The Logic and Metaphysics of Leibniz

Roger Bishop Jones

Abstract

Formal models of aspects of the Logic and Metaphysics of Gottfried Wilhelm Leibniz.

Created 2009/9/2

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

http://www.rbjones.com/rbjpub/pp/doc/t032.pdf

Id: t032a.tex,v 1.4 2010/08/08 15:50:44 rbj Exp

Contents

1 Prelude

This document is intended to form a chapter of An Analytic History of Philosophical Logic [4, 6, 5] and is made available as a separate document prior to completion of that work.

For an introduction to and overview of the work as a whole see [2].

2 Introduction

This document has not really been started yet.

Here are some notes on what it might contain.

For the purposes of this analytic history the single most important concern is what Leibniz contributed to out understanding of the concept of logical truth. For this we consider primarily the most fundamental parts of his metaphysics, which we do partly through the perspective of Bertrand Russell [7], whose own philosophy of Logical Atomism was influenced by Leibniz and will be considered later.

Leibniz contributed also to our ideas about the applications of logic, through his "universal characteristic" and "calculus ratiocinator". Some kind of analysis of his ideas in this area would be nice.

3 Leibniz On Identity

This just shows the triviality of the identity of indiscernibles in our logical context, and raises the question, what more substantive point is Leibniz making and has it any substance?

It is clear that Leibniz's intention was not to forumlate such trivial principles as are found here. He intends that distinct individuals always differ in some substantive way (using that term informally). The real problem here is whether this can be captured formally in HOL, and this section at present does not make offer any elightenment on that topic.

SML

```
open_theory "misc2";
force_new_theory "leibniz01";
set_pc "misc2";
```

In higher order logic an identity of indiscernables (though probably not Leibniz's) is a trivial principle.

Its formulation is:

$$\forall x\ y \bullet (\forall P \bullet P(x) \Leftrightarrow P(y)) \Rightarrow x = y$$

Here is a long-winded transcript of a ProofPower proof session:

SML

```
set_goal([], ⌜∀x y• (∀P• P(x) ⇔ P(y)) ⇒ x = y⌝);
```

ProofPower output

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

(* ?⊢ *)  ⌜∀ x y• (∀ P• P x ⇔ P y) ⇒ x = y⌝
```

Strip the goal.

SML

```
a (REPEAT strip_tac);
```

ProofPower output

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

(*  1 *) ⌜∀ P• P x ⇔ P y⌝

(* ?⊢ *) ⌜x = y⌝
```

Instantiate the assumption using the predicate ⌜$= y⌝. [1]:

SML

```
a (spec_asm_tac ⌜∀ P• P x ⇔ P y⌝ ⌜$= y⌝);
```

ProofPower output

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

(*  2 *) ⌜∀ P• P x ⇔ P y⌝
(*  1 *) ⌜y = x⌝

(* ?⊢ *) ⌜x = y⌝
```

The instantiation yields:

$$
\begin{aligned}
&\ulcorner y = x \Leftrightarrow y = y \urcorner \\
&\Leftrightarrow \ulcorner (y = x \Rightarrow y = y) \wedge (y = y \Rightarrow y = x) \urcorner \\
&\Leftrightarrow \ulcorner (\neg y = x \vee y = y) \wedge (y = x \vee \neg y = y) \urcorner \\
&\Leftrightarrow \ulcorner (\neg y = x \vee T) \wedge (y = x \vee F) \urcorner \\
&\Leftrightarrow \ulcorner (y = x) \urcorner
\end{aligned}
$$

of which the last is the new assumption shown above.

Rewrite the conclusion with the assumptions (giving $\ulcorner x = x \urcorner$ which is automatically discharged).

SML

```
a (asm_rewrite_tac[]);
```

ProofPower output

```
Tactic produced 0 subgoals:
Current and main goal achieved
```

Save the theorem.

SML

