The Logic of Opacity

Andrew Bacon
Jeffrey Sanford Russell *

June 25, 2017

Abstract

We explore the view that Frege's puzzle is a source of straightforward counterexamples to Leibniz's law. Taking this seriously requires us to revise the classical logic of quantifiers and identity; we work out the options, in the context of higher-order logic. The logics we arrive at provide the resources for a straightforward semantics of attitude reports that is consistent with the Millian thesis that the meaning of a name is just the thing it stands for. We provide models to show that some of these logics are non-degenerate.

1 Opacity and Export

Having incomplete knowledge of classical astronomy (and being unacquainted with the literature on Frege's Puzzle) Asher assents to 'Hesperus is bright', but not to 'Phosphorus is bright'. It's natural, then, to describe Asher's doxastic state like this: Asher believes that Hesperus is bright, but Asher does not believe that Phosphorus is bright. But of course, Hesperus and Phosphorus are the very same thing: the planet Venus. Taking this familiar story at face-value, we might formalize these claims as follows:

1. $a = b$
2. Ba

*Thanks to Peter Fritz, Jeremy Goodman, Cian Dorr, the participants in the Oslo Higher-Order Metaphysics Workshop, and an anonymous referee for helpful comments.

3. $\neg Bb$

(Here a stands for 'Hesperus', b stands for 'Phosphorus', and B stands for the predicate 'λx (Asher believes x is bright)'.) These three claims together — we'll call them *the Triad* — conflict with an instance of Leibniz's law:

L() $a = b \rightarrow Ba \rightarrow Bb$

(Here, and throughout, we associate arrows to the right.)

There are many well-known replies to this puzzle. Some say it rests on equivocation (see Dorr 2014). For instance, if 'believes' is context-sensitive, and this sensitivity is resolved differently for each of the two belief reports, then it would be inappropriate to represent both reports using a single predicate letter B in statements 2 and 3 (e.g. Crimmins and Perry 1989). Alternatively, perhaps the term 'Hesperus' has a different denotation in the identity statement than in the belief ascription, making it inappropriate to use a single constant symbol a in both 1 and 2. (For instance, Frege 2010 claimed that, while 'Hesperus' ordinarily denotes a planet, in the context of the sentence 'Asher believes Hesperus is bright', 'Hesperus' instead denotes a certain sense.) Another reply is that, while it's true that Asher believes that Hesperus is bright, it doesn't follow from this that Hesperus is such that Asher believes *it* to be bright — or in other words, it doesn't follow that λx (Asher believes x is bright) applies to Hesperus. In that case, our regimentation of 2 in subject-predicate form would not accurately reflect the fact that Asher believes that Hesperus is bright. In general, philosophers who take this line hold that *β-reduction*, which transforms the complex predication $(\lambda x\, \phi)a$ to the substitution instance $\phi[a/x]$, does not preserve truth-value (see Kripke 2005; Salmon 2010).

Let's set these replies aside. In this paper, we'll explore views that take this argument against an instance of Leibniz's law at face value. In particular, we'll suppose that the Triad perspicuously and univocally formalizes certain truths about this situation. (We'll also take β-reduction for granted more generally.) Taking these things for granted, we must reject L(). The predicate B is *opaque*, in the sense (derived from Quine 1960, sec. 30) that applying B to co-referential terms does not always produce sentences with the same truth-value. We'll be investigating the logic of opacity.

While the canonical examples of opacity involve belief (or similar attitudes), there are many other potential applications — though particular examples will of course be controversial. Some hold that the statue Goliath just is the clay of which it is made — and yet one would like to say that Goliath could not survive being squashed, while the clay could (e.g. Gibbard 1975). Some hold that Kilimanjaro

just is a certain precise chunk of rock — and yet one would like to say that Kilimanjaro is determinately a mountain, while that precise chunk of rock is not (e.g. McGee 1997). Like Frege's puzzle of belief, these other puzzles might also move one to take at face value the apparent opacity of predicates involving possibility, change, or determinacy. (If the data Saul 1997 presents is to be taken at face value, involving an apparent contrast between 'Clark Kent went into the phone booth, and Superman came out' and 'Superman went into the phone booth, and Clark Kent came out', then opacity might turn out to be very pervasive indeed.) The logical questions we are investigating are not parochial to any of these subject matters; but for the sake of concreteness we'll continue to focus on the example of belief.

We'll be working in the context of higher-order logic: let's begin with some preliminaries about what that is. English has expressions of various grammatical categories, such as nouns, adjectives, and articles. The formal language we'll be working in has a simplified hierarchy of grammatical categories — or *types*. The basic types are e, for singular terms (for first-order 'entities'), and t for truth-evaluable expressions — sentences and formulas. A predicate, such as 'λx (x is bright)' or 'λx (Asher believes x is bright)', has neither of these types, but it can be applied to a type e expression to produce a type t expression. Accordingly, its type is called $e \to t$. In general, expressions of type $\sigma \to \tau$ can be applied to expressions of type σ to yield expressions of type τ. For example, the first-order quantifer $\exists_e$ has the type $(e \to t) \to t$, which is to say that it can be applied to a predicate like λx (Asher believes x is bright) to produce the sentence $\exists_e x$ (Asher believes x is bright). (As is customary, in our notation we suppress lambdas immediately after quantifiers.)

The language of higher-order logic is so-called because it provides us with resources not just for first-order generalization, but also for generalization at every type. Besides the first-order quantifier $\exists_e$, we also have a quantifier $\exists_t$ which allows us to generalize in sentence position, a quantifier $\exists_{e \to t}$ for generalization in predicate position, and so on. Some philosophers have scrupled at this kind of expressive promiscuity (e.g Quine 1986), but it seems to us that higher-order generalization is both intelligible and salutary (following Frege 2000; e.g. Prior 1961; Shapiro 1991; Williamson 2003). We'll typically use capital letters for higher-order variables, and suppress the type subscripts on quantifiers when it is clear in context how to fill them in.

We are investigating the logic of identity: but just as higher-order logic allows generalization at types other than e, it is natural to consider *identity* for types other than e. For example, one might say that to be an attorney *just is* to be a lawyer. This

claim is readily formalized using an identity predicate for type $e \to t$:

$$\lambda x(x \text{ is an attorney}) =_{e \to t} \lambda x(x \text{ is a lawyer})$$

(So the predicate $=_{e \to t}$ itself has type $(e \to t) \to (e \to t) \to t$.) Likewise, one may think that since Hesperus just is Phosphorus, for Hesperus to be bright just is for Phosphorus to be bright — which again we may readily formalize:

$$Fa =_t Fb$$

(where F stands for 'bright'). Again, we will normally omit the type subscripts. As with higher-order quantification, some philosophers might be inclined to scruple over higher-order identities; but here too we think that expressive promiscuity is nothing to be afraid of (following Rayo 2013; Dorr 2016). Indeed, in both cases working with this kind of generality over types turns out to be very useful. (Furthermore — though we shouldn't prejudge this point here — it turns out there are tight connections between identity at each type and quantification at higher types. Thus even the theory of first-order identity is intimately tied to higher-order quantification. But we will to come to this later on.)

The view we are considering is that the Leibnizian claim L() is false, because the predicate 'believed to be bright' is opaque: it applies to Hesperus, but does not apply Phosphorus, despite the fact that they are identical. In contrast, 'bright' is (plausibly) a *transparent* predicate: whether Hesperus is bright is simply a matter of what a certain planet is like. If Hesperus is bright, then whatever is identical to Hesperus is also bright. More generally, we can characterize this distinction as follows:

Definition 1. A predicate ϕ of type $\sigma \to t$ is *(materially) transparent* iff

$$\alpha = \beta \ \to \ \phi\alpha \to \phi\beta$$

is true for all terms α, β of type σ. Otherwise ϕ is *(materially) opaque*.

Opacity and transparency don't just make sense for type $e \to t$ predicates, but also for higher-order predicates. For example, perhaps Asher believes that Cassandra is a lawyer, but not that Cassandra is an attorney. That is to say, the higher-order predicate λX (Asher believes X(Cassandra)) is plausibly opaque. In contrast, consider the predicate λX(there is an X) (that is, the existential quantifier $\exists_e$, which has type $(e \to t) \to t$). Plausibly this is like 'bright' rather than 'believed

to be bright': it follows from the fact that being an attorney is being a lawyer, and the fact that there is an attorney, that there is a lawyer.

To better understand the Leibnizian claim L(), let's examine how it can be derived from a certain higher-order generalization.

L(*xyX*) $\forall x \forall y \forall X(x = y \rightarrow Xx \rightarrow Xy)$

L(*xy*) $\forall x \forall y(x = y \rightarrow Bx \rightarrow By)$

L() $a = b \rightarrow Ba \rightarrow Bb$

Both steps of this argument involve applications of universal instantiation. The step from L(*xyX*) to L(*xy*) instantiates the second-order variable *X*, and the step from L(*xy*) to L() instantiates two first-order variables *x* and *y*. Accordingly, there are three ways to reject L().

The first way departs most radically from a Leibnizian view of identity, by rejecting L(*xyX*). It's natural to worry that someone who takes this line has lost contact with the original subject matter — for in this case there would be a relation that seems even more identity-like than identity, namely that of *Leibniz equivalence*:

$$\lambda x \lambda y \forall X(Xx \rightarrow Xy)$$

This relation clearly does satisfy L(*xyX*), in the sense that replacing = with Leibniz equivalence in L(*xyX*) yields a logical truth. (Indeed, if one were to reject L(*xyX*) without tampering with universal instantiation, then Leibniz equivalence would also satisfy L() in the same sense, and thus precisely play the role of identity in classical logic.) If rejecting L(*xyX*) is to amount to a solution to the puzzle, one would also have to say that there are objects — Hesperus and Phosphorus — which are identical, but not Leibniz equivalent. That is, although they are not different objects, there are differences between them — there are properties that apply to one but not the other. We find this view barely intelligible. So, while we will still consider views that reject even L(*xyX*), our main focus will be on views that accept at least this much of the standard logic of identity: there is at least this much right about the Leibnizian idea that one thing has the same properties as itself.

The second way of resisting the argument departs from the classical logic of quantification, blocking the step from L(*xy*) to L() by rejecting the inference from $\forall x\, \phi(x)$ to $\phi(a)$ (or similarly for *y* and *b*). Given the duality of existential and universal quantification, this is equivalent to saying that one cannot infer $\exists x\, \phi(x)$ from $\phi(a)$. In this case we'll say for short that *a* is not *exportable*.

Definition 2. A term α of type σ is *exportable* iff

$$\forall x\, \phi x \rightarrow \phi\alpha$$

is true for every predicate ϕ of type $\sigma \rightarrow t$.

Failures of exportability are familiar territory for philosophers working on attitude reports. It's commonly held that quantifying into intentional contexts can be treacherous (e.g. Quine 1956; Kaplan 1968). For example, even if the inspector believes that Jack the Ripper committed the Whitechapel murders, many are reluctant to admit that *there is some person* that the inspector believes to have committed the murders. If this inference is invalid, then 'Jack the Ripper' is not an exportable name. Similarly one might think that 'Hesperus' and 'Phosphorus' fail to export from contexts involving belief, thus blocking the move from L(*xy*) to L().

We should contrast this kind of exportability failure with another kind that arises from non-being. While it seems as though one can truly assert that Zeus is a mythological god, it's dubious to conclude that there is some being which is a mythological god — since it's dubious that there is any such being as Zeus. But the kind of exportability failure presently under consideration is not like this: it's not dubious at all that there is such a thing as Hesperus — that is, the planet Venus. And even if we assume that there is such a person as Jack the Ripper, this does not make the conclusion that the inspector has beliefs about him more plausible. If 'Hesperus' and 'Jack the Ripper' are not exportable, this is not due to non-being. In general, we must distinguish *exportability* from *existence*.

The third way of resisting the argument for L() makes the analogous move for the second-order generalization: that is, one might say that 'λx (Asher believes x is bright)' is not an exportable *predicate*. As in the case of first-order export failures, one way to motivate this higher-order exportability failure would be by appeal to non-being. For example, plausibly there are no such properties as being dephlogisticated, or being absolutely at rest. One might similarly say that there is no such property as being believed to be bright. But this is not forced on us. We'll explore the alternative, where 'believed to be bright' is treated uniformly with with terms like 'Hesperus' and 'Phosphorus'. While 'believed to be bright' may not in general be exportable, this is not because there is no such property as being believed to be bright. Rather, we shall assume that there is such a property, and see where this assumption leads.

