$ ),
|     ([ $\Gamma \neg \{ \} \in \{M; S; P\}^\top$ ,  $\Gamma P \subseteq M^\top$ ,  $\Gamma M \subseteq \sim S^\top$ ],  $\Gamma \neg S \subseteq P^\top$ )]
| : (TERM list * TERM) list

```

We now apply this to the non-u-p syllogisms, saving the results in the theory.

The following code generates the goals for proving the above syllogisms from the previously defined data structure describing the valid syllogisms, generates and checks formal proofs and saves the resulting theorems in the current theory.

```

SML
| val st2_mapkit:mapkit =
|   {
|     mkAT = fn s => mk_var(s,  $\Gamma$ :TermS $^\top$ ),
|     mkAP = st_mkAP,
|     mkAS = fn (ps, c) => ( $\Gamma \neg (\{ \} : a SET) \in \{M; S; P\}^\top :: ps, c$ );
|   }

```

```

SML
| val st2_tac = REPEAT (POP_ASM_T ante_tac)
|   THEN PC_T1 "hol1" rewrite_tac [] THEN prove_tac[];

```

```

SML
| proveSylls st2_mapkit st2_tac "" (syllogism_data2 @ syllogism_data3);

```

The full set of syllogisms may be found in the theory (Appendix A.2).

4.4 Propositional Interpretation

SML

```
| open_theory "aristotle";
| force_new_theory "syllog2";
```

The term “Mortal” may be thought of as being mapped to the proposition “x is Mortal”.

Thus the Barbara syllogism:

- All As are Bs
- All Bs are Cs
- =====
- All As are Cs

is rendered in the propositional calculus as:

$$(A \Rightarrow B \wedge B \Rightarrow C) \Rightarrow (A \Rightarrow C)$$

where A, B and C are boolean values or propositions.

However, since we are working here in a sequent calculus we can render the theorems closely to the original as:

```
| [A => B,
|   B => C]
| ⊢ A => C
```

The propositional logic mapping is determined as follows:

SML

```
| declare_type_abbrev("TermP", [], ⌈:BOOL⌋);
|
| val pl_mapkit:mapkit =
|   let fun a (su, pr) = ⌈_MLsu⌋ => ⌈_MLpr⌋⌋
|       fun e (su, pr) = ⌈_MLsu⌋ => ¬ ⌈_MLpr⌋⌋
|       fun i (su, pr) = ⌈¬ (⌈_MLsu⌋ => ¬ ⌈_MLpr⌋)⌋
|       fun u (su, pr) = ⌈¬ (⌈_MLsu⌋ => ⌈_MLpr⌋)⌋
|   in { mkAT = fn s => mk_var(s, ⌈:TermP⌋),
|       mkAP = fn s => case s of
|         "a" => a
|       |   "e" => e
|       |   "i" => i
|       |   "o" => u,
|       mkAS = fn x => x }
|   end;
```

giving the following list of “goals”:

```

val it =
  [([M ⇒ P⌈, [S ⇒ M⌈, [S ⇒ P⌈),
    ([M ⇒ ¬ P⌈, [S ⇒ M⌈, [S ⇒ ¬ P⌈),
    ([M ⇒ P⌈, [¬ (S ⇒ ¬ M)⌈, [¬ (S ⇒ ¬ P)⌈),
    ([M ⇒ ¬ P⌈, [¬ (S ⇒ ¬ M)⌈, [¬ (S ⇒ P)⌈),
    ([P ⇒ ¬ M⌈, [S ⇒ M⌈, [S ⇒ ¬ P⌈),
    ([P ⇒ M⌈, [S ⇒ ¬ M⌈, [S ⇒ ¬ P⌈),
    ([P ⇒ ¬ M⌈, [¬ (S ⇒ ¬ M)⌈, [¬ (S ⇒ P)⌈),
    ([P ⇒ M⌈, [¬ (S ⇒ M)⌈, [¬ (S ⇒ P)⌈),
    ([¬ (M ⇒ ¬ P)⌈, [M ⇒ S⌈, [¬ (S ⇒ ¬ P)⌈),
    ([M ⇒ P⌈, [¬ (M ⇒ ¬ S)⌈, [¬ (S ⇒ ¬ P)⌈),
    ([¬ (M ⇒ P)⌈, [M ⇒ S⌈, [¬ (S ⇒ P)⌈),
    ([M ⇒ ¬ P⌈, [¬ (M ⇒ ¬ S)⌈, [¬ (S ⇒ P)⌈),
    ([P ⇒ M⌈, [M ⇒ ¬ S⌈, [S ⇒ ¬ P⌈),
    ([¬ (P ⇒ ¬ M)⌈, [M ⇒ S⌈, [¬ (S ⇒ ¬ P)⌈),
    ([P ⇒ ¬ M⌈, [¬ (M ⇒ ¬ S)⌈, [¬ (S ⇒ P)⌈]] : (TERM list * TERM) list

```

The following tactic suffices for proving propositional tautologies.

```

SML
val pl_tac = REPEAT (POP_ASM_T ante_tac)
              THEN REPEAT strip_tac;

```

We now apply this to the non-u-p syllogisms, saving the results in the theory.

```

SML
val valid_psylls = proveSylls pl_mapkit pl_tac "" syllogism_data1;

```

This is the resulting value.

```

val valid_psylls =
  [("Barbara", M ⇒ P, S ⇒ M ⊢ S ⇒ P),
   ("Celarent", M ⇒ ¬ P, S ⇒ M ⊢ S ⇒ ¬ P),
   ("Darii", M ⇒ P, ¬ (S ⇒ ¬ M) ⊢ ¬ (S ⇒ ¬ P)),
   ("Ferio", M ⇒ ¬ P, ¬ (S ⇒ ¬ M) ⊢ ¬ (S ⇒ P)),
   ("Cesare", P ⇒ ¬ M, S ⇒ M ⊢ S ⇒ ¬ P),
   ("Camestres", P ⇒ M, S ⇒ ¬ M ⊢ S ⇒ ¬ P),
   ("Festino", P ⇒ ¬ M, ¬ (S ⇒ ¬ M) ⊢ ¬ (S ⇒ P)),
   ("Baroco", P ⇒ M, ¬ (S ⇒ M) ⊢ ¬ (S ⇒ P)),
   ("Disamis", ¬ (M ⇒ ¬ P), M ⇒ S ⊢ ¬ (S ⇒ ¬ P)),
   ("Datisi", M ⇒ P, ¬ (M ⇒ ¬ S) ⊢ ¬ (S ⇒ ¬ P)),
   ("Bocardo", ¬ (M ⇒ P), M ⇒ S ⊢ ¬ (S ⇒ P)),
   ("Ferison", M ⇒ ¬ P, ¬ (M ⇒ ¬ S) ⊢ ¬ (S ⇒ P)),
   ("Camenes", P ⇒ M, M ⇒ ¬ S ⊢ S ⇒ ¬ P),
   ("Dimaris", ¬ (P ⇒ ¬ M), M ⇒ S ⊢ ¬ (S ⇒ ¬ P)),
   ("Fresison", P ⇒ ¬ M, ¬ (M ⇒ ¬ S) ⊢ ¬ (S ⇒ P))] : (string * THM) list

```

.. which is a list of name/theorem pairs of the tautologies corresponding to each syllogism.

The theorems are also displayed in the theory listing in Appendix [A.3](#)

Some words about the very limited significance of this little exercise would be appropriate here!

4.5 Naive Interpretation in Predicate Calculus

SML

```
| open_theory "aristotle";
| force_new_theory "syllog3";
```

4.5.1 Semantics

Aristotle’s syllogistic logic is concerned with inferences between judgements considered as predications. A predication in Aristotle affirms a *predicate* of some *subject*, but by contrast with more recent notions of predication the subject need not be an individual, the kinds of things which appear as predicates may also appear as subjects, and the relationship expressed seems closer to a modern eye to set inclusion than to what we now regard as predication. Since subject and predicate are for present purposes the same kind of thing, it is useful to have a name for that kind of thing, and I will use the name *property*.

There are four kinds of predication which we have here to account for, which we will do by offering definitions which provide a good model for syllogistic logic, i.e. one in which the syllogisms held to be true by Aristotle are in fact true. Before providing these definitions we must decide what kind of thing are the terms which are related by Aristotelian predication.

In HOL the most natural answer to this is “boolean valued functions” which are objects of type $\ulcorner : 'a \rightarrow \text{BOOL} \urcorner$ for some type of individuals which we can leave open by using the type variable $\ulcorner : 'a \urcorner$. This provides a simple model of Aristotle’s non-u-p syllogistic reasoning. Four of the syllogisms which Aristotle considered valid fail under this conception of predicate, because among the objects of type $\ulcorner : 'a \rightarrow \text{BOOL} \urcorner$ is the function $\ulcorner \lambda x : 'a \bullet F \urcorner$ which corresponds to a predicate with empty extension and does not admit inference from the universal to the existential (unless the universal is interpreted specially).

SML

```
| declare_type_abbrev("Term2", [], \ulcorner : 'a \rightarrow \text{BOOL} \urcorner);
```

4.5.2 Predication

“o” is already in use for functional composition, so we will use “u” instead and then use an alias to permit us to write this as “o” (type inference will usually resolve any ambiguity).

The predication operators are defined as follows:

SML

```
| declare_infix (300, "a");
| declare_infix (300, "e");
| declare_infix (300, "i");
| declare_infix (300, "u");
```

HOL Constant

```
| $a : Term2 \rightarrow Term2 \rightarrow \text{BOOL}
```

```
| \forall A B \bullet A a B \Leftrightarrow \forall x \bullet A x \Rightarrow B x
```

HOL Constant

```
| $e : Term2 → Term2 → BOOL
```

```
| ∀A B• A e B ⇔ ∀x• A x ⇒ ¬ B x
```

HOL Constant

```
| $i : Term2 → Term2 → BOOL
```

```
| ∀A B• A i B ⇔ ∃x• A x ∧ B x
```

HOL Constant

```
| $u : Term2 → Term2 → BOOL
```

```
| ∀A B• A u B ⇔ ∃x• A x ∧ ¬ B x
```

SML

```
| declare_alias("o", ⌈$u⌋);
```

Note that as defined above these come in complementary pairs, a being the negation of o and e of i . If we had negation we could manage with just two predication operators.

4.5.3 The Laws of Immediate Inference

Though in the source of this kind of “literate script” are to be found the scripts for generating and checking the proofs of all the theorems presente, it will not be my practice to expose these scripts in the printed version of the document. These scripts are not usually intelligible other than in that intimate man-machine dialogue which they mediate, and sufficient knowledge for most purposes of the structure of the proof will be found in the detailed lemmas proven (since the level of proof automation is modest).

However, I will begin by exposing some of the scripts used for obtaining proofs of syllogisms in this model, to give the reader an impression of the level of complexity and kind of obscurity involved in this kind of formal work, I will not attempt sufficient explanation to make these scripts intelligible, they are best understood in the interactive environment, all the scripts are available for readers who want to run them.

Most readers are expected to skip over the gory details, the philosophical points at stake do not depend on the details of the proofs.

Before addressing the laws of immediate inference ⁴ I devise a tactic for automating simple proofs in this domain.

The following elementary tactic expands the goal by applying the definitions of the operators and then invokes a general tactic for the predicate calculus. A rule is also defined using that tactic for direct rather than interactive proof.

SML

```
| val syll_tac = asm_prove_tac (map get_spec [⌈$a⌋, ⌈$e⌋, ⌈$i⌋, ⌈$u⌋]);  
| fun syll_rule g = tac_proof (g, syll_tac);
```

⁴in which I followed Strawson [11], though I can now cite [Aristotle, Prior Analytic, Book 1, Part 2.](#) [1]

Simple Conversion Using the above tactic thus:

```
SML
| val e_conv_thm = save_thm ("e_conv_thm", syll_rule([ $\lceil A e B \rceil$ ,  $\lceil B e A \rceil$ ]);
| val i_conv_thm = save_thm ("i_conv_thm", syll_rule([ $\lceil A i B \rceil$ ,  $\lceil B i A \rceil$ ]);
```

The following two theorems are obtained.

```
| val e_conv_thm = A e B  $\vdash$  B e A : THM
| val i_conv_thm = A i B  $\vdash$  B i A : THM
```

The following two theorems show that the other obvious conversions are false.

```
| a_not_conv_thm =  $\vdash \exists A B \bullet A a B \wedge \neg B a A$ 
| o_not_conv_thm =  $\vdash \exists A B \bullet A o B \wedge \neg B o A$ 
```

Conversion Per Accidens These don't work here because they rely upon the u-p syllogisms.

Obversion For these we need to define an operation of complementation on terms.

HOL Constant

```
| Complement : Term2  $\rightarrow$  Term2
```

```
|  $\forall A \alpha \bullet (\text{Complement } A) \alpha \Leftrightarrow \neg (A \alpha)$ 
```

We will use “ \sim ” as a shorthand for “complement”.

SML

```
| declare_alias (" $\sim$ ",  $\lceil \text{Complement} \rceil$ );
```

```
| ae_obv_thm = A a B  $\vdash$  A e  $\sim$  B
| ea_obv_thm = A e B  $\vdash$  A a  $\sim$  B
| io_obv_thm = A i B  $\vdash$  A o  $\sim$  B
| oi_obv_thm = A o B  $\vdash$  A i  $\sim$  B
```

Contraposition and Inversion

4.5.4 The Square of Opposition

```
| ao_contrad_thm =  $\vdash A a B \Leftrightarrow \neg A o B$ 
| ei_contrad_thm =  $\vdash A e B \Leftrightarrow \neg A i B$ 
|  $\neg$ ae_contrar_thm =  $\vdash \neg (\forall A B \bullet \neg (A a B \wedge A e B))$ 
|  $\neg$ io_subcont_thm =  $\vdash \neg (\forall A B \bullet A i B \vee A o B)$ 
|  $\neg$ ai_subalt_thm =  $\vdash \neg (\forall A B \bullet A a B \Rightarrow A i B)$ 
|  $\neg$ eo_subalt_thm =  $\vdash \neg (\forall A B \bullet A e B \Rightarrow A o B)$ 
```

4.5.5 The Syllogisms

The fifteen valid non-u-p syllogisms are true under this semantics and can be proven formally with ease.

4.5.6 Generating Syllogisms

First we make a *mapkit*.

SML

```
|val s1mapkit:mapkit = mkSimpMapkit ⌈:Term2⌋ [⌈$a⌋,⌈$e⌋,⌈$i⌋,⌈$u⌋];
```

Then we apply this in generating and proving the non u-p syllogisms.

SML

```
|proveGoals syll_tac "" (mkGoals s1mapkit syllogism_data1);
```

This is the resulting value.

```
|val valid_sylls = [  
  ("Barbara", M a P, S a M ⊢ S a P),  
  ("Celarent", M e P, S a M ⊢ S e P),  
  ("Darii", M a P, S i M ⊢ S i P),  
  ("Ferio", M e P, S i M ⊢ S o P),  
  ("Cesare", P e M, S a M ⊢ S e P),  
  ("Camestres", P a M, S e M ⊢ S e P),  
  ("Festino", P e M, S i M ⊢ S o P),  
  ("Baroco", P a M, S o M ⊢ S o P),  
  ("Disamis", M i P, M a S ⊢ S i P),  
  ("Datisi", M a P, M i S ⊢ S i P),  
  ("Bocardo", M o P, M a S ⊢ S o P),  
  ("Ferison", M e P, M i S ⊢ S o P),  
  ("Camenes", P a M, M e S ⊢ S e P),  
  ("Dimaris", P i M, M a S ⊢ S i P),  
  ("Fresison", P e M, M i S ⊢ S o P)  
]: (string * THM) list
```

The theorems are also displayed in the theory listing in [Appendix A.4](#)

4.6 Predicate Calculus Without Empty Terms

There is more than one way in which the semantics of the syllogism can be modified to make the inference from “All As are Bs” to “Some As are Bs” sound. One way would be to change the meaning of “All”. This would interfere with the square of opposition by making diagonal entries no longer contradictories. From this I initially inferred that the exclusion of empty predicates is a better approach. ⁵

We can then prove valid 24 forms of syllogism.

For the most concise statement of the results of the exercise, the reader should refer directly to the theory listing in [Appendix A.5](#).

SML

```
|open_theory "aristotle";  
|force_new_theory "syllog4";
```

⁵This seemed then to be endorsed by Robin Smith in the [Stanford Encyclopaedia of Philosophy](#) [9].

4.6.1 Semantics

The key to getting the u-p syllogisms into the model is the adoption of a type for the variables in the syllogisms which does not include empty predicates. We could do this by defining a new type which is a sub-type of the propositional functions, but it is simpler to use another type-abbreviation as follows.

Instead of using a propositional function, which might be unsatisfiable, we use an ordered pair. The pair consists of one value, a value for which the predicate is true, and a propositional function. The predicate (v, pf) is then to be considered true of some value x *either* if x is v *or* if pf is true of x .

SML

```
| declare_type_abbrev("Term3", [],  $\ulcorner 'a \times ('a \rightarrow BOOL) \urcorner$ );
```

4.6.2 Predication

To work with this new type for the predicates we define a function which will convert this kind of predicate into the old kind, as follows:

HOL Constant

```
|  $p : Term3 \rightarrow ('a \rightarrow BOOL)$ 
|-----
|  $\forall A \bullet p A = \lambda x \bullet let (v, f) = A in x = v \vee f x$ 
```

The resulting values have the same type as the old, but they will never have empty extension.

The following principle can be proven (proof omitted):

```
|  $p\_exists\_lemma =$ 
|  $\vdash \forall A \bullet \exists v \bullet p A v$ 
```

This principle is what we need to prove the u-p syllogisms.

It should be noted that there is no complementation operation on terms of this type and that the obversions will therefore fail.

We then proceed in a similar manner to the first model, using the function p to convert the new kind of predicate into the old.

The predication operators are then defined. Note that the differences are small and uniform. The type $\ulcorner : PROP \urcorner$ is changed to $\ulcorner : Term3 \urcorner$ and the function p is invoked before applying a predicate.

SML

```
| declare_infix (300, "a");
| declare_infix (300, "e");
| declare_infix (300, "i");
| declare_infix (300, "u");
```

HOL Constant

```
|  $\$a : Term3 \rightarrow Term3 \rightarrow BOOL$ 
|-----
|  $\forall A B \bullet A a B \Leftrightarrow \forall x \bullet p A x \Rightarrow p B x$ 
```

HOL Constant

```
| $e : Term3 → Term3 → BOOL
```

```
| ∀A B• A e B ⇔ ∀x• p A x ⇒ ¬ p B x
```

HOL Constant

```
| $i : Term3 → Term3 → BOOL
```

```
| ∀A B• A i B ⇔ ∃x• p A x ∧ p B x
```

HOL Constant

```
| $u : Term3 → Term3 → BOOL
```

```
| ∀A B• A u B ⇔ ∃x• p A x ∧ ¬ p B x
```

SML

```
| declare_alias("o", [ "$u" ]);
```

With these defined we can now produce a ‘mapkit’ for translating the syllogisms under this semantics.

SML

```
| val s2_mapkit:mapkit = mkSimpMapkit [":Term3"] [ "$a", "$e", "$i", "$u" ];
```

4.6.3 Laws of Immediate Inference

The same tactic used for proof of the syllogisms in the previous model still works with this model (with the new definitions), but does not prove the u-p syllogisms.

To obtain proofs of these other syllogisms we need to make use of the lemma we proved about p , p - \exists _lemma. This we do by instantiating it for each of the variables which appear in the syllogisms and supplying these for use in the proof.

SML

```
| val syll_tac2 =  
|   (MAP_EVERY (fn x => strip_asm_tac (∀_elim x p_∃_lemma))  
|     [":M:Term3", ":P:Term3", ":S:Term3", ":A:Term3", ":B:Term3"])  
|   THEN asm_prove_tac (map get_spec [ "$a", "$e", "$i", "$u" ]);  
  
| fun syll_rule2 g = tac_proof(g, syll_tac2);
```

SML

```
| val e_conv = ([":A e B", ":B e A"]);  
| val i_conv = ([":A i B", ":B i A"]);
```

SML