```
val leibniz_identity = save_pop_thm "leibniz_identity";
```

[1] This is the predicate "equal to y", or $\ulcorner \lambda x.y = x \urcorner$

ProofPower output

$$\vdash \textit{Now 0 goals on the main goal stack}$$
$$\vdash \textit{val leibniz\_identity} = \vdash \forall\ x\ y\bullet\ (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y) \Rightarrow x = y : \textit{THM}$$

So, in this context, that indiscernibles are identical is an elementary consequence of the fact that for every entity 'e' there is a predicate 'equal to e' which is satisfied only by e.

Leibniz intended by his principle something more substantial, which is harder to capture.

4 The Calculus Ratiocinator

This section is primarily based on what is said about Leibniz in the book written by Lukasievicz on Aristotles syllogistic, I have not checked this out against Leibniz's own writings, though it seems plausible from what I have read.

Leibniz's calculus is an arithmetisation of Aristotle's syllogistic. That such an arithmetisation will have the power which Leibniz attributes to it is certainly not the case, but my concern here is just to build a model which is similar to the arithmetic interpretation and allows us to check the extent to which it properly captures the relevant parts of Aristotle's logic. Since useful groundwork on this is done in my formal treatment of Aristotle[3], I will make use of some of that material by making this document logically dependent upon it, and making the theory which is here developed a child of one of the my models of Aristotle.

4.1 Leibniz's Interpretation of Aristotle's Syllogistic

The rationale for Leibniz's interpretation of propositions depends upon his conceptual atomism. This is the idea that concepts can be classified as either *simple* or *complex* and that complex concepts are defined ultimately (though possibly indirectly) in terms of simple concepts, by limited means. The limited means consist of negation of simple concepts and conjunction. It is further assumed that simple concepts are logically independent of each other, and that none of them is always true or always false (possibly this should be read necessarily true or necessarily false).

Given this simple idea of what concepts are, conceptual inclusion is decidable provided that we know which concepts are simple and we know the definitions of all the complex concepts. Conceptual inclusion corresponds to the A form of proposition, the I form is also decidable, and the other two are defined in terms of those two.

Leibniz arithmetises this by assuming that each simple concept is given a unique prime number, and that complex concepts are then represented by two numbers. The first of these two numbers is the product of the primes of the simple concepts which occur positively in the definition of the complex concept (when this has been expanded out so that it no longer mentions any complex concepts and therefore consists of a conjunction of simple concepts or their negations). The second number is the product of the primes which code the simple concepts whose negations appear in the expanded definition.

Arithetisation is not essential, any equivalent way of coding up the information about which simple concepts or negations of simple concepts appear in the definition of a complex concept will do, and reasoning will be simpler if the problem of obtaining prime factorisations is sidestepped. We need not know how simple concepts are represented, and we can represent a conjunction as a list of conjuncts.

There is a small awkwardness if we want equality of concepts to be the same thing as equality of the representation and hence obtain:

$A\ a\ B\ and\ B\ a\ A\ entails\ A = B$

If we just used two pairs of lists of simple concepts then the same concept would have multiple representatives, and pairs of lists which overlapped would not represent concepts at all. We could use a pair of sets of simple concepts, but then we have the possibility of infinite sets and we still might have overlap. A function from simple concepts into the type BOOL+ONE where the BOOL component represents negative or positive presence of the simple concept gets the identity correct but might allow infinite numbers of simple concepts. This is a possible point of divergence from Leibniz, but I'm going to try this one.

SML

```
open_theory "aristotle";
force_new_theory "leibniz02";
```

4.1.1 Semantics

We will allow that any type be used as the simple concepts, so that all the rules we can establish will be correct in the finite case and in the infinite case.

SML