These responses raise the possibility of two distinct varieties of export failure (besides the non-being kind). First, there is 'Hesperus'-type export failure, where certain expressions fail to export from *within* an opaque context. Second, there

may be export failure for expressions which *create* opaque contexts, like 'believed to be bright'. Does opacity require both kinds of export failure, or can we get by with just one of them? It appears that either failure on its own would block the route to L() we just considered: 'Hesperus'-type export failure could block the move from L(*xy*) to L(), or 'belief'-type export failure could block the move from L(*xyX*) to L(*xy*). But in fact, we can argue for the existence of both kinds. (These arguments deploy some quantificational reasoning which we will discuss in detail in section 3. For now, we will keep those details in the background.)

First, we will argue that 'belief'-type export failure is insufficient on its own. The argument turns on the following fact. Consider a transparent predicate, such as 'bright'. Since Hesperus is something, and Hesperus is bright, it does follow that something is bright. This inference is fine even if 'Hesperus' is not exportable in general. More generally, the key observation is that as long as a is something, then even if a is not generally exportable, it is still always possible to export a from a transparent context. That is to say, we can prove that if ϕ is transparent, then

$$\exists x\, x = a \rightarrow \forall x\, \phi x \rightarrow \phi a$$

is true — whether or not a is exportable in general.[1] Note that this holds at every type: for example, it holds when a is not a name but rather a predicate of type $e \rightarrow t$, and ϕ is a transparent higher-order predicate of type $(e \rightarrow t) \rightarrow t$.

Now, note that to make the move from L(*xyX*) to L(*xy*), we instantiate the second-order variable X with the predicate B in a context that involves nothing but logical vocabulary:

$$\lambda X \forall x \forall y (x = y \rightarrow Xx \rightarrow Xy)$$

Plausibly, purely logical predicates like this one are transparent. If this is so, then we can apply our observation: even if B is not exportable in general, it can still be instantiated in transparent contexts, and so it is still legitimate to move from L(*xyX*) to L(*xy*), given our assumption that there is such a property as B. So this argument

[1] *Proof sketch.* If ϕ is transparent, we have $\forall x(x = a \rightarrow \phi x \rightarrow \phi a)$. Then the quantificational logic we present in section 3 guarantees

$$\forall x(\phi x \rightarrow x = a \rightarrow \phi a)$$

and subsequently

$$\forall x\, \phi x \rightarrow \exists x\, x = a \rightarrow \phi a$$

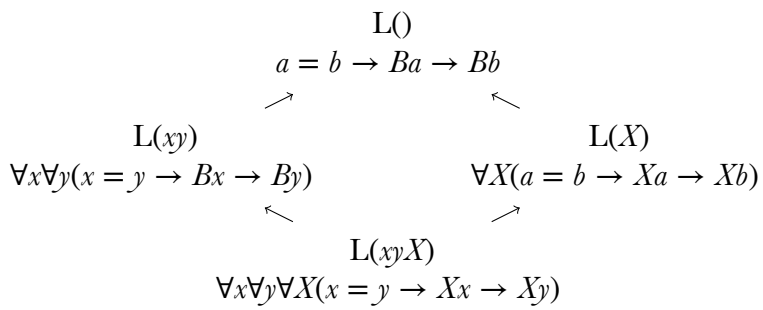

Figure 1: Two routes from the generalization L(xyX) to the instance L().

for L() cannot be blocked by 'belief'-type export failure after all: it must be blocked by 'Hesperus'-type export failure instead.[2]

Is 'Hesperus'-type export failure enough on its own? It is not, because there is another route from L(xyX) to L() which also must be blocked.

L(xyX)**.** $\forall x \forall y \forall X(x = y \rightarrow Xx \rightarrow Xy)$

L(X)**.** $\forall X(a = b \rightarrow Xa \rightarrow Xb)$

L()**.** $a = b \rightarrow Ba \rightarrow Bb$.

This argument differs from the first regarding the order in which the variables are instantiated: in this case, first we instantiate the first-order variables, and then the second-order variable. (See figure 1.)

Again, it initially seems that either instantiation step may fail. But again, transparency considerations push us to the view that the step from L(xyX) to L(X) is legitimate, regardless of whether the names a and b are exportable in general. For again, we can move from L(xyX) to L(X) while only instantiating predicates which are purely logical, and thus plausibly transparent.[3] The upshot is that, if purely logical predicates are transparent, then this second route to L() cannot be blocked

[2] We'll also consider an independent route to L(xy) in section 3.

[3] The argument in this case is a bit more delicate, because it involves instantiating two variables rather than one. But we can get around this with a bit of trickery. This predicate is purely logical:

$$\lambda x \forall y \forall X(x = y \rightarrow Xx \rightarrow Xy)$$

So we can instantiate it with a:

$$\forall y \forall X(a = y \rightarrow Xa \rightarrow Xy) \tag{1}$$

by appeal to 'Hesperus'-type export failure. If L(*xyX*) is true and L() is false, the culprit must be 'belief'-type export failure in this case. In other words, even if Hesperus is believed to be bright and Phosphorus is not, it doesn't follow that there is some property that Hesperus has but Phosphorus lacks. This conclusion is independent of whether 'Hesperus' and 'Phosphorus' are exportable in general.

To sum up, the route to L() through L(*xy*) must be blocked by 'Hesperus'-type export failure, and the route through L(*X*) must be blocked by 'belief'-type export failure. So we need both kinds of export failure to save opacity.

Note that in general these two types of export failure need not apply only to type e and type $e \to t$ expressions, respectively. Rather, 'Hesperus'-type export failure can arise for expressions of any type. An example we considered earlier is that Asher might believe Cassandra is a lawyer without believing Cassandra is an attorney. Even so, there may be no property which Asher believes Cassandra to have and not have. So 'lawyer' and 'attorney' are examples of higher-order 'Hesperus'-type export failure. Likewise, 'belief'-type export failures can arise for expressions of any predicative type. For example, this plausibly applies to the $t \to t$ operator 'λp (Asher believes that p)'. In section 4 we will argue that for any non-exportable name a, the higher predicate $\lambda X\, Xa$ of type $(e \to t) \to t$ is another example. A key contrast is that 'belief'-type export failure arises only for opaque predicates, whereas one could in principle have a completely transparent predicate that is subject to 'Hesperus'-type export failures — such as 'attorney'. (We will argue in section 4 that for any non-exportable name a, the predicate $\lambda x\, x = a$ is transparent, but not exportable.)

So far we have been specifically discussing the Leibnizian principles L(), L(*xy*), L(*X*), and L(*xyX*) that involve the names 'Hesperus' and 'Phosphorus' and the predicate 'believed to be bright'. Obviously this is just one example of opacity: in what follows we'll consider the general schemata of which those principles were instances. From here on out, we'll use the label L(*X*) for the general schema $\forall X(\alpha = \beta \to X\alpha \to X\beta)$, where α and β are schematic letters that can be instantiated with

---

We can apply similar reasoning to b, to also derive:

$$\forall x \forall X(x = b \;\to\; Xx \to Xb) \tag{2}$$

By unobjectionable quantificational reasoning (see section 3) we can combine (1) and (2):

$$\forall z \forall X((a = z \;\to\; Xa \to Xz) \wedge (z = b \;\to\; Xz \to Xb)) \tag{3}$$

Furthermore, if $a = b$ and a is something, then there is something which is both a and b: that is, $\exists z(a = z \wedge z = b)$. Putting this together with (3), we can deduce L(*X*).

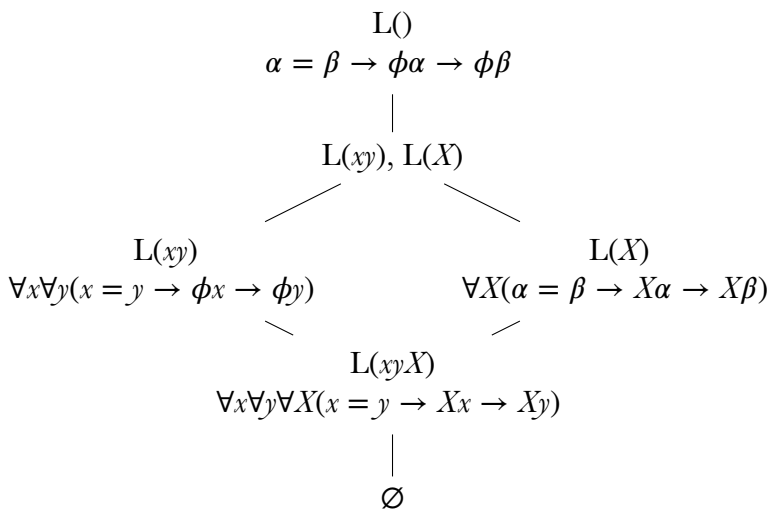

Figure 2: Leibnizian schemata partially ordered by strength.

arbitrary terms. (In other words, the L(X) schema says that each *exportable* predicate is *transparent*. If ϕ is exportable, then by definition it can be used to instantiate L(X). This yields $\alpha = \beta \to \phi\alpha \to \phi\beta$, which is to say that ϕ is transparent.) We can similarly generalize the other three Leibnizian principles (see figure 2).

We will not only treat the terms in these principles as schematic, but also their types. For example, the L(X) schema has instances where α and β have type $(e \to t)$ and the variable X has type $(e \to t) \to t$. Thus one instance of L(X) says that if being a lawyer is the same as being an attorney, then every higher-order property of lawyerhood is shared by attorneyhood. (But as always, we should be cautious about *instantiating* this generalization. For example, we ought not instantiate 'every higher-order property' with λX (Asher believes X(Cassandra)).)

These schematic principles are partially ordered by strength as in figure 2. We can move from each node in the diagram to lower nodes by universal generalization — which we suppose is valid — and to higher nodes by universal instantiation — which we do not. (The obvious exception to this is the upper 'junction' node. As we mentioned earlier, we will comment on details of the quantificational logic in section 3.)

2 Opaque Semantics

Frege's puzzle is often taught as part of the philosophy of language. When encountered in this context, it's natural to describe it as a puzzle about how to assign semantic values to certain sentences — *belief reports*. People coming at it this way naturally take up a 'semantics first' methodology: the primary job is to engineer the right gizmos to serve as semantic values for names, predicates, and operators like 'believes', which when finally assembled and running will spit out the right pattern of felicity for sentences including 'Asher believes Hesperus is bright' and the rest of the Triad. Thus much of the debate over Frege's puzzle centers on questions about the nature of propositions, the relation of names to what is named, and so on.

In contrast, our approach is 'logic first'. The primary questions we are asking are not semantic questions about linguistic expressions, but rather logical questions about planets and what is believed about them. But while semantic engineering is not our primary concern, logic does have implications for semantic theorizing — as indeed it does for all theorizing.

A common thought (received from Frege 2010) is that the phenomenon of opacity calls for 'fine-grained' semantic values: for example, the proposition expressed by 'Hesperus is bright' must be distinct from that expressed by 'Phosphorus is bright', and accordingly the semantic value of 'Hesperus' is distinct from that of 'Phosphorus'. Here's a way of unpacking this idea (see also McDowell 1977).

First, some notation. For any expression α of type σ, let $\langle\alpha\rangle$ be the quotation name for the expression α, and let $[\![\langle\alpha\rangle]\!]$ be the semantic value of this expression. We'll take advantage of our background type theory here: let's suppose that the semantic value of a name has type e, the semantic value of a predicate has type $e \to t$, and in general $[\![\langle\alpha\rangle]\!]$ has the same type as α.

A natural view — sometimes called the *Millian Thesis* — says that the semantic value of a name just is the thing it names, rather than anything intensional, fine-grained, or otherwise fancy.

Disquotation.

$$[\![\langle a\rangle]\!] = a$$
$$[\![\langle b\rangle]\!] = b$$

(Here again a and b stand for 'Hesperus' and 'Phosphorus'.) But as attractive as this simple picture is, many have been deterred from it by something like the following reasoning. First, and most obviously,

Identity.

$$a = b$$

Whether or not semantics is fancy, there is just one planet Venus. Second, though, if both belief reports in the Triad are correct, then it seems that we should also accept

Difference.

$$[\![\langle Ba \rangle]\!] \neq [\![\langle Bb \rangle]\!]$$

(Here B again stands for the predicate 'believed to be bright'.) For (we are supposing) the first sentence Ba is true and the second sentence Bb is false, and surely this reflects some difference between their semantic values. But these three claims appear to conflict with a standard assumption of semantic theorizing: Frege's doctrine that the meanings of the parts determine the meaning of the whole. In particular,

Compositionality.

$$[\![\langle Ba \rangle]\!] = [\![\langle B \rangle]\!][\![\langle a \rangle]\!]$$
$$[\![\langle Bb \rangle]\!] = [\![\langle B \rangle]\!][\![\langle b \rangle]\!]$$

(Recall that we are taking semantic values to respect types, so the semantic value of the predicate B is itself predicable.) For by Identity and Disquotation, it follows that

$$[\![\langle a \rangle]\!] = [\![\langle b \rangle]\!]$$

(This is supposing that identity is at least an equivalence relation — a point to which we will return in section 4.) Then by Leibniz's law,

$$[\![\langle B \rangle]\!][\![\langle a \rangle]\!] = [\![\langle B \rangle]\!][\![\langle b \rangle]\!]$$

So, by Compositionality (and again the fact that identity is an equivalence),

$$[\![\langle Ba \rangle]\!] = [\![\langle Bb \rangle]\!]$$

This contradicts Difference.