```
| val e_conv_thm = save_thm ("e_conv_thm", syll_rule2 e_conv);  
| val i_conv_thm = save_thm ("i_conv_thm", syll_rule2 i_conv);
```

```
| val e_conv_thm = A e B ⊢ B e A : THM  
| val i_conv_thm = A i B ⊢ B i A : THM
```

Simple Conversion In this version of the semantics, “a” and “o” conversion is neither provable nor refutable. In the previous version, since the universe is a HOL type there is at least one individual, and contradictory predicates are allowed, we can use these two to disprove the two conversions. With this semantics there is no empty predicate, and we cannot know that there are two distinct predicates.

SML

```
| val sg_03 = ([⊢ A a B⊥], ⊢ B i A⊥);  
| val sg_04 = ([⊢ A e B⊥], ⊢ B u A⊥);
```

SML

```
| val ai_conv_thm = save_thm ("ai_conv_thm", syll_rule2 sg_03);  
| val eo_conv_thm = save_thm ("eo_conv_thm", syll_rule2 sg_04);
```

```
| val ai_conv_thm = A a B ⊢ B i A : THM  
| val eo_conv_thm = A e B ⊢ B o A : THM
```

Conversion Per Accidens

Obversion We have been unable to define a complementation operation and the obversions listed by Strawson cannot even be expressed in this representation.

SML

```
| val sg_09 = ([]: TERM list, ⊢ A a B ⇔ ¬ A u B⊥);  
| val sg_10 = ([]: TERM list, ⊢ A e B ⇔ ¬ A i B⊥);  
| val sg_11 = ([]: TERM list, ⊢ ¬ (A a B ∧ A e B)⊥);  
| val sg_12 = ([]: TERM list, ⊢ A i B ∨ A u B⊥);  
| val sg_13 = ([⊢ A a B⊥], ⊢ A i B⊥);  
| val sg_14 = ([⊢ A e B⊥], ⊢ A u B⊥);
```

SML

```
| val sg_09_thm = save_thm ("sg_09_thm", syll_rule2 sg_09);  
| val sg_10_thm = save_thm ("sg_10_thm", syll_rule2 sg_10);  
| val sg_11_thm = save_thm ("sg_11_thm", syll_rule2 sg_11);  
| val sg_12_thm = save_thm ("sg_12_thm", syll_rule2 sg_12);  
| val sg_13_thm = save_thm ("sg_13_thm", syll_rule2 sg_13);  
| val sg_14_thm = save_thm ("sg_14_thm", syll_rule2 sg_14);
```

The Square of Opposition

4.6.4 The Valid Syllogisms

The valid syllogisms have been described in Section 4.2.1.

All twenty four syllogisms are true under this semantics and have been proven. The actual theorems are shown in the theory listing in Appendix A.5.

To implement a mapping corresponding to the above semantics we must create a matching *mapkit* as follows:

```
SML
|val mods_mapkit:mapkit = mkSimpMapkit [Term3] [⌈$a⌋, ⌈$e⌋, ⌈$i⌋, ⌈$u⌋];
```

4.6.5 Proving the Syllogisms

The resulting translation yields goals which look exactly like the previous versions but have the meanings defined in this context. They are proven and stored in the theory listing (see Appendix A.5) by the following:

```
SML
|proveSylls mods_mapkit syll_tac2 "" (syllogism_data1 @ syllogism_data2 @ syllogism_data3);
```

4.7 Existential Import in Universals

A more complete description of this interpretation, which is taken from Strawson [11] though not endorsed by him, is that existential import in relation both to the subject and the complement of the predicate, is present in universals and is absent from particular assertions. This combination is though by Strawson to be unsatisfactory primarily I believe because of its poor correspondence with any plausible account of the ordinary usage of the terms involved, however it appears also to be in poor correspondence with Aristotle. The difficulties can be traced to Strawson's acceptance of all four obversions, which appear not to have been endorsed by Aristotle. These obversions lead to the equivalence of contrapositives, and create a symmetry between subject and predicate in consequence of which presuppositions or implications of non emptyness attach both to subject and predicate.

My aim here is to confirm what Strawson says about this, which is that it satisfies all 14 laws and 28 syllogisms. He writes as if it were the only interpretation of this kind (i.e. not involving existential presuppositions rather than implications) which meet this requirement, on which I am sceptical but have not come to a definite conclusion.

```
SML
|open_theory "aristotle";
|force_new_theory "syllog5";
```

4.7.1 Semantics

Aristotle's syllogistic logic is concerned with inferences between judgements considered as predications. A predication in Aristotle affirms a *predicate* of some *subject*, but by contrast with more recent notions of predication the subject need not be an individual, the kinds of things which appear as predicates may also appear as subjects, and the relationship expressed seems closer to a modern eye to set inclusion than to what we now regard as predication. Since subject and predicate are for

present purposes the same kind of thing, it is useful to have a name for that kind of thing, and I will use the name *property*.

There are four kinds of predication which we have here to account for, which we will do by offering definitions which provide a good model for syllogistic logic, i.e. one in which the syllogisms held to be true by Aristotle are in fact true. Before providing these definitions we must decide what kind of thing are the terms which are related by Aristotelian predication.

In HOL the most natural answer to this is “boolean valued functions” which are objects of type $\ulcorner 'a \rightarrow \text{BOOL} \urcorner$ for some type of individuals which we can leave open by using the type variable $\ulcorner 'a \urcorner$. This provides a simple model of Aristotle’s non-u-p syllogistic reasoning. Four of the syllogisms which Aristotle considered valid fail under this conception of predicate, because among the objects of type $\ulcorner 'a \rightarrow \text{BOOL} \urcorner$ is the function $\ulcorner \lambda x : 'a \bullet F \urcorner$ which corresponds to a predicate with empty extension and does not admit inference from the universal to the existential (unless the universal is interpreted specially).

SML

```
|declare_type_abbrev("Term2", [], \ulcorner 'a \rightarrow \text{BOOL} \urcorner);
```

4.7.2 Predication

“o” is already in use for functional composition, so we will use “u” instead and then use an alias to permit us to write this as “o” (type inference will usually resolve any ambiguity).

To render these in HOL we first declare the relevant letters as infix operators:

They predication operators are defined as follows:

SML

```
|declare_infix (300, "a");
|declare_infix (300, "e");
|declare_infix (300, "i");
|declare_infix (300, "u");
```

HOL Constant

```
|$a : Term2 \rightarrow Term2 \rightarrow \text{BOOL}
|-----
|\forall A B \bullet A a B \Leftrightarrow (\forall x \bullet A x \Rightarrow B x) \wedge (\exists x \bullet A x) \wedge (\exists x \bullet \neg B x)
```

HOL Constant

```
|$e : Term2 \rightarrow Term2 \rightarrow \text{BOOL}
|-----
|\forall A B \bullet A e B \Leftrightarrow (\forall x \bullet A x \Rightarrow \neg B x) \wedge (\exists x \bullet A x) \wedge (\exists x \bullet B x)
```

HOL Constant

```
|$i : Term2 \rightarrow Term2 \rightarrow \text{BOOL}
|-----
|\forall A B \bullet A i B \Leftrightarrow (\exists x \bullet A x \wedge B x) \vee \neg (\exists x \bullet A x) \vee \neg (\exists x \bullet B x)
```

HOL Constant

```
| $u : Term2 → Term2 → BOOL
```

```
| ∀A B• A u B ⇔ (∃x• A x ∧ ¬ B x) ∨ ¬ (∃x• A x) ∨ ¬ (∃x• ¬ B x)
```

SML

```
| declare_alias("o", ⌈$u⌋);
```

Note that as defined above these come in complementary pairs, a being the negation of o and e of i . If we had negation we could manage with just two predication operators.

4.7.3 The Laws of Immediate Inference

Though in the source of this kind of “literate script” are to be found the scripts for generating and checking the proofs of all the theorems presente, it will not be my practice to expose these scripts in the printed version of the document. These scripts are not usually intelligible other than in that intimate man-machine dialogue which they mediate, and sufficient knowledge for most purposes of the structure of the proof will be found in the detailed lemmas proven (since the level of proof automation is modest).

However, I will begin by exposing some of the scripts used for obtaining proofs of syllogisms in this model, to give the reader an impression of the level of complexity and kind of obscurity involved in this kind of formal work, I will not attempt sufficient explanation to make these scripts intelligible, they are best understood in the interactive environment, all the scripts are available for readers who want to run them.

Most readers are expected to skip over the gory details, the philosophical points at stake do not depend on the details of the proofs.

Before addressing the laws of immediate inference ⁶ I devise a tactic for automating simple proofs in this domain.

The following elementary tactic expands the goal by applying the definitions of the operators and then invokes a general tactic for the predicate calculus. A rule is also defined using that tactic for direct rather than interactive proof.

SML

```
| val syll_tac3 = asm_prove_tac (map get_spec [⌈$a⌋, ⌈$e⌋, ⌈$i⌋, ⌈$u⌋]);  
| fun syll_rule3 g = tac_proof (g, syll_tac3);
```

SML

```
| val e_conv_thm = save_thm("e_conv_thm", syll_rule3 ([⌈A e B⌋, ⌈B e A⌋]);  
| val i_conv_thm = save_thm("i_conv_thm", syll_rule3 ([⌈A i B⌋, ⌈B i A⌋]);
```

Simple Conversion The following two theorems are obtained.

```
| e_conv_thm = A e B ⊢ B e A  
| i_conv_thm = A i B ⊢ B i A
```

⁶in which I followed Strawson [11], though I can now cite [Aristotle, Prior Analytic, Book 1, Part 2](#). [1]

The following two theorems show that the other obvious conversions are also false. Note that the theorems are not polymorphic, they are proven specifically for terms of type $\mathbb{N} \rightarrow \text{BOOL}$ (though any type of more than one element would do).

```
|¬a_conv_thm = ⊢ ∃ A B • A a B ∧ ¬ B a A
|¬o_conv_thm = ⊢ ∃ A B • A o B ∧ ¬ B o A
```

Conversion Per Accidens These come out OK.

SML

```
|val ai_conv_thm = save_thm ("ai_conv_thm", syll_rule3([⊢ A a B⊣, ⊢ B i A⊣]);
|val eo_conv_thm = save_thm ("eo_conv_thm", syll_rule3([⊢ A e B⊣, ⊢ B u A⊣]);
```

```
|val ai_conv_thm = A a B ⊢ B i A : THM
|val eo_conv_thm = A e B ⊢ B o A : THM
```

Obversion For these we need to define an operation of complementation on terms.

HOL Constant

```
| Complement : Term2 → Term2
```

```
| ∀A α • (Complement A) α ⇔ ¬ (A α)
```

We will use “ \sim ” as a shorthand for “Complement”.

SML

```
|declare_alias ("~", ⊢ Complement⊣);
```

```
|ae_obv_thm = A a B ⊢ A e ~ B
|ea_obv_thm = A e B ⊢ A a ~ B
|io_obv_thm = A i B ⊢ A o ~ B
|oi_obv_thm = A o B ⊢ A i ~ B
```

Contraposition and Inversion

4.7.4 The Square of Opposition

This is complete with this semantics.

```
|ao_contrad_thm = ⊢ A a B ⇔ ¬ A o B
|ei_contrad_thm = ⊢ A e B ⇔ ¬ A i B
|ae_contrar_thm = ⊢ ¬ (A a B ∧ A e B)
|io_subcont_thm = ⊢ A i B ∨ A o B
|ai_subalt_thm = ⊢ A a B ⇒ A i B
|eo_subalt_thm = ⊢ A e B ⇒ A o B
```

4.7.5 The Syllogisms

First we make a *mapkit*.

SML

```
|val s3mapkit:mapkit = mkSimpMapkit [Term2] [a, e, i, u];
```

Then we apply this in generating and proving the syllogisms.

SML

```
|proveGoals syll_tac3 "" (mkGoals s3mapkit syllogism_data1);
|proveGoals syll_tac3 "" (mkGoals s3mapkit syllogism_data2);
|proveGoals syll_tac3 "" (mkGoals s3mapkit syllogism_data3);
```

Three of the 24 generally accepted syllogisms prove unsound under this semantics: Camenes, Dimaris and Fresison.

```
|¬_Fresison = ⊢ ¬ (∀ P M S • P e M ∧ M i S ⇒ S o P)
|¬_Dimaris  = ⊢ ¬ (∀ P M S • P i M ∧ M a S ⇒ S i P)
|¬_Camenes  = ⊢ ¬ (∀ P M S • P a M ∧ M e S ⇒ S e P)
```

for the record the counterexamples which disprove these syllogisms are all combinations of universal (U) and empty (E) terms as follows.

	<i>P</i>	<i>M</i>	<i>S</i>
<i>Fresison</i>	<i>U</i>	<i>E</i>	<i>U</i>
<i>Dimaris</i>	<i>E</i>	<i>U</i>	<i>U</i>
<i>Camenes</i>	<i>U</i>	<i>U</i>	<i>E</i>

The theorems are also displayed in the theory listing in Appendix [A.6](#)

4.8 Existential Import in Affirmations

This is my present best attempt at an interpretation which correponds closely to Aristotle.

SML

```
|open_theory "aristotle";
|force_new_theory "syllog6";
```

4.8.1 Semantics

Aristotle's syllogistic logic is concerned with inferences between judgements considered as predications. A predication in Aristotle affirms a *predicate* of some *subject*, but by contrast with more recent notions of predication the subject need not be an individual, the kinds of things which appear as predicates may also appear as subjects, and the relationship expressed seems closer to a modern eye to set inclusion than to what we now regard as predication. Since subject and predicate are for present purposes the same kind of thing, it is useful to have a name for that kind of thing, and I will use the name *property*.

There are four kinds of predication which we have here to account for, which we will do by offering definitions which provide a good model for syllogistic logic, i.e. one in which the syllogisms held to

be true by Aristotle are in fact true. Before providing these definitions we must decide what kind of thing are the terms which are related by Aristotelian predication.

In HOL the most natural answer to this is “boolean valued functions” which are objects of type $\ulcorner 'a \rightarrow \text{BOOL} \urcorner$ for some type of individuals which we can leave open by using the type variable $\ulcorner 'a \urcorner$. This provides a simple model of Aristotle’s non-u-p syllogistic reasoning. Four of the syllogisms which Aristotle considered valid fail under this conception of predicate, because among the objects of type $\ulcorner 'a \rightarrow \text{BOOL} \urcorner$ is the function $\ulcorner \lambda x : 'a \bullet F \urcorner$ which corresponds to a predicate with empty extension and does not admit inference from the universal to the existential (unless the universal is interpreted specially).

SML

```
|declare_type_abbrev("Term2", [],  $\ulcorner 'a \rightarrow \text{BOOL} \urcorner$ );
```

4.8.2 Predication

“o” is already in use for functional composition, so we will use “u” instead and then use an alias to permit us to write this as “o” (type inference will usually resolve any ambiguity).

To render these in HOL we first declare the relevant letters as infix operators:

They predication operators are defined as follows:

SML

```
|declare_infix (300, "a");
|declare_infix (300, "e");
|declare_infix (300, "i");
|declare_infix (300, "u");
```

HOL Constant

```
|$a : Term2  $\rightarrow$  Term2  $\rightarrow$  BOOL
|-----
| $\forall A B \bullet A a B \Leftrightarrow (\forall x \bullet A x \Rightarrow B x) \wedge \exists x \bullet A x$ 
```

HOL Constant

```
|$e : Term2  $\rightarrow$  Term2  $\rightarrow$  BOOL
|-----
| $\forall A B \bullet A e B \Leftrightarrow (\forall x \bullet A x \Rightarrow \neg B x)$ 
```

HOL Constant

```
|$i : Term2  $\rightarrow$  Term2  $\rightarrow$  BOOL
|-----
| $\forall A B \bullet A i B \Leftrightarrow (\exists x \bullet A x \wedge B x)$ 
```

HOL Constant

```
|$u : Term2  $\rightarrow$  Term2  $\rightarrow$  BOOL
|-----
| $\forall A B \bullet A u B \Leftrightarrow (\exists x \bullet A x \wedge \neg B x) \vee \neg (\exists x \bullet A x)$ 
```

SML

```
|declare_alias("o", "⌈$u⌋");
```

Note that as defined above these come in complementary pairs, a being the negation of o and e of i . If we had negation we could manage with just two predication operators.

4.8.3 The Laws of Immediate Inference

Though in the source of this kind of “literate script” are to be found the scripts for generating and checking the proofs of all the theorems presente, it will not be my practice to expose these scripts in the printed version of the document. These scripts are not usually intelligible other than in that intimate man-machine dialogue which they mediate, and sufficient knowledge for most purposes of the structure of the proof will be found in the detailed lemmas proven (since the level of proof automation is modest).

However, I will begin by exposing some of the scripts used for obtaining proofs of syllogisms in this model, to give the reader an impression of the level of complexity and kind of obscurity involved in this kind of formal work, I will not attempt sufficient explanation to make these scripts intelligible, they are best understood in the interactive environment, all the scripts are available for readers who want to run them.

Most readers are expected to skip over the gory details, the philosophical points at stake do not depend on the details of the proofs.

Before addressing the laws of immediate inference ⁷ I devise a tactic for automating simple proofs in this domain.

The following elementary tactic expands the goal by applying the definitions of the operators and then invokes a general tactic for the predicate calculus. A rule is also defined using that tactic for direct rather than interactive proof.

SML

```
|val syll_tac6 = asm_prove_tac (map get_spec [⌈$a⌋, ⌈$e⌋, ⌈$i⌋, ⌈$u⌋]);  
|fun syll_rule6 g = tac_proof (g, syll_tac6);  
|val syll_tac6b = REPEAT (POP_ASM_T ante_tac)  
|   THEN rewrite_tac (map get_spec [⌈$a⌋, ⌈$e⌋, ⌈$i⌋, ⌈$u⌋])  
|   THEN contr_tac THEN asm_fc_tac[];  
|fun syll_rule6b g = tac_proof (g, syll_tac6b);
```

SML

```
|val e_conv_thm = save_thm ("e_conv_thm", syll_rule6([⌈A e B⌋, ⌈B e A⌋]);  
|val i_conv_thm = save_thm ("i_conv_thm", syll_rule6([⌈A i B⌋, ⌈B i A⌋]);  
  
|val e_conv_thm = A e B ⊢ B e A : THM  
|val i_conv_thm = A i B ⊢ B i A : THM
```

Simple Conversion

⁷in which I followed Strawson [11], though I can now cite [Aristotle, Prior Analytic, Book 1, Part 2](#). [1]

Conversion Per Accidens These are OK.

SML

```
| val ai_conv_thm = save_thm ("ai_conv_thm", syll_rule6(⟦A a B⟧, ⟦B i A⟧));
| val eo_conv_thm = save_thm ("eo_conv_thm", syll_rule6(⟦A e B⟧, ⟦B u A⟧));
```

```
| val ai_conv_thm = A a B ⊢ B i A : THM
| val eo_conv_thm = A e B ⊢ B o A : THM
```

Obversion For these we need to define an operation of complementation on terms.

HOL Constant

```
| Complement : Term2 → Term2
```

```
| ∀A α • (Complement A) α ⇔ ¬ (A α)
```

We will use “~” as a shorthand for “Complement”.

SML

```
| declare_alias ("~", ⟦Complement⟧);
```

Only two of the obversions are valid.

```
| val ae_obv_thm = A a B ⊢ A e ~ B : THM
| val iu_obv_thm = A i B ⊢ A o ~ B : THM
```

4.8.4 The Square of Opposition

This is complete with this semantics.