```
declare_type_abbrev("TermL", [], ⌜:'a → TTV⌝);
```

TTV is a type with just three elements which may be thought of as truth values. Their names are $pTrue$, $pFalse$ and pU. I will use $pTrue$ to make a positively occurring simple concept, $pFalse$ to mark a negated simple concept. pU marks a simple concept which does not occur in the relevant complex concept.

The predicate *Some* will be true iff a TTV does not have the value pU.

HOL Constant

$$\boldsymbol{Some} : TTV \to BOOL$$

$$\forall x \bullet\ Some\ x = \neg\ x = pU$$

The predicate $IsPos$ will be true iff a BOOL+ONE has the value $pTrue$.

HOL Constant

$$\boldsymbol{IsPos} : TTV \to BOOL$$

$$\forall x \bullet\ IsPos\ x \Leftrightarrow x = pTrue$$

4.1.2 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

$$\$\boldsymbol{a} : TermL \to TermL \to BOOL$$

$$\forall A\ B \bullet\ A\ a\ B \Leftrightarrow \forall x \bullet\ (B\ x = pTrue \Rightarrow A\ x = pTrue) \wedge (B\ x = pFalse \Rightarrow A\ x = pFalse)$$

HOL Constant

$$\$\boldsymbol{i} : TermL \to TermL \to BOOL$$

$$\forall A\ B \bullet\ A\ i\ B \Leftrightarrow \forall x \bullet\ A\ x = B\ x \Rightarrow A\ x = pU$$

HOL Constant

$$\$\boldsymbol{e} : TermL \to TermL \to BOOL$$

$$\forall A\ B \bullet\ A\ e\ B \Leftrightarrow \neg\ A\ i\ B$$

HOL Constant

$$\$\boldsymbol{u} : TermL \to TermL \to BOOL$$

$$\forall A\ B \bullet\ A\ u\ B \Leftrightarrow \neg\ A\ a\ B$$

SML

$$declare\_alias("o", \ulcorner\$u\urcorner);$$

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.

4.1.3 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 impession 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 [2] 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_tacL = REPEAT (POP_ASM_T ante_tac)
      THEN rewrite_tac (map get_spec [⌜$a⌝, ⌜$e⌝, ⌜$i⌝, ⌜$u⌝])
      THEN REPEAT strip_tac THEN all_asm_fc_tac []
      THEN_TRY asm_rewrite_tac[];
fun syll_ruleL g = tac_proof (g, syll_tacL);
val syll_tacLb = 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_ruleLb g = tac_proof (g, syll_tacLb);
```

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

Simple Conversion

Conversion Per Accidens These are OK.

```
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 : TermL → TermL
```

```
∀A α• (Complement A) α =
          if A α = pTrue then pFalse
          else if A α = pFalse then pTrue
          else pU
```

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
```

[2] in which I followed Strawson [8], though I can now cite Aristotle, Prior Analytic, Book 1, Part 2. [1]

4.1.4 The Square of Opposition

$ao\_contrad\_thm = \vdash A\ a\ B \Leftrightarrow \neg\ A\ o\ B$
$ei\_contrad\_thm = \vdash A\ e\ B \Leftrightarrow \neg\ A\ i\ B$
$ae\_contrar\_thm = \vdash \neg\ (A\ a\ B \wedge A\ e\ B)$
$io\_subcont\_thm = \vdash A\ i\ B \vee A\ o\ B$
$ai\_subalt\_thm = \vdash A\ a\ B \Rightarrow A\ i\ B$
$eo\_subalt\_thm = \vdash A\ e\ B \Rightarrow A\ o\ B$

4.1.5 The Syllogisms

First we make a *mapkit*.

SML

```
val sLmapkit:mapkit = mkSimpMapkit ⌜:TermL⌝ [⌜$a⌝,⌜$e⌝,⌜$i⌝,⌜$u⌝];
```

Then we apply this in generating and proving the syllogisms.

SML

```
proveGoals syll_tacL "" (mkGoals sLmapkit syllogism_data1);
proveGoals syll_tacL "" (mkGoals sLmapkit syllogism_data2);
proveGoals syll_tacL "" (mkGoals sLmapkit syllogism_data3);
```

The theorems are also displayed in the theory listing in Appendix B

5 Metaphysics

This is an adaptation of the model of Aristotelian logic and metaphysics in section **??**, to reflect the most crucial differences between Aristotle and Leibniz.

Russell [7] represents Leibniz as having adhered rather strictly to Aristotle's logic with bad consequences for his metaphysics, in particular he sees the idea of monads as having arisen from the idea that all propositions have subject/predicate form. This is something into which I hope to look more closely in due course.

However, our analysis of Aristole suggests that if his metaphysics is so construed as to make his logic sound, then existence is necessary, and this view is incorporated into our model (which may go too far in this matter). For Leibniz however the position on existence is pretty clear. Existence of substantial individuals is the only thing which is contingent, all else is necessary.

The following model is therefore an exploration of what happens if we tweak the underlying model to ensure exactly that. One thing that we should expect, is that the syllogisms which exhibit the "existential fallacy" will no longer be sound. The second thing which seems to flow from that is the irrelevance of the essential/accidental distinction which is possibly the most important feature of Aristotelian metaphysics. In **??** I assume that this distinction is intended to correspond to that between necessary and contingent truth (though this may be tendentious, I am not aware of explicit textual support for it). This can no longer be done, and I therefore abandon Aristotelian metaphysics altogether returning to a treatment of the syllogism less influenced by metaphysics.

The connection with Grice and Codd is no longer relevant so that material also is excised.

SML