In light of this kind of argument, many theorists reject Disquotation. For instance, the denotation of a name might be an individual concept — a function from worlds to individuals — or it might be something like a word, or a rule, or a mental image. (In a sense, neo-Fregeans about meaning fall in this camp, but

the classification is delicate because, following Frege 2010, these theorists typically hold that expressions have more than one kind of semantic value, none of which is privileged. For Frege, 'Hesperus' and 'Phosphorus' have different (customary) *senses*, despite having the same (customary) *referent*. See also e.g. Chalmers 2006, sec. 3.5.)

If we are to maintain Compositionality and Identity, failures of Disquotation will have to be pervasive, rather than isolated to the case of names. For according to this view, $a = b$, but $[\![\langle a \rangle]\!] \neq [\![\langle b \rangle]\!]$. But also, by another instance of Compositionality (this time for the identity predicate = rather than the predicate B),

$$[\![\langle a = b \rangle]\!] = [\![\langle = \rangle]\!][\![\langle a \rangle]\!][\![\langle b \rangle]\!]$$

So, given Identity, $[\![\langle = \rangle]\!]$ holds between $[\![\langle a \rangle]\!]$ and $[\![\langle b \rangle]\!]$, even though $[\![\langle a \rangle]\!]$ and $[\![\langle b \rangle]\!]$ are distinct. So it would follow by similar reasoning that

$$[\![\langle = \rangle]\!] \neq (=)$$

That is, the semantic value of the identity predicate is not just *being identical*.

But for those who take opacity seriously, there is another way. The argument for Difference relied on taking the Triad at face value: it is really *true* that Asher believes Hesperus is bright, and *false* that Asher believes Phosphorus is bright. Anyone who accepts this much — and who also accepts that Hesperus just is Phosphorus — is committed to giving up the form of Leibniz's law we have called L(). But the semantic argument above relies on precisely this form of Leibniz's law. (In particular, it relies on the instance that says that if $[\![\langle a \rangle]\!] = [\![\langle b \rangle]\!]$, then since $[\![\langle B \rangle]\!][\![\langle a \rangle]\!] = [\![\langle B \rangle]\!][\![\langle a \rangle]\!]$, it follows that $[\![\langle B \rangle]\!][\![\langle a \rangle]\!] = [\![\langle B \rangle]\!][\![\langle b \rangle]\!]$.) So anyone who is persuaded that Difference is true should also hold that the argument against Disquotation is invalid. Thus no one should be persuaded that the argument is sound.

Anyone who takes opacity at face value concedes that the property of being believed to be bright applies to Hesperus but not Phosphorus, even though they are identical. It's not clear why someone who concedes this much would resist the analogous claim about the semantic value of 'believed to be bright'. (Indeed, one might reasonably think that the property of being believed to be bright just *is* the semantic value of the predicate.) Semantic theorizing is governed by the same logic as anything else. It is natural to take the above argument to show that semantic expressions can be just as opaque as their corresponding object language predicates. Disquotation, Compositionality, Identity, and Difference are not inconsistent: rather, together they imply that $[\![\langle B \rangle]\!]$ is *opaque*.

Taking opacity seriously thus opens up the possibility of a distinctive kind of semantic theory: one that upholds the Millian Thesis that the semantic value of a name is exhausted by its referent, without giving up Compositionality, Identity, or Difference. The key idea of this approach is that what's good for the object language is good for the meta-language: not only is there opacity in what a person believes, but also in what a word means.

A more general moral is that *opacity* is independent of *fineness of grain*. One might be tempted by the thought that a theory of propositions that are tightly coupled to the cognitive significance of sentences would somehow get one off the hook for violations of Leibniz's law, while still allowing one to affirm the Triad. This thought can't be right: the Triad is inconsistent with L(), and this inconsistency won't magically go away just because we go on to say many other things about what propositions are like. Here's one form this tempting thought might take. Perhaps the proposition that Hesperus is bright is distinct from the proposition that Phosphorus is bright (though Hesperus and Phosphorus are one and the same); in that case Asher could believe one of these propositions without believing the other, and not thereby fall into a violation of Leibniz's law. But the violation has already been committed: if $a = b$, then another instance of L() says that, since $Fa = Fa$, $Fa = Fb$ as well. That is, since Hesperus is Phosphorus, L() implies that the proposition that Hesperus is bright is identical to the proposition that Phosphorus is bright. Whatever your theory of propositions, you have take your lumps: either there is something wrong with the Triad or there is something wrong with L().

Fine-grained propositions don't provide a way of keeping L(), but neither are they required for views that give up L(). Indeed, opacity is compatible with propositions being very coarsely individuated: for example, propositions might be sets of metaphysically possible worlds, or even just truth-values. This point will be justified more fully in what follows. (But it is a bit delicate when further constraints on identity are imposed; see our qualification at the end of section 4.)

Since fineness of grain is largely independent of opacity, we will remain officially neutral on the question of what propositions are like, and whether or not there are many distinct logically equivalent propositions.

3 Leibnizian Logics

As we discussed in section 1, accepting opacity means rejecting L(), as well as the classical rules of universal instantiation and existential generalization. But if we knock out these pillars of classical logic, is there any structure left that can stand on

its own? In this section we'll identify a minimal core logic that is acceptable to the opacity-lover; in the following section we'll consider a natural way of strengthening this.

We'll begin with the quantifiers. Since we can't allow arbitrary instances of universal instantiation or existential generalization, a natural starting point is free logic, which admits exceptions to these principles. Rather than the classical instantiation principle

$$\forall x\, \phi x \to \phi \alpha \tag{4}$$

we make use of the following weaker version of universal instantiation (from Lambert 1963 and Kripke 1963):

$$\forall y(\forall x\, \phi x \to \phi y) \tag{5}$$

While free logic is usually discussed in the context of first-order logic, we intend our discussion to apply equally to the higher-order quantifiers; here, for example, x and y may be variables of any type. Note that (5) does not imply the classical principle (4): deriving it would require an instance of the very principle of universal instantiation that we repudiate.

We should be careful to distinguish (5) from an alternative version of universal instantiation that some systems of free logic take for granted: if α is something and everything is ϕ, then α is ϕ.

$$\exists x(x = \alpha) \to \forall x\, \phi x \to \phi \alpha \tag{6}$$

In the presence of Leibniz's law L(), (6) is a consequence of our other quantificational principles.[4] But of course, we are not assuming L() — nor should we assume (6). Earlier we mentioned cases like 'Jack the Ripper' that seemed like they might

[4] The following is obtained by universal generalization from L():

$$\forall x(x = \alpha \;\to\; \phi x \to \phi \alpha)$$

Rearranging the conditionals and distributing the quantifier,

$$\forall x\, \phi x \to \forall x(x = \alpha \to \phi \alpha)$$

Then, distributing the quantifier again and using duality,

$$\forall x\, \phi x \to \exists x\, x = \alpha \to \exists x\, \phi \alpha$$

The last existential quantifier is vacuous, so we can drop it, leaving us with something obviously equivalent to (6).

be counterexamples to (6). Given that we are rejecting L(), we also have a more theoretical reason for rejecting (6). In section 1 we argued for L(xy) (and we shall give a second argument below):

$$\forall x \forall y(x = y \rightarrow Bx \rightarrow By) \tag{7}$$

Hesperus and Phosphorus are each something (indeed, the same thing), so if (6) were correct we could instantiate (7) with 'Hesperus', and then instantiate the result with 'Phosphorus'. Since Hesperus is Phosphorus, this would imply that if Hesperus is believed to be bright then so is Phosphorus, contradicting the original Triad. In general, if L(xy) is true but L() has false instances, then (6) also has false instances.

In fact, we will assume each instance of this schema:

Being. $\exists x\, x = \alpha$

(where α may have any type). With (6) this would obviously entail (4), collapsing back to classical logic; but without (6) there is no such danger. Plausibly Being has counterexamples: for plausibly there is no such thing as Zeus and there is no such property as being dephlogisticated. Even so, we are setting aside these cases to focus on the distinctive logical issues raised by opacity.

Even for those who are suspicious of Being, another related principle is very plausible. For instance, someone might doubt that there is any such property as being believed to be bright; even so, it is still natural to think that there is at least a property *coextensive* with being believed to be bright. Suppose the only things that Asher believes to be bright are the sun, the moon and Venus. Surely there is at least the property of being one of those three things.[5] (Does Asher really believe Venus is bright? This is not settled by things we have said so far.) This is an instance of a more general principle: for each predicate in our language there is a coextensive property.[6]

Comprehension. $\exists X \forall y(Xy \leftrightarrow \phi y)$

[5] That is,

$$\exists x \exists y \exists z(x = \text{Sun} \wedge y = \text{Moon} \wedge z = \text{Venus} \wedge \exists X(X = \lambda w(w = x \vee w = y \vee w = z)))$$

[6] We might also assume versions of this principle that apply to relations of arbitrary arity. We don't need these here, and in fact they will follow from principles we introduce in section 4.

We will also assume Comprehension in what follows. (Note that in the current context Comprehension does not obviously follow from Being: for in fact, we have not yet shown that if $X = \phi$ then X and ϕ are coextensive. Don't worry — we'll get this in section 4.)

Comprehension has an important consequence for the Leibnizian principles: given Comprehension, L(*xyX*) implies the superficially stronger principle L(*xy*). Let ϕ be an arbitrary predicate. By Comprehension there is some property X which is coextensive with ϕ. By L(*xyX*), for any x and y that are identical, if x has X then so does y. Since ϕ is coextensive with X, if ϕx then Xx; and likewise, if Xy, then ϕy. So, chaining this together, if x and y are identical, then if ϕx, then ϕy. This is just what L(*xy*) requires.

One final logical note concerns λ-abstraction. At the outset we took it for granted that λx (Asher believes that x is bright) applies to Hesperus if and only if Asher believes Hesperus is bright. In fact, we will suppose even more. Not only are these two claims materially equivalent, but in fact they are the very same proposition; even further, they are intersubstitutable, in the sense that whatever may be asserted of one may be asserted of the other. In general we are supposing that $\beta\eta$-equivalent expressions are intersubstitutable. That is: $(\lambda x\, \phi)\alpha$ is intersubstitutable with the substitution instance $\phi[\alpha/x]$ (*β-reduction*), and, when x is not free in ϕ, $\lambda x\, \phi x$ is intersubstitutable with ϕ (*η-reduction*). Note that this is a fairly strong principle; some might be interested in exploring how thing go without it. For example, some structured-proposition-theorists might wish to distinguish the propositions that John loves Mary (*Lab*), that the loving relation applies to John and Mary ($(\lambda x\, \lambda y\, Lxy)ab$), that John has the property of loving Mary ($(\lambda x\, Lxb)a$), and so on. But at this stage of our investigation the simplifying power of full-fledged $\beta\eta$-reduction is to be welcomed.

To sum up, in what follows we will rely on the following *Background Logic*.

$\beta\eta$-Reduction. $\phi[\alpha/x] \rightarrow \phi[\beta/x]$, whenever α and β are $\beta\eta$-equivalent.

Propositional Logic All substitution instances of propositional tautologies.

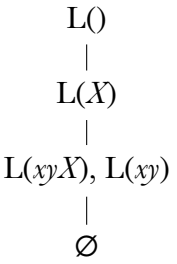

Figure 3: Leibnizian schemas, given Comprehension.

Free Logic. [7]

$\forall x(\forall y\, \phi y \rightarrow \phi x)$ (Quantified Instantiation)
$\forall x(\phi \rightarrow \psi) \rightarrow \forall x\, \phi \rightarrow \forall x\, \psi$ (Normality)
$\forall x \forall y\, \phi \rightarrow \forall y \forall x\, \phi$ (Quantifier Exchange)
$\phi \rightarrow \forall x\, \phi$ when x does not occur free in ϕ (Vacuous Quantification)

Being. $\exists x\, x = \alpha$

Comprehension. $\exists X \forall y(Xy \leftrightarrow \phi y)$

Note that instances of these schematic principles can contain free variables. When we affirm such an instance, that should be taken as an affirmation of its universal closure (that is, the result of prefixing it with enough universal quantifiers to bind every free variable). We also suppose that modus ponens preserves truth.