```
| ao_contrad_thm = ⊢ A a B ⇔ ¬ A o B
| ei_contrad_thm = ⊢ A e B ⇔ ¬ A i B
| ae_contrar_thm = ⊢ ¬ (A a B ∧ A e B)
| io_subcont_thm = ⊢ A i B ∨ A o B
| ai_subalt_thm = ⊢ A a B ⇒ A i B
| eo_subalt_thm = ⊢ A e B ⇒ A o B
```

4.8.5 The Syllogisms

First we make a *mapkit*.

SML

```
| val s6mapkit:mapkit = mkSimpMapkit ⟦:Term2⟧ [⟦$a⟧,⟦$e⟧,⟦$i⟧,⟦$u⟧];
```

Then we apply this in generating and proving the syllogisms.

SML

```
| proveGoals syll_tac6 "" (mkGoals s6mapkit syllogism_data1);
| proveGoals syll_tac6 "" (mkGoals s6mapkit syllogism_data2);
| proveGoals syll_tac6 "" (mkGoals s6mapkit syllogism_data3);
```

The theorems are also displayed in the theory listing in [Appendix A.7](#)

4.9 Modal Syllogisms

The language of syllogistic logic does not have operators over propositions. The only operators are the ones which apply predicates to subjects.

The modalities are perhaps therefore better thought of as kinds of judgements rather than as operations on propositions. This would give us three kinds of judgement, which assert a predication contingently, necessarily or possibly.

It is natural to consider the modal aspects in terms of possible worlds, and I will model it first in those terms (not knowing whether this will provide a good model of Aristotle's conception of modality). The propositional functions could then be modelled as functions from possible worlds to non-empty propositional functions.

The following first attempt at a modal syllogism is poorly representative of Aristotle, since, it appears, considered two definitions of possibility in terms of necessity, but based his work on modal syllogisms primarily on one which does not correspond to the modern usage which is implicit in the treatment given here. It is to be expected therefore that any examination of the results obtained here will differ from Aristotle's views substantially because of this different conception of necessity. In due course I will add another kind of modal judgement which will be closer to the one principally investigated by Aristotle.

Aristotle's principle definition of 'possibly P' was:

not necessarily P and not necessarily not P

This corresponds more closely with the concept 'contingent' or its modern counterpart, 'synthetic' than with contemporary usage of 'possible', though the contemporary rendering which corresponds to Aristotle's other notion of possibility is a pre-Kripkean one in which we assume a fixed set of possible worlds with modal operators quantifying over the whole.

I adapt the treatment of u-p syllogisms by treating predicates as parameterised by a possible world.

Rather than using a type variable (which is what I did for the two preceding treatments) I will use two new type constants for individual substances and possible worlds.

It may suffice for the reader to refer directly to the theory listing in Appendix [A.8](#).

SML

```
|open_theory "aristotle";  
|force_new_theory "modsyllog";
```

4.9.1 Semantics

The complexity required in the semantics of modal operators depends upon other features of the language in which they occur. Because the language of the syllogism is very simple, having neither propositional operators nor variables for individuals a very simple semantics may suffice. When we come to consider the metaphysics there will be some increase in the complexity of the other features of the language, and also a greater premium on getting the semantics to correspond intuitively with the content of the metaphysics, but at this stage we will adopt the simplest semantic model which seems likely to secure the results expressible in our restricted language.

So are now talking about predicates parameterised by possible worlds. Furthermore, we will model this with a fixed set to individuals, independent of the possible world. Possible worlds differ only in the extension of predicates.

First some new types, “I” for individual substances, “W” for possible worlds:

SML

```
|new_type ("I",0);
|new_type ("W",0);
```

Then a type abbreviation for the predicates:

SML

```
|declare_type_abbrev("MPROP", [],  $\ulcorner : W \rightarrow I \times (I \rightarrow \text{BOOL}) \urcorner$ );
```

4.9.2 Predication

To work with this new type for the predicates we define a function which will convert this kind of predicate into the old kind, as follows:

HOL Constant

```
|  $p : \text{MPROP} \rightarrow (W \rightarrow I \rightarrow \text{BOOL})$ 
|-----
|  $\forall A \bullet p A = \lambda w x \bullet \text{let } (v,f) = A w \text{ in } x = v \vee f x$ 
```

The following principle can be proven (proof omitted):

```
|  $p\_exists\_lemma =$ 
|  $\ulcorner \forall A w \bullet \exists v \bullet p A w v$ 
```

This principle is what we need to prove the u-p syllogisms.

We then proceed in a similar manner to the other models, using the function p to convert the new kind of predicate into the old.

The predication operators are then defined. Note that the differences are small and uniform. The type $\ulcorner : \text{Term3} \urcorner$ is changed to $\ulcorner : \text{MPROP} \urcorner$ and the function p is invoked before applying a predicate.

Now we think of a predication as being a set of possible worlds, or Boolean valued function over possible worlds.

SML

```
|declare_infix (300, "a");
|declare_infix (300, "e");
|declare_infix (300, "i");
|declare_infix (300, "u");
```

HOL Constant

```
|  $\$a : \text{MPROP} \rightarrow \text{MPROP} \rightarrow W \rightarrow \text{BOOL}$ 
|-----
|  $\forall A B w \bullet (A a B) w \Leftrightarrow \forall x \bullet p A w x \Rightarrow p B w x$ 
```

HOL Constant

$\$e : MPROP \rightarrow MPROP \rightarrow W \rightarrow BOOL$

$\forall A B w \bullet (A e B) w \Leftrightarrow \forall x \bullet p A w x \Rightarrow \neg p B w x$

HOL Constant

$\$i : MPROP \rightarrow MPROP \rightarrow W \rightarrow BOOL$

$\forall A B w \bullet (A i B) w \Leftrightarrow \exists x \bullet p A w x \wedge p B w x$

HOL Constant

$\$u : MPROP \rightarrow MPROP \rightarrow W \rightarrow BOOL$

$\forall A B w \bullet (A u B) w \Leftrightarrow \exists x \bullet p A w x \wedge \neg p B w x$

SML

`declare_alias("o", "⌈$u⌋");`

We now have to define some additional constants for the forms of judgement, which will assert the predications either of the actual world or of some or all possible worlds.

First I define a constant (rather loosely) to be the actual world:

HOL Constant

$\mathbf{actual_world} : W$

T

Then the two modal judgement forms:

HOL Constant

$\diamond : (W \rightarrow BOOL) \rightarrow BOOL$

$\forall s \bullet \diamond s \Leftrightarrow \exists w \bullet s w$

HOL Constant

$\square : (W \rightarrow BOOL) \rightarrow BOOL$

$\forall s \bullet \square s \Leftrightarrow \forall w \bullet s w$

Aristotle's other notion of possibility is:

HOL Constant

$\diamond_a : (W \rightarrow BOOL) \rightarrow BOOL$

$\forall s \bullet \diamond_a s \Leftrightarrow \neg (\forall w \bullet s w) \wedge \neg (\forall w \bullet \neg s w)$

Finally the non-modal judgements also need a judgement forming constant.

HOL Constant

$$\begin{array}{|l} \vdash : (W \rightarrow \text{BOOL}) \rightarrow \text{BOOL} \\ \hline \forall s \bullet \vdash s \Leftrightarrow s \text{ actual_world} \end{array}$$

Special difficulties are raised by reasoning with \diamond_a and to help with these difficulties it is useful to have negation as a kind of propositional operator in this modal logic.

SML

`declare_prefix (350, "¬m");`

HOL Constant

$$\begin{array}{|l} \$\neg_m : (W \rightarrow \text{BOOL}) \rightarrow (W \rightarrow \text{BOOL}) \\ \hline \forall x \bullet \neg_m x = \lambda w \bullet \neg (x w) \end{array}$$

Though this constant is distinct from the non-modal negation, we might as well drop the subscript where no ambiguity arises.

SML

`declare_alias ("¬", "[$¬m]");`

4.9.3 Laws of Immediate Inference

Before looking at the conversions there are some general rules which may be helpful for us though these probably are not in Aristotle.

$$\begin{array}{|l} \Box \diamond_thm = \Box X \vdash \diamond X \\ \Box \vdash_thm = \Box X \vdash \vdash X \\ \vdash \diamond_thm = \vdash X \vdash \diamond X \\ \diamond_a \diamond_thm = \diamond_a X \vdash \diamond X \\ \diamond_a \neg \Box_thm = \diamond_a X \vdash \neg \Box X \\ \diamond_a \neg_m_thm = \vdash \diamond_a X \Leftrightarrow \diamond_a (\neg_m X) \end{array}$$

$$\begin{array}{|l} e_conv_thm = \vdash (A e B) \vdash \vdash (B e A) \\ i_conv_thm = \vdash (A i B) \vdash \vdash (B i A) \\ \Box e_conv_thm = \Box (A e B) \vdash \Box (B e A) \\ \Box i_conv_thm = \Box (A i B) \vdash \Box (B i A) \\ \diamond e_conv_thm = \diamond (A e B) \vdash \diamond (B e A) \\ \diamond i_conv_thm = \diamond (A i B) \vdash \diamond (B i A) \\ \diamond_a e_conv_thm = \diamond_a (A e B) \vdash \diamond_a (B e A) \\ \diamond_a i_conv_thm = \diamond_a (A i B) \vdash \diamond_a (B i A) \\ \diamond_a ao_conv_thm = \vdash \diamond_a (A a B) \Leftrightarrow \diamond_a (A o B) \\ \diamond_a ei_conv_thm = \vdash \diamond_a (A e B) \Leftrightarrow \diamond_a (A i B) \\ \Box \diamond e_conv_thm = \Box (A e B) \vdash \diamond (B e A) \\ \Box \vdash e_conv_thm = \Box (A e B) \vdash \vdash (B e A) \end{array}$$

Simple Conversion In this version of the semantics, “a” and “o” conversion is neither provable nor refutable. In the previous version (the one not admitting the u-p syllogisms), since the universe is a HOL type there is at least one individual, and contradictory predicates are allowed, we can use these two to disprove the two conversions. With this semantics there is no empty predicate, and we cannot know that there are two distinct predicates.

```

| $\vDash ai\_conv\_thm = \vDash (A a B) \vdash \vDash (B i A)$ 
| $\vDash eo\_conv\_thm = \vDash (A e B) \vdash \vDash (B o A)$ 
| $\Box \diamond ai\_conv\_thm = \Box (A a B) \vdash \diamond (B i A)$ 
| $\Box \diamond eo\_conv\_thm = \Box (A e B) \vdash \diamond (B o A)$ 

```

Conversion Per Accidens

4.9.4 The Valid Modal Syllogisms

The valid syllogisms have been described in Section 4.2.1.

All nineteen syllogisms supposed valid by Aristotle are true under this semantics and have been proven. A further five⁸ have also been proven, giving a total of 24. When combinations of modal operators are added to this the number gets quite large, so, rather than proving all the valid cases I will prove sufficient to enable the rest to be automatically proven.

This will involve some theorems which are not strictly syllogistic.

The actual theorems proved are shown in the theory listing in Appendix A.8.

Because of the modal operators the generation of the syllogisms is more complicated. The generation functions are adapted to allow a single modal operator to be applied to each of the premises and the conclusion.

SML

```

|fun mk_pred q s p = mk_app(mk_app (q, s), p);
|
|fun mk_syll vt (a,b,c,d) (q1, q2, q3) =
|   ([mk_pred q1 a b, mk_pred q2 c d],
|     mk_pred q3 (mk_var("S", vt)) (mk_var("P", vt)));
|
|fun mk_relt t = mk_ctype ("→", [t, mk_ctype ("→", [t,  $\ulcorner$ BOOL $\urcorner$ ]]));
|
|fun mk_syllp vt (s, n) =
|   mk_syll vt (nth (n-1) (figurest vt)) (optrip_from_text (mk_relt vt) s);

```

⁸Which I got from Strawson [11].

SML

```
|fun syll_prove msp suff tac (a,n) =  
|  let val thm = tac_proof (msp (a,n), tac) handle _ => t_thm  
|  in (concat [a, suff], thm)  
|  end;  
  
|fun syll_prove_and_store msp suff tac (a,n) =  
|  let val res = syll_prove msp suff tac (a,n);  
|    val _ = save_thm res  
|  in res  
|  end;
```

SML

```
|fun map_goal f (st, t) = (map f st, f t);  
  
|fun mk_modt vt = mk_ctype ("→", [vt,  
|  mk_ctype ("→", [vt, (mk_ctype ("→", [⊢:W⊢, ⊢:BOOL⊢]))]));  
  
|fun mk_modsyll vt (s, n) =  
|  mk_syll vt (nth (n-1) (figurest vt)) (optrip_from_text (mk_modt vt) s);  
  
|fun modgoal (mo1, mo2, mo3) ([p1,p2], c) =  
|  ([mk_app (mo1, p1), mk_app (mo2, p2)], mk_app (mo3, c));  
  
|fun mk_modsyllp mot p = modgoal mot (mk_modsyll ⊢:MPROP⊢ p);
```

This defines the function *mk_modsyllp* whose type is shown:

```
|val mk_modsyllp = fn: TERM * TERM * TERM -> string * int -> TERM list * TERM
```

in which the *TERM* parameters are modal operators the next argument is a pair consisting of a string which is the name of a syllogism and a number which is the number of the figure. The result is a goal for proof.

An example of its use is:

SML

```
|mk_modsyllp (⊢□⊢, ⊢≡⊢, ⊢◇⊢) ("Barbara", 1);
```

which yields:

```
|val it = ([⊢□ (M a P)⊢, ⊢□ (S a M)⊢, ⊢□ (S a P)⊢) : TERM list * TERM
```

4.9.5 General Results

The logic of the modal operators is completely independent of the logic of the syllogism. The relevant results can be stated and proven in HOL concisely, but these statements are not in the language of the syllogism.

There are in effect just seven modal truths, each of which appears in 24 forms, one for each of the valid non-modal syllogisms.

Rather than proving all 192 theorems (counting the non-modal truths in this modal model), I prove the eight proformas expressed in HOL. From these eight all 192 theorems can be obtained by proving (a special form of) one of the valid syllogisms and instantiating one of the general modal rules using it.

I omit the details of the metalanguage scripts which automate all this.

The following lists the valid modal forms. In each tuple the three entries give the modalities of the first and second premise and the conclusion respectively. Taking any valid syllogism and applying modal operators using one of the patterns in this table will give a valid modal syllogism.

SML

```
val mod_gen_params =
  [( $\Box$ ,  $\Box$ ,  $\Box$ ),
   ( $\Box$ ,  $\Box$ ,  $\Diamond$ ),
   ( $\Box$ ,  $\Box$ ,  $\neg$ ),
   ( $\Box$ ,  $\Diamond$ ,  $\Diamond$ ),
   ( $\Box$ ,  $\Diamond$ ,  $\neg$ ),
   ( $\Box$ ,  $\neg$ ,  $\neg$ ),
   ( $\neg$ ,  $\Box$ ,  $\neg$ ),
   ( $\neg$ ,  $\neg$ ,  $\neg$ )];
```

The set of general HOL theorems which facilitate the proofs of these modal syllogisms is as follows:

ProofPower Theorems

```
val mod_gen_thms =
  [ $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \Box CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \Diamond CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \neg CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Diamond FP \wedge \Box SP \Rightarrow \Diamond CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Diamond SP \Rightarrow \Diamond CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \neg SP \Rightarrow \neg CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \neg FP \wedge \Box SP \Rightarrow \neg CS$ ,
    $\vdash \forall FP SP CS \bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \neg FP \wedge \neg SP \Rightarrow \neg CS$ ]
: THM list
```

In the above theorems the variables FP , SP , CS , stand respectively for *first premise*, *second premise*, *conclusion of syllogism* and range over modal propositions (which have type $\Box : W \rightarrow \text{BOOL}$).

4.9.6 Proving the Syllogisms

I then prove the 24 non-modal syllogisms in the required form and infer forward using the above 8 theorems to obtain a total of 192 theorems true in this model of the modal syllogism.

Details of scripts omitted.

The automated proof the yields the expected 192 modal syllogisms, of which we display only the first few (and do not save them in the theory):

```

| val valid_G_modsylls =
|    $\vDash (P a M), \vDash (M e S) \vdash \vDash (\$ o P), \vDash (P a M), \Box (M e S) \vdash \vDash (\$ o P),$ 
|      $\Box (P a M), \vDash (M e S) \vdash \vDash (\$ o P), \Box (P a M), \Diamond (M e S) \vdash \Diamond (\$ o P),$ 
|      $\Diamond (P a M), \Box (M e S) \vdash \Diamond (\$ o P), \Box (P a M), \Box (M e S) \vdash \vDash (\$ o P),$ 
|      $\Box (P a M), \Box (M e S) \vdash \Diamond (\$ o P), \Box (P a M), \Box (M e S) \vdash \Box (\$ o P),$ 
|      $\vDash (P a M), \vDash (\$ e M) \vdash \vDash (\$ o P), \vDash (P a M), \Box (\$ e M) \vdash \vDash (\$ o P),$ 
|      $\Box (P a M), \vDash (\$ e M) \vdash \vDash (\$ o P), \Box (P a M), \Diamond (\$ e M) \vdash \Diamond (\$ o P),$ 
|      $\Diamond (P a M), \Box (\$ e M) \vdash \Diamond (\$ o P), \Box (P a M), \Box (\$ e M) \vdash \vDash (\$ o P),$ 
|      $\Box (P a M), \Box (\$ e M) \vdash \Diamond (\$ o P), \Box (P a M), \Box (\$ e M) \vdash \Box (\$ o P),$ 
|      $\vDash (P e M), \vDash (\$ a M) \vdash \vDash (\$ o P), \vDash (P e M), \Box (\$ a M) \vdash \vDash (\$ o P),$ 
|      $\Box (P e M), \vDash (\$ a M) \vdash \vDash (\$ o P), \Box (P e M), \Diamond (\$ a M) \vdash \Diamond (\$ o P),$ 
| ...
|
| SML
| length valid_G_modsylls;
|
| val it = 192 : int

```

4.10 Demonstrative Truth

An important part of Aristotle’s philosophy is his concept of demonstrative science.

A proof is demonstrative if it proceeds from first principles and is deductively sound. Truths established in this way are necessary because the first principles must be essential and hence necessary and sound deduction preserves necessity.

If we understand Hume’s “intuitively certain” as a reference to the criteria for Aristotle’s first principles, and understand Hume as using the term ‘demonstrative’ in the same sense as Aristotle, then it is plausible that Hume’s “truths of reason” are the same as Aristotle’s truths of demonstrative science.

It is tempting to use the word demonstrative for ‘truths of reason’ though in Aristotle and Hume the first principles do not count as demonstrable. This follows modern logic in using concepts such as ‘theorem’ and ‘valid sentence’ which apply to logical axioms as well as results deduced from them. It is tempting also to identify these concepts with the concept of analyticity. This last point is aided by the connection in Aristotle between essential truth (which must be possessed by the first principles) and definition, which seems close, and which distinguishes his accidental predications from essential predications. A weak point in this is the question of the first principles of the various sciences, and the doubts one can reasonably have about whether Aristotle’s essential truth must be analytic.

However, we now expect deductive systems to be incomplete, and hence that not all analytic propositions are provable. However, the incompleteness may arise from adopting a fixed set of first principles, rather from incompleteness of deduction. In this case, the availability of an open set of first principles makes completeness in principle possible. To sustain this principle in relation to set theory, for example, we would have to regard ourselves as having some definition of the concept of set relative to which the present axioms (say those of ZFC) are essential, and relative to which extensions as necessary to prove progressively more difficult results can also be seen to be essential. This is not entirely implausible. This is close to the rationale for large cardinal axioms. The informal description of the cumulative hierarchy as the domain of set theory involves the idea that the construction of

well-founded sets from other well-founded sets of lesser rank proceeds indefinitely, and hence any axiom which states that a set of a certain rank exists must be true.

Does formal modelling contribute anything to this discussion?

The above discussion involves ideas which belong to Aristotle's metaphysics rather than his logic. So a fuller formal analysis of these ideas will have to wait until we get to the *Metaphysics*.

However some aspects may be considered here. For example, we need to know that from necessary premises only necessary premises are derivable by syllogisms.

5 Metaphysics (II)

In this section I offer a single model integrating Modal Syllogisms with the distinction between essential and accidental predication.

My interest is primarily in the extent to which may be found in Aristotle's philosophy a precursor of Hume's fork or the modern distinctions between analytic and synthetic or necessary and contingent propositions. I see three Aristotelian ideas which have some relevance.

- the distinction between necessary and contingent propositions
- the distinction between essential and accidental predication
- the notion of demonstrability

It is only when we combine the syllogism with the metaphysics that we can explore the relationship between these various concepts.

I will give higher priority in this model to good structure while remaining faithful to Aristotle. It is not the purpose of this model to further investigate the position in relation to Aristotle of Grice, Codd or Speranza.

I have constructed this model to give a good correspondence between necessary truth and essential predication. If the model is successful in that respect it remains to consider whether it is consistent with the philosophy of Aristotle. I do not know whether Aristotle talked about the relationship between essence and necessity.

I also hope that the model may help to explore the question of whether demonstrable truth, or rather the truths which are either "intuitively or demonstrably certain" to use Hume's words, coincides with necessary truth.

SML