```
open_theory "misc2";
force_new_theory "leibniz03";
force_new_pc "'leibniz03";
```

5.1 The Subject Matter

Once the essential/accidental distinction is discarded, we are left with a metaphysic in which the key distinction is between individual substances and predicates.

We take an individual to be something which is only truly predicable of itself, and other predicates as collections of individuals, once again accounting for (essential) predication as set inclusion (having represented an individual as a unit set). This subset relation is fixed, but the individual substances which are the extensions of predicates are themselves contingent.

SML

```
new_type ("ISUB", 0);
```

Let us take a new type for a fixed population of predicates.

SML

```
new_type ("PRED", 0);
```

Whose extension is fixed.

HOL Constant

$$\boldsymbol{extension} : PRED \to ISUB\ \mathbb{P}$$

$$\vdash\ T$$

However, the extension is defined in terms of individual substances whose existence is contingent, and so we still have the possibility of distinguishing essential predication from accidental, according to whether inclusion obtains on the full extension, or merely on the extensions restricted to the individuals which actually exist.

A possible world is therefore a collection of individuals.

SML

```
declare_type_abbrev ("W", [], ⌜:ISUB ℙ⌝);
```

We to distinguish one possible world which is the actual world:

HOL Constant

$$\boldsymbol{actual\_world} : W$$

$$\vdash\ T$$

Subjects and predicates are just things of type *PRED*.

5.2 Predication

Though the retreat from accidental predication simplifies matters I will retain a presentation of the syllogism similar to that in Section **??**, for the sake of its readability.

So I separate out the quantifier by defining *All* and *Some* appropriately, and retain the postfix negator even though only one kind of predication is now available. (in fact I could define the two kinds of predication because the I still have available two kinds of extension, but the modal operators suffice to express the distinction between the two kinds of predication).

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

$$declare\_type\_abbrev(\texttt{"}COPULA\texttt{"},\ [],\ \ulcorner{:}ISUB \to PRED \to (W \to BOOL)\urcorner);$$

The first parameter is an individual substance rather than a PRED, the quantifying operato will arrange for each of the relevant individuals to be supplied.

SML

$$declare\_type\_abbrev\ (\texttt{"}MPROP\texttt{"},\ [],\ \ulcorner{:}W \to BOOL\urcorner);$$

Propositions

Complementation The distinction between affirmative and negative is achieved by a postfix negation so we can say "is not", or "are not" (which in this models would be synonyms, so we will go with "are" only.

SML

$$declare\_postfix\ (100,\ \texttt{"}not\texttt{"});$$

HOL Constant

$$\$\boldsymbol{not} : COPULA \to COPULA$$

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

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

$$\boldsymbol{All} : PRED \to (ISUB \to PRED \to MPROP) \to PRED \to MPROP$$

$$\forall\ s\ r\ p\bullet\ All\ s\ r\ p = \lambda w\bullet\ \forall z\bullet\ z \in extension\ s \wedge z \in w \Rightarrow r\ z\ p\ w$$

HOL Constant

$$\boldsymbol{Some} : PRED \to (ISUB \to PRED \to MPROP) \to PRED \to MPROP$$

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

Predication 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

$$\boldsymbol{are} : ISUB \to PRED \to MPROP$$

$$\forall\ i\ t\bullet\ are\ i\ t = \lambda w\bullet\ i \in extension\ t$$

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

HOL Constant

$$\Diamond : MPROP \to MPROP$$

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

HOL Constant

$$\Box : MPROP \to MPROP$$

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

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

$$\neg_{\boldsymbol{a}} : MPROP \to MPROP$$

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

SML

$$declare\_alias\ (\texttt{"}\neg\texttt{"},\ \ulcorner\neg_a\urcorner);$$

SML

$declare\_infix(220,\ "\wedge_a");$

HOL Constant

$\$\wedge_{\boldsymbol{a}} : MPROP \to MPROP \to MPROP$

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

SML

$declare\_alias\ ("\wedge", \ulcorner\$\wedge_a\urcorner);$

SML

$declare\_infix(210,\ "\Rightarrow_a");$

HOL Constant

$\$\Rightarrow_{\boldsymbol{a}} : MPROP \to MPROP \to MPROP$

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

SML

$declare\_alias\ ("\Rightarrow", \ulcorner\$\Rightarrow_a\urcorner);$

SML

$declare\_infix(200,\ "\Leftrightarrow_a");$

HOL Constant

$\$\Leftrightarrow_{\boldsymbol{a}} : MPROP \to MPROP \to MPROP$

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

SML

$declare\_alias\ ("\Leftrightarrow", \ulcorner\$\Leftrightarrow_a\urcorner);$

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

$declare\_binder\ "\forall_a";$

HOL Constant

$\$\forall_{\boldsymbol{a}} : (PRED \to MPROP) \to MPROP$

$\forall mpf\bullet\ \$\forall_a\ mpf\ =\ \lambda w\bullet\ \forall t\bullet\ mpf\ \ t\ \ w$

SML