With this background logic in place we are in a position to precisely describe the relationship between our different Leibnizian principles. In section 1 we presented a lattice of six different forms of Leibniz's law. Given Comprehension (and our background logic of quantification) you can't have L(xyX) without L(xy), and so these six positions collapse to four: see figure 3. Again, each of the higher principles implies those below it in the diagram.

To show that there are really even four different positions, it is useful to provide *models*. It is important to distinguish the models that we will introduce here from the project of giving realistic semantics (of the sort we briefly discussed in section 2).

[7] In the presence of L(), the Quantifier Exchange axiom is derivable from the other axioms of first-order free logic. However, since in the absence of L(), it is not derivable, it must be taken as an axiom (see Fine 1983). The situation with the higher-order quantifiers is analogous.

We are regarding these models instrumentally as devices for demonstrating certain facts about logical consequence — in particular, the logical independence of some of our Leibnizian principles from L() and from one another. We are not taking them to exhibit what our language is really 'about'. For example, we can have an identity sentence $a = b$ which counts as true in a certain model, even though a and b denote-in-the-model certain mathematical objects that are really distinct.

Here is the basic idea for how these models will work. In models of classical logic one standardly uses a *domain* — a certain set — for two purposes. First, the domain includes the objects over which the quantifiers are taken to range; second, it includes the objects which are available to be denoted by terms. In free logic one standardly separates these two purposes: one introduces an *inner domain* as well as a more inclusive *outer domain*. The quantifiers range over the inner domain, but terms can denote arbitrary objects in the outer domain. Thus if ϕ is a predicate that is satisfied by just those things in the inner domain, and α denotes some outer element, then $\forall x\, \phi x$ will come out true in the model, while $\phi\alpha$ still comes out false.

We'll extend this idea by also providing non-standard truth conditions for identity statements. Standardly, an identity statement $a = b$ is taken to be true in a model if and only if a and b denote the very same element of the domain. We will relax this condition, and instead provide some equivalence relation on the outer domain which suffices for the truth of identity statements. For each outer element d there will be some equivalent inner element d', guaranteeing the truth of Being.

We also need to extend these two ideas to higher-order logic. Standard models of higher-order logic provide a whole family of different domains, one for each type. Since we also allow counterexamples to higher-order instantiation, we extend the 'outer domain' idea from the first-order domain to arbitrary types. Thus for every type σ a model will include an inner domain $M\sigma$ and an outer domain $\mathcal{N}\sigma$. Likewise, to interpret higher-order identities, we will provide an equivalence relation on each domain $\mathcal{N}\sigma$. Depending on which Leibnizian principles we want the model to capture, we will need to impose certain constraints on how identity is related to higher-order domains. For example, if a model is to satisfy L(X), then the inner domain $M(e \to t)$ should not include any properties that distinguish between an outer object d and its 'identical' inner object d'.

To be more specific, for any types σ and τ, the outer domain $\mathcal{N}(\sigma \to \tau)$ will consist of functions from $\mathcal{N}\sigma$ to $\mathcal{N}\tau$ (though not necessarily all such functions).[8] The inner domain $M(\sigma \to \tau)$ will be some subset of $\mathcal{N}(\sigma \to \tau)$: so note that the inner functions don't just take inner objects as arguments. (This is important for making

[8] Thus $\mathcal{N}$ is a 'Henkin model' (Henkin 1950).

sense of expressions like 'there is some property that Hesperus has'.)

Theorem 1. *There exist Leibniz models that satisfy our Background Logic, as well as:*

(i) $L(xyX)$, $L(xy)$, *and* $L(X)$ *but not* $L()$.

(ii) $L(xyX)$ *and* $L(xy)$, *but not* $L(X)$ *or* $L()$.

As we discussed above, given Comprehension, we cannot have $\mathrm{L}(xyX)$ without $\mathrm{L}(xy)$. It is also straightforward to construct models that satisfy none of the Leibnizian principles, by interpreting the identity symbol as some arbitrary relation, but we won't bother to spell out the details.

The proof (together with more explicit definitions) is given in appendix A; but here's one point about our strategy worth highlighting. In order to construct models it is helpful to consider some apparently stronger Leibnizian principles. L() implies that the proposition that Hesperus is bright is materially equivalent to the proposition that Phosphorus is bright. But (as we mentioned in section 2) it implies more: the proposition that Hesperus is bright *is* the proposition that Phosphorus is bright. In general L() implies:

$\mathbf{L}()^{=}\ \ \alpha = \beta \ \rightarrow\ \phi\alpha = \phi\beta$

(Recall: since $\phi\alpha = \phi\alpha$, we can apply L() to the predicate $\lambda x(\phi\alpha = \phi x)$.) We can also go the other direction, from $\mathrm{L}()^{=}$ to L(), if we assume that identical propositions have the same truth value:

Truth Value. $\phi = \psi \ \rightarrow\ (\phi \leftrightarrow \psi)$

(In section 4 we will derive this principle from something more general.) One reason this principle $\mathrm{L}()^{=}$ is technically useful is that, unlike L(), it straightforwardly generalises to the case where ϕ has some type other than $\sigma \rightarrow t$. In general, $\mathrm{L}()^{=}$ makes sense when ϕ has any type of the form $\sigma \rightarrow \tau$ (where the identity on the left side of the conditional is $=_{\sigma}$, and the identity on the right side is $=_{\tau}$).

Likewise each of our other Leibnizian principles has a corresponding 'strict' principle which is not about material equivalence but about identity:

$\mathbf{L}(xyX)^{=}\ \ \forall x\forall y\forall X(x = y \rightarrow Xx = Xy)$

$\mathbf{L}(X)^{=}\ \ \forall X(\alpha = \beta \rightarrow X\alpha = X\beta)$

$\mathbf{L}(xy)^{=}\ \ \forall x\forall y(x = y \rightarrow \phi x = \phi y)$

Again given Truth Value each of these principles about identity entails the corresponding principle about material equivalence. The converse implication also goes through similarly for $\mathrm{L}(xy)^{=}$. The cases of $\mathrm{L}(xyX)^{=}$ and $\mathrm{L}(X)^{=}$ are a bit more delicate: our derivations of these principles from $\mathrm{L}(xyX)$ and $\mathrm{L}(X)$ rely on ideas we will introduce in section 4.[9] But we can ignore these niceties if we make a simplifying assumption:

Extensionality. $(\phi \leftrightarrow \psi) \rightarrow \phi = \psi$

Of course propositions might be finer grained than this — there might be more than two of them. In general, there are models of our Background Logic that do not satisfy Extensionality. The inner domain for type t can be an arbitrary set, representing arbitrary proliferation of distinct but equivalent propositions. The only special feature that we generally require of the domain of type t is that it has some distinguished subset of elements representing *true* propositions, as well as interpretations of the logical constants that have the right truth-functional profile. For example, there is a *negation* element in $\mathcal{N}(t \rightarrow t)$ which, applied to any true element of $\mathcal{N}t$, produces an element which is not true, and vice versa. These general models are compatible with propositions being highly structured, like sentences, or very unstructured, like sets of worlds of truth values. But for the purpose of providing models that separate these Leibnizian principles from one another, it is enough to provide extensional models, with the special feature that any pair of true elements is in the extension of the identity symbol, and likewise any pair of untrue elements.

Each of our identity principles correspond to a natural constraint on models. Let $\sim_\sigma$ be the equivalence relation on $\mathcal{N}\sigma$ representing identity. Then, for example, $\mathrm{L}(X)^{=}$ corresponds to the following constraint:

> For any inner function $f \in M(\sigma \rightarrow \tau)$, and any outer elements $d, d' \in \mathcal{N}\sigma$, if $d \sim_\sigma d'$ then $fd \sim_\tau fd'$

That is to say, $\mathrm{L}(X)^{=}$ requires that each inner function preserves identities.

Thus, to prove Theorem 1, it is enough to provide *extensional* models of our Background Logic that satisfy (i) both $\mathrm{L}(xy)^{=}$ and $\mathrm{L}(X)^{=}$ without $\mathrm{L}()^{=}$, and (ii) $\mathrm{L}(xy)^{=}$ without $\mathrm{L}(X)^{=}$. To do this, it is enough to ensure that the corresponding model-theoretic constraints hold. We show how to do this in the appendix.

[9] The case of $\mathrm{L}(xyX)$ relies on the assumption that the logical predicate $\lambda y(Xx = Xy)$ is exportable: this is an instance of the principle Pure Export that we will introduce later. The case of $\mathrm{L}(X)$ is trickier, since even Pure Export does not guarantee that $\lambda x(X\alpha = Xx)$ is exportable — since it includes the non-logical expression α.

4 Purity

At this point many questions about identity remain open. We've noted that our Background Logic does not settle which of the Leibnizian principles L(), L(X), L(xy), and L(xyX) are true; more generally it settles very little indeed about what identity can be like. Is identity even an equivalence relation? Could a true proposition be identical to a false one? Could identical properties apply to different things? Is the property of being Hesperus identical to the property of being Phosphorus? In fact, while the answers to these questions would be settled by L(), what we have put forward so far leaves these questions unanswered. Similarly, we'd like to get a clearer view of the logic of exportability and transparency. If a is exportable, is $\lambda x\, x = a$? Is $\lambda X\, Xa$? Is the constant function $\lambda x\, a$? Are these transparent?

In this section we will make some advance from this state of ignorance: there is a natural position that not only settles the questions we just raised, but also entails all but the strongest of our Leibnizian principles. The key idea is that *purely logical* expressions have a special status. While certain 'cognitively thick' expressions, like 'believes', 'attorney' or 'Hesperus', may be contaminated by intentionality, the language of pure logic — things we can say using variables, λ-abstraction, quantifiers and so on — are immune from this contamination. The general idea is that even if we go so far as to deny L(), we would rather not be pushed into tampering with familiar facts of *pure* logic as well. This idea — which we'll call *Purity* — will be elaborated in three different ways.

First, we would like to affirm all the truths of classical higher-order logic which contain only logical vocabulary. It would be very odd if our views on Frege's puzzle forced us to take a heterodox stand on points of pure mathematics, and many mathematical facts are statable in purely logical terms. For example, there are infinitely many prime numbers. Indeed, the formalization of this claim is a logical consequence of the second-order Peano axioms, which provides us with a corresponding truth of pure higher-order logic.

$$\forall X(\mathsf{PA}\, X \rightarrow \mathsf{Primes}\, X)$$

(Here X is a relation variable of type $e \rightarrow e \rightarrow t$, PA is a pure predicate of type $(e \rightarrow e \rightarrow t) \rightarrow t$ which formalizes the standard properties of the ordering of the natural numbers, and similarly Primes formalizes the claim that this structure includes infinitely many primes.) It would be a shame to give up these arithmetical and logical insights. Similar considerations apply to many claims from group theory, analysis, and so on.

In section 3 we introduced the idea of models for type theory. *Classical* models have several distinctive features. (For details see the appendix.) First, there is no distinction between an inner and outer domain: there is just one domain for each type. Second, the domains for higher types are *full*: the domain of type $\sigma \rightarrow \tau$ includes every function from the domain of type σ to the domain of type τ. Third, the identity predicate receives a standard interpretation: namely, the same denotation as Leibniz equivalence, $\lambda x \lambda y \forall X(Xx \rightarrow Xy)$. A sentence is a *classical logical truth* if and only if it is true in all classical models, in this sense. An expression is *purely logical* if and only if it contains no non-logical constants.

The preceding considerations motivate the following principle:

Pure Truth. Each purely logical classical logical truth is true.