```
|open_theory "aristotle";  
|force_new_theory "syllmetap";
```

5.1 Semantics

In my first metaphysical model the main question in relations to subject matter was "what are subjects and predicates", to which a model of Aristotelian categories gives an answer. The introduction of modality makes it necessary to consider something like possible worlds. These were left

uninterpreted in the modal treatment of the syllogism, but now that we expose the distinction between essential and accidental predication is it desirable to identify possible worlds with that which is accidental.

What is accidental is the extension of individual attributes, and this gives our concept of possible world.

It is convenient at this point to consider the question of extensionality.

According to Grice/Codd/Speranza, essential predicates are extensional:

$$A \text{ izz } B \wedge B \text{ izz } A \Rightarrow A = B$$

but I know no reason to suppose that accidental predication is, and it seems counter-intuitive that it should be. Consequently the modelling of an accidental predicate using a BOOLEAN valued function in HOL (in which functions are extensional) is inappropriate. So I will separate out the extension from the individual attribute.

Individuals belong to categories, and are collected together in groups which determine the nature of essential predication within that category. The simplest way of getting the right structure is to use some type of tags to differentiate individuals within a category, using the same collection of tags in each category (with the unintended effect of ensuring that each category has the same number of individuals, which I hope will have a significant effect on the resulting theory). The individuals are then represented by an ordered pair consisting of a category and a tag.

For most purposes the number of categories is not important, though we must have a category of substances, so a completely undifferentiated new type might have sufficed. However, it turns out that some things don't work, and its useful to have at least one non-substantial category in order to prove that they don't work. So I introduce a new type of non-substantial (attribute) categories (so we get at least one) and then make the type of categories by adding one more.

SML

```
|new_type ("ACAT", 0);
|new_type ("TAG", 0);
```

SML

```
|declare_type_abbrev("CAT", [], ⌈:ONE+ACAT⌋);
```

One of the categories will be the category of substance, it doesn't matter which one but we might as well use the odd One on the left of the sum (so you can test for substance using *IsL*).

HOL Constant

```
| Category_of_Substance : CAT
|-----
| Category_of_Substance = InL One
```

An individual will therefore be modelled as an ordered pair consisting of a tag and a category. This is captured by the following type abbreviation.

SML

```
|declare_type_abbrev ("I", [], ⌈:CAT × TAG⌋);
```

A possible world is then an assignment of extensions to individual attributes, where an extension is a set of particulars. Since we only want individual attributes, we do not use type *I* for the domain.

Particulars always belong to the category of substance, so we only need a set of tags in the result, the category is implicit.

SML

```
|declare_type_abbrev ("W", [],  $\ulcorner$ :ACAT  $\times$  TAG  $\rightarrow$  TAG SET $\urcorner$ );
```

This does include an assignement of extensions to particulars, but this plays no role, only intefering with the identity criteria for possible worlds, which do not feature in the theory.

I need to distinguish one possible world which is the actual world:

HOL Constant

```
|actual_world : W
```

```
|T
```

Since the individuals are pairs it might be handy to have appropriately named projection funtions which extract the two components:

HOL Constant

```
|category : I  $\rightarrow$  CAT;
```

```
|tag : I  $\rightarrow$  TAG
```

```
| $\forall$ ct • category ct = Fst ct
```

```
| $\wedge$  tag ct = Snd ct
```

Finally the question of what subjects and predicates are can be determined. I will call them *TermMs* and they are either a set of particulars or a set of attributes. To allow for complementation I add a boolean component, which if true indicates a complement.

The sets in this case must be non-empty if we are to retain the u-p syllogisms (in default of a different universal). The method used in my model of the modal syllogism in Section 4.9 will not do here, because (at least according to Code) we need an extensionality result, so I have instead introduced ⁹ for this purpose a new type (*NESET*) of non-empty sets which will give us both the u-p syllogisms and extensionality (for essential predication, not for accidental predication).

A term is therefore modelled as either a non-empty set of individual substances or a non-empty set of individual attributes.

SML

```
|declare_type_abbrev ("TermM", [],  $\ulcorner$ :(CAT  $\times$  TAG NESET) $\urcorner$ );
```

HOL Constant

```
|mk_SubstTerm : TAG  $\mathbb{P}$   $\rightarrow$  TermM
```

```
| $\forall$ s • mk_SubstTerm s = (Category_of_Substance, NeSet s)
```

HOL Constant

```
|mk_AttrTerm : CAT  $\times$  TAG  $\mathbb{P}$   $\rightarrow$  TermM
```

```
| $\forall$ s • mk_AttrTerm s = (Fst s, NeSet (Snd s))
```

⁹The definition has been placed in a separate document, [7].

It may be useful to have a name for the predicate encompassing all substance.

HOL Constant

Substance : *TermM*

Substance = *mk_SubsTerm Universe*

5.2 Predication

The syllogism comes with four kinds of predication (a, e, i, o), and the metaphysics with three (izz, hazz and izz or hazz). Combining these would give twelve combinations.

To simplify a bit I will separate out the quantifier but defining *All* and *Some* appropriately, and provide a postfix negator for izz an hazz.

I will then treat the modal operators as operators over propositions, and introduce the syllogism as a kind of judgement.

The type of the primitive copulas is:

SML

`|declare_type_abbrev("COPULA", [], [⌈:I → TermM → (W → BOOL)⌋]);`

The first parameter is an individual substance or attribute rather than a TermM, the quantifying operator will arrange for each of the relevant individuals or attributes to be supplied.

SML

`|declare_type_abbrev ("MPROP", [], [⌈:W → BOOL⌋]);`

Propositions

Complementation The distinction between affirmative and negative is achieved by a postfix negation so we can say “izz not”, “hazz not” or “are not”.

SML

`|declare_postfix (100, "not");`

HOL Constant

\$not : *COPULA* → *COPULA*

$\forall pred \bullet pred\ not = \lambda pa\ t\ w \bullet \neg\ pred\ pa\ t\ w$

Quantifiers The following function is used by both quantifiers to check if something is in the range of quantification.

Think of a TermM as denoting a set of individuals, this is a test for membership of that set. The complications are because substances and attributes have different types in this model.

SML

`|declare_infix(300, "InTermM");`

HOL Constant

$\$InTermM : I \rightarrow TermM \rightarrow BOOL$

$\forall c1t\ c2ts \bullet c1t\ InTermM\ c2ts \Leftrightarrow Fst\ c1t = Fst\ c2ts \wedge Snd\ c1t \in PeSet\ (Snd\ c2ts)$

$interm_exists_lemma =$

$\vdash \forall t \bullet \exists j \bullet j\ InTermM\ t$

We then use that membership test in defining the quantifiers. The quantifiers expect to be supplied with a copula and a term. The quantifier then predicates using the copula the term of everything or something in the domain of quantification (which is the subject term). The copulas are defined below.

HOL Constant

$All : TermM \rightarrow (I \rightarrow TermM \rightarrow MPROP) \rightarrow TermM \rightarrow MPROP$

$\forall s\ r\ p \bullet All\ s\ r\ p = \lambda w \bullet \forall z \bullet z\ InTermM\ s \Rightarrow r\ z\ p\ w$

HOL Constant

$Some : TermM \rightarrow (I \rightarrow TermM \rightarrow MPROP) \rightarrow TermM \rightarrow MPROP$

$\forall s\ r\ p \bullet Some\ s\ r\ p = \lambda w \bullet \exists z \bullet z\ InTermM\ s \wedge r\ z\ p\ w$

Predicators For essential predication it is necessary that the individual and the predicate are both of the same category and then reduces under our model to set membership. In effect, since the non-substantial individuals are tagged with their category, we need only deal separately with the distinction between substantial and non-substantial and the set inclusion will ensure a match in the non-substantial categories.

HOL Constant

$izz : I \rightarrow TermM \rightarrow MPROP$

$\forall j\ t \bullet izz\ j\ t = \lambda w \bullet j\ InTermM\ t$

For accidental predication the subject term must be substantial and the predicate may not be. We then need some member of the predicate to be attributable to the substance.

HOL Constant

$hazz : I \rightarrow TermM \rightarrow MPROP$

$\forall c1t\ c2ts \bullet hazz\ c1t\ c2ts = \lambda w \bullet$

$Fst\ c1t = Category_of_Substance$

$\wedge \neg Fst\ c2ts = Category_of_Substance$

$\wedge (\exists b \bullet b \in PeSet\ (Snd\ c2ts) \wedge (Snd\ c1t) \in w\ (OutR(Fst\ c2ts), b))$

$not_izz_and_hazz_lemma1 =$

$\vdash \forall pa\ t\ w \bullet \neg (izz\ pa\ t\ w \wedge hazz\ pa\ t\ w)$

HOL Constant

$\mathit{are} : I \rightarrow \mathit{TermM} \rightarrow \mathit{MPROP}$

$\forall pa\ t \bullet \mathit{are}\ pa\ t = \lambda w \bullet \mathit{izz}\ pa\ t\ w \vee \mathit{hazz}\ pa\ t\ w$

$\mathit{are_izz_neq_hazz_lemma} =$

$\vdash \forall pa\ t\ w \bullet \mathit{are}\ pa\ t\ w \Leftrightarrow \neg (\mathit{izz}\ pa\ t\ w \Leftrightarrow \mathit{hazz}\ pa\ t\ w)$

$\mathit{All_are_izz_or_hazz_lemma} =$

$\vdash \forall A\ B\ w \bullet \mathit{All}\ A\ \mathit{are}\ B\ w \Leftrightarrow \mathit{All}\ A\ \mathit{izz}\ B\ w \vee \mathit{All}\ A\ \mathit{hazz}\ B\ w$

$\mathit{Some_are_izz_or_hazz_lemma} =$

$\vdash \forall A\ B\ w \bullet \mathit{Some}\ A\ \mathit{are}\ B\ w \Leftrightarrow \mathit{Some}\ A\ \mathit{izz}\ B\ w \vee \mathit{Some}\ A\ \mathit{hazz}\ B\ w$

$\mathit{All_are_not_lemma} =$

$\vdash \forall A\ B\ w \bullet \mathit{All}\ A\ (\mathit{are\ not})\ B\ w \Leftrightarrow \mathit{All}\ A\ (\mathit{izz\ not})\ B\ w \wedge \mathit{All}\ A\ (\mathit{hazz\ not})\ B\ w$

Modal Operators In this model the modal operators are operators over propositions.

HOL Constant

$\diamond : \mathit{MPROP} \rightarrow \mathit{MPROP}$

$\forall p \bullet \diamond\ p = \lambda w \bullet \exists w' \bullet p\ w'$

HOL Constant

$\square : \mathit{MPROP} \rightarrow \mathit{MPROP}$

$\forall p \bullet \square\ p = \lambda w \bullet \forall w' \bullet p\ w'$

5.3 Propositional Operators

Though the truth functional propositional operators do not feature in the syllogism it is nevertheless useful to have them in giving a full account of Aristotle's logic and they are therefore here defined.

That these propositional operators are "truth functional", in a context in which propositions are not regarded as denoting truth values requires a little explanation perhaps. Our propositions are families of truth values indexed by possible worlds, i.e. functions from possible worlds to truth values, or in the context of a two valued logic (which Aristotle's seems to be), sets of possible worlds. In this context the usual truth functional operators can be expressed by mapping the usual operator over the set of possible worlds, i.e. the result in every possible world is the result of applying the truth functional operator to the values of the propositions in that possible world. These also correspond to the obvious set theoretic operation if the propositions are thought of as sets of possible worlds, i.e. intersetion for conjunction, complementation for negation.

The symbols for the operators are already in use, so we define the operations using decorated variants of the symbols and use an alias to allow the undecorated symbol to be used.

HOL Constant

$\neg_a : MPROP \rightarrow MPROP$

$\forall p \bullet \neg_a p = \lambda w \bullet \neg (p w)$

SML

`declare_alias ("¬", "¬a");`

SML

`declare_infix(220, "∧a");`

HOL Constant

$\$ \wedge_a : MPROP \rightarrow MPROP \rightarrow MPROP$

$\forall p \ q \bullet (p \wedge_a q) = \lambda w \bullet (p w) \wedge (q w)$

SML

`declare_alias ("∧", "$∧a");`

SML

`declare_infix(210, "⇒a");`

HOL Constant

$\$ \Rightarrow_a : MPROP \rightarrow MPROP \rightarrow MPROP$

$\forall p \ q \bullet (p \Rightarrow_a q) = \lambda w \bullet p w \Rightarrow q w$

SML

`declare_alias ("⇒", "$⇒a");`

SML

`declare_infix(200, "⇔a");`

HOL Constant

$\$ \Leftrightarrow_a : MPROP \rightarrow MPROP \rightarrow MPROP$

$\forall p \ q \bullet (p \Leftrightarrow_a q) = \lambda w \bullet p w \Leftrightarrow q w$

SML

`declare_alias ("⇔", "$⇔a");`

5.4 Quantification

The Grice/Code analysis makes use of quantifiers, particularly existential quantification. To verify the formulae in this context we therefore need to define modal version of the quantifiers.

SML

```
| declare_binder "∀a";
```

HOL Constant

```
| $∀a : (TermM → MPROP) → MPROP
|-----
| ∀mpf • $∀a mpf = λw • ∀t • mpf t w
```

SML

```
| declare_alias ("∀", «$∀a»);
```

SML

```
| declare_binder "∃a";
```

HOL Constant

```
| $∃a : (TermM → MPROP) → MPROP
|-----
| ∀mpf • $∃a mpf = λw • ∃t • mpf t w
```

SML

```
| declare_alias ("∃", «$∃a»);
```

5.5 Judgements

I'm not yet clear what to offer here, so for the present I will define two kinds of sequent, which will be displayed with the symbols \models and \Vdash . the former being a kind of contingent material implication and the latter a necessary implication.

Both form of judgement seem suitable for expressing the rules of the syllogism at first glance but which can also be used for conversions.

The first expresses a contingent entailment, that if some arbitrary finite (possibly empty) collection of premisses are contingently true, then some conclusion will also be true. Since the consequence is material, and the premisses might be contingent, the conclusion might also be contingent. One might hope that if the rules of the syllogism are applied and the premisses are necessary, then so will be the conclusions.

SML

```
| declare_infix(100, "⊨");
```

HOL Constant

```
| $⊨ : MPROP LIST → MPROP → BOOL
|-----
| ∀lp c • lp ⊨ c ⇔ Fold (λp t • p actual_world ∧ t) lp T ⇒ c actual_world
```

This one says that in every possible world the premises entail the conclusion (still material).

SML

```
|declare_infix(100, "⊨");
```

HOL Constant

```
|$⊨ : MPROP LIST → MPROP → BOOL
```

```
|∀lp c• lp ⊨ c ⇔ ∀w• Fold (λp t• p w ∧ t) lp T ⇒ c w
```

In the present context the choice between the two is probably immaterial, since we know no more about the actual world than any other, so anything that we can prove to be true contingently, we can also prove to be true necessarily.

5.6 Conversions

Premisses, their Modes and Conversions See: [Prior Analytics Book 1 Part 2 Paragraph 2](#).

First then take a universal negative with the terms A and B.

If no B is A, neither can any A be B. For if some A (say C) were B, it would not be true that no B is A; for C is a B.

But if every B is A then some A is B. For if no A were B, then no B could be A. But we assumed that every B is A.

Similarly too, if the premiss is particular. For if some B is A, then some of the As must be B. For if none were, then no B would be A. But if some B is not A, there is no necessity that some of the As should not be B; e.g. let B stand for animal and A for man. Not every animal is a man; but every man is an animal.

These work out fine for *izz*, so I will do those first, and then show that they fail for *hazz* and *is*.

The first and third conversions are most useful when expressed as an equation, since our proof system is based primarily on rewriting using equations.

```
|izz_not_lemma =
|  ⊢ All B (izz not) A = All A (izz not) B
```

```
|some_izz_lemma =
|  ⊢ Some B izz A = Some A izz B
```

These we also supply as our Aristotelian judgements, together with the second which does not give an equation. The second conversion embodies the u-p syllogisms.

```
|izz_conv1 = ⊢
|  [All B (izz not) A] ⊨ All A (izz not) B
```

```
|izz_conv2 = ⊢
|  [All B izz A] ⊨ Some A izz B
```

```
|izz_conv3 = ⊢
|  [Some B izz A] ⊨ Some A izz B
```


Now we look at *hazz*.

The following theorems state that the two equational conversions are both false for *hazz*.

$$\begin{array}{|l} \mathbf{not_hazz_not_lemma} = \\ \vdash \neg (\forall A B \bullet \text{All } B \text{ (hazz not) } A = \text{All } A \text{ (hazz not) } B) \end{array}$$

$$\begin{array}{|l} \mathbf{not_some_hazz_lemma} = \\ \vdash \neg (\forall A B \bullet \text{Some } B \text{ hazz } A = \text{Some } A \text{ hazz } B) \end{array}$$

Aristotle's second conversion also fails for *hazz*, because it incorporates an application of the third in effect. If we simplify by removing the final flip we get:

$$\begin{array}{|l} \mathbf{hazz_conv2} = \\ \vdash [\text{All } A \text{ hazz } B] \Vdash \text{Some } A \text{ hazz } B \end{array}$$

Since *is* is the conjunction of *izz* and *hazz* it is likely that it would yeild similar results to *hazz*.

5.7 Modal Conversions

Prior Analytics Book 1 Part 3 See: [Universal and Possible Premisses and their Conversions](#).

These are the conversions in relation to necessity and possibility described by Aristotle:

1. If it is necessary that no B is A, it is necessary also that no A is B.
2. If all or some B is A of necessity, it is necessary also that some A is B.
3. If it is possible that all or some B is A, it will be possible that some A is B.
4. and so on

So in this section Aristotle only offers variants of the previous conversions with either “possible” or “necessary” attached to both premiss and conclusion.

We can prove generally that modal operators can be introduced into a conversion:

$$\begin{array}{|l} \mathbf{\diamond_conv} = \\ \vdash [P] \Vdash Q \Rightarrow [\diamond P] \Vdash \diamond Q \end{array}$$

$$\begin{array}{|l} \mathbf{\square_conv} = \\ \vdash [P] \Vdash Q \Rightarrow [\square P] \Vdash \square Q \end{array}$$

$$\mathbf{\square_izz_thm} = \vdash [\square (\text{All } A \text{ izz } B)] \Vdash \text{All } A \text{ izz } B$$

$$\mathbf{\square_hazz_thm} = \vdash [\square (\text{All } A \text{ hazz } B)] \Vdash \text{All } A \text{ izz } B$$

$$\mathbf{izz_square_thm} = \vdash [\text{All } A \text{ izz } B] \Vdash \square (\text{All } A \text{ izz } B)$$

$$\mathbf{not_square_hazz_thm} = \vdash [] \Vdash (\neg \square (\text{All } A \text{ hazz } B))$$

$$\mathbf{\square_izz_thm2} = \vdash [\square (\text{All } A \text{ izz } B)] \Vdash \text{All } A \text{ izz } B$$

$$\mathbf{\square_hazz_thm2} = \vdash [\square (\text{All } A \text{ hazz } B)] \Vdash \text{All } A \text{ izz } B$$

$$\mathbf{izz_square_thm2} = \vdash [\text{All } A \text{ izz } B] \Vdash \square (\text{All } A \text{ izz } B)$$

$$\mathbf{not_square_hazz_thm2} = \vdash [] \Vdash (\neg \square (\text{All } A \text{ hazz } B))$$

\Box_hazz_thm is a bit odd. Really what I wanted to prove was that no accidental predication is necessary, but I have no negation in the syllogism, so I just proved that if an accidental predication were necessary then it would be essential. Then I went back and defined negation so permitting a direct denial that any accidental predication is necessary.

There are many theorems which one would naturally prove at this point, to facilitate further proofs and proof automation, which are not expressible syllogistically. Proof automation depends heavily on the demonstration of equations, so that proof may proceed by rewriting. But syllogisms are not suitable for this.

The natural way to proceed in such a case is to continue in this theory doing things which support proofs of syllogisms without being restrained to syllogisms, and then to have a separate theory in which the syllogistic claims are presented. Some reflection is desirable on what the philosophical objectives are and what course will best contribute to those purposes.

Here are some general modal results which I have not noticed in Aristotle as yet.