```
declare_alias ("∀", ⌜$∀a⌝);
```

SML

```
declare_binder "∃a";
```

HOL Constant

```
$∃a : (PRED → MPROP) → MPROP
```

$$\forall mpf\bullet\ \$\exists_a\ mpf\ =\ \lambda w\bullet\ \exists t\bullet\ mpf\ \ t\ \ w$$

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 $\models$ asnd Π. 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 premises 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 premises are necessary, then so will be the conclusions.

SML

```
declare_infix(100, "⊨");
```

HOL Constant

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

$$\forall lp\ \ c\bullet\ lp\ \models\ c\ \Leftrightarrow\ Fold\ (\lambda p\ \ t\bullet\ p\ \ actual\_world\ \wedge\ t)\ \ lp\ \ T\ \Rightarrow\ c\ \ actual\_world$$

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

SML

```
declare_infix(100, "Π");
```

HOL Constant

```
$Π : MPROP LIST → MPROP → BOOL
```

$$\forall lp\ \ c\bullet\ lp\ \ \Pi\ \ c\ \Leftrightarrow\ \forall w\bullet\ Fold\ (\lambda p\ \ t\bullet\ p\ \ w\ \ \wedge\ t)\ \ lp\ \ T\ \Rightarrow\ c\ \ w$$

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.

```
are_not_lemma =
    ⊢ All B (are not) A = All A (are not) B
some_are_lemma =
    ⊢ Some B are A = Some A are B
```

These we also supply as our Aristotelian judgements, together with the second which does not give an equation. The second conversion embodies the existential fallacy, and therefore is not provable here.