So, for example, our formalization of the claim that there are infinitely many primes in any natural-number structure comes out true. This says nothing, though, about impure logical truths. So Pure Truth is still consistent with failures of universal instantiation, L(), L(X) and L(xy). On the other hand, observe that L(xyX) is a purely logical statement

$$\forall x \forall y \forall X(x = y \rightarrow Xx \rightarrow Xy)$$

and so it is settled in the affirmative by Pure Truth.[10]

Our simple statement of Pure Truth belies hidden complexity. For in fact (unlike the schematic principles we have been discussing) there is no systematic way of enumerating precisely what the classical logical truths are. This follows immediately from the incompleteness of higher-order logic (which in turn follows from Gödel's first incompleteness theorem; see Shapiro 1991, ch. 4). The classical logical truths are not *recursively axiomatizable*; accordingly, the purity package we are outlining is not recursively axiomatizable either: the set of sentences it comprises is highly complex. That's just how it goes: the truth about any sufficiently complex

[10] It should be noted that there may be counterexamples to Pure Truth that have nothing to do with opacity, but rather with very general issues concerning set-theoretic semantics for higher-order languages. For example, there is a pure sentence of higher-order logic that states that the universe has the size of the first inaccessible cardinal; if this sentence were true, it would be false in every classical model. (Roughly, this is because classical models are set-theoretic entities, and thus they all have domains that are strictly smaller than the total number of things.) The principle that says that these sorts of scenarios do not arise is called *Kreisel's principle*, and it is highly non-trivial: it implies the existence of many large cardinals (see Shapiro 1987). Since these issues concerning the size of the universe aren't our direct concern here, we will simply take Kreisel's principle for granted without further argument.

subject matter is complex, and as it turns out the logic of properties is sufficiently complex in the relevant sense. This has nothing special to do with opacity; it is just a feature of working in a higher-order context.

Our formulation of Pure Truth has the consequence that identity *just is* Leibniz equivalence:

The Identity Identity. $(=) \;=\; \lambda x \lambda y \forall X(Xx \rightarrow Xy)$

Some might take exception to the Identity Identity on the grounds that propositions, properties, and relations are highly structured, 'fine-grained' entities. For example, one might think that 'snow is white and grass is green' expresses an equivalent but distinct proposition from 'grass is green and snow is white'. So likewise in this case one might think that while identity is *equivalent* to Leibniz equivalence — in the sense that they each apply to the same pairs of elements of any classical model — they are distinct relations even so.

In a sense, we hard-coded the Identity Identity into our formulation of the Pure Truth principle, in our definition of a 'standard interpretation' for the identity symbol. This could be avoided by suitably modifying the Pure Truth principle; for instance we might allow intepretations of identity with the standard application conditions, without *identifying* the interpretation of identity with the interpretation of Leibniz equivalence.[11] But while this move is available, it does not seem well-motivated: even for those with fine-grained views of relations, it's hard to see what the value of drawing this particular distinction would be. If there are indeed many different relations that are all logically equivalent to Leibniz equivalence and to one another, it's hard to see the question of which of these relations is *really* identity as deep and important. From this vantage, using the label 'identity' for the relation of Leibniz equivalence seems like a natural and unobjectionable choice.

In addition to Pure Truth, considerations of purity motivate another principle. We have distinguished between exportable and non-exportable expressions; for instance 'Hesperus' and 'Jack the Ripper' do not always license existential generalization or universal instantiation, but other expressions do — variables, *par excellence*. The second Purity idea is that, just as variables are exportable, so are all other purely logical expressions as well.

[11] In the context of the general model theory for higher-order logic introduced in section 3, the type t domain, representing the set of 'propositions', is an arbitrary set, with some subset representing the 'true' propositions. So in general there will be many different functions that take each identity-pair (d, d) to some 'true' element of the type-t domain, and each non-identity pair (d, d') to some 'false' element. Leibniz equivalence denotes one of these functions, but one might interpret the identity symbol as any other such function.

Pure Export. $\forall x\, \phi x \rightarrow \phi\alpha$, where α is purely logical.

Note that (as before) we are affirming instances of Pure Export where α may have any type. Note also that (as before) by affirming this schema, we are also affirming the universal closure of each instance. For example, the expression $\lambda X\, Xy$ is exportable, which is to say that every instance of $\forall y(\forall x\, \phi x \rightarrow \phi(\lambda X\, Xy))$ is true. (To put that a bit more roughly: for any thing, the higher-order property of applying to that thing is exportable.) Our principle of Quantified Instantiation $\forall y(\forall x\, \phi x \rightarrow \phi y)$ is another instance of Pure Export, where α is simply the variable y. Indeed, we can think of Pure Export as an alternative principle of universal instantiation, which is intermediate between the classical principle and the weaker quantified principle.

Another consequence of Pure Export is that exportability is closed under application: if ϕ (of type $\sigma \rightarrow t$) and α (of type σ) are each exportable then so is $\phi\alpha$ (of type τ).[12] One more consequence that will prove important is that Leibniz equivalence itself, being purely logical, is exportable.[13]

[12] Let ψ be a predicate (of type $\tau \rightarrow t$). This is an instance of Pure Export:

$$\forall X \forall y(\forall x\, \psi x \rightarrow \psi(Xy))$$

If ϕ is exportable, we can instantiate it for X:

$$\forall y(\forall x\, \psi x \rightarrow \psi(\phi y))$$

Likewise, if α is exportable, we conclude:

$$\forall x\, \psi x \rightarrow \psi(\phi\alpha)$$

That is, $\phi\alpha$ is exportable.

Alternatively, we could treat this fact as an axiom, and then derive Pure Export from some *basis* for the logical expressions. For instance, suppose that the combinators S and K are each exportable (that is $\lambda x \lambda y \lambda z\, xz(yz)$ and $\lambda x \lambda y\, x$), as are the logical constants. Suppose further that exportability is closed under application. Then Pure Export follows. (Note that these combinators are really an infinite family of terms of various types. There is an S combinator for each type of the form $(\rho \rightarrow \sigma \rightarrow \tau) \rightarrow (\rho \rightarrow \sigma) \rightarrow \rho \rightarrow \tau$, and similarly a K combinator for each type of the form $\sigma \rightarrow \tau \rightarrow \sigma$.)

[13] Recall that L(xy) says that exportable co-denoting terms are intersubstitutable. If we combine this principle with Pure Export, we can conclude that *logical* co-denoting terms are intersubstitutable: that is, for pure terms α and β, if $\alpha = \beta$ and $\phi\alpha$ then $\phi\beta$. One might worry that this would commit us to *logical omniscience*: anyone who believes one logical truth believes them all. But this does not follow. It would follow from the principle that *logically equivalent* terms are intersubstitutable. But to be logically equivalent is not to be logical, and equivalent (that is, co-denoting). Both parts are wrong. First, there are many logical truths which are not purely logical sentences (like 'if snow is white then snow is white') and many purely logical sentences which are not logical truths

Our third Purity principle is an idea we invoked in section 1: namely, that purely logical predicates are *transparent*. As we noted earlier, while being believed to be bright may apply to Hesperus without applying to Phosphorus, whether Hesperus is *bright* is simply a matter of what a certain planet is like: 'bright' is transparent. The idea here is that logical predicates are like brightness rather than belief.

Pure Transparency. $\alpha = \beta \to \phi\alpha \to \phi\beta$, where ϕ is a purely logical predicate.

In section 1 we appealed to this principle to argue for L(xy) and L(X): purely logical predicates are transparent.

For example, since $\lambda p\, p$ is purely logical, we can derive for any sentences ϕ and ψ

$$\phi = \psi \to (\lambda p\, p)\phi \to (\lambda p\, p)\psi$$

So by β-reduction,

$$\phi = \psi \to \phi \to \psi$$

In other words, if ϕ is true and ψ is not they are distinct, which answers one of the questions we raised earlier: the Truth Value principle follows from Pure Transparency.

Pure Transparency is closely related to the Leibnizian principle L(X). Indeed, L(X) follows from an instance of Pure Transparency where ϕ is simply the variable X. (We also gave an alternative argument in section 1 that only relied on the transparency of *closed* pure terms.) We can also go in the other direction, if we assume Pure Export: in that case we can instantiate the higher-order variable X in L(X) with an arbitrary pure predicate.

Note that Pure Transparency only applies to predicates of one argument (terms of type $\sigma \to t$). In striking contrast, a parallel principle for binary predicates (of type $\sigma \to \tau \to t$) is untenable:

$$\alpha = \alpha' \to \beta = \beta' \to \phi\alpha\beta \to \phi\alpha'\beta' \quad \text{where } \phi \text{ is purely logical} \tag{8}$$

This principle is incompatible with opacity. Just consider this instance, where the binary logical predicate is application, $\lambda X \lambda y\, Xy$:

$$\phi = \phi \to \alpha = \beta \to (\lambda X \lambda y\, Xy)\phi\alpha \to (\lambda X \lambda y\, Xy)\phi\beta$$

(like 'there are exactly three things'). Pure Export is not concerned with logical truths, but with expressions which only involve logical vocabulary. Second, for all we've said, logically equivalent sentences (whether or not they are pure) need not express the same proposition. This would be to take a stand on a question on which we have been officially neutral — the question of fineness of grain.

By β-reduction this implies the strongest Leibnizian principle L().

In general we should take care to distinguish between cases that involve application to one argument and those that involve multiple arguments. One might think that the two-argument principle (8) could be derived by applying Pure Transparency twice.

$$\alpha = \alpha' \rightarrow \phi\alpha\beta \rightarrow \phi\alpha'\beta$$
$$\beta = \beta' \rightarrow \phi\alpha'\beta \rightarrow \phi\alpha'\beta'$$

But in fact neither of these is an instance of Pure Transparency, because neither of the corresponding predicates $\lambda x\, \phi x \beta$ or $\lambda x\, \phi \alpha' x$ are pure. Once a pure binary predicate has had one of its arguments filled with an impure term, the resulting monadic predicate is no longer pure — and thus it is not suitable for instantiating Pure Transparency.[14]

The combination of these three principles — Pure Truth, Pure Export and Pure Transparency — we call *Purity*. Purity is powerful: it can be used to settle the questions about identity, exportability and opacity raised at the beginning of this section, and it implies each of the Leibnizian principles other than L(). In fact we don't need all three principles as independent assumptions. Given our background logic, Pure Truth and Pure Export suffice on their own, and alternatively so do Pure Truth and Pure Transparency. In either case we can derive the third Purity principle, thus showing that they are not independent, but rather form a mutually supporting package.

Call the following schematic principles the *Nice Principles*:

Pure Truth. Each purely logical classical logical truth.

Pure Export. $\forall x\, \phi x \rightarrow \phi\alpha$, where α is a purely logical expression.

[14] Sometimes one simulates binary predication by applying suitable monadic predicates to *ordered pairs*. Suppose we enrich our type theory with ordered-pair types $\sigma \times \tau$, with a pairing term pair of type $\sigma \rightarrow \tau \rightarrow \sigma \times \tau$, and projection terms π_1 and π_2 of types $\sigma \times \tau \rightarrow \sigma$ and $\sigma \times \tau \rightarrow \tau$, respectively. If these pairing operations also count as logical, then Purity demands that we reject a standard principle about the identity of ordered pairs:

$$\alpha = \alpha' \rightarrow \beta = \beta' \rightarrow \text{pair}\,\alpha\beta = \text{pair}\,\alpha'\beta'$$

If we accepted this principle, then given $B = B$ and $a = b$ we would have $\text{pair}\,B\,a = \text{pair}\,B\,b$. But then, consider the expression $\lambda x\,((\pi_1 x)(\pi_2 x))$ (which applies the first element of a pair to the second element). If this is logical, and thus transparent, it would follow that $Ba = Bb$, contradicting the Triad.

Pure Transparency. $\alpha = \beta \rightarrow \phi\alpha \rightarrow \phi\beta$, where ϕ is a purely logical predicate.

Equivalence.

$$\alpha = \alpha \qquad \text{(Reflexivity)}$$
$$\alpha = \beta \rightarrow \alpha = \gamma \rightarrow \beta = \gamma \qquad \text{(The Euclidean Property)}$$

Truth Value. $\alpha = \beta \rightarrow \alpha \rightarrow \beta$

Functionality. $\phi = \psi \leftrightarrow \forall x\,(\phi x = \psi x)$

Leibniz. L(xyX), L(X), L(xy), L(xyX)$^{=}$, L(X)$^{=}$, and L(xy)$^{=}$.

The only principle we have not yet discussed is Functionality. Informally, this says that the inhabitants of functional types are completely determined by their behavior with respect to application. If ϕ and ψ, both of type $\sigma \rightarrow \tau$, produce equal values for each input, then they are themselves equal, and conversely, if ϕ and ψ are the same, they produce the same values for each input. (In classical contexts the left-to-right direction is not typically stated explicitly, since it follows from L().)

The above principles aren't anything like a complete and independent set of axioms. (Indeed, we're about to show that they are very far from independent.) Rather, these principles make a handy starter kit for reasoning in this quasi-Leibnizian setting. Moreover, we need not go on carefully distinguishing each of these variant forms of Leibniz's law, since all of them follow from our Purity principles.

Theorem 2. *Pure Truth and Pure Export together imply each of the Nice Principles (given our Background Logic, stated in section 3).*

Theorem 3. *Pure Truth and Pure Transparency together imply each of the Nice Principles (given our Background Logic).*

These are proved in appendix B.