```

|  $\Box\_elim\_thm =$ 
|    $\vdash [\Box P] \vDash P$ 
|  $\Diamond\_intro\_thm =$ 
|    $\vdash [P] \vDash \Diamond P$ 
|  $\Box\_Diamond\_thm =$ 
|    $\vdash [\Box P] \vDash \Diamond P$ 

```

5.8 Other Conversions

The following conversions relate to the square of opposition, but I have not yet discovered where they appear in Aristotle. They work for all the copulas, so I have used a free variable for the copulas.

```

|  $\neg\_All\_conv\_thm =$ 
|    $\vdash (\neg All A cop B) = Some A (cop not) B$ 
|  $\neg\_All\_not\_conv\_thm2 =$ 
|    $\vdash (\neg All A (cop not) B) = Some A cop B$ 
|  $\neg\_Some\_conv\_thm =$ 
|    $\vdash (\neg Some A cop B) = All A (cop not) B$ 
|  $\neg\_Some\_not\_conv\_thm =$ 
|    $\vdash (\neg Some A (cop not) B) = All A cop B$ 

```

They are contraries out of Aristotles square of opposition

Normally theorems like this would be proved closed, but it looks more Aristotelian without the quantifiers and we can imagine that they are schemata. To use them it will usually be desirable to close them, which is easily done, e.g.:

SML

```

|  $all\_forall\_intro \neg\_Some\_not\_conv\_thm;$ 

```

ProofPower output

```

|  $val it = \vdash \forall A cop B \bullet (\neg Some A (cop not) B) = All A cop B : THM$ 

```

5.9 Syllogisms for Essential Predication

Though the usual syllogisms are not valid for predication in general, the problems are confined to accidental predication. We can, by methods similar to those used above obtain automatic proofs of the 24 valid syllogisms restricted to essential predication.

The details are omitted, but the 24 izz syllogisms have been proven and stored in the theory, see: Appendix [A.9](#).

5.10 Some Accidental Syllogisms

5.11 Grice and Code

I now review the Grice/Code analysis under the revised interpretation of *izz* and *hazz*.

On my first attempt I did not notice till rather late that the material covered both Aristotle's and Plato's metaphysics, which are not wholly compatible. To do this using only conservative extension we have to use a different context for the parts of the treatment which might be incompatible. Three new theories will therefore be introduced, respectively covering:

gocom material common to Aristotle and Plato

gcaris material specific to Aristotle

geplato material specific to Plato

All three theories are in the context of theory *syllmetap* which is an integrated model of both the modal syllogism and the metaphysics, incorporating the u-p syllogisms, which makes slightly more sense in the context of the metaphysics than otherwise.

5.11.1 Common Material

SML

```
|force_new_theory "gcon";
```

The following results are now provable:

```
|FP1 =   ⊢ [] ⊢ All A izz A
|FP2 =   ⊢ [All A izz B; All B izz C] ⊢ All A izz C
|FP3 =   ⊢ [All A hazz B] ⊢ ¬ (All A izz B)
|FP4a =  ⊢ [All A hazz B; All B izz C] ⊢ All A hazz C
|FP4 =   ⊢ [] ⊢ All A hazz B ⇔ (∃a C • All A hazz C ∧ All C izz B)
```

These are not very Aristotelian. It would seem more Aristotelian to have:

```
|FP3b =
|   ⊢ [All A hazz B] ⊢ Some A (izz not) B
```

The above rendition of FP4 may not be true to the intention of Code. I possibly he might have intended that C be an individual.

To prove that more interesting result I need to define “individual”.

HOL Constant

$$\begin{array}{|l} \mathbf{individual} : \mathit{TermM} \rightarrow \mathit{BOOL} \\ \hline \forall A \bullet \mathit{individual} A \Leftrightarrow \exists b \bullet \mathit{Snd} A = \mathit{NeSet}\{b\} \end{array}$$

Since the above is a regular predicate, we need something to convert an ordinary proposition into a modal proposition to use it in the context of modal syllogisms.

HOL Constant

$$\begin{array}{|l} \mathbf{Mp} : \mathit{BOOL} \rightarrow \mathit{MPROP} \\ \hline \forall p \bullet \mathit{Mp} p = \lambda w \bullet p \end{array}$$

The revised principle is then:

$$\mathit{FP4b} = ?\vdash \square \vdash (\mathit{All} A \mathit{hazz} B \Leftrightarrow \exists C \bullet \mathit{Mp}(\mathit{individual} C) \wedge (\mathit{All} A \mathit{hazz} C) \wedge (\mathit{All} C \mathit{izz} B))$$

However, this is false in our model, since there need be no single attribute which is possessed by every substance which izz A. Consider the claim that all paints have colour. This may be true even if there is no individual colour which every paint hazz. However, if this stronger claim is not what Code intended, then what did he mean? Surely not the theorem I actually proved as FP4, since that is too trivial to be worth mentioning.

If we require A to be an individual we get a result:

$$\begin{array}{|l} \mathbf{FP4c} = \\ \vdash [\mathit{Mp} (\mathit{individual} A)] \\ \vdash (\mathit{All} A \mathit{hazz} B \\ \Leftrightarrow (\exists C \bullet \mathit{Mp} (\mathit{individual} C) \wedge \mathit{All} A \mathit{hazz} C \wedge \mathit{All} C \mathit{izz} B)) \end{array}$$

HOL Constant

$$\begin{array}{|l} \mathbf{Individual} : \mathit{TermM} \rightarrow \mathit{MPROP} \\ \hline \forall A \bullet \mathit{Individual} A = \square \forall_a B \bullet \mathit{All} B \mathit{izz} A \Rightarrow_a \mathit{All} A \mathit{izz} B \end{array}$$

Now on the face of it, in the context of our present model, the modal operator in this definition is irrelevant. This because all essential predication is necessary.

The following theorem confirms that intuition.

$$\begin{array}{|l} \mathit{individual_lemma1} = \\ \vdash \forall A \bullet \mathit{Individual} A = (\forall B \bullet \mathit{All} B \mathit{izz} A \Rightarrow \mathit{All} A \mathit{izz} B) \end{array}$$

We can also show that the Code definition is equivalent to our own:

$$\begin{array}{|l} \mathit{individual_lemma2} = \\ \vdash \mathit{Individual} A = \mathit{Mp} (\mathit{individual} A) \end{array}$$

In addition to the apparently spurious invocation of necessity, the Code definition depends upon the u-p syllogisms.

Our own primitive definition is couched in terms of the underlying model, and so in terms of that model we would have to regard as a primitive rather than a defined concept (this may be the best way to think of it).

Now we come to the Code definition of particular:

HOL Constant

$$\begin{array}{|l} \mathit{particular} : \mathit{TermM} \rightarrow \mathit{MPROP} \\ \hline \forall A \bullet \mathit{particular} A = \Box \forall_a B \bullet \mathit{All} B \text{ are } A \Rightarrow_a \mathit{All} A \text{ izz } B \wedge_a \mathit{All} B \text{ izz } A \end{array}$$

A particular is an individual substance and one would have thought that a definition closer to saying that directly might have been a good idea. In this case in our model the modal operator is not redundant, because without it the definiens would be true if A were an individual attribute which contingently has an empty extension (i.e. is true of no substance), which is possible in this model. However, this cannot be true of necessity unless A is substantial.

Code might have used a similar device to define substantial:

HOL Constant

$$\begin{array}{|l} \mathit{substantial} : \mathit{TermM} \rightarrow \mathit{MPROP} \\ \hline \forall A \bullet \mathit{substantial} A = \Box \forall_a B \bullet \mathit{All} B \text{ are } A \Rightarrow_a \mathit{All} B \text{ izz } A \end{array}$$

and then defined a particular as a substantial individual. Alternatively substantial might be taken as primitive to avoid the use of a modal operator. Again, Code's definition relies on the u-p syllogisms.

Code's definition of universal is:

HOL Constant

$$\begin{array}{|l} \mathit{universal} : \mathit{TermM} \rightarrow \mathit{MPROP} \\ \hline \forall A \bullet \mathit{universal} A = \Diamond \exists_a B \bullet \mathit{All} B \text{ are } A \wedge_a \neg_a (\mathit{All} A \text{ izz } B \wedge_a \mathit{All} B \text{ izz } A) \end{array}$$

I think the intension is that a universal is anything except a particular, in which case that would be a better way to define it. However, in this model, this definition will be true of any non-individual, unless the $\lceil \Diamond \rceil$ is changed to $\lceil \Box \rceil$.

Ontological Theorems T1-T3 are essentially redundant definitions so I will omit them and not use the additional vocabulary on this pass.

6 Conclusions

7 Postscript

These are the kinds of thing which *might* happen in future issues of this document:

- If I get constructive feedback (pointing out errors counts, telling me its a waste of time doesn't, explaining bits of Aristotle I'm getting wrong would be great) then I will probably do something about it.
- There is a good chance I will get to know something about what Aristotle really said and make improvements arising from this. In that case I would like to put some hyperlinks into my online hypertext of Aristotle's relevant works, connecting the specific features of the model with the passages in Aristotle which they reflect.
- I am interested in the methods, and I may spend more time trying to describe them in a way which might be intelligible to philosophers.
- If this ever becomes a good approximation to Aristotle's position there would then be some philosophical analysis of the metaphysics (by contrast with a purely logical analysis), and somewhere there are points to be made about the kind of analysis which I am aiming for.

There are three stages I envisage in the process of getting philosophical insights from this kind of work:

formal modelling I now have a series of models. Of the Aristotelian Syllogism the final model (Sections 4.8, A.7) may well be good enough, but the subsequent models (modal syllogisms and the metaphysics) still follow the same approach as earlier models of the syllogism which I no longer consider ideal, notably the presumption against empty predicates. I therefore have in mind that Sections 4.9, and 5 will be brought into line with the treatment in Section 4.8. I also regret in my treatment of the modal syllogisms treating the modal operators as forms of judgement, so I might recant on that and make them propositional operators. When those changes are done a closer scrutiny of the model of the metaphysics will be needed, after which I can re-examine the Grice/Code material and see whether I can now make any better sense of it, as well as considering the real current target of my investigation, which is the comparative analysis of Plato and Aristotle on the nature of key concepts such as necessity and contingency, essential and accidental predication, and demonstrative truth.

model verification The models need checking against Aristotle's writings.

formal analysis Further formal work is needed to come to an understanding of these models.

philosophical analysis After the models are formulated and verified, and formal analysis has deepened our understanding of these models, we may then be in a position to obtain some interesting philosophical insights.

A Theory Listings

A.1 The Theory ariscat

Parents

aristotle

Constants

<i>CatSubs</i>	<i>CAT</i>
<i>Cat</i>	$CATM \rightarrow CAT$
<i>IndvSet</i>	$CATM \rightarrow ISUB \mathbb{P}$
<i>AttrSet</i>	$CATM \rightarrow (ISUB \rightarrow BOOL) \mathbb{P}$
<i>CatSet</i>	$CATM \rightarrow (ISUB + (ISUB \rightarrow BOOL)) \mathbb{P}$
<i>\$izz</i>	$(CATM, BOOL) BR$
<i>\$hazz</i>	$(CATM, BOOL) BR$
<i>\$essentially-predicable-of</i>	$(CATM, BOOL) BR$
<i>\$accidentally-predicable-of</i>	$(CATM, BOOL) BR$
<i>\$predicable-of</i>	$(CATM, BOOL) BR$
<i>individual</i>	$CATM \rightarrow BOOL$
<i>particular</i>	$CATM \rightarrow BOOL$
<i>universal</i>	$CATM \rightarrow BOOL$
<i>SomeThing</i>	$CATM \rightarrow BOOL$
<i>form</i>	$CATM \rightarrow BOOL$

Types

ACAT
ISUB

Type Abbreviations

<i>CATM</i>	<i>CATM</i>
<i>CAT</i>	<i>CAT</i>

Fixity

Right Infix 300:

accidentally-predicable-of *izz*
essentially-predicable-of *predicable-of*
hazz

Definitions

CatSubs	$\vdash \text{CatSubs} = \text{InL One}$
Cat	$\vdash \forall x$ <ul style="list-style-type: none"> • $\text{Cat } x$ $= (\text{if } \text{IsL } x \text{ then } \text{CatSubs} \text{ else } \text{InR } (\text{Fst } (\text{OutR } x)))$
IndvSet	$\vdash \forall x \bullet \text{IndvSet } x = \text{OutL } x$
AttrSet	$\vdash \forall x \bullet \text{AttrSet } x = \text{Snd } (\text{OutR } x)$
CatSet	$\vdash \forall x$ <ul style="list-style-type: none"> • $\text{CatSet } x$ $= (\text{if } \text{Cat } x = \text{CatSubs}$ $\text{ then } \{y \mid \exists z \bullet z \in \text{IndvSet } x \wedge y = \text{InL } z\}$ $\text{ else } \{y \mid \exists z \bullet z \in \text{AttrSet } x \wedge y = \text{InR } z\})$
izz	$\vdash \forall A B \bullet A \text{ izz } B \Leftrightarrow \text{Cat } A = \text{Cat } B \wedge \text{CatSet } A \subseteq \text{CatSet } B$
hazz	$\vdash \forall A B$ <ul style="list-style-type: none"> • $A \text{ hazz } B$ $\Leftrightarrow \text{Cat } A = \text{CatSubs}$ $\wedge \neg \text{Cat } B = \text{CatSubs}$ $\wedge (\exists a$ <ul style="list-style-type: none"> • $a \in \text{AttrSet } B \wedge (\forall s \bullet s \in \text{IndvSet } A \Rightarrow a s)$
essentially_predicable_of	$\vdash \forall A B \bullet A \text{ essentially_predicable_of } B \Leftrightarrow B \text{ izz } A$
accidentally_predicable_of	$\vdash \forall A B \bullet A \text{ accidentally_predicable_of } B \Leftrightarrow B \text{ hazz } A$
predicable_of	$\vdash \forall A B$ <ul style="list-style-type: none"> • $A \text{ predicable_of } B$ $\Leftrightarrow A \text{ essentially_predicable_of } B$ $\vee A \text{ accidentally_predicable_of } B$
individual	$\vdash \forall A \bullet \text{individual } A \Leftrightarrow (\exists a \bullet \text{CatSet } A = \{a\})$
particular	$\vdash \forall A \bullet \text{particular } A \Leftrightarrow \text{individual } A \wedge \text{Cat } A = \text{CatSubs}$
universal	$\vdash \forall A \bullet \text{universal } A \Leftrightarrow \neg \text{particular } A$
SomeThing	$\vdash \forall x \bullet \text{SomeThing } x \Rightarrow \text{individual } x$
form	$\vdash \forall x \bullet \text{form } x \Rightarrow \text{SomeThing } x \wedge \text{universal } x$

Theorems

catm_eq_lemma	$\vdash \forall A B \bullet A = B \Leftrightarrow \text{Cat } A = \text{Cat } B \wedge \text{CatSet } A = \text{CatSet } B$
izz_lemma1	$\vdash \forall A B$ <ul style="list-style-type: none"> • $\text{Cat } A = \text{CatSubs}$ $\Rightarrow (A \text{ izz } B$ $\Leftrightarrow \text{Cat } B = \text{CatSubs} \wedge \text{IndvSet } A \subseteq \text{IndvSet } B)$
izz_lemma2	$\vdash \forall A B$ <ul style="list-style-type: none"> • $\text{Cat } B = \text{CatSubs}$ $\Rightarrow (A \text{ izz } B$ $\Leftrightarrow \text{Cat } A = \text{CatSubs} \wedge \text{IndvSet } A \subseteq \text{IndvSet } B)$
izz_lemma3	$\vdash \forall A B$ <ul style="list-style-type: none"> • $\neg \text{Cat } A = \text{CatSubs}$ $\Rightarrow (A \text{ izz } B$ $\Leftrightarrow \text{Cat } B = \text{Cat } A \wedge \text{AttrSet } A \subseteq \text{AttrSet } B)$

izz_lemma4 $\vdash \forall A B$
 $\bullet \neg \text{Cat } B = \text{CatSubs}$
 $\Rightarrow (A \text{ izz } B)$
 $\Leftrightarrow \text{Cat } B = \text{Cat } A \wedge \text{AttrSet } A \subseteq \text{AttrSet } B)$

l01 $\vdash \forall A \bullet A \text{ izz } A$

l01b $\vdash \forall A \bullet \neg A \text{ hazz } A$

l02 $\vdash \forall A B C \bullet A \text{ izz } B \wedge B \text{ izz } C \Rightarrow A \text{ izz } C$

l03 $\vdash \forall A B \bullet A \text{ hazz } B \Rightarrow \neg A \text{ izz } B$

l03b $\vdash \forall A B \bullet A \text{ izz } B \Rightarrow \neg A \text{ hazz } B$

l03c $\vdash \forall A B \bullet A \text{ hazz } B \Rightarrow \neg A = B$

l04a $\vdash \forall A B C \bullet A \text{ hazz } B \wedge B \text{ izz } C \Rightarrow A \text{ hazz } C$

l04 $\vdash \forall A B \bullet A \text{ hazz } B \Leftrightarrow (\exists C \bullet A \text{ hazz } C \wedge C \text{ izz } B)$

l04b $\vdash \forall A B C \bullet A \text{ izz } B \wedge B \text{ hazz } C \Rightarrow A \text{ hazz } C$

l06n $\vdash \neg (\forall A B$
 $\bullet A \text{ hazz } B \wedge \text{particular } A \Rightarrow (\exists C \bullet C \neq A \wedge A \text{ izz } B))$

l07 $\vdash \forall A B$
 $\bullet A \text{ predicable_of } B$
 $\Leftrightarrow B \text{ izz } A \vee (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$

l08 $\vdash \forall A \bullet A \text{ essentially_predicable_of } B \Leftrightarrow B \text{ izz } A$

l09 $\vdash \forall A$
 $\bullet A \text{ accidentally_predicable_of } B$
 $\Leftrightarrow (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$

l10 $\vdash \forall A B \bullet A = B \Leftrightarrow A \text{ izz } B \wedge B \text{ izz } A$

l16 $\vdash \forall A B$
 $\bullet A \text{ predicable_of } B$
 $\Leftrightarrow B \text{ izz } A \vee (\exists C \bullet B \text{ hazz } C \wedge C \text{ izz } A)$

l17 $\vdash \forall A \bullet A \text{ essentially_predicable_of } A$

l18 $\vdash \forall A \bullet A \text{ accidentally_predicable_of } B \Rightarrow A \neq B$

l19 $\vdash \forall A \bullet \neg A \text{ accidentally_predicable_of } A$

l20 $\vdash \forall A \bullet \text{particular } A \Rightarrow \text{individual } A$

l22 $\vdash \neg (\exists A \bullet \text{particular } A \wedge \text{form } A)$

l23b $\vdash \forall A \bullet \text{form } A \Rightarrow \text{individual } A$

l24a $\vdash \forall A B$
 $\bullet \text{particular } A \wedge \text{individual } B \wedge A \text{ izz } B$
 $\Rightarrow \text{particular } B$

l24 $\vdash \forall A \bullet \text{particular } A \Rightarrow \neg (\exists B \bullet \text{form } B \wedge A \text{ izz } B)$

l26 $\vdash \forall A B$
 $\bullet \text{form } A \wedge \text{particular } B$
 $\Rightarrow (A \text{ predicable_of } B \Leftrightarrow B \text{ hazz } A)$

l05b $\vdash \forall x \bullet \text{form } x \Rightarrow \text{universal } x$

A.2 The Theory syllog1

Parents

aristotle

Type Abbreviations

TermS *TermS*

Theorems

<i>Barbara</i>	$M \subseteq P, S \subseteq M \vdash S \subseteq P$
<i>Celarent</i>	$M \subseteq \sim P, S \subseteq M \vdash S \subseteq \sim P$
<i>Darii</i>	$M \subseteq P, \neg S \subseteq \sim M \vdash \neg S \subseteq \sim P$
<i>Ferio</i>	$M \subseteq \sim P, \neg S \subseteq \sim M \vdash \neg S \subseteq P$
<i>Cesare</i>	$P \subseteq \sim M, S \subseteq M \vdash S \subseteq \sim P$
<i>Camestres</i>	$P \subseteq M, S \subseteq \sim M \vdash S \subseteq \sim P$
<i>Festino</i>	$P \subseteq \sim M, \neg S \subseteq \sim M \vdash \neg S \subseteq P$
<i>Baroco</i>	$P \subseteq M, \neg S \subseteq M \vdash \neg S \subseteq P$
<i>Disamis</i>	$\neg M \subseteq \sim P, M \subseteq S \vdash \neg S \subseteq \sim P$
<i>Datisi</i>	$M \subseteq P, \neg M \subseteq \sim S \vdash \neg S \subseteq \sim P$
<i>Bocardo</i>	$\neg M \subseteq P, M \subseteq S \vdash \neg S \subseteq P$
<i>Ferison</i>	$M \subseteq \sim P, \neg M \subseteq \sim S \vdash \neg S \subseteq P$
<i>Camenes</i>	$P \subseteq M, M \subseteq \sim S \vdash S \subseteq \sim P$
<i>Dimaris</i>	$\neg P \subseteq \sim M, M \subseteq S \vdash \neg S \subseteq \sim P$
<i>Fresison</i>	$P \subseteq \sim M, \neg M \subseteq \sim S \vdash \neg S \subseteq P$
<i>Darapti</i>	$\neg \{ \} \in \{M; S; P\}, M \subseteq P, M \subseteq S \vdash \neg S \subseteq \sim P$
<i>Felapton</i>	$\neg \{ \} \in \{M; S; P\}, M \subseteq \sim P, M \subseteq S \vdash \neg S \subseteq P$
<i>Bramantip</i>	$\neg \{ \} \in \{M; S; P\}, P \subseteq M, M \subseteq S \vdash \neg S \subseteq \sim P$
<i>Fesapo</i>	$\neg \{ \} \in \{M; S; P\}, P \subseteq \sim M, M \subseteq S \vdash \neg S \subseteq P$
<i>Barbari</i>	$\neg \{ \} \in \{M; S; P\}, M \subseteq P, S \subseteq M \vdash \neg S \subseteq \sim P$
<i>Celaront</i>	$\neg \{ \} \in \{M; S; P\}, M \subseteq \sim P, S \subseteq M \vdash \neg S \subseteq P$
<i>Cesaro</i>	$\neg \{ \} \in \{M; S; P\}, P \subseteq \sim M, S \subseteq M \vdash \neg S \subseteq P$
<i>Camestrop</i>	$\neg \{ \} \in \{M; S; P\}, P \subseteq M, S \subseteq \sim M \vdash \neg S \subseteq P$
<i>Camenop</i>	$\neg \{ \} \in \{M; S; P\}, P \subseteq M, M \subseteq \sim S \vdash \neg S \subseteq P$

A.3 The Theory syllog2

Parents

aristotle

Type Abbreviations

TermP *TermP*

Theorems

<i>Barbara</i>	$M \Rightarrow P, S \Rightarrow M \vdash S \Rightarrow P$
<i>Celarent</i>	$M \Rightarrow \neg P, S \Rightarrow M \vdash S \Rightarrow \neg P$
<i>Darii</i>	$M \Rightarrow P, \neg (S \Rightarrow \neg M) \vdash \neg (S \Rightarrow \neg P)$
<i>Ferio</i>	$M \Rightarrow \neg P, \neg (S \Rightarrow \neg M) \vdash \neg (S \Rightarrow P)$
<i>Cesare</i>	$P \Rightarrow \neg M, S \Rightarrow M \vdash S \Rightarrow \neg P$
<i>Camestres</i>	$P \Rightarrow M, S \Rightarrow \neg M \vdash S \Rightarrow \neg P$
<i>Festino</i>	$P \Rightarrow \neg M, \neg (S \Rightarrow \neg M) \vdash \neg (S \Rightarrow P)$
<i>Baroco</i>	$P \Rightarrow M, \neg (S \Rightarrow M) \vdash \neg (S \Rightarrow P)$
<i>Disamis</i>	$\neg (M \Rightarrow \neg P), M \Rightarrow S \vdash \neg (S \Rightarrow \neg P)$