```
are_conv1 = ⊢
    [All B (are not) A] Π All A (are not) B

are_conv2 = ⊢
    [All B izz A] Π Some A izz B

are_conv3 = ⊢
    [Some B izz A] Π Some A izz B
```

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:

$\Diamond\_\boldsymbol{conv} =$
$\quad \vdash [P]\ \Pi\ Q \Rightarrow [\Diamond\ P]\ \Pi\ \Diamond\ Q$

$\Box\_\boldsymbol{conv} =$
$\quad \vdash [P]\ \Pi\ Q \Rightarrow [\Box\ P]\ \Pi\ \Box\ Q$

Now according to Leibniz all predication is necessary, only existence is contingent. However, the contingency of existence means that this must be interpreted as a claim about predicates applied only to individuals.

$\Diamond\boldsymbol{AllBareA\_thm} = \vdash [] \ \Pi\ \Diamond\ (All\ B\ are\ A)$

The upshot is that to show that our model captures the necessity of predication (in the sense in which this is conceivable), we need a way to talk about individuals.

HOL Constant
$\boldsymbol{individual} : PRED \to MPROP$

$\forall A \bullet\ individual\ A = \lambda w \bullet\ \exists a \bullet\ extension\ A = \{a\}$

$\Diamond\boldsymbol{AarenotA\_thm} = \vdash [] \ \Pi\ \Diamond\ (All\ A\ (are\ not)\ A)$

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.

$\Box\_\boldsymbol{elim\_thm} =$
$\quad \vdash [\Box\ P] \models P$
$\Diamond\_\boldsymbol{intro\_thm} =$
$\quad \vdash [P] \models \Diamond\ P$
$\Box\_\Diamond\_\boldsymbol{thm} =$
$\quad \vdash [\Box\ P] \models \Diamond\ P$

5.8 Other Conversions

The following coversions 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 less Aristotelian without the quantifiers and we can imagine 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

The abolition of accidental predication should result in a syllogistic logic which corresponds to Aristotle, though the contingency of existence means that the existential fallacies really are fallacies.

We can, by methods similar to those used above obtain automatic proofs of the syllogisms which are valid in this model.

The details are omitted, but the valid syllogisms have been proven and stored in the theory, see: Appendix C.

6 Postscript

A The Theory leibniz01

A.1 Parents

misc2

A.2 Theorems

leibniz_identity

$$\vdash \forall\ x\ y\bullet\ (\forall\ P\bullet\ P\ x \Leftrightarrow P\ y) \Rightarrow x = y$$

B The Theory leibniz02

B.1 Parents

aristotle

B.2 Constants

Some	$TTV \to BOOL$
IsPos	$TTV \to BOOL$
$\$\boldsymbol{a}$	$(TermL,\ BOOL)\ BR$
$\$\boldsymbol{i}$	$(TermL,\ BOOL)\ BR$
$\$\boldsymbol{e}$	$(TermL,\ BOOL)\ BR$
$\$\boldsymbol{u}$	$(TermL,\ BOOL)\ BR$
Complement	$TermL \to TermL$

B.3 Aliases

$\boldsymbol{o}$	$\$u : (TermL,\ BOOL)\ BR$
$\sim$	$Complement : TermL \to TermL$

B.4 Type Abbreviations

TermL	*TermL*

B.5 Fixity

Right Infix 300:

$\boldsymbol{a}$ $\boldsymbol{e}$ $\boldsymbol{i}$ $\boldsymbol{u}$

B.6 Definitions

Some	$\vdash \forall\ x\bullet\ Some\ x \Leftrightarrow \neg\ x = pU$
IsPos	$\vdash \forall\ x\bullet\ IsPos\ x \Leftrightarrow x = pTrue$
$\boldsymbol{a}$	$\vdash \forall\ A\ B$ $\bullet\ A\ a\ B$ $\Leftrightarrow (\forall\ x$ $\bullet\ (B\ x = pTrue \Rightarrow A\ x = pTrue)$ $\wedge\ (B\ x = pFalse \Rightarrow A\ x = pFalse))$
$\boldsymbol{i}$	$\vdash \forall\ A\ B\bullet\ A\ i\ B \Leftrightarrow (\forall\ x\bullet\ A\ x = B\ x \Rightarrow A\ x = pU)$
$\boldsymbol{e}$	$\vdash \forall\ A\ B\bullet\ A\ e\ B \Leftrightarrow \neg\ A\ i\ B$