Recall that in section 1 we argued for L(X) and L(xy) from L(xyX) and the idea that logical predicates are transparent — that is, the idea of Pure Transparency. So Theorem 3 reinforces and extends this point: each of the Leibnizian principles L(xyX), L(xy), and L(X) follows from Purity.

Answering the questions we raised at the beginning of this section — for example whether $\lambda X\,Xa$ is exportable whenever a is, and likewise whether this is transparent — is now an easy exercise. (In fact, this is both exportable and transparent.)

Another application is that, if Hesperus is Phosphorus, then the property of being Hesperus is the property of being Phosphorus. This follows from instantiating the higher-order variable in $\mathrm{L}(X)^{=}$with the purely logical predicate $\lambda y \lambda x\, x = y$:

$$a = b \;\rightarrow\; (\lambda x\, x = a) = (\lambda x\, x = b)$$

We can thus make good on our promise from section 1 to give an example of a predicate which is transparent but not exportable. Suppose that Asher believes Hesperus is Hesperus, but does not believe Hesperus is Phosphorus; or to put it another way, Asher believes $\lambda x\, x = a$ applies to a, but Asher does not believe $\lambda x\, x = b$ applies to a. But these are the same property: so at least one of $\lambda x\, x = a$ and $\lambda x\, x = b$ is not exportable, since otherwise they would be intersubstitutable (by $\mathrm{L}(xy)$). On the other hand, they are both transparent. This follows from Equivalence:

$$\alpha = \beta \;\rightarrow\; (\lambda x\, x = a)\alpha \rightarrow (\lambda x\, x = a)\beta$$

(Likewise for $\lambda x\, x = b$.)

Let's highlight one further perspective on Purity: we can generalize Pure Truth from sentences to monadic predicates. Say that a predicate ϕ (of type $\sigma \rightarrow t$) is *universal* iff it is purely logical and and applies to every element in the σ-domain of every classical model.

Pure Properties. $\phi\alpha$, where ϕ is a universal predicate.

Pure Properties turns out to be a consequence of Purity (see appendix B). But also, conversely, Pure Properties encapsulates the entire package on its own:

Theorem 4. *Pure Properties implies the Nice Principles, given our Background Logic.*

In fact, this result relies on much less of our Background Logic than the other purity results, because much of our Background Logic immediately follows from Pure Properties. Specifically, each of the principles that have only one schematic variable immediately follows from Pure Properties, because each of them follows by β-reduction from the application of a universal predicate to one arbitrary term. This covers everything except for $\beta\eta$-Reduction, Propositional Logic, and Normality. Thus, Pure Properties together with just these three additional assumptions suffice to derive all of our Background Logic, Purity, and the Nice Principles.

Note that one should resist the temptation to generalise even further: the principle which is analogous to Pure Properties for universal *relations* is inconsistent with opacity. Consider this two-place universal relation:

$$\lambda x \lambda X \forall y(x = y \rightarrow Xx \rightarrow Xy)$$

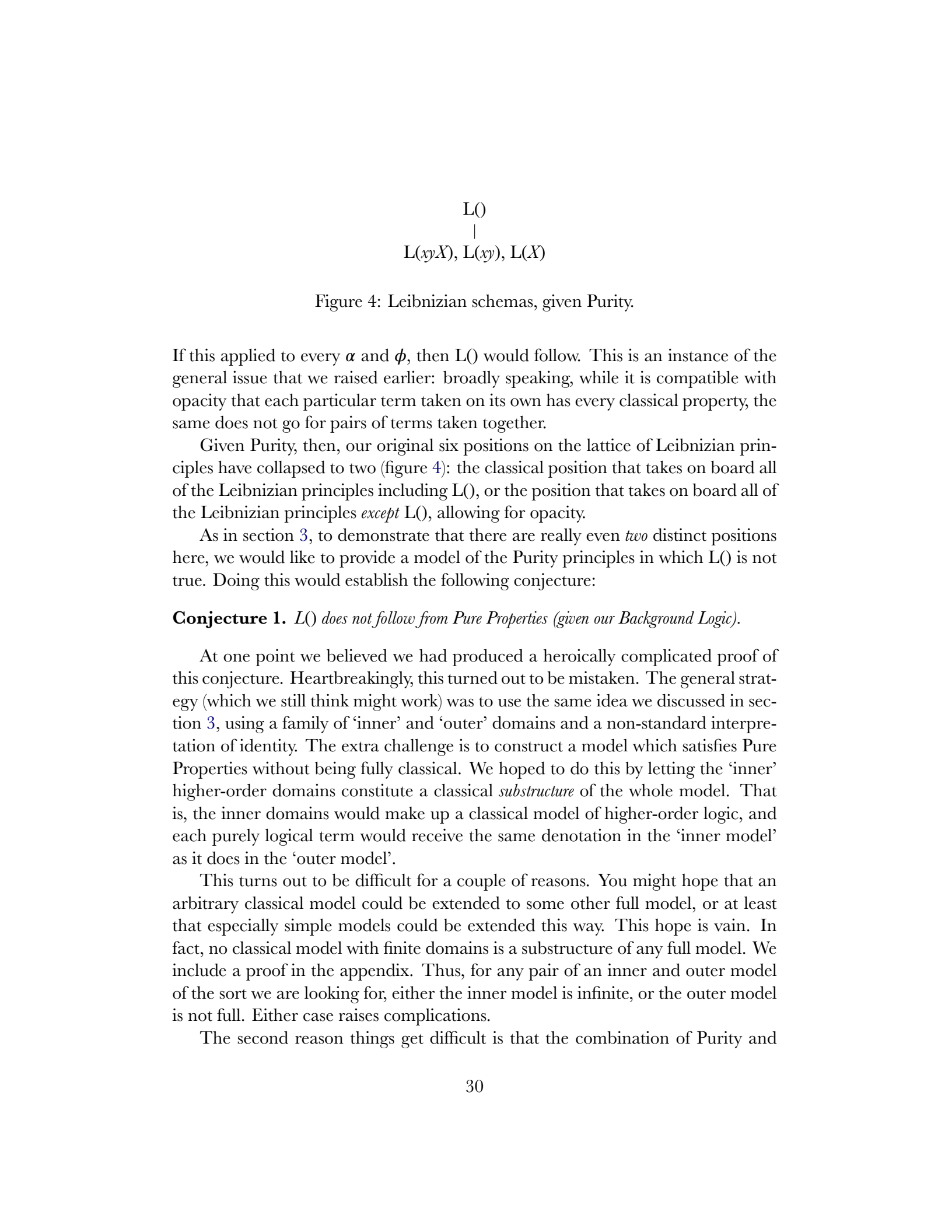

Figure 4: Leibnizian schemas, given Purity.

If this applied to every α and ϕ, then L() would follow. This is an instance of the general issue that we raised earlier: broadly speaking, while it is compatible with opacity that each particular term taken on its own has every classical property, the same does not go for pairs of terms taken together.

Given Purity, then, our original six positions on the lattice of Leibnizian principles have collapsed to two (figure 4): the classical position that takes on board all of the Leibnizian principles including L(), or the position that takes on board all of the Leibnizian principles *except* L(), allowing for opacity.

As in section 3, to demonstrate that there are really even *two* distinct positions here, we would like to provide a model of the Purity principles in which L() is not true. Doing this would establish the following conjecture:

Conjecture 1. *L() does not follow from Pure Properties (given our Background Logic).*

At one point we believed we had produced a heroically complicated proof of this conjecture. Heartbreakingly, this turned out to be mistaken. The general strategy (which we still think might work) was to use the same idea we discussed in section 3, using a family of 'inner' and 'outer' domains and a non-standard interpretation of identity. The extra challenge is to construct a model which satisfies Pure Properties without being fully classical. We hoped to do this by letting the 'inner' higher-order domains constitute a classical *substructure* of the whole model. That is, the inner domains would make up a classical model of higher-order logic, and each purely logical term would receive the same denotation in the 'inner model' as it does in the 'outer model'.

This turns out to be difficult for a couple of reasons. You might hope that an arbitrary classical model could be extended to some other full model, or at least that especially simple models could be extended this way. This hope is vain. In fact, no classical model with finite domains is a substructure of any full model. We include a proof in the appendix. Thus, for any pair of an inner and outer model of the sort we are looking for, either the inner model is infinite, or the outer model is not full. Either case raises complications.

The second reason things get difficult is that the combination of Purity and

opacity seems to require (somewhat to our dismay) that propositions and relations are fine-grained. We can illustrate the problem by supposing that relations are very coarse-grained: suppose that there are no distinct coextensive (higher-order) relations. Since L(*xyX*) tells us that the relation

$$\lambda x\, \lambda y\, \lambda X (x = y \to Xx \to Xy) \tag{9}$$

is coextensive with

$$\lambda x\, \lambda y\, \lambda X \top \tag{10}$$

if relations are determined extensionally then these relations are identical. But as we've shown, Purity implies that for purely logical terms α and β, if $\alpha = \beta$ is true then α and β are intersubstitutable. So (9) and (10) are intersubstitutable. For any terms α, β, and ϕ, we know that $\lambda Z Z \alpha \beta \phi$ applies to (10), so it also applies to (9). But then (by β-reduction) it follows that there is no opacity at all. Moreover, this same kind of argument goes through using assumptions much weaker than extensionality. Indeed, if we think of L(*xyX*) as a logical truth, then in a natural sense Purity and opacity require us to distinguish logically equivalent relations.

At this point, then, it remains an open question whether the full Purity package is really compatible with opacity.

So far we've been using the terms 'exportable' and 'transparent' as metalinguistic predicates of expressions. One may wish to introduce these notions to the object language, using a predicate E_σ of type $\sigma \to t$ such that $E_\sigma \alpha$ is true whenever α is an exportable expression of type σ. Similarly one could introduce a transparency predicate T_σ of type $(\sigma \to t) \to t$. We see no obstacle to enriching the language in this way, but one must take care: the predicates E and T are themselves neither exportable nor transparent, if there is any opacity at all. Suppose a is not exportable. By Being there is some x such that $x = a$. Furthermore by Pure Export, x is exportable. So 'exportable' is not transparent. Similarly, suppose that F is not transparent. By Being there is some property X such that $X = F$. By Pure Transparency X is transparent. So 'transparent' is not transparent. Finally, L(X) says that exportable predicates are transparent, so neither E nor T is exportable either. One consequence of these observations is that if there is any opacity, then neither E nor T is definable in purely logical terms (since by Pure Export any such definition would be exportable, and similarly by Pure Transparency it would be transparent). Of course, the *existence* predicate $\lambda x\, \exists y\, y = x$ won't do to define exportability — as we have discussed, being something does not suffice for being exportable. What this shows is that no other logical predicate will do, either.

One question we have not attempted to answer is, in general, which predicates are opaque, and which are transparent? The reader may feel unsettled by this silence. Applications of Leibniz's law are pervasive in the sciences and in ordinary reasoning, but these applications are not sanctioned as logically valid insofar as they involve logically impure expressions. Since Pure Transparency doesn't help us in these cases, how are we to know when we are reasoning with transparent predicates? According to the view defended in this paper, this is not a question of logic, but of philosophy. By analogy, the free logician tells us that existential generalization is not generally valid — but that isn't to say that one can't conclude from the fact that seven is a prime number that there are prime numbers; for in fact seven is something, unlike Zeus or Pegasus. This contrast between what is (such as seven) and what is not (such as Zeus) is not given to us by logic alone. The same goes for the contrast between transparent and opaque predicates. The question of which predicates are 'worldly', rather than 'representational', is important and difficult. Some cases seem relatively clear — belief belongs on one side and mathematics on the other — but many cases are contested, such as contingency, vagueness, chance, or value. The logic of opacity provides a framework for asking these questions, but does not answer them.

A Leibniz Models

Definition 3. The (relational) **types** are inductively defined as follows: e is a type, and $\sigma_1 \to \ldots \to \sigma_n \to t$ is a type for any relational types $\sigma_1, \ldots, \sigma_n$. (For $n = 0$, this is the type t.) Let $\sigma^n \to t$ abbreviate the type $\sigma \to \ldots \to \sigma \to t$.

A **typed family of sets** is a function from types to sets.

If X and Y are typed families of sets, a **typed family of functions** $f : X \to Y$ is a function from types σ to functions from $X\sigma$ to $Y\sigma$.