<i>Datisi</i>	$M \Rightarrow P, \neg (M \Rightarrow \neg S) \vdash \neg (S \Rightarrow \neg P)$
<i>Bocardo</i>	$\neg (M \Rightarrow P), M \Rightarrow S \vdash \neg (S \Rightarrow P)$
<i>Ferison</i>	$M \Rightarrow \neg P, \neg (M \Rightarrow \neg S) \vdash \neg (S \Rightarrow P)$
<i>Camenes</i>	$P \Rightarrow M, M \Rightarrow \neg S \vdash S \Rightarrow \neg P$
<i>Dimaris</i>	$\neg (P \Rightarrow \neg M), M \Rightarrow S \vdash \neg (S \Rightarrow \neg P)$
<i>Fresison</i>	$P \Rightarrow \neg M, \neg (M \Rightarrow \neg S) \vdash \neg (S \Rightarrow P)$

A.4 The Theory syllog3

Parents

aristotle

Constants

$\$a$ $(Term2, BOOL) BR$
 $\$e$ $(Term2, BOOL) BR$
 $\$i$ $(Term2, BOOL) BR$
 $\$u$ $(Term2, BOOL) BR$
Complement $Term2 \rightarrow Term2$

Aliases

o $\$u : (Term2, BOOL) BR$
 \sim $Complement : Term2 \rightarrow Term2$

Type Abbreviations

Term2 $Term2$

Fixity

Right Infix 300:

a e i u

Definitions

a $\vdash \forall A B \bullet A a B \Leftrightarrow (\forall x \bullet A x \Rightarrow B x)$
 e $\vdash \forall A B \bullet A e B \Leftrightarrow (\forall x \bullet A x \Rightarrow \neg B x)$
 i $\vdash \forall A B \bullet A i B \Leftrightarrow (\exists x \bullet A x \wedge B x)$
 u $\vdash \forall A B \bullet A o B \Leftrightarrow (\exists x \bullet A x \wedge \neg B x)$
Complement $\vdash \forall A \alpha \bullet \sim A \alpha \Leftrightarrow \neg A \alpha$

Theorems

<i>e_conv_thm</i>	$A e B \vdash B e A$
<i>i_conv_thm</i>	$A i B \vdash B i A$
<i>a_not_conv_thm</i>	$\vdash \exists A B \bullet A a B \wedge \neg B a A$
<i>o_not_conv_thm</i>	$\vdash \exists A B \bullet A o B \wedge \neg B o A$
<i>ae_obv_thm</i>	$A a B \vdash A e \sim B$
<i>ea_obv_thm</i>	$A e B \vdash A a \sim B$
<i>io_obv_thm</i>	$A i B \vdash A o \sim B$
<i>oi_obv_thm</i>	$A o B \vdash A i \sim B$
<i>ao_contrad_thm</i>	$\vdash A a B \Leftrightarrow \neg A o B$
<i>ei_contrad_thm</i>	$\vdash A e B \Leftrightarrow \neg A i B$
<i>¬ae_contrar_thm</i>	$\vdash \neg (\forall A B \bullet \neg (A a B \wedge A e B))$
<i>¬IU_thm</i>	$\vdash \neg (\forall A B \bullet A i B \vee A o B)$
<i>¬ai_subalt_thm</i>	$\vdash \neg (\forall A B \bullet A a B \Rightarrow A i B)$
<i>¬eo_subalt_thm</i>	$\vdash \neg (\forall A B \bullet A e B \Rightarrow A o B)$
<i>Barbara</i>	$M a P, S a M \vdash S a P$
<i>Celarent</i>	$M e P, S a M \vdash S e P$
<i>Darii</i>	$M a P, S i M \vdash S i P$
<i>Ferio</i>	$M e P, S i M \vdash S o P$
<i>Cesare</i>	$P e M, S a M \vdash S e P$
<i>Camestres</i>	$P a M, S e M \vdash S e P$
<i>Festino</i>	$P e M, S i M \vdash S o P$
<i>Baroco</i>	$P a M, S o M \vdash S o P$
<i>Disamis</i>	$M i P, M a S \vdash S i P$
<i>Datisi</i>	$M a P, M i S \vdash S i P$
<i>Bocardo</i>	$M o P, M a S \vdash S o P$
<i>Ferison</i>	$M e P, M i S \vdash S o P$
<i>Camenes</i>	$P a M, M e S \vdash S e P$
<i>Dimaris</i>	$P i M, M a S \vdash S i P$
<i>Fresison</i>	$P e M, M i S \vdash S o P$

A.5 The Theory syllog4

Parents

aristotle

Constants

p $Term3 \rightarrow 'a \rightarrow BOOL$
 $\$a$ $(Term3, BOOL) BR$
 $\$e$ $(Term3, BOOL) BR$
 $\$i$ $(Term3, BOOL) BR$
 $\$u$ $(Term3, BOOL) BR$

Aliases

o $\$u : (Term3, BOOL) BR$

Type Abbreviations

$Term3$ $Term3$

Fixity

Right Infix 300:

a e i u

Definitions

p $\vdash \forall A \bullet p A = (\lambda x \bullet (let (v, f) = A in x = v \vee f x))$
 a $\vdash \forall A B \bullet A a B \Leftrightarrow (\forall x \bullet p A x \Rightarrow p B x)$
 e $\vdash \forall A B \bullet A e B \Leftrightarrow (\forall x \bullet p A x \Rightarrow \neg p B x)$
 i $\vdash \forall A B \bullet A i B \Leftrightarrow (\exists x \bullet p A x \wedge p B x)$
 u $\vdash \forall A B \bullet A o B \Leftrightarrow (\exists x \bullet p A x \wedge \neg p B x)$

Theorems

<i>p_∃_lemma</i>	$\vdash \forall A \bullet \exists v \bullet p A v$
<i>e_conv_thm</i>	$A e B \vdash B e A$
<i>i_conv_thm</i>	$A i B \vdash B i A$
<i>ai_conv_thm</i>	$A a B \vdash B i A$
<i>eo_conv_thm</i>	$A e B \vdash B o A$
<i>sg_09_thm</i>	$\vdash A a B \Leftrightarrow \neg A o B$
<i>sg_10_thm</i>	$\vdash A e B \Leftrightarrow \neg A i B$
<i>sg_11_thm</i>	$\vdash \neg (A a B \wedge A e B)$
<i>sg_12_thm</i>	$\vdash A i B \vee A o B$
<i>sg_13_thm</i>	$A a B \vdash A i B$
<i>sg_14_thm</i>	$A e B \vdash A o B$
<i>Barbara</i>	$M a P, S a M \vdash S a P$
<i>Celarent</i>	$M e P, S a M \vdash S e P$
<i>Darii</i>	$M a P, S i M \vdash S i P$
<i>Ferio</i>	$M e P, S i M \vdash S o P$
<i>Cesare</i>	$P e M, S a M \vdash S e P$
<i>Camestres</i>	$P a M, S e M \vdash S e P$
<i>Festino</i>	$P e M, S i M \vdash S o P$
<i>Baroco</i>	$P a M, S o M \vdash S o P$
<i>Disamis</i>	$M i P, M a S \vdash S i P$
<i>Datisi</i>	$M a P, M i S \vdash S i P$
<i>Bocardo</i>	$M o P, M a S \vdash S o P$
<i>Ferison</i>	$M e P, M i S \vdash S o P$
<i>Camenes</i>	$P a M, M e S \vdash S e P$
<i>Dimaris</i>	$P i M, M a S \vdash S i P$
<i>Fresison</i>	$P e M, M i S \vdash S o P$
<i>Darapti</i>	$M a P, M a S \vdash S i P$
<i>Felapton</i>	$M e P, M a S \vdash S o P$
<i>Bramantip</i>	$P a M, M a S \vdash S i P$
<i>Fesapo</i>	$P e M, M a S \vdash S o P$
<i>Barbari</i>	$M a P, S a M \vdash S i P$
<i>Celaront</i>	$M e P, S a M \vdash S o P$
<i>Cesaro</i>	$P e M, S a M \vdash S o P$
<i>Camestrop</i>	$P a M, S e M \vdash S o P$
<i>Camenop</i>	$P a M, M e S \vdash S o P$

A.6 The Theory syllog5

Parents

aristotle

Constants

$\$a$ (*Term2*, *BOOL*) *BR*
 $\$e$ (*Term2*, *BOOL*) *BR*
 $\$i$ (*Term2*, *BOOL*) *BR*
 $\$u$ (*Term2*, *BOOL*) *BR*
Complement *Term2* \rightarrow *Term2*

Aliases

o $\$u$: (*Term2*, *BOOL*) *BR*
 \sim *Complement* : *Term2* \rightarrow *Term2*

Type Abbreviations

Term2 *Term2*

Fixity

Right Infix 300:

a *e* *i* *u*

Definitions

a $\vdash \forall A B$
 • $A a B$
 $\Leftrightarrow (\forall x \bullet A x \Rightarrow B x) \wedge (\exists x \bullet A x) \wedge (\exists x \bullet \neg B x)$
e $\vdash \forall A B$
 • $A e B$
 $\Leftrightarrow (\forall x \bullet A x \Rightarrow \neg B x) \wedge (\exists x \bullet A x) \wedge (\exists x \bullet B x)$
i $\vdash \forall A B$
 • $A i B$
 $\Leftrightarrow (\exists x \bullet A x \wedge B x) \vee \neg (\exists x \bullet A x) \vee \neg (\exists x \bullet B x)$
u $\vdash \forall A B$
 • $A o B$
 $\Leftrightarrow (\exists x \bullet A x \wedge \neg B x)$
 $\vee \neg (\exists x \bullet A x)$
 $\vee \neg (\exists x \bullet \neg B x)$
Complement $\vdash \forall A \alpha \bullet \sim A \alpha \Leftrightarrow \neg A \alpha$

Theorems

<i>e_conv_thm</i>	$A e B \vdash B e A$
<i>i_conv_thm</i>	$A i B \vdash B i A$
$\neg a_conv_thm$	$\vdash \exists A B \bullet A a B \wedge \neg B a A$
$\neg o_conv_thm$	$\vdash \exists A B \bullet A o B \wedge \neg B o A$
<i>ai_conv_thm</i>	$A a B \vdash B i A$
<i>eo_conv_thm</i>	$A e B \vdash B o A$
<i>ae_obv_thm</i>	$A a B \vdash A e \sim B$
<i>ea_obv_thm</i>	$A e B \vdash A a \sim B$
<i>io_obv_thm</i>	$A i B \vdash A o \sim B$
<i>oi_obv_thm</i>	$A o B \vdash A i \sim B$
<i>ao_contrad_thm</i>	$\vdash A a B \Leftrightarrow \neg A o B$
<i>ei_contrad_thm</i>	$\vdash A e B \Leftrightarrow \neg A i B$
<i>ae_contrar_thm</i>	$\vdash \neg (A a B \wedge A e B)$
<i>io_subcont_thm</i>	$\vdash A i B \vee A o B$
<i>ai_subalt_thm</i>	$\vdash A a B \Rightarrow A i B$
<i>eo_subalt_thm</i>	$\vdash A e B \Rightarrow A o B$
<i>Barbara</i>	$M a P, S a M \vdash S a P$
<i>Celarent</i>	$M e P, S a M \vdash S e P$
<i>Darii</i>	$M a P, S i M \vdash S i P$
<i>Ferio</i>	$M e P, S i M \vdash S o P$
<i>Cesare</i>	$P e M, S a M \vdash S e P$
<i>Camestres</i>	$P a M, S e M \vdash S e P$
<i>Festino</i>	$P e M, S i M \vdash S o P$
<i>Baroco</i>	$P a M, S o M \vdash S o P$
<i>Disamis</i>	$M i P, M a S \vdash S i P$
<i>Datisi</i>	$M a P, M i S \vdash S i P$
<i>Bocardo</i>	$M o P, M a S \vdash S o P$
<i>Ferison</i>	$M e P, M i S \vdash S o P$
<i>Camenes</i>	$P a M, M e S \vdash S e P$
<i>Dimaris</i>	$P i M, M a S \vdash S i P$
<i>Fresison</i>	$P e M, M i S \vdash S o P$
<i>Darapti</i>	$M a P, M a S \vdash S i P$
<i>Felapton</i>	$M e P, M a S \vdash S o P$
<i>Bramantip</i>	$P a M, M a S \vdash S i P$
<i>Fesapo</i>	$P e M, M a S \vdash S o P$
<i>Barbari</i>	$M a P, S a M \vdash S i P$
<i>Celaront</i>	$M e P, S a M \vdash S o P$
<i>Cesaro</i>	$P e M, S a M \vdash S o P$
<i>Camestrop</i>	$P a M, S e M \vdash S o P$
<i>Camenop</i>	$P a M, M e S \vdash S o P$

A.7 The Theory syllog6

Parents

aristotle

Constants

$\$a$ $(Term2, BOOL) BR$
 $\$e$ $(Term2, BOOL) BR$
 $\$i$ $(Term2, BOOL) BR$
 $\$u$ $(Term2, BOOL) BR$
Complement $Term2 \rightarrow Term2$

Aliases

o $\$u : (Term2, BOOL) BR$
 \sim $Complement : Term2 \rightarrow Term2$

Type Abbreviations

Term2 $Term2$

Fixity

Right Infix 300:

a e i u

Definitions

a $\vdash \forall A B \bullet A a B \Leftrightarrow (\forall x \bullet A x \Rightarrow B x) \wedge (\exists x \bullet A x)$
 e $\vdash \forall A B \bullet A e B \Leftrightarrow (\forall x \bullet A x \Rightarrow \neg B x)$
 i $\vdash \forall A B \bullet A i B \Leftrightarrow (\exists x \bullet A x \wedge B x)$
 u $\vdash \forall A B \bullet A o B \Leftrightarrow (\exists x \bullet A x \wedge \neg B x) \vee \neg (\exists x \bullet A x)$
Complement $\vdash \forall A \alpha \bullet \sim A \alpha \Leftrightarrow \neg A \alpha$

Theorems

<i>e_conv_thm</i>	$A e B \vdash B e A$
<i>i_conv_thm</i>	$A i B \vdash B i A$
<i>ai_conv_thm</i>	$A a B \vdash B i A$
<i>eo_conv_thm</i>	$A e B \vdash B o A$
<i>ae_obv_thm</i>	$A a B \vdash A e \sim B$
<i>iu_obv_thm</i>	$A i B \vdash A o \sim B$
<i>ao_contrad_thm</i>	$\vdash A a B \Leftrightarrow \neg A o B$
<i>ei_contrad_thm</i>	$\vdash A e B \Leftrightarrow \neg A i B$
<i>ae_contrar_thm</i>	$\vdash \neg (A a B \wedge A e B)$
<i>io_subcont_thm</i>	$\vdash A i B \vee A o B$
<i>ai_subalt_thm</i>	$\vdash A a B \Rightarrow A i B$
<i>eo_subalt_thm</i>	$\vdash A e B \Rightarrow A o B$
<i>Barbara</i>	$M a P, S a M \vdash S a P$
<i>Celarent</i>	$M e P, S a M \vdash S e P$
<i>Darii</i>	$M a P, S i M \vdash S i P$
<i>Ferio</i>	$M e P, S i M \vdash S o P$
<i>Cesare</i>	$P e M, S a M \vdash S e P$
<i>Camestres</i>	$P a M, S e M \vdash S e P$
<i>Festino</i>	$P e M, S i M \vdash S o P$
<i>Baroco</i>	$P a M, S o M \vdash S o P$
<i>Disamis</i>	$M i P, M a S \vdash S i P$
<i>Datisi</i>	$M a P, M i S \vdash S i P$
<i>Bocardo</i>	$M o P, M a S \vdash S o P$
<i>Ferison</i>	$M e P, M i S \vdash S o P$
<i>Camenes</i>	$P a M, M e S \vdash S e P$
<i>Dimaris</i>	$P i M, M a S \vdash S i P$
<i>Fresison</i>	$P e M, M i S \vdash S o P$
<i>Darapti</i>	$M a P, M a S \vdash S i P$
<i>Felapton</i>	$M e P, M a S \vdash S o P$
<i>Bramantip</i>	$P a M, M a S \vdash S i P$
<i>Fesapo</i>	$P e M, M a S \vdash S o P$
<i>Barbari</i>	$M a P, S a M \vdash S i P$
<i>Celaront</i>	$M e P, S a M \vdash S o P$
<i>Cesaro</i>	$P e M, S a M \vdash S o P$
<i>Camestrop</i>	$P a M, S e M \vdash S o P$
<i>Camenop</i>	$P a M, M e S \vdash S o P$

A.8 The Theory modsyllog

Parents

aristotle

Constants

p	$MPROP \rightarrow W \rightarrow I \rightarrow BOOL$
$\$a$	$(MPROP, W \rightarrow BOOL) BR$
$\$e$	$(MPROP, W \rightarrow BOOL) BR$
$\$i$	$(MPROP, W \rightarrow BOOL) BR$
$\$u$	$(MPROP, W \rightarrow BOOL) BR$
<i>actual_world</i>	W
\diamond	$(W \rightarrow BOOL) \rightarrow BOOL$
\square	$(W \rightarrow BOOL) \rightarrow BOOL$
\diamond_a	$(W \rightarrow BOOL) \rightarrow BOOL$
\models	$(W \rightarrow BOOL) \rightarrow BOOL$
$\$\neg_m$	$(W \rightarrow BOOL) \rightarrow W \rightarrow BOOL$

Aliases

o	$\$u : (MPROP, W \rightarrow BOOL) BR$
\neg	$\$\neg_m : (W \rightarrow BOOL) \rightarrow W \rightarrow BOOL$

Types

I
 W

Type Abbreviations

$MPROP$ $MPROP$

Fixity

Right Infix 300:

a e i u

Prefix 350:

\neg_m

Definitions

p	$\vdash \forall A \bullet p A = (\lambda w x \bullet (let (v, f) = A w in x = v \vee f x))$
a	$\vdash \forall A B w \bullet (A a B) w \Leftrightarrow (\forall x \bullet p A w x \Rightarrow p B w x)$
e	$\vdash \forall A B w \bullet (A e B) w \Leftrightarrow (\forall x \bullet p A w x \Rightarrow \neg p B w x)$
i	$\vdash \forall A B w \bullet (A i B) w \Leftrightarrow (\exists x \bullet p A w x \wedge p B w x)$
u	$\vdash \forall A B w \bullet (A o B) w \Leftrightarrow (\exists x \bullet p A w x \wedge \neg p B w x)$
<i>actual_world</i>	$\vdash T$
\diamond	$\vdash \forall s \bullet \diamond s \Leftrightarrow (\exists w \bullet s w)$
\square	$\vdash \forall s \bullet \square s \Leftrightarrow (\forall w \bullet s w)$
\diamond_a	$\vdash \forall s \bullet \diamond_a s \Leftrightarrow \neg (\forall w \bullet s w) \wedge \neg (\forall w \bullet \neg s w)$
\models	$\vdash \forall s \bullet \models s \Leftrightarrow s \text{ actual_world}$
\neg_m	$\vdash \forall x \bullet (\neg x) = (\lambda w \bullet \neg x w)$

Theorems

<i>p_∃_lemma</i>	$\vdash \forall A w \bullet \exists v \bullet p A w v$
$\Box \Diamond$ _thm	$\Box X \vdash \Diamond X$
$\Box \Vdash$ _thm	$\Box X \vdash \Vdash X$
$\Vdash \Diamond$ _thm	$\Vdash X \vdash \Diamond X$
$\Diamond_a \Diamond$ _thm	$\Diamond_a X \vdash \Diamond X$
$\Diamond_a \neg \Box$ _thm	$\Diamond_a X \vdash \neg \Box X$
$\Diamond_a \neg_m$ _thm	$\vdash \Diamond_a X \Leftrightarrow \Diamond_a (\neg X)$
<i>e_conv_thm</i>	$\Vdash (A e B) \vdash \Vdash (B e A)$
<i>i_conv_thm</i>	$\Vdash (A i B) \vdash \Vdash (B i A)$
$\Box e_conv_thm$	$\Box (A e B) \vdash \Box (B e A)$
$\Box i_conv_thm$	$\Box (A i B) \vdash \Box (B i A)$
$\Diamond e_conv_thm$	$\Diamond (A e B) \vdash \Diamond (B e A)$
$\Diamond i_conv_thm$	$\Diamond (A i B) \vdash \Diamond (B i A)$
$\Diamond_a e_conv_thm$	$\Diamond_a (A e B) \vdash \Diamond_a (B e A)$
$\Diamond_a i_conv_thm$	$\Diamond_a (A i B) \vdash \Diamond_a (B i A)$
$\Diamond_a ao_conv_thm$	$\vdash \Diamond_a (A a B) \Leftrightarrow \Diamond_a (A o B)$
$\Diamond_a ei_conv_thm$	$\vdash \Diamond_a (A e B) \Leftrightarrow \Diamond_a (A i B)$
$\Box \Diamond e_conv_thm$	$\Box (A e B) \vdash \Diamond (B e A)$
$\Box \Vdash e_conv_thm$	$\Box (A e B) \vdash \Vdash (B e A)$
$\Vdash ai_conv_thm$	$\Vdash (A a B) \vdash \Vdash (B i A)$
$\Vdash eo_conv_thm$	$\Vdash (A e B) \vdash \Vdash (B o A)$
$\Box \Diamond ai_conv_thm$	$\Box (A a B) \vdash \Diamond (B i A)$
$\Box \Diamond eo_conv_thm$	$\Box (A e B) \vdash \Diamond (B o A)$
<i>mod_gen_□□□</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \Box CS$
<i>mod_gen_□□◇</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \Diamond CS$
<i>mod_gen_□□\Vdash</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Box SP \Rightarrow \Vdash CS$
<i>mod_gen_◇□◇</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Diamond FP \wedge \Box SP \Rightarrow \Diamond CS$
<i>mod_gen_□◇◇</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Diamond SP \Rightarrow \Diamond CS$
<i>mod_gen_□\Vdash\Vdash</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Box FP \wedge \Vdash SP \Rightarrow \Vdash CS$
<i>mod_gen_\Vdash□\Vdash</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Vdash FP \wedge \Box SP \Rightarrow \Vdash CS$
<i>mod_gen_\Vdash\Vdash\Vdash</i>	$\vdash \forall FP SP CS$ $\bullet (\forall w \bullet FP w \wedge SP w \Rightarrow CS w) \Rightarrow \Vdash FP \wedge \Vdash SP \Rightarrow \Vdash CS$

A.9 The Theory *syllmetap*

Parents

aristotle

Children

gcccon

Constants

Category_of_Substance

	<i>CAT</i>
<i>actual_world</i>	<i>W</i>
<i>tag</i>	$I \rightarrow TAG$
<i>category</i>	$I \rightarrow CAT$
<i>mk_SubstTerm</i>	$TAG \mathbb{P} \rightarrow TermM$
<i>mk_AttrTerm</i>	$CAT \times TAG \mathbb{P} \rightarrow TermM$
<i>Substance</i>	<i>TermM</i>
<i>\$not</i>	$COPULA \rightarrow COPULA$
<i>\$InTermM</i>	$I \rightarrow TermM \rightarrow BOOL$
<i>All</i>	$TermM \rightarrow COPULA \rightarrow TermM \rightarrow MPROP$
<i>Some</i>	$TermM \rightarrow COPULA \rightarrow TermM \rightarrow MPROP$
<i>izz</i>	<i>COPULA</i>
<i>hazz</i>	<i>COPULA</i>
<i>are</i>	<i>COPULA</i>
\diamond	$MPROP \rightarrow MPROP$
\square	$MPROP \rightarrow MPROP$
\neg_a	$MPROP \rightarrow MPROP$
$\$\wedge_a$	$(MPROP, MPROP) BR$
$\$\Rightarrow_a$	$(MPROP, MPROP) BR$
$\$\Leftrightarrow_a$	$(MPROP, MPROP) BR$
$\$\forall_a$	$(TermM \rightarrow MPROP) \rightarrow MPROP$
$\$\exists_a$	$(TermM \rightarrow MPROP) \rightarrow MPROP$
$\$\models$	$MPROP LIST \rightarrow MPROP \rightarrow BOOL$
$\$\Vdash$	$MPROP LIST \rightarrow MPROP \rightarrow BOOL$

Aliases

\neg	$\neg_a : MPROP \rightarrow MPROP$
\wedge	$\$\wedge_a : (MPROP, MPROP) BR$
\Rightarrow	$\$\Rightarrow_a : (MPROP, MPROP) BR$
\Leftrightarrow	$\$\Leftrightarrow_a : (MPROP, MPROP) BR$
\forall	$\$\forall_a : (TermM \rightarrow MPROP) \rightarrow MPROP$
\exists	$\$\exists_a : (TermM \rightarrow MPROP) \rightarrow MPROP$

Types

ACAT

TAG

Type Abbreviations

<i>CAT</i>	<i>CAT</i>
<i>I</i>	<i>I</i>
<i>W</i>	<i>W</i>
<i>TermM</i>	<i>TermM</i>
<i>COPULA</i>	<i>COPULA</i>
<i>MPROP</i>	<i>MPROP</i>

Fixity

<i>Binder:</i>	\forall_a	\exists_a
<i>Right Infix 100:</i>	\Vdash	\Vdash
<i>Right Infix 200:</i>	\Leftrightarrow_a	
<i>Right Infix 210:</i>	\Rightarrow_a	
<i>Right Infix 220:</i>	\wedge_a	
<i>Right Infix 300:</i>	InTermM	
<i>Postfix 100:</i>	not	

Definitions

Category_of_Substance

$\vdash \text{Category_of_Substance} = \text{InL One}$

actual_world $\vdash T$

category

tag $\vdash \forall ct \bullet \text{category } ct = \text{Fst } ct \wedge \text{tag } ct = \text{Snd } ct$

mk_SubstTerm $\vdash \forall s \bullet \text{mk_SubstTerm } s = (\text{Category_of_Substance}, \text{NeSet } s)$

mk_AttrTerm $\vdash \forall s \bullet \text{mk_AttrTerm } s = (\text{Fst } s, \text{NeSet } (\text{Snd } s))$

Substance $\vdash \text{Substance} = \text{mk_SubstTerm Universe}$

not $\vdash \forall \text{pred} \bullet (\text{pred not}) = (\lambda pa \ t \ w \bullet \neg \text{pred } pa \ t \ w)$

InTermM $\vdash \forall c1t \ c2ts$

• $c1t \text{ InTermM } c2ts$

$\Leftrightarrow \text{Fst } c1t = \text{Fst } c2ts \wedge \text{Snd } c1t \in \text{PeSet } (\text{Snd } c2ts)$

All $\vdash \forall s \ r \ p$

• $\text{All } s \ r \ p = (\lambda w \bullet \forall z \bullet z \text{ InTermM } s \Rightarrow r \ z \ p \ w)$