$\boldsymbol{u}$	$\vdash \forall\ A\ B\bullet\ A\ o\ B \Leftrightarrow \neg\ A\ a\ B$
Complement	$\vdash \forall\ A\ \alpha$ $\bullet \sim A\ \alpha$ $= (if\ A\ \alpha = pTrue$ $then\ pFalse$ $else\ if\ A\ \alpha = pFalse$ $then\ pTrue$ $else\ pU)$

B.7 Theorems

e_conv_thm $A\ e\ B \vdash B\ e\ A$
i_conv_thm $A\ i\ B \vdash B\ i\ A$

C The Theory leibniz03

C.1 Parents

misc2

C.2 Constants

extension	$PRED \to W$
actual_world	W
$***not***	$COPULA \to COPULA$
All	$PRED \to COPULA \to PRED \to MPROP$
Some	$PRED \to COPULA \to PRED \to MPROP$
are	$COPULA$
$\Diamond$	$MPROP \to MPROP$
$\Box$	$MPROP \to MPROP$
$\neg_{a}$	$MPROP \to MPROP$
$\$\wedge_{a}$	$(MPROP,\ MPROP)\ BR$
$\$\Rightarrow_{a}$	$(MPROP,\ MPROP)\ BR$
$\$\Leftrightarrow_{a}$	$(MPROP,\ MPROP)\ BR$
$\$\forall_{a}$	$(PRED \to MPROP) \to MPROP$
$\$\exists_{a}$	$(PRED \to MPROP) \to MPROP$
$\$\models$	$MPROP\ LIST \to MPROP \to BOOL$
$\$\mathbf{\Pi}$	$MPROP\ LIST \to MPROP \to BOOL$
individual	$PRED \to MPROP$

C.3 Aliases

$\neg$	$\neg_{a} : MPROP \to MPROP$
$\wedge$	$\$\wedge_{a} : (MPROP,\ MPROP)\ BR$
$\Rightarrow$	$\$\Rightarrow_{a} : (MPROP,\ MPROP)\ BR$
$\Leftrightarrow$	$\$\Leftrightarrow_{a} : (MPROP,\ MPROP)\ BR$
$\forall$	$\$\forall_{a} : (PRED \to MPROP) \to MPROP$
$\exists$	$\$\exists_{a} : (PRED \to MPROP) \to MPROP$

C.4 Types

ISUB
PRED

C.5 Type Abbreviations

W	W
COPULA	$COPULA$
MPROP	$MPROP$

C.6 Fixity

Binder: $\forall_a$ $\exists_a$
Right Infix 100:
$\mathbf{\Pi}$ $\models$
Right Infix 200:
$\Leftrightarrow_a$
Right Infix 210:
$\Rightarrow_a$
Right Infix 220:
$\wedge_a$
Postfix 100: ***not***

C.7 Definitions

actual_world
extension $\vdash T$
not $\vdash \forall\ pred\bullet\ (pred\ not) = (\lambda\ pa\ t\ w\bullet \neg\ pred\ pa\ t\ w)$
All $\vdash \forall\ s\ r\ p$
$\bullet\ All\ s\ r\ p$
$= (\lambda\ w\bullet \forall\ z\bullet\ z \in extension\ s \wedge z \in w \Rightarrow r\ z\ p\ w)$
Some $\vdash \forall\ s\ r\ p$
$\bullet\ Some\ s\ r\ p$
$= (\lambda\ w\bullet \exists\ z\bullet\ z \in extension\ s \wedge z \in w \wedge r\ z\ p\ w)$
are $\vdash \forall\ i\ t\bullet\ are\ i\ t = (\lambda\ w\bullet\ i \in extension\ t)$
$\Diamond$ $\vdash \forall\ p\bullet \Diamond\ p = (\lambda\ w\bullet \exists\ w'\bullet\ p\ w')$
$\Box$ $\vdash \forall\ p\bullet \Box\ p = (\lambda\ w\bullet \forall\ w'\bullet\ p\ w')$
$\neg_a$ $\vdash \forall\ p\bullet (\neg\ p) = (\lambda\ w\bullet \neg\ p\ w)$
$\wedge_a$ $\vdash \forall\ p\ q\bullet (p \wedge q) = (\lambda\ w\bullet\ p\ w \wedge q\ w)$
$\Rightarrow_a$ $\vdash \forall\ p\ q\bullet (p \Rightarrow q) = (\lambda\ w\bullet\ p\ w \Rightarrow q\ w)$
$\Leftrightarrow_a$ $\vdash \forall\ p\ q\bullet (p \Leftrightarrow q) = (\lambda\ w\bullet\ p\ w \Leftrightarrow q\ w)$
$\forall_a$ $\vdash \forall\ mpf\bullet\ \$\forall\ mpf = (\lambda\ w\bullet \forall\ t\bullet\ mpf\ t\ w)$
$\exists_a$ $\vdash \forall\ mpf\bullet\ \$\exists\ mpf = (\lambda\ w\bullet \exists\ t\bullet\ mpf\ t\ w)$
$\models$ $\vdash \forall\ lp\ c$
$\bullet\ lp \models c$
$\Leftrightarrow Fold\ (\lambda\ p\ t\bullet\ p\ actual\_world \wedge t)\ lp\ T$
$\Rightarrow c\ actual\_world$
$\mathbf{\Pi}$ $\vdash \forall\ lp\ c$
$\bullet\ lp\ \Pi\ c \Leftrightarrow (\forall\ w\bullet\ Fold\ (\lambda\ p\ t\bullet\ p\ w \wedge t)\ lp\ T \Rightarrow c\ w)$
individual $\vdash \forall\ A\bullet\ individual\ A = (\lambda\ w\bullet \exists\ a\bullet\ extension\ A = \{a\})$

C.8 Theorems

are_not_lemma