Definition 4. Let V be a typed family of countably infinite sets: the **variables**. Let C be a typed family of sets: the **constants**. The **terms** (based on C) are the elements of a typed family of sets L, inductively defined as follows, using the shorthand $a : \sigma$ to mean that a is a term of type σ:

$c : \sigma$	for each constant $c \in C\sigma$
$x : \sigma$	for each variable $x \in V\sigma$
$ab : \tau$	for terms $a : \sigma \to \tau$ and $b : \sigma$
$\lambda x\, a : \sigma \to \tau$	for each variable $x \in V\sigma$ and term $a : \tau$

We assume in what follows that the constants include the following **logical constants**:

$$\begin{aligned} &\rightarrow : t \rightarrow t \rightarrow t \\ &\forall_\sigma : (\sigma \rightarrow t) \rightarrow t && \text{for each type } \sigma \\ &=_\sigma : \sigma \rightarrow \sigma \rightarrow t && \text{for each type } \sigma \end{aligned}$$

We assume for simplicity that the other logical connectives, such as conjunction, negation, and the existential quantifier, are defined in terms of these in some standard way; this allows us to interpret arbitrary formulas of higher-order logic. For the sake of verifying the *truth* of the principles we are interested in, it doesn't really matter which of the various available definitions we use, as long as they are truth-conditionally standard.[15] A term is **purely logical** iff it does not contain any non-logical constants.

In section 3 we said that the higher-order domains of our models contained functions; but in fact, it is technically convenient to use a slightly broader notion of things that can be 'applied' (as in, e.g. Benzmüller, Brown, and Kohlhase 2004, p. 1033).

Definition 5. An **applicative structure** (for constants C) consists of

1. A typed family of sets M — the **domains**,
2. An operation $*_{\sigma\rightarrow\tau} : M(\sigma \rightarrow \tau) \rightarrow M\sigma \rightarrow M\tau$ for each types σ and τ,
3. For each **variable assignment** — each typed family of functions $g : V \rightarrow M$ from variables to domains — a typed family of functions $M_g : L \rightarrow M$ that takes each type-σ term to an element of the type-σ domain. These are required to have the following properties:

$$\begin{aligned} M_g(x) &= gx && \text{for each variable } x \\ M_g(c) &= M_{g'}(c) && \text{for each constant } c \text{ and any assignments } g \text{ and } g' \\ M_g(ab) &= M_g a * M_g b && \text{for } a : \sigma \rightarrow \tau \text{ and } b : \sigma \\ M_g(\lambda x\, a) * d &= M_{g[x\mapsto d]} a && \text{for } a : \tau \text{ and } x : \sigma, \text{ for each } d \in M\sigma \end{aligned}$$

   Here $g[x \mapsto d]$ is the assignment that agrees with g on all variables other than x, and maps x to d.

[15] To see that this is a complete basis, note that $\neg\phi$ can be defined as $\phi \rightarrow \forall_t p\, p$. For some of the systems we will discuss, identity could be defined in terms of Leibniz equivalence, rather than taken as an additional primitive constant.

For a closed term a, let Ma be $M_g a$ for an arbitrary assignment g.

Definition 6. Let M be an applicative structure.

- M is **functional** iff the operation $*$ is injective (considered as a function from $M(\sigma \to \tau)$ to the set of functions $M\sigma \to M\tau$).
- M is **full** iff $*$ is surjective (in the same sense).
- M is **non-trivial** iff $M\sigma$ contains at least two elements for each type σ.
- M is **extensional** iff $Mt = \{0, 1\}$.

Definition 7. A **Leibniz structure** consists of

1. An applicative structure $\mathcal{N}$, the **outer structure**,
2. A typed family of subsets $M\sigma \subseteq \mathcal{N}\sigma$, the **inner domains**,
3. A typed family of functions $r : \mathcal{N} \to M$, the **representative functions**, and
4. A function $v : Mt \to 2$, the **valuation** (where $2 = \{0, 1\}$ with the usual ordering).

These are required to obey the following constraints. We say $p \in \mathcal{N}t$ is **true** iff $v(r_t p) = 1$; we say $p \leq q$ iff $v(r_t p) \leq v(r_t q)$. We say $a \sim_\sigma b$ iff $r_\sigma a = r_\sigma b$.

1. For each inner element $a \in M\sigma$,
$$r_\sigma a = a$$
2. For each outer function $F \in \mathcal{N}(\sigma \to \tau)$ and inner element $d \in M\sigma$,
$$r_{\sigma\to\tau} F * d \sim_\tau F * d$$
3. For $p, q \in \mathcal{N}t$,
$$\mathcal{N}(\to) * p * q \text{ is true iff } p \leq q$$
4. For each outer function $F \in \mathcal{N}(\sigma \to \tau)$,
$$\mathcal{N}(\forall_\sigma) * F \text{ is true iff } F * d \text{ is true for every inner element } d \in M\sigma$$

5. For $a, b \in \mathcal{N}\sigma$,

$$\mathcal{N}(=_\sigma) * a * b \text{ is true iff } a \sim_\sigma b$$

As we discussed in section 3, a Leibniz structure includes a family of equivalence relations for each type; these correspond to object-language identity according to constraint 5. Here we define these equivalence relations in terms of a family of representative functions $r : \mathcal{N} \to M$. This approach kills several birds with one stone. It guarantees that $\sim_\sigma$ is an equivalence relation just from the structure of its definition. It guarantees the Being condition, since each outer element d is equivalent to its representative inner element $r_\sigma d$. It guarantees L(xy), since no two inner elements are equivalent to one another. (If d and d' are each inner elements such that $d \sim d'$, then in fact $d = d'$; so for an arbitrary outer element $F \in \mathcal{N}(\sigma \to \tau)$, clearly $F * d \sim F * d'$.) In combination with our definition of truth in terms of a valuation function on the *inner* domain of propositions, this approach also automatically guarantees the Truth Value condition, that truths are only equivalent to truths.

Condition 2 of the definition guarantees one direction of Functionality: for any outer function, its equivalent inner function is *functionally equivalent* to it. This in turn implies Comprehension. Conditions 3 and 4 of the definition have the effect of ensuring that classical propositional logic and standard free logic hold.

We can make these points more precise with the following lemmas.

Definition 8. A term $\phi : t$ is **true** in a Leibniz structure $(M, \mathcal{N}, r, v)$ iff for each variable assignment g, the element $\mathcal{N}_g\phi \in \mathcal{N}t$ is true. That is, $v(r_t(\mathcal{N}_g\phi)) = 1$.

Lemma 1. *For any Leibniz structure S, truth-in-S is closed under modus ponens, as well as universal generalization in the following form: if $\phi \to \psi$ is true in S and x is not free in ϕ, then $\phi \to \forall x\, \psi$ is true in S.*

Proof. This follows straightforwardly from conditions (3) and (4) of Definition 7. □

Lemma 2. *Let S be a Leibniz structure.*

1. *Each Background Logic principle (section 3) is true in S.*
2. *Each instance of $L(xyX)^=$ and $L(xy)^=$ is true in S.*
3. *Each instance of $L(X)^=$ is true in S iff*

$$F * r_\sigma d \sim_\tau F * d \quad \text{for every } F \in M(\sigma \to t) \text{ and } d \in \mathcal{N}\sigma \qquad \text{(LX)}$$

4. *Extensionality is true in S iff M is extensional.*

Lemma 2 is straightforward to prove from the definitions. Note that property (LX) says that each inner function F preserves equivalence.

Lemma 3. *There exist Leibniz structures that have property* (LX)*, and there exist Leibniz structures that do not have property* (LX)*. In particular, there exist extensional Leibniz structures of these two sorts.*

Proof. Let $\mathcal{N}$ be any full and functional applicative structure such that $\mathcal{N}e$ and $\mathcal{N}t$ each have at least three elements. We will simultaneously define $M \subseteq \mathcal{N}$ and $r : \mathcal{N} \to M$ by induction on types. For the base types, let Me and Mt be arbitrary proper subsets of $\mathcal{N}e$ and $\mathcal{N}t$ that each contain at least two elements. Let $r_e : \mathcal{N}e \to Me$ and $r_t : \mathcal{N}t \to Mt$ be any functions that fix Me and Mt, respectively.

For any types σ and τ, we will define $r_{\sigma\to\tau}$ as follows. For each function $F : M\sigma \to M\tau$, there are many elements $\overline{F} \in \mathcal{N}(\sigma \to \tau)$ that extend F, in the sense that $\overline{F} * d = Fd$ for any $d \in M\sigma$. We will pick one such element $\overline{F}$ in one of two ways — depending on whether we want to guarantee that (LX) holds or that it does not. To guarantee (LX), we choose $\overline{F}$ such that

$$\overline{F} * d \sim_\tau F(r_\sigma d) \quad \text{for every } d \in \mathcal{N}\sigma$$

For the alternative, we ensure that for at least one function F, the extension $\overline{F}$ does not satisfy this. Then, for any function $G \in \mathcal{N}(\sigma \to \tau)$, let $r_{\sigma\to\tau}G$ be $\overline{F}$, where F is the function that takes each $d \in M\sigma$ to $r_\tau(G * d)$. Let $M(\sigma \to \tau)$ be the range of $r_{\sigma\to\tau}$, or in other words, the set $\{\overline{F} \mid F : M\sigma \to M\tau\}$.

It's straightforward to check that r meets conditions (1) and (2) of Definition 7.

We also need to define the valuation function $v : Mt \to 2$. We can let v be any such function such that at least one element is mapped to 0 and at least one is mapped to 1. Finally, we can define the interpretations of the logical constants: for each pair of propositions $p, q \in \mathcal{N}t$, we can pick some arbitrary element $p \to q \in \mathcal{N}t$ with the appropriate truth-value, and likewise for the quantifiers (using the fact that the outer model $\mathcal{N}$ is full, and so appropriate elements of $\mathcal{N}(t \to t \to t)$ and $\mathcal{N}((\sigma \to t) \to t)$ exist). This guarantees conditions (3), (4), and (5) of the definition. □

Theorem 1 in section 3 immediately follows from Lemmas 1, 2, and 3.

B Purity Theorems

In section 4 we stated that certain Nice Principles are derivable from each of three variations on the Purity package. Here we will show how this can be done.

First, let's give the precise definition of the notion of a classical structure we adverted to in section 4.

Definition 9. An applicative structure M is **classical** iff M is full and functional, and additionally, for each type σ,

$$M(=_\sigma) \quad = \quad M(\lambda x \lambda y \forall_{\sigma \to t} X(Xx \to Xy))$$

(This constraint corresponds to the Identity Identity.)

Theorem 2. *Pure Truth and Pure Export together imply each of the Nice Principles.*

Proof Sketch. We noted above that L(xyX) is an immediate consequence of Pure Truth. We have also already shown in section 3 that given Comprehension (part of our Background Logic), L(xyX) implies L(xy). This means that if a and b are each *exportable* terms, and $a = b$, then a and b are intersubstitutable: if ϕa, then ϕb.

Recall that Pure Truth (as we have formulated it) also implies the Identity Identity: the identity relation $=$ is identical to Leibniz equivalence $\lambda x \lambda y \forall X(Xx \to Xy)$. Furthermore, Pure Export says that identity and Leibniz equivalence are each exportable. Using this fact together with L(xy) lets us prove that identity is well-behaved. For instance, $\lambda X Xaa$ applies to Leibniz equivalence (this just says that a has any property that a has). It then follows by L(xy) that this also applies to identity, which (using β-reduction) implies that $a = a$. A similar argument shows that identity has the Euclidean property, by considering higher-order properties of the form $\lambda X(Xab \to Xac \to Xbc)$. This proves Equivalence.

The same style of argument also lets us prove L(X) from L(xy) and the Identity Identity, this time using the property $\lambda Y \forall X(Yab \to Xa \to Xb)$. Again, Leibniz equivalence has this property as a matter of straightforward quantificational logic, so by L(xy) and Pure Export, identity has it as well, which is what L(X) requires. As we mentioned earlier, L(X) (given Pure Export) also suffices for Pure Transparency. We also already noted that Pure Transparency implies Truth Value.

Next we will prove Functionality:

$$\phi = \psi \leftrightarrow \forall x (\phi x = \psi x)$$

We can start by applying Pure Transparency to the pure predicate $\lambda Y \forall x (Yx = Xx)$, to conclude:

$$\forall X(\phi = X \to \forall x(\phi x = Xx)) \tag{11}$$

By Being, there is some property X equal to ϕ and some property Y equal to ψ. Then (11), with Equivalence, implies:

$$\forall x(\phi x = \psi x) \leftrightarrow \forall x(Xx = Yx)$$

Furthermore, by Equivalence $\phi = \psi$ iff $X = Y$, which in turn holds iff $\forall x(Xx = Yx)$ (the latter is an instance of Pure Truth). Putting these biconditionals together gives us Functionality.