Some $\vdash \forall s \ r \ p$

• $\text{Some } s \ r \ p = (\lambda w \bullet \exists z \bullet z \text{ InTermM } s \wedge r \ z \ p \ w)$

izz $\vdash \forall j \ t \bullet \text{izz } j \ t = (\lambda w \bullet j \text{ InTermM } t)$

hazz $\vdash \forall c1t \ c2ts$

• $\text{hazz } c1t \ c2ts$

$= (\lambda w$

• $\text{Fst } c1t = \text{Category_of_Substance}$

$\wedge \neg \text{Fst } c2ts = \text{Category_of_Substance}$

$\wedge (\exists b$

• $b \in \text{PeSet } (\text{Snd } c2ts)$

$\wedge \text{Snd } c1t \in w (\text{OutR } (\text{Fst } c2ts), b))$

are	$\vdash \forall pa\ t \bullet are\ pa\ t = (\lambda w \bullet izz\ pa\ t\ w \vee hazz\ pa\ t\ w)$
\diamond	$\vdash \forall p \bullet \diamond p = (\lambda w \bullet \exists w' \bullet p\ w')$
\square	$\vdash \forall p \bullet \square p = (\lambda w \bullet \forall w' \bullet p\ w')$
\neg_a	$\vdash \forall p \bullet (\neg p) = (\lambda w \bullet \neg p\ w)$
\wedge_a	$\vdash \forall p\ q \bullet (p \wedge q) = (\lambda w \bullet p\ w \wedge q\ w)$
\Rightarrow_a	$\vdash \forall p\ q \bullet (p \Rightarrow q) = (\lambda w \bullet p\ w \Rightarrow q\ w)$
\Leftrightarrow_a	$\vdash \forall p\ q \bullet (p \Leftrightarrow q) = (\lambda w \bullet p\ w \Leftrightarrow q\ w)$
\forall_a	$\vdash \forall mpf \bullet \$\forall mpf = (\lambda w \bullet \forall t \bullet mpf\ t\ w)$
\exists_a	$\vdash \forall mpf \bullet \$\exists mpf = (\lambda w \bullet \exists t \bullet mpf\ t\ w)$
\vDash	$\vdash \forall lp\ c$ <ul style="list-style-type: none"> • $lp \vDash c$ $\Leftrightarrow Fold\ (\lambda p\ t \bullet p\ actual_world \wedge t)\ lp\ T$ $\Rightarrow c\ actual_world$
\Vdash	$\vdash \forall lp\ c$ <ul style="list-style-type: none"> • $lp \Vdash c \Leftrightarrow (\forall w \bullet Fold\ (\lambda p\ t \bullet p\ w \wedge t)\ lp\ T \Rightarrow c\ w)$

Theorems

interm_∃_lemma

$$\vdash \forall t \bullet \exists j \bullet j\ InTermM\ t$$

not_izz_and_hazz_lemma1

$$\vdash \forall pa\ t\ w \bullet \neg (izz\ pa\ t\ w \wedge hazz\ pa\ t\ w)$$

are_izz_neq_hazz_lemma

$$\vdash \forall pa\ t\ w \bullet are\ pa\ t\ w \Leftrightarrow \neg (izz\ pa\ t\ w \Leftrightarrow hazz\ pa\ t\ w)$$

All_are_izz_or_hazz_lemma

$$\vdash \forall A\ B\ w$$

- $All\ A\ are\ B\ w \Leftrightarrow All\ A\ izz\ B\ w \vee All\ A\ hazz\ B\ w$

Some_are_izz_or_hazz_lemma

$$\vdash \forall A\ B\ w$$

- $Some\ A\ are\ B\ w \Leftrightarrow Some\ A\ izz\ B\ w \vee Some\ A\ hazz\ B\ w$

All_are_not_lemma

$$\vdash \forall A\ B\ w$$

- $All\ A\ (are\ not)\ B\ w$
 $\Leftrightarrow All\ A\ (izz\ not)\ B\ w \wedge All\ A\ (hazz\ not)\ B\ w$

izz_not_lemma

$$\vdash All\ B\ (izz\ not)\ A = All\ A\ (izz\ not)\ B$$

some_izz_lemma

$$\vdash Some\ B\ izz\ A = Some\ A\ izz\ B$$

$$\text{izz_conv1} \quad \vdash [All\ B\ (izz\ not)\ A] \Vdash All\ A\ (izz\ not)\ B$$

$$\text{izz_conv2} \quad \vdash [All\ B\ izz\ A] \Vdash Some\ A\ izz\ B$$

$$\text{izz_conv3} \quad \vdash [Some\ B\ izz\ A] \Vdash Some\ A\ izz\ B$$

not_hazz_not_lemma

$$\vdash \neg (\forall A\ B \bullet All\ B\ (hazz\ not)\ A = All\ A\ (hazz\ not)\ B)$$

not_some_hazz_lemma

$$\vdash \neg (\forall A\ B \bullet Some\ B\ hazz\ A = Some\ A\ hazz\ B)$$

$$\text{hazz_conv2} \quad \vdash [All\ A\ hazz\ B] \Vdash Some\ A\ hazz\ B$$

$$\diamond_conv \quad \vdash [P] \Vdash Q \Rightarrow [\diamond P] \Vdash \diamond Q$$

$$\square_conv \quad \vdash [P] \Vdash Q \Rightarrow [\square P] \Vdash \square Q$$

$$\square_izz_thm \quad \vdash [\square (All\ A\ izz\ B)] \vDash All\ A\ izz\ B$$

$$\square_hazz_thm \quad \vdash [\square (All\ A\ hazz\ B)] \vDash All\ A\ izz\ B$$

$$\text{izz_}\square_thm \quad \vdash [All\ A\ izz\ B] \vDash \square (All\ A\ izz\ B)$$

not_}\square_hazz_thm

$\vdash [] \models (\neg \square (All\ A\ hazz\ B))$
 \square_izz_thm2 $\vdash [\square (All\ A\ izz\ B)] \Vdash All\ A\ izz\ B$
 \square_hazz_thm2 $\vdash [\square (All\ A\ hazz\ B)] \Vdash All\ A\ izz\ B$
 izz_square_thm2 $\vdash [All\ A\ izz\ B] \Vdash \square (All\ A\ izz\ B)$
 $not_square_hazz_thm2$
 $\vdash [] \Vdash (\neg \square (All\ A\ hazz\ B))$
 \square_elim_thm $\vdash [\square P] \models P$
 \square_intro_thm $\vdash [P] \models \diamond P$
 $\square_diamond_thm$ $\vdash [\square P] \models \diamond P$
 $\neg_All_conv_thm$
 $\vdash (\neg All\ A\ cop\ B) = Some\ A\ (cop\ not)\ B$
 $\neg_All_not_conv_thm2$
 $\vdash (\neg All\ A\ (cop\ not)\ B) = Some\ A\ cop\ B$
 $\neg_Some_conv_thm$
 $\vdash (\neg Some\ A\ cop\ B) = All\ A\ (cop\ not)\ B$
 $\neg_Some_not_conv_thm$
 $\vdash (\neg Some\ A\ (cop\ not)\ B) = All\ A\ cop\ B$
 $Barbara_izz$ $\vdash [All\ M\ izz\ P; All\ S\ izz\ M] \Vdash All\ S\ izz\ P$
 $Celarent_izz$ $\vdash [All\ M\ (izz\ not)\ P; All\ S\ izz\ M] \Vdash All\ S\ (izz\ not)\ P$
 $Darii_izz$ $\vdash [All\ M\ izz\ P; Some\ S\ izz\ M] \Vdash Some\ S\ izz\ P$
 $Ferio_izz$ $\vdash [All\ M\ (izz\ not)\ P; Some\ S\ izz\ M]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Cesare_izz$ $\vdash [All\ P\ (izz\ not)\ M; All\ S\ izz\ M] \Vdash All\ S\ (izz\ not)\ P$
 $Camestres_izz$
 $\vdash [All\ P\ izz\ M; All\ S\ (izz\ not)\ M] \Vdash All\ S\ (izz\ not)\ P$
 $Festino_izz$ $\vdash [All\ P\ (izz\ not)\ M; Some\ S\ izz\ M]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Baroco_izz$ $\vdash [All\ P\ izz\ M; Some\ S\ (izz\ not)\ M]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Disamis_izz$ $\vdash [Some\ M\ izz\ P; All\ M\ izz\ S] \Vdash Some\ S\ izz\ P$
 $Datisi_izz$ $\vdash [All\ M\ izz\ P; Some\ M\ izz\ S] \Vdash Some\ S\ izz\ P$
 $Bocardo_izz$ $\vdash [Some\ M\ (izz\ not)\ P; All\ M\ izz\ S]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Ferison_izz$ $\vdash [All\ M\ (izz\ not)\ P; Some\ M\ izz\ S]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Camenes_izz$ $\vdash [All\ P\ izz\ M; All\ M\ (izz\ not)\ S] \Vdash All\ S\ (izz\ not)\ P$
 $Dimaris_izz$ $\vdash [Some\ P\ izz\ M; All\ M\ izz\ S] \Vdash Some\ S\ izz\ P$
 $Fresison_izz$ $\vdash [All\ P\ (izz\ not)\ M; Some\ M\ izz\ S]$
 $\quad \Vdash Some\ S\ (izz\ not)\ P$
 $Darapti_izz$ $\vdash [All\ M\ izz\ P; All\ M\ izz\ S] \Vdash Some\ S\ izz\ P$
 $Felapton_izz$ $\vdash [All\ M\ (izz\ not)\ P; All\ M\ izz\ S] \Vdash Some\ S\ (izz\ not)\ P$
 $Bramantip_izz$
 $\vdash [All\ P\ izz\ M; All\ M\ izz\ S] \Vdash Some\ S\ izz\ P$
 $Fesapo_izz$ $\vdash [All\ P\ (izz\ not)\ M; All\ M\ izz\ S] \Vdash Some\ S\ (izz\ not)\ P$
 $Barbari_izz$ $\vdash [All\ M\ izz\ P; All\ S\ izz\ M] \Vdash Some\ S\ izz\ P$
 $Celaront_izz$ $\vdash [All\ M\ (izz\ not)\ P; All\ S\ izz\ M] \Vdash Some\ S\ (izz\ not)\ P$
 $Cesaro_izz$ $\vdash [All\ P\ (izz\ not)\ M; All\ S\ izz\ M] \Vdash Some\ S\ (izz\ not)\ P$
 $Camestrop_izz$
 $\vdash [All\ P\ izz\ M; All\ S\ (izz\ not)\ M] \Vdash Some\ S\ (izz\ not)\ P$
 $Camenop_izz$ $\vdash [All\ P\ izz\ M; All\ M\ (izz\ not)\ S] \Vdash Some\ S\ (izz\ not)\ P$

A.10 The Theory gcon

Parents

syllmetap

Constants

individual $TermM \rightarrow BOOL$
Mp $BOOL \rightarrow MPROP$
Individual $TermM \rightarrow MPROP$
particular $TermM \rightarrow MPROP$
substantial $TermM \rightarrow MPROP$
universal $TermM \rightarrow MPROP$

Definitions

individual $\vdash \forall A \bullet \text{individual } A \Leftrightarrow (\exists b \bullet Snd\ A = NeSet\ \{b\})$
Mp $\vdash \forall p \bullet Mp\ p = (\lambda w \bullet p)$
Individual $\vdash \forall A$
 • *Individual* $A = \square (\forall B \bullet All\ B\ izz\ A \Rightarrow All\ A\ izz\ B)$
particular $\vdash \forall A$
 • *particular* A
 = \square
 $(\forall B \bullet All\ B\ are\ A \Rightarrow All\ A\ izz\ B \wedge All\ B\ izz\ A)$
substantial $\vdash \forall A$
 • *substantial* $A = \square (\forall B \bullet All\ B\ are\ A \Rightarrow All\ B\ izz\ A)$
universal $\vdash \forall A$
 • *universal* A
 = \diamond
 $(\exists B$
 • *All B are A*
 $\wedge \neg (All\ A\ izz\ B \wedge All\ B\ izz\ A))$

Theorems

FP1 $\vdash \square \Vdash All\ A\ izz\ A$
FP2 $\vdash [All\ A\ izz\ B; All\ B\ izz\ C] \Vdash All\ A\ izz\ C$
FP3 $\vdash [All\ A\ hazz\ B] \Vdash (\neg All\ A\ izz\ B)$
FP3b $\vdash [All\ A\ hazz\ B] \Vdash Some\ A\ (izz\ not)\ B$
FP4a $\vdash [All\ A\ hazz\ B; All\ B\ izz\ C] \Vdash All\ A\ hazz\ C$
FP4 $\vdash \square$
 $\Vdash (All\ A\ hazz\ B$
 $\Leftrightarrow (\exists C \bullet All\ A\ hazz\ C \wedge All\ C\ izz\ B))$
FP4c $\vdash [Mp\ (individual\ A)]$
 $\Vdash (All\ A\ hazz\ B$
 $\Leftrightarrow (\exists C$
 • *Mp (individual C)*
 $\wedge All\ A\ hazz\ C$
 $\wedge All\ C\ izz\ B))$

individual_lemma1

$\vdash \forall A \bullet \text{Individual } A = (\forall B \bullet \text{All } B \text{ izz } A \Rightarrow \text{All } A \text{ izz } B)$
individual_lemma2
 $\vdash \text{Individual } A = \text{Mp } (\text{individual } A)$

Bibliography

- [1] Aristotle. *Categories. On Interpretation. Prior Analytics*, volume 325 of *Loeb Classical Library*. Harvard University Press. 1938.
see also: <http://texts.rbjones.com/rbjpub/philos/classics/aristotl/oi.htm>.
- [2] Aristotle. *Metaphysics*. Penguin Books. 1999.
see also: <http://texts.rbjones.com/rbjpub/philos/classics/aristotl/mi.htm>.
- [3] Alan Code. Aristotle, Essence and Accident. In *Philosophical Grounds of Rationality: Intentions, Categories, Ends*. Oxford University Press, 1988.
- [4] H.P. Grice. Aristotle on the Multiplicity of Being. *Pacific Philosophical Quarterly*, 69:175–200, 1988.
- [5] Roger Bishop Jones. Analyses of Analysis: Part I - Introduction. *RBJones.com*, 2009-.
<http://www.rbjones.com/rbjpub/pp/doc/t029.pdf>.
- [6] Roger Bishop Jones. *Analyses of Analysis: Part I - Exegetical Analysis*. *RBJones.com*. 2009.
<http://www.rbjones.com/rbjpub/pp/doc/b001.pdf>.
- [7] Roger Bishop Jones. Miscellaneous Theory Supplements. *RBJones.com*, 2010.
<http://www.rbjones.com/rbjpub/pp/doc/t006.pdf>.
- [8] Terence Parsons. *Basics of Medieval Formal Logic*. URL <http://www.humnet.ucla.edu/humnet/phil/faculty/tparsons/Medieval%20Book/contents.htm>, work in progress.
- [9] Robin Smith. *Aristotle's Logic*. The Stanford Encyclopedia of Philosophy, spring 2009 edition.
URL <http://plato.stanford.edu/archives/spr2009/entries/aristotle-logic/#Syl>.
- [10] Paul Vincent Spade. *Thoughts, Words and Things: An Introduction to Late Mediaeval Logic and Semantic Theory*. PVSDC, 1.1th edition. URL http://pvspade.com/Logic/docs/thoughts1_1a.pdf, 2002.
- [11] P.F. Strawson. *Introduction to Logical Theory*. Methuen. URL <http://www.archive.org/details/introductiontolo010626mbp>, 1952.
- [12] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Routledge Kegan and Paul. 1961.

i	e_conv	conversion
ii	i_conv	
iii	ai_conv	
iv	eo_conv	
v	ae_obv	obversion
vi	ea_obv	
vii	io_obv	
viii	oi_obv	
ix	ao_contrad	contradiction
x	ei_contrad	
xi	ae_contrar	contrary
xii	io_subcontrar	subcontrary
xiii	ai_subalt	subalternate
xiv	eo_subalt	

Table 4: Laws of Immediate Inference

Index

\square	50, 61, 85, 87, 89
$\square\diamond_thm$	86
$\square\diamond ai_conv_thm$	86
$\square\diamond e_conv_thm$	86
$\square\diamond eo_conv_thm$	86
$\square_ \diamond_thm$	66, 90
\square_conv	65, 89
\square_elim_thm	66, 90
\square_hazz_thm	65, 89
\square_hazz_thm2	65, 90
\square_intro_thm	90
\square_izz_thm	65, 89
\square_izz_thm2	65, 90
$\square \models_thm$	86
$\square \models e_conv_thm$	86
$\square e_conv_thm$	86
$\square i_conv_thm$	86
\diamond	50, 61, 85, 87, 89
\diamond_conv	65, 89
\diamond_intro_thm	66
$\diamond e_conv_thm$	86
$\diamond i_conv_thm$	86
\diamond_a	50, 85
$\diamond_a \diamond_thm$	86
$\diamond_a ao_conv_thm$	86
$\diamond_a e_conv_thm$	86
$\diamond_a ei_conv_thm$	86
$\diamond_a i_conv_thm$	86
$\diamond_a \neg \square_thm$	86
$\diamond_a \neg m_thm$	86
\Leftrightarrow	87
\Leftrightarrow_a	62, 87–89
\Rightarrow	87
\Rightarrow_a	62, 87–89
\Vdash	64, 87–89
\exists	87
\exists_a	63, 87–89
\forall	87
\forall_a	63, 87–89
\wedge	87
\wedge_a	62, 87–89
\neg	85, 87
$\neg IU_thm$	78
$\neg All_conv_thm$	90
$\neg All_not_conv_thm2$	90
$\neg Some_conv_thm$	90
$\neg Some_not_conv_thm$	90
$\neg a_conv_thm$	82
$\neg ae_contrar_thm$	78
$\neg ai_subalt_thm$	78
$\neg eo_subalt_thm$	78
$\neg o_conv_thm$	82
\neg_a	62, 87, 89
\neg_m	51, 85
\neq	5
\sim	77, 81, 83
\vDash	51, 63, 85, 87–89
$\vDash \diamond_thm$	86
$\vDash ai_conv_thm$	86
$\vDash eo_conv_thm$	86
a	33, 37, 41, 45, 49, 77, 79, 81, 83, 85
$a_not_conv_thm$	78
$ACAT$	8, 72, 87
$accidentally_predicable_of$	12, 72, 73
$actual_world$	50, 58, 85, 87, 88
$ae_contrar_thm$	82, 84
ae_obv_thm	78, 82, 84
ai_conv_thm	43, 47, 80, 82, 84
ai_subalt_thm	82, 84
All	60, 87, 88
$All_are_izz_or_hazz_lemma$	61, 89
$All_are_not_lemma$	61, 89
$ao_contrad_thm$	78, 82, 84
are	61, 87, 89
$are_izz_neq_hazz_lemma$	61, 89
$Aristotle$	4
<i>organon</i>	5
$AttrSet$	9, 72, 73
$Barbara$	75, 76, 78, 80, 82, 84
$Barbara_izz$	90
$Barbari$	75, 80, 82, 84
$Barbari_izz$	90
$Baroco$	75, 76, 78, 80, 82, 84
$Baroco_izz$	90
$Bocardo$	75, 76, 78, 80, 82, 84
$Bocardo_izz$	90
$Bramantip$	75, 80, 82, 84
$Bramantip_izz$	90
$Camenes$	75, 76, 78, 80, 82, 84
$Camenes_izz$	90
$Camenop$	75, 80, 82, 84
$Camenop_izz$	90
$Camestres$	75, 76, 78, 80, 82, 84
$Camestres_izz$	90
$Camestrop$	75, 80, 82, 84
$Camestrop_izz$	90
CAT	8, 72, 88
Cat	9, 72, 73
$category$	58, 87, 88
$Category_of_Substance$	57, 87, 88
$CATM$	8, 72
$catm_eq_lemma$	73
$CatSet$	9, 72, 73
$CatSubs$	8, 72, 73
$Celarent$	75, 76, 78, 80, 82, 84
$Celarent_izz$	90
$Celaront$	75, 80, 82, 84
$Celaront_izz$	90

<i>Cesare</i>	75, 76, 78, 80, 82, 84	<i>individual</i>	12, 68, 72, 73, 91
<i>Cesare_izz</i>	90	<i>individual_lemma1</i>	91
<i>Cesaro</i>	75, 80, 82, 84	<i>individual_lemma2</i>	92
<i>Cesaro_izz</i>	90	<i>IndvSet</i>	9, 72, 73
<i>Code</i>	4	<i>interm_∃_lemma</i>	89
<i>Complement</i>	35, 43, 47, 77, 81, 83	<i>InTermM</i>	60, 87, 88
<i>COPULA</i>	88	<i>io_obv_thm</i>	78, 82
 		<i>io_subcont_thm</i>	82, 84
<i>Darapti</i>	75, 80, 82, 84	<i>ISUB</i>	8, 72
<i>Darapti_izz</i>	90	<i>iu_obv_thm</i>	84
<i>Darii</i>	75, 76, 78, 80, 82, 84	<i>izz</i>	9, 60, 72, 73, 87, 88
<i>Darii_izz</i>	90	<i>izz_□_thm</i>	65, 89
<i>Datisi</i>	75, 76, 78, 80, 82, 84	<i>izz_□_thm2</i>	65, 90
<i>Datisi_izz</i>	90	<i>izz_conv1</i>	64, 89
<i>Dimaris</i>	75, 76, 78, 80, 82, 84	<i>izz_conv2</i>	64, 89
<i>Dimaris_izz</i>	90	<i>izz_conv3</i>	64, 89
<i>Disamis</i>	75, 76, 78, 80, 82, 84	<i>izz_lemma1</i>	73
<i>Disamis_izz</i>	90	<i>izz_lemma2</i>	73
 		<i>izz_lemma3</i>	73
<i>e</i>	34, 38, 41, 45, 50, 77, 79, 81, 83, 85	<i>izz_lemma4</i>	74
<i>e_conv</i>	38	<i>izz_not_lemma</i>	64, 89
<i>e_conv_thm</i>	42, 46, 78, 80, 82, 84, 86	<i>l01</i>	74
<i>ea_obv_thm</i>	78, 82	<i>l01b</i>	74
<i>ei_contrad_thm</i>	78, 82, 84	<i>l02</i>	74
<i>eo_conv_thm</i>	43, 47, 80, 82, 84	<i>l03</i>	74
<i>eo_subalt_thm</i>	82, 84	<i>l03b</i>	74
<i>essentially_predicable_of</i>	12, 72, 73	<i>l03c</i>	74
 		<i>l04</i>	74
<i>Felapton</i>	75, 80, 82, 84	<i>l04a</i>	74
<i>Felapton_izz</i>	90	<i>l04b</i>	74
<i>Ferio</i>	75, 76, 78, 80, 82, 84	<i>l05b</i>	74
<i>Ferio_izz</i>	90	<i>l06n</i>	74
<i>Ferison</i>	75, 76, 78, 80, 82, 84	<i>l07</i>	74
<i>Ferison_izz</i>	90	<i>l08</i>	74
<i>Fesapo</i>	75, 80, 82, 84	<i>l09</i>	74
<i>Fesapo_izz</i>	90	<i>l10</i>	74
<i>Festino</i>	75, 76, 78, 80, 82, 84	<i>l16</i>	74
<i>Festino_izz</i>	90	<i>l17</i>	74
<i>figureS</i>	22	<i>l18</i>	74
<i>figurest</i>	22	<i>l19</i>	74
<i>form</i>	13, 72, 73	<i>l20</i>	74
<i>FP1</i>	67, 91	<i>l22</i>	74
<i>FP2</i>	67, 91	<i>l23b</i>	74
<i>FP3</i>	67, 91	<i>l24</i>	74
<i>FP3b</i>	67, 91	<i>l24a</i>	74
<i>FP4</i>	67, 91	<i>l26</i>	74
<i>FP4a</i>	67, 91	 	
<i>FP4c</i>	68, 91	<i>map_goal</i>	53
<i>Fresison</i>	75, 76, 78, 80, 82, 84	<i>mk_AttrTerm</i>	58, 87, 88
<i>Fresison_izz</i>	90	<i>mk_modsyll</i>	53
 		<i>mk_modsyllp</i>	53
<i>hazz</i>	10, 60, 72, 73, 87, 88	<i>mk_modt</i>	53
<i>hazz_conv2</i>	65, 89	<i>mk_pred</i>	52
 		<i>mk_relt</i>	52
<i>I</i>	49, 85, 88	<i>mk_SubsTerm</i>	58, 87, 88
<i>i</i>	34, 38, 41, 45, 50, 77, 79, 81, 83, 85	<i>mk_syll</i>	52
<i>i_conv</i>	38	<i>mk_syllp</i>	52
<i>i_conv_thm</i>	42, 46, 78, 80, 82, 84, 86	<i>mkATvar</i>	22
<i>Individual</i>	68, 91		

<i>mkGoals</i>	24	<i>syll_prove_and_store</i>	53
<i>mkSimpMapkit</i>	25	<i>syll_rule</i>	34
<i>mkSyll</i>	24	<i>syll_rule2</i>	38
<i>mod_gen_□□□</i>	86	<i>syll_rule3</i>	42
<i>mod_gen_□□◇</i>	86	<i>syll_rule6</i>	46
<i>mod_gen_□□⊢</i>	86	<i>syll_rule6b</i>	46
<i>mod_gen_□◇◇</i>	86	<i>syll_tac</i>	34
<i>mod_gen_□⊢⊢</i>	86	<i>syll_tac2</i>	38
<i>mod_gen_◇□◇</i>	86	<i>syll_tac3</i>	42
<i>mod_gen_⊢□⊢</i>	86	<i>syll_tac6</i>	46
<i>mod_gen_⊢⊢⊢</i>	86	<i>syll_tac6b</i>	46
<i>modgoal</i>	53	<i>sylllogism_data1</i>	23
<i>mods_mapkit</i>	40	<i>sylllogism_data2</i>	24
<i>Mp</i>	68, 91	<i>sylllogism_data3</i>	24
<i>MPROP</i>	85, 88	<i>TAG</i>	87
<i>not</i>	59, 87, 88	<i>tag</i>	58, 87, 88
<i>not_□_hazz_thm</i>	65, 89	<i>Term2</i>	77, 81, 83
<i>not_□_hazz_thm2</i>	65, 90	<i>Term3</i>	79
<i>not_hazz_not_lemma</i>	65, 89	<i>TermM</i>	58, 88
<i>not_izz_and_hazz_lemma1</i>	89	<i>TermP</i>	76
<i>not_some_hazz_lemma</i>	65, 89	<i>TermS</i>	75
<i>nthfig</i>	22	<i>u</i>	34, 38, 42, 45, 50, 77, 79, 81, 83, 85
<i>o</i>	77, 79, 81, 83, 85	<i>u - p</i>	18, 23
<i>o_not_conv_thm</i>	78	<i>universal</i>	12, 69, 72, 73, 91
<i>oi_obv_thm</i>	78, 82	<i>universal - particular</i>	18, 23
<i>op_from_char</i>	25	<i>vowels_from_string</i>	24
<i>optrip_from_text</i>	25	<i>W</i>	49, 85, 88
<i>p</i>	37, 49, 79, 85		
<i>p_∃_lemma</i>	80, 86		
<i>particular</i>	12, 69, 72, 73, 91		
<i>Plato</i>	4		
<i>predicable_of</i>	12, 72, 73		
<i>proveGoals</i>	24		
<i>proveSylls</i>	25		
<i>sg_03</i>	39		
<i>sg_04</i>	39		
<i>sg_09</i>	39		
<i>sg_09_thm</i>	80		
<i>sg_10</i>	39		
<i>sg_10_thm</i>	80		
<i>sg_11</i>	39		
<i>sg_11_thm</i>	80		
<i>sg_12</i>	39		
<i>sg_12_thm</i>	80		
<i>sg_13</i>	39		
<i>sg_13_thm</i>	80		
<i>sg_14</i>	39		
<i>sg_14_thm</i>	80		
<i>Some</i>	60, 87, 88		
<i>Some_are_izz_or_hazz_lemma</i>	61, 89		
<i>some_izz_lemma</i>	64, 89		
<i>Something</i>	13, 72, 73		
<i>Substance</i>	59, 87, 88		
<i>substantial</i>	69, 91		
<i>syll_prove</i>	53		