$\vdash All\ B\ (are\ not)\ A = All\ A\ (are\ not)\ B$
some_are_lemma
$\vdash Some\ B\ are\ A = Some\ A\ are\ B$
are_conv1 $\vdash [All\ B\ (are\ not)\ A]\ \Pi\ All\ A\ (are\ not)\ B$
are_conv3 $\vdash [Some\ B\ are\ A]\ \Pi\ Some\ A\ are\ B$
$\Diamond$***_conv*** $\vdash [P]\ \Pi\ Q \Rightarrow [\Diamond\ P]\ \Pi\ \Diamond\ Q$
$\Box$***_conv*** $\vdash [P]\ \Pi\ Q \Rightarrow [\Box\ P]\ \Pi\ \Box\ Q$

$\Diamond$***AllBareA_thm***
$\vdash [] \ \Pi \ \Diamond \ (All \ B \ are \ A)$

$\Diamond$***AarenotA_thm***
$\vdash [] \models \Diamond \ (All \ A \ (are \ not) \ A)$

$\Box$***_elim_thm*** $\vdash [\Box \ P] \models P$

$\Box$***_intro_thm*** $\vdash [P] \models \Diamond \ P$

$\Box$***_$\Diamond$_thm*** $\vdash [\Box \ P] \models \Diamond \ P$

$\neg$***_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_are $\vdash [All \ M \ are \ P; \ All \ \mathbb{S} \ are \ M] \ \Pi \ All \ \mathbb{S} \ are \ P$

Celarent_are $\vdash [All \ M \ (are \ not) \ P; \ All \ \mathbb{S} \ are \ M] \ \Pi \ All \ \mathbb{S} \ (are \ not) \ P$

Darii_are $\vdash [All \ M \ are \ P; \ Some \ \mathbb{S} \ are \ M] \ \Pi \ Some \ \mathbb{S} \ are \ P$

Ferio_are $\vdash [All \ M \ (are \ not) \ P; \ Some \ \mathbb{S} \ are \ M]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

Cesare_are $\vdash [All \ P \ (are \ not) \ M; \ All \ \mathbb{S} \ are \ M] \ \Pi \ All \ \mathbb{S} \ (are \ not) \ P$

Camestres_are
$\vdash [All \ P \ are \ M; \ All \ \mathbb{S} \ (are \ not) \ M] \ \Pi \ All \ \mathbb{S} \ (are \ not) \ P$

Festino_are $\vdash [All \ P \ (are \ not) \ M; \ Some \ \mathbb{S} \ are \ M]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

Baroco_are $\vdash [All \ P \ are \ M; \ Some \ \mathbb{S} \ (are \ not) \ M]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

Disamis_are $\vdash [Some \ M \ are \ P; \ All \ M \ are \ \mathbb{S}] \ \Pi \ Some \ \mathbb{S} \ are \ P$

Datisi_are $\vdash [All \ M \ are \ P; \ Some \ M \ are \ \mathbb{S}] \ \Pi \ Some \ \mathbb{S} \ are \ P$

Bocardo_are $\vdash [Some \ M \ (are \ not) \ P; \ All \ M \ are \ \mathbb{S}]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

Ferison_are $\vdash [All \ M \ (are \ not) \ P; \ Some \ M \ are \ \mathbb{S}]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

Camenes_are $\vdash [All \ P \ are \ M; \ All \ M \ (are \ not) \ \mathbb{S}] \ \Pi \ All \ \mathbb{S} \ (are \ not) \ P$

Dimaris_are $\vdash [Some \ P \ are \ M; \ All \ M \ are \ \mathbb{S}] \ \Pi \ Some \ \mathbb{S} \ are \ P$

Fresison_are $\vdash [All \ P \ (are \ not) \ M; \ Some \ M \ are \ \mathbb{S}]$
$\Pi \ Some \ \mathbb{S} \ (are \ not) \ P$

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

[3] Roger Bishop Jones. Aristotle's Logic and Metaphysics. *RBJones.com*, 2009-.
http://www.rbjones.com/rbjpub/pp/doc/t028.pdf.

[4] Roger Bishop Jones. *Analyses of Analysis: Part I - Exegetical Analysis*. RBJones.com. 2009.
http://www.rbjones.com/rbjpub/pp/doc/b001.pdf.

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

[6] Roger Bishop Jones. *Analyses of Analysis: Part III - Theory Listings*. RBJones.com. 2009.
http://www.rbjones.com/rbjpub/pp/doc/b002.pdf.

[7] Bertrand Russell. *The Philosophy of Leibniz*. George Allen & Unwin. 1900.

[8] P.F. Strawson. *Introduction to Logical Theory*. Methuen. URL
http://www.archive.org/details/introductiontolo010626mbp, 1952.

Index