Deriving the strict Leibnizian principles $\mathrm{L}(xyX)^{=}$, $\mathrm{L}(xy)^{=}$ and $\mathrm{L}(X)^{=}$ is left as an exercise.[16] □

Theorem 3. *Pure Truth and Pure Transparency together imply each of the Nice Principles.*

Proof. In light of Theorem 2, it suffices to prove Pure Export from these assumptions. Let ϕ be an arbitrary predicate, and let α be an arbitrary pure term. Note that in this case the predicate

$$\lambda Y(\forall x\, Yx \rightarrow Y\alpha) \tag{12}$$

is pure and thus transparent by Pure Transparency. By Being, there is some property X identical to ϕ. Note also that X has the property (12), by Pure Truth. So, since $X = \phi$ and (12) is transparent, it follows that this predicate also applies to ϕ. By β-reduction we can then conclude

$$\forall x\, \phi x \rightarrow \phi\alpha$$

which is just what Pure Export says. □

Call a predicate $\phi : \sigma \rightarrow t$ **universal** iff it is purely logical and applies to every element in the σ-domain of every classical structure. Consider the following schema:

Pure Properties. $\phi\alpha$, where $\phi : \sigma \rightarrow t$ is universal and $\alpha : \sigma$.

Theorem 4. *Pure Properties is equivalent to the Nice Principles.*

Proof. First we show that Pure Truth and Pure Transparency imply Pure Properties. Suppose $\phi : \sigma \rightarrow t$ is universal, and $\alpha : \sigma$ is arbitrary. By Pure Transparency we have $\forall x(x = \alpha \rightarrow \phi x \rightarrow \phi\alpha)$, and then by quantificational logic $\forall x\, \phi x \rightarrow \exists x(x =$

[16] Hint: the case of $\mathrm{L}(X)^{=}$ is the trickiest. Use the fact that Being implies $\alpha = \beta \rightarrow \exists x(x = \alpha \wedge x = \beta)$.

$\alpha) \to \phi\alpha$. Since ϕ is universal, $\forall x\, \phi x$ is true in every classical model, and thus true by Pure Truth; and $\exists x\, x = \alpha$ is true by Being. Thus $\phi\alpha$ is true.

For the converse, it suffices to show that Pure Properties implies Pure Truth and Pure Export. These immediately follow from applying Pure Properties to the following universal predicates:

$$\lambda x\, \phi \qquad \text{where } \phi \text{ is a purely logical sentence true in all classical structures} \tag{13}$$

$$\lambda X (\forall y\, Xy \to X\alpha) \qquad \text{where } \alpha \text{ is purely logical} \tag{14}$$

□

C Models of Purity

Our final ambition is to provide models for Purity without L(). Unfortunately, we have not yet succeeded in this ambition, but here we provide some notes on our progress which we hope will be helpful.

First, note that Pure Export requires that each purely logical expression denotes something in the *inner* domain of our structure. Thus the inner domains aren't just an arbitrary family of sets, but rather M is itself a smaller applicative structure living within the larger structure $\mathcal{N}$. Furthermore, since exportability is preserved under application, this means that application within this smaller structure M agrees with application on the larger structure $\mathcal{N}$. That is, Purity requires that M is a *substructure* of $\mathcal{N}$, in the following sense.

Definition 10. Let M and $\mathcal{N}$ be applicative structures. Then M is a **substructure** of $\mathcal{N}$ iff

1. $M\sigma \subseteq \mathcal{N}\sigma$ for each type σ
2. Each application operation $*_{\sigma\to\tau}$ in M is the restriction of the corresponding operation in $\mathcal{N}$ to $M(\sigma \to \tau)$ and $M\sigma$
3. For each term a and assignment g of values in M, $M_g a = \mathcal{N}_g a$.

Our strategy has been to construct a pair M and $\mathcal{N}$ where M is a classical proper substructure of $\mathcal{N}$.[17] It is helpful to understand a constraint that makes this task a bit more difficult than it might otherwise seem.

[17] Note, though, that this is not the only way in which we might proceed. An alternative would be to use an inner model which is not classical, but still satisfies Pure Truth. There are such models, by the Löwenheim-Skolem Theorem.

Definition 11. An applicative structure M is **finitary** iff $M\sigma$ is finite for each type σ.

Theorem 5. *No finitary, full, and functional applicative structure is a proper substructure of any full structure.*

Proof. Consider the **Church numerals**, which are terms of type $(t \to t) \to t \to t$ given as follows:

$$c_0 = \lambda X \lambda x\, x$$
$$c_{n+1} = \lambda X \lambda x\, X(c_n X x)$$

That is, the nth Church function takes a $t \to t$ function and composes it with itself n times.

If the domain of type t is infinite, then these terms each denote distinct functions (in which case they are useful for the purpose that Church put them to, of encoding arithmetic using expressions of the pure λ-calculus). But if the t-domain is *finite*, then the denotations of some of these numerals collapse: there are infinitely many terms, and only finitely many functions for them to denote. Furthermore, which of these numerals collapse depends on the size of the type-t domain. For any numbers $0 < m < n$, there are functions c_i and c_j that denote the same function $(Mt \to Mt) \to Mt \to Mt$ when $|Mt| = m$, but are distinguished when $|Mt| = n$.[18]

Now suppose that M is a full, functional, and finitary applicative structure, which is a proper substructure of a full structure $\mathcal{N}$. Since $\mathcal{N}$ is full and $0 < |Mt| < |\mathcal{N}t|$, it follows that there are c_i and c_j such that $M(c_i) = M(c_j)$, but $\mathcal{N}(c_i) \neq \mathcal{N}(c_j)$. But since M is a substructure of $\mathcal{N}$, $M(c_i) = \mathcal{N}(c_i)$ and $M(c_j) = \mathcal{N}(c_j)$. This is a contradiction. □

The upshot is that any model of Purity without L() has to have one of two features: either the outer model is not full, and thus not simply given by the space of functions, or else the inner model is infinite.

Definition 12. A **purity structure** is a Leibniz structure $(M, \mathcal{N}, r, v)$, where M does not just provide a family of *subsets* of the $\mathcal{N}$-domains, but rather M is a classical *substructure* of $\mathcal{N}$, and which furthermore satisfies this constraint:

$$r_\tau(F * d) = F * r_\sigma d \quad \text{for each inner } F \in M(\sigma \to t) \text{ and outer } d \in \mathcal{N}\sigma \qquad (*)$$

[18] Thanks to Eric Wofsey for a proof of this: see http://math.stackexchange.com/q/2208850/ . In particular, for $0 < m < n$, we can let i and j be $n - 2$ and $m! + n - 2$.

Constraint (*) says that the representative function r commutes with inner predicates. This is a simplified form of the property (LX) that encodes $\mathrm{L}(X)^{=}$: it reflects the object-language claim that for exportable X and y, if $a = y$, then $Xa = Xy$. This alternative formulation is possible because we are now also assuming that exportability is closed under application (since the inner domains, $M\sigma$, form a model). So since F and $r_\sigma d$ are each 'inner' elements, it follows that $F * r_\sigma d$ is also an inner element; thus it is not merely *equivalent* to the inner element $r_\tau(F * d)$, but the very same model-theoretic object.

Note that of the original five conditions in the definition of a Leibniz structure (Definition 7), conditions 3, 4, and 5 follow from assigning classical denotations to $\rightarrow$, $\forall_\sigma$, and $=_\sigma$ in the inner structure M, and condition 2 follows from (*) together with condition 1 of Definition 7.[19] Thus to construct a purity structure, it will be enough to construct a classical substructure M of an applicative structure $\mathcal{N}$, with a family of functions $r : \mathcal{N} \rightarrow M$ that satisfies (*) and fixes M (condition 1 of Definition 7).

Lemma 4. *If S is a purity structure, then each instance of Pure Properties is true in S. Thus S satisfies each of the Nice Principles.*

Proof. Suppose ϕ is a universal predicate, and let $a = \mathcal{N}\alpha$ and $F = M\phi = \mathcal{N}\phi$. Since ϕ is universal and M is classical, $v(F * d) = 1$ for each $d \in M\sigma$. In particular, $r_\sigma a \in M\sigma$, and so

$$v(r_t(\mathcal{N}(\phi\alpha))) = v(r_t(F * a)) = v(F * r_\sigma a) = 1$$

That is, $\phi\alpha$ is true in S. □

Conjecture 2. *There exists a purity structure.*

References

[1] Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase. "Higher-Order Semantics and Extensionality". In: *Journal of Symbolic Logic* 69.4 (2004), pp. 1027–1088.

[19] For $d \in M\sigma$, consider $G = M_{[y \mapsto d]}(\lambda XXy)$. For any $F \in \mathcal{N}(\sigma \rightarrow \tau)$, $F * d = G * F$. Thus using (*), the fact that M is closed under application, and the fact that r_τ fixes each element of M_τ,

$$r_\tau(F * d) = r_\tau(G * F) = G * r_{\sigma\rightarrow\tau}F = r_{\sigma\rightarrow\tau}F * d = r_\tau(r_{\sigma\rightarrow\tau}F * d)$$

[2] David J. Chalmers. "Two-Dimensional Semantics". In: *Oxford Handbook of the Philosophy of Language*. Ed. by E. Lepore and B. Smith. Oxford University Press, 2006.

[3] Mark Crimmins and John Perry. "The Prince and the Phone Booth: Reporting Puzzling Beliefs". In: *Journal of Philosophy* 86.12 (1989), pp. 685–711.

[4] Cian Dorr. "To Be F Is To Be G". In: *Philosophical Perspectives* 30.1 (Dec. 1, 2016), pp. 39–134.

[5] Cian Dorr. "Transparency and the Context-Sensitivity of Attitude Reports". In: *Empty Representations: Reference and Non-existence*. Ed. by Manuel García-Carpintero and Genoveva Martí. Oxford University Press, 2014, pp. 25–66.

[6] Kit Fine. "The Permutation Principle in Quantificational Logic". In: *Journal of Philosophical Logic* 12.1 (1983), pp. 33–37.

[7] Gottlob Frege. *Conceptual Notation and Related Articles*. Trans. by Terrell Ward Bynum. 1 edition. Oxford: Clarendon Press, 2000. 306 pp.

[8] Gottlob Frege. "On Sense and Reference". In: *Arguing About Language*. Ed. by Darragh Byrne and Max Kölbel. Routledge, 2010, pp. 36–56.

[9] Allan Gibbard. "Contingent Identity". In: *Journal of Philosophical Logic* 4.2 (1975), pp. 187–222.

[10] Leon Henkin. "Completeness in the Theory of Types". In: *Journal of Symbolic Logic* 15.2 (1950), pp. 81–91.

[11] David Kaplan. "Quantifying In". In: *Synthese* 19.1-2 (1968), pp. 178–214.

[12] Saul A. Kripke. "Russell's Notion of Scope". In: *Mind* 114.456 (2005), pp. 1005–1037.

[13] Saul A. Kripke. "Semantical Considerations on Modal Logic". In: *Acta Philosophica Fennica* 16.1963 (1963), pp. 83–94.

[14] Karel Lambert. "Existential Import Revisited". In: *Notre Dame Journal of Formal Logic* 4.4 (1963), pp. 288–292.

[15] John McDowell. "On the Sense and Reference of a Proper Name". In: *Mind* 86.342 (1977), pp. 159–185.

[16] Vann McGee. "'Kilimanjaro'". In: *Canadian Journal of Philosophy* 27.sup1 (1997), pp. 141–163.

[17] A. N. Prior. "On a Family of Paradoxes". In: *Notre Dame Journal of Formal Logic* 2.1 (1961), pp. 16–32.

[18] W. V. Quine. *Philosophy of Logic*. Harvard University Press, 1986.

[19] W. V. Quine. "Quantifiers and Propositional Attitudes". In: *Journal of Philosophy* 53.5 (1956), pp. 177–187.

[20] W. V. Quine. *Word and Object*. The MIT Press, 1960.

[21] Augustin Rayo. *The Construction of Logical Space*. Oxford University Press, 2013.

[22] Nathan Salmon. "Lambda in Sentences with Designators". In: *Journal of Philosophy* 107.9 (2010), pp. 445–468.

[23] Jennifer M. Saul. "Substitution and Simple Sentences". In: *Analysis* 57.2 (1997), pp. 102–108.

[24] Stewart Shapiro. *Foundations Without Foundationalism: A Case for Second-Order Logic*. Oxford University Press, 1991.

[25] Stewart Shapiro. "Principles of Reflection and Second-Order Logic". In: *Journal of Philosophical Logic* 16.3 (1987), pp. 309–333.

[26] Timothy Williamson. "Everything". In: *Philosophical Perspectives* 17.1 (2003), pp. 415–